ProCurve Networking
HP Innovation

6400cl
5300xl
4200vl
3400cl

Advanced Traffic
Management Guide

ProCurve Switches
E.10.02 (Series 5300xl)
L.10.XX (Series 4200vl)
M.08.73 (Series 3400/6400cl)

www.procurve.com

ProCurve
Series 6400cl Switches
Series 5300xl Switches
Series 4200vl Switches
Series 3400cl Switches

Advanced Traffic Management Guide

Applicable Products

ProCurve Switch 5308XL	(J4819A)
ProCurve Switch 5372XL	(J4848A)
ProCurve Switch 5348XL	(J4849A)
ProCurve Switch 5304XL	(J4850A)
ProCurve Switch 4204vl	(J8770A)
ProCurve Switch 4208vl	(J8773A)
ProCurve Switch 4202vl-72	(J8772A)
ProCurve Switch 4202vl-48G	(J8771A)
ProCurve Switch 3400cl-24G	(J4905A)
ProCurve Switch 3400cl-48G	(J4906A)
ProCurve Switch 10G CX4 6400cl-6XG	(J8433A)
ProCurve Switch 10G X2 6400cl-6XG	(J8474A)

Trademark Credits

Microsoft, Windows, Windows 95, and Microsoft Windows NT are US registered trademarks of Microsoft Corporation. Internet Explorer is a trademark of Microsoft Corporation. Ethernet is a registered trademark of Xerox Corporation. Netscape is a registered trademark of Netscape Corporation. Cisco® is a trademark of Cisco Systems, Inc.

Contents

Product Documentation

1 Getting Started

2 Static Virtual LANs (VLANs)

6 Spanning-Tree Operation

8 Quality of Service (QoS): Managing Bandwidth More Effectively

9 Access Control Lists (ACLs) for the Series 5300xl Switches

10 Access Control Lists (ACLs) for the Series 3400cl and Series 6400cl Switches

11 IP Routing Features

12 Router Redundancy Using XRRP

13 Stack Management for the Series 3400cl, 6400cl, and 4200vl Switches

Index

—This page unused intentionally—

Product Documentation

About Your Switch Manual Set

Printed Publications

The two publications listed below are printed and shipped with your switch.
The latest version of each is also available in PDF format on the ProCurve
Networking Web sit, as described in the above Note.

- *Read Me First*—Provides software update information, product notes,
  and other information.
- *Installation and Getting Started Guide*—Explains how to prepare for
  and perform the physical installation and connect the switch to your
  network.

Electronic Publications

The latest version of each of the publications listed below is available in PDF
format on the ProCurve Web site, as described in the above Note.

- *Management and Configuration Guide*—Describes how to configure,
  manage, and monitor switch operation.
- *Advanced Traffic Management Guide*—Explains how to configure traffic
  management features such as STP, QoS, and IP routing.
- *Access Security Guide*—Explains how to configure access security fea-
  tures and user authentication on the switch.
- *Release Notes*—Describe new features, fixes, and enhancements that
  become available between revisions of the main product guide.

Feature Index

For the manual set supporting your switch model, the following feature index indicates which manual to consult for information on a given software feature and which switches support that feature.

Feature	Management and Configuration	Advanced Traffic Management	Access Security Guide	Supported on 5300xl	Supported on 4200vl	Supported on 3400cl/ 6400cl
802.1Q VLAN Tagging		X		yes	yes	yes
802.1X Port-Based Priority	X			yes	yes	yes
802.1X Multiple Authenticated Clients per port			X	yes	no	yes
ACLs		X		yes	no	yes
AAA Authentication			X	yes	yes	yes
Authorized IP Managers			X	yes	yes	yes
Authorized Manager List (web, telnet, TFTP)			X	yes	yes	yes
Auto MDIX Configuration	X			yes	yes	yes
BOOTP	X			yes	yes	yes
Config File	X			yes	yes	yes
Console Access	X			yes	yes	yes
Copy Command	X			yes	yes	yes
CoS (Class of Service)		X		yes	yes	yes
Debug	X			yes	yes	yes
DHCP Configuration		X		yes	yes	yes
DHCP Option 82		X		yes	yes	no
DHCP/Bootp Operation	X			yes	yes	yes
Diagnostic Tools	X			yes	yes	yes
Downloading Software	X			yes	yes	yes

Feature	Management and Configuration	Advanced Traffic Management	Access Security Guide	Supported on 5300xl	Supported on 4200vl	Supported on 3400cl/ 6400cl
Eavesdrop Protection			X	yes	yes	no
Event Log	X			yes	yes	yes
Factory Default Settings	X			yes	yes	yes
Flow Control (802.3x)	X			yes	yes	yes
File Management	X			yes	yes	yes
File Transfers	X			yes	yes	yes
Friendly Port Names	X			yes	yes	yes
Guaranteed Minimum Bandwidth (GMB)	X			yes	no	yes
GVRP		X		yes	yes	yes
IGMP		X		yes	yes	yes
Delayed Group Flush		X		yes	yes	yes
Interface Access (Telnet, Console/ Serial, Web)	X			yes	yes	yes
IP Addressing	X			yes	yes	yes
IP Routing		X		yes	yes	yes
Jumbo Support		X		no	no	yes
LACP	X			yes	yes	yes
Link	X			yes	yes	yes
LLDP	X			yes	yes	yes
LLDP-Med	X			yes	yes	no
MAC Address Management	X			yes	yes	yes
MAC Lockdown			X	yes	yes	yes
MAC Lockout			X	yes	yes	yes
MAC-based Authentication			X	yes	yes	yes
MAC authentication RADIUS support			X	yes	yes	yes
Management VLAN		X		yes	yes	yes

Feature	Management and Configuration	Advanced Traffic Management	Access Security Guide	Supported on 5300xl	Supported on 4200vl	Supported on 3400cl/ 6400cl
Meshing		X		yes	no	yes
Monitoring and Analysis	X			yes	yes	yes
Multicast Filtering			X	yes	no	no
Multiple Configuration Files	X			yes	yes	yes
Network Management Applications	X			yes	SNMP only	SNMP only
OpenView Device Management	X			yes	yes	yes
OSPF		X		yes	no	yes
Passwords			X	yes	yes	yes
Password Clear Protection			X	yes	yes	yes
PCM	X			yes	yes	yes
PIM Dense		X		yes	no	no
Ping	X			yes	yes	yes
Port Configuration	X			yes	yes	yes
Port Monitoring		X		yes	yes	yes
Port Security			X	yes	yes	yes
Port Status	X			yes	yes	yes
Port Trunking (LACP)	X			yes	yes	yes
Port-Based Access Control			X	yes	yes	yes
Port-Based Priority (802.1Q)	X			yes	yes	yes
Power over Ethernet (PoE)	X			yes	no	no
Protocol Filters			X	yes	no	no
Protocol VLANS		X		yes	no	yes
Quality of Service (QoS)		X		yes	yes	yes
RADIUS Authentication and Accounting			X	yes	yes	yes
Rate-limiting	X			yes	no	yes
RIP		X		yes	no	yes

Feature	Management and Configuration	Advanced Traffic Management	Access Security Guide	Supported on 5300xl	Supported on 4200vl	Supported on 3400cl/ 6400cl
RMON 1,2,3,9	X			yes	yes	yes
Routing		X		yes	yes	yes
Routing - IP Static		X		yes	yes	yes
Secure Copy	X			yes	yes	yes
SFLOW				yes	yes	yes
SFTP	X			yes	yes	yes
SNMPv3	X			yes	yes	yes
Software Downloads (SCP/SFTP, TFTP, Xmodem)	X X			yes	yes	yes
Source-Port Filters			X	yes	yes	yes
Spanning Tree (STP, RSTP, MSTP)		X		yes	yes	yes
SSHv2 (Secure Shell) Encryption			X	yes	yes	yes
SSL (Secure Socket Layer)			X	yes	yes	yes
Stack Management (Stacking)		X		no	yes	yes
Syslog	X			yes	yes	yes
System Information	X			yes	yes	yes
TACACS+ Authentication			X	yes	yes	yes
Telnet Access	X			yes	yes	yes
TFTP	X			yes	yes	yes
Time Protocols (TimeP, SNTP)	X			yes	yes	yes
Traffic/Security Filters			X	yes	yes	yes
Troubleshooting	X			yes	yes	yes
UDP Forwarder		X		yes	yes	yes
Virtual Stacking		X		no	yes	yes
Virus Throttling (connection-rate filtering)			X	yes	no	no
VLANs		X		yes	yes	yes
VLAN Mirroring (1 static VLAN)		X		yes	yes	no

Product Documentation

Feature	Management and Configuration	Advanced Traffic Management	Access Security Guide	Supported on 5300xl	Supported on 4200vl	Supported on 3400cl/ 6400cl
Voice VLAN		X		yes	yes	yes
Web Authentication RADIUS Support			X	yes	yes	yes
Web-based Authentication			X	yes	yes	yes
Web UI	X			yes	yes	yes
Xmodem	X			yes	yes	yes
XRRP		X		yes	no	yes

Getting Started

Contents

Introduction

This *Advanced Traffic Management Guide* is intended for use with the following switches:

- ProCurve Switch 10G CX4 6400cl-6xg
- ProCurve Switch 5304xl
- ProCurve Switch 5308xl
- ProCurve Switch 4204vl
- ProCurve Switch 4202vl-48G
- ProCurve Switch 3400cl-24G

- ProCurve Switch 10G X2 6400cl-6xg
- ProCurve Switch 5348xl
- ProCurve Switch 5372xl
- ProCurve Switch 4208vl
- ProCurve Switch 4202vl-72
- ProCurve Switch 3400cl-48G

This guide describes how to use the command line interface (CLI), Menu interface, and web browser to configure, manage, monitor, and troubleshoot switch operation.

For an overview of other product documentation for the above switches, refer to *"Product Documentation"* on page xxiii.

The *Product Documentation CD-ROM* shipped with the switch includes a copy of this guide. You can also download a copy from the ProCurve Networking web site, **www.procurve.com**.

Conventions

This guide uses the following conventions for command syntax and displayed information.

Feature Descriptions by Model

In cases where a software feature is not available in all of the switch models covered by this guide, the section heading specifically indicates which product or product series offer the feature.

For example, (the switch is highlighted here in **bold italics**):

"QoS Pass-Through Mode on the **Series 5300xl and 4200vl Switches**".

Command Syntax Statements

Syntax: ip < default-gateway < *ip-addr* >> | routing >

Syntax: show interfaces [*port-list*]

- Vertical bars (|) separate alternative, mutually exclusive elements.
- Square brackets ([ ]) indicate optional elements.
- Braces (< >) enclose required elements.
- Braces within square brackets ([< >]) indicate a required element within an optional choice.
- Boldface indicates use of a CLI command, part of a CLI command syntax, or other displayed element in general text. For example:

    "Use the **copy tftp** command to download the key from a TFTP server."
- Italics indicate variables for which you must supply a value when executing the command. For example, in this command syntax, you must provide one or more port numbers:

    Syntax: aaa port-access authenticator < *port-list* >

Command Prompts

In the default configuration, your switch displays one of the following CLI prompts:

```
ProCurve 6400cl#
ProCurve 5304xl#
ProCurve 5308xl#
ProCurve 4204vl#
ProCurve 4208vl#
ProCurve 3400-24cl#
ProCurve 3400-48cl#
```

To simplify recognition, this guide uses **ProCurve** to represent command prompts for all models. For example:

```
ProCurve#
```

(You can use the **hostname** command to change the text in the CLI prompt.)

Screen Simulations

Displayed Text. Figures containing simulated screen text and command output look like this:

```
ProCurve> show version
Image stamp:   /sw/code/build/info
               September 30 2005 13:43:13
               E.08.22
               139

ProCurve>
```
Figure 1-1.   Example of a Figure Showing a Simulated Screen

In some cases, brief command-output sequences appear without figure identification. For example:

```
ProCurve(config)# clear public-key
ProCurve(config)# show ip client-public-key
show_client_public_key: cannot stat keyfile
```

Port Identity Examples

This guide describes software applicable to both chassis-based and stackable ProCurve switches. Where port identities are needed in an example, this guide uses the chassis-based port identity system, such as "A1", "B3-B5", "C7", etc.

However, unless otherwise noted, such examples apply equally to the stack-able switches, which typically use only numbers, such as "1", "3-5", "15", etc. for port identities.

Keys

Simulations of actual keys use a bold, sans-serif typeface with square brackets. For example, the Tab key appears as **[Tab]** and the "Y" key appears as **[Y]**.

Sources for More Information

For additional information about switch operation and features not covered in this guide, consult the following sources:

■ For information on which product manual to consult on a given software feature, refer to the chapter *"Product Documentation"*.

N o t e   For the latest version of all ProCurve switch documentation, including Release Notes covering recently added features, visit the ProCurve Network-ing Web Site at **www.procurve.com**, click on **Technical support**, and then click on **Product Manuals (all)**.

■ Software Release Notes—Release notes are posted on the ProCurve Networking web site and provide information on new software updates:

   • New features and how to configure and use them

   • Software management, including downloading software to the switch

   • Software fixes addressed in current and previous releases

   To view and download a copy of the latest software release notes for your switch, refer to "Getting Documentation From the Web" on page 1-6.

■ Product Notes and Software Update Information—The printed *Read Me First* shipped with your switch provides software update information, product notes, and other information. For the latest version, refer to "Getting Documentation From the Web" on page 1-6.

■ Installation and Getting Started Guide—Use the *Installation and Getting Started Guide* shipped with your switch to prepare for and perform the physical installation. This guide also steps you through connecting the switch to your network and assigning IP addressing, as well as describing the LED indications for correct operation and trouble analysis. A PDF version of this guide is also provided on the *Product Documentation CD-*

ROM shipped with the switch. And you can download a copy from the ProCurve Networking Web Site. (See "Getting Documentation From the Web" on page 1-6.)

■ Advanced Traffic Management Guide—Use the *Advanced Traffic Management Guide* for information on:

- VLANs: Static port-based and protocol VLANs, and dynamic GVRP VLANs
- Multicast traffic control (IGMP) and Protocol-Independent Multicast routing (PIM-DM)
- Spanning-Tree: 802.1D (STP), 802.1w (RSTP), and 802.1s (MSTP)
- Meshing
- Quality-of-Service (QoS)
- Access Control Lists (ACLs)
- IP routing
- Static NAT for intranet applications (*Series 5300xl switches only*)
- XRRP (XL Router Redundancy Protocol)

■ Access Security Guide—Use the *Access Security Guide* for information on:

- Local username and password security
- Web-Based and MAC-based authentication
- RADIUS and TACACS+ authentication
- SSH (Secure Shell) and SSL (Secure Socket Layer) operation
- 802.1x port-based access control
- Port security operation with MAC-based control
- Authorized IP Manager security
- Key Management System (KMS)

Getting Documentation From the Web

1. Go to the ProCurve Networking Web Site at

   www.procurve.com

2. Click on **Technical support**.
3. Click on **Product manuals**.
4. Click on the product for which you want to view or download a manual.

Figure 1-2. **Example of How To Locate Product Manuals on the ProCurve Networking Web Site**

Figure 1-3. **Listing of ProCurve Manuals on the ProCurve Networking Web Site**

Online Help

If you need information on specific parameters in the menu interface, refer to the online help provided in the interface. For example:

```
==========================- CONSOLE - MANAGER MODE -=======================
                    Switch Configuration - Internet (IP) Service


 Default Gateway : 10.35.204.1
 Default TTL     : 64

 IP Config [DHCP/Bootp] : Manual
 IP Address  : 10.35.204.104
 Subnet Mask : 255.255.240.0                       Online Help
                                                   for Menu

 Actions->   Cancel     Edit      Save      Help

Display help information.
Use arrow keys to change action selection and <Enter> to execute action.
```

If you need information on a specific command in the CLI, type the command name followed by "help". For example:

```
ProCurve# write help
Usage: write <memory|terminal>

Description: View or save the running configuration of the switch.

            write terminal - displays the running configuration of the
                             switch on the terminal
            write memory   - saves the running configuration of the
                             switch to flash. The saved configuration
                             becomes the boot-up configuration of the switch
```

If you need information on specific features in the web browser interface use the online help available for the web browser interface. For more information on web browser Help options, see the section "Online Help for the Web Browser Interface" in the Management and Configuration Guide.

If you need further information on Hewlett-Packard switch technology, visit the ProCurve Networking web site at:

www.procurve.com

Need Only a Quick Start?

IP Addressing

If you just want to give the switch an IP address so that it can communicate on your network, or if you are not using VLANs, HP recommends that you use the Switch Setup screen to quickly configure IP addressing. To do so, do one of the following:

■ Enter **setup** at the CLI Manager level prompt.

        Procurve# setup

■ In the Main Menu of the Menu interface, select

    8. Run Setup

For more on using the Switch Setup screen, see the *Installation and Getting Started Guide* you received with the switch.

To Set Up and Install the Switch in Your Network

Use the ProCurve *Installation and Getting Started Guide* (shipped with the switch) for the following:

■ Notes, cautions, and warnings related to installing and using the switch and its related modules

■ Instructions for physically installing the switch in your network

■ Quickly assigning an IP address and subnet mask, set a Manager password, and (optionally) configure other basic features.

■ Interpreting LED behavior.

For the latest version of the *Installation and Getting Started Guide* for your switch, refer to "Getting Documentation From the Web" on page 1-6.

—This page is intentionally unused—

2

Static Virtual LANs (VLANs)

Contents

Overview

This chapter describes how to configure and use static, port-based and protocol-based VLANs on the switches covered by this manual.

For general information on how to use the switch's built-in interfaces, refer to these chapters in the *Management and Configuration Guide* for your switch:

- Chapter 3, "Using the Menu Interface"
- Chapter 4, "Using the Command Line Interface (CLI)"
- Chapter 5, "Using the Web Browser Interface
- Chapter 6, "Switch Memory and Configuration"

Introduction

VLAN Features

Feature	Default	Menu	CLI	Web
view existing VLANs	n/a	page 2-23 thru 2-28	page 2-29	page 2-39
configuring static VLANs	default VLAN with VID = 1	page 2-23 thru 2-28	page 2-28	page 2-39

VLANs enable you to group users by logical function instead of physical location. This helps to control bandwidth usage within your network by allowing you to group high-bandwidth users on low-traffic segments and to organize users from different LAN segments according to their need for common resources and/or their use of individual protocols. You can also improve traffic control at the edge of your network by separating traffic of different protocol types. VLANs can also enhance your network security by creating separate subnets to help control in-band access to specific network resources.

General VLAN Operation

A VLAN is comprised of multiple ports operating as members of the same subnet (broadcast domain). Ports on multiple devices can belong to the same VLAN, and traffic moving between ports in the same VLAN is bridged (or "switched"). (Traffic moving between different VLANs must be routed.) A s*tatic* VLAN is an 802.1Q-compliant VLAN configured with one or more ports that remain members regardless of traffic usage. (A *dynamic* VLAN is an 802.1Q-compliant VLAN membership that the switch temporarily creates on a port to provide a link to another port in the same VLAN on another device.)

This chapter describes *static* VLANs configured for port-based or protocol-based operation. Static VLANs are configured with a name, VLAN ID number (VID), and port members. (For *dynamic* VLANs, refer to chapter 3, "GVRP" .)

By default, the switches covered by this guide are 802.1Q VLAN-enabled and allow up to 256 static and dynamic VLANs. (The default static VLAN setting is 8). 802.1Q compatibility enables you to assign each switch port to multiple VLANs, if needed.

Types of Static VLANs Available in the Switch

Port-Based VLANs

This type of static VLAN creates a specific layer-2 broadcast domain comprised of member ports that bridge IPv4 traffic among themselves. Port-Based VLAN traffic is routable on the switches covered by this guide.

Protocol-Based VLANs

This type of static VLAN creates a layer-3 broadcast domain for traffic of a particular protocol, and is comprised of member ports that bridge traffic of the specified protocol type among themselves. Some protocol types are routable on the switches covered by this manual. Refer to table 2-1 on page 2-7.

Designated VLANs

The switch uses these static, port-based VLAN types to separate switch management traffic from other network traffic. While these VLANs are not limited to management traffic only, they can provide improved security and availability for management traffic.

■ **The Default VLAN:** This port-based VLAN is always present in the switch and, in the default configuration, includes all ports as members (page 2-45).

■ **The Primary VLAN:** The switch uses this port-based VLAN to run certain features and management functions, including DHCP/Bootp responses for switch management. In the default configuration, the Default VLAN is also the Primary VLAN. However, you can designate another, port-based, non-default VLAN, as the Primary VLAN (page 2-45).

■ **The Secure Management VLAN:** This optional, port-based VLAN establishes an isolated network for managing the ProCurve switches that support this feature. Access to this VLAN and to the switch's management functions are available only through ports configured as members (page 2-46).

■ **Voice VLANs:** This optional, port-based VLAN type enables you to separate, prioritize, and authenticate voice traffic moving through your network, and to avoid the possibility of broadcast storms affecting VoIP (Voice-over-IP) operation (page 2-51).

N o t e   In a multiple-VLAN environment that includes some older switch models there may be problems related to the same MAC address appearing on different ports and VLANs on the same switch. In such cases the solution is to impose some cabling and VLAN restrictions. For more on this topic, refer to "Multiple VLAN Considerations" on page 2-18.

Terminology

Dynamic VLAN: An 802.1Q VLAN membership temporarily created on a port linked to another device, where both devices are running GVRP. (See also Static VLAN.) For more information, refer to chapter 3, "GVRP" .

Static VLAN: A port-based or protocol-based VLAN configured in switch memory. (**See also Dynamic VLAN**.)

Tagged Packet: A packet that carries an IEEE 802.1Q VLAN ID (VID), which is a two-byte extension that precedes the source MAC address field of an ethernet frame. A VLAN tag is layer 2 data and is transparent to higher layers.

Tagged VLAN: A VLAN that complies with the 802.1Q standard, including priority settings, and allows a port to join multiple VLANs. (See also **Untagged VLAN**.)

Untagged Packet: A packet that does not carry an IEEE 802.1Q VLAN ID (VID).

Untagged VLAN: A VLAN that does not use or forward 802.1Q VLAN tagging, including priority settings. A port can be a member of only one untagged VLAN of a given type (port-based and the various protocol-based types). (**See also Tagged VLAN**.)

VID: The acronym for a VLAN Identification Number. Each 802.1Q-compliant VLAN must have its own unique VID number, and that VLAN must be given the same VID in every device in which it is configured.

Static VLAN Operation

A group of networked ports assigned to a VLAN form a broadcast domain that is separate from other VLANs that may be configured on the switch. On a given switch, packets are bridged between source and destination ports that belong to the same VLAN. Thus, all ports passing traffic for a particular subnet address should be configured to the same VLAN. Cross-domain broadcast traffic in the switch is eliminated and bandwidth is saved by not allowing packets to flood out all ports.

Table 2-1. Comparative Operation of Port-Based and Protocol-Based VLANs

	Port-Based VLANs	Protocol-Based VLANs
IP Addressing	Usually configured with at least one unique IP address. You can create a port-based VLAN without an IP address. However, this limits the switch features available to ports on that VLAN. (Refer to "How IP Addressing Affects Switch Operation" in the chapter on configuring IP addressing in the ***Basic Management and Configuration Guide*** for the switch.) You can also use multiple IP addresses to create multiple subnets within the same VLAN. (For more on this topic, refer to the chapter on configuring IP addressing in the ***Basic Management and Configuration Guide*** for the switch.)	You can configure IP addresses on all protocol VLANs. However, IP addressing is used only on IPv4 and IPv6 protocol VLANs.
Untagged VLAN Membership	A port can be a member of one untagged, port-based VLAN. All other port-based VLAN assignments for that port must be tagged.	A port can be an untagged member of one protocol VLAN of a specific protocol type (such as IPX or IPv6). If the same protocol type is configured in multiple protocol VLANs, then a port can be an untagged member of only one of those protocol VLANs. For example, if you have two protocol VLANs, 100 and 200, and both include IPX, then a port can be an untagged member of either VLAN 100 or VLAN 200, but not both VLANs. A port's untagged VLAN memberships can include up to three different protocol types. This means that a port can be an untagged member of one of the following: • Three single-protocol VLANs • Two protocol VLANs where one VLAN includes a single protocol and the other includes two protocols • One protocol VLAN where the VLAN includes three protocols

	Port-Based VLANs	**Protocol-Based VLANs**
Tagged VLAN Membership	A port can be a tagged member of any port-based VLAN. See above.	A port can be a tagged member of any protocol-based VLAN. See above.
Routing	The switch can internally route IP (IPv4) traffic between port-based VLANs and between port-based and IPv4 protocol-based VLANs if the switch configuration enables IP routing. If the switch is not configured to route traffic internally between port-based VLANs, then an external router must be used to move traffic between VLANs.	If the switch configuration enables IP routing, the switch can internally route IPv4 traffic as follows: • Between multiple IPv4 protocol-based VLANs • Between IPv4 protocol-based VLANs and port-based VLANs. Other protocol-based VLANs require an external router for moving traffic between VLANs. **Note:** NETbeui, SNA, and DEClat are non-routable protocols. End stations intended to receive traffic in these protocols must be attached to the same physical network. **Note:** The Series 3400cl and Series 6400cl switches do not support SNA and DEClat protocol VLANs.
Commands for Configuring Static VLANs	vlan < *VID* > [tagged | untagged < [e] *port-list* >]	vlan < *VID* > protocol < ipx | ipv4 | ipv6 | arp | appletalk | sna | declat | netbeui > vlan < *VID* > [tagged | untagged < [e] *port-list* >]

VLAN Environments

You can configure different VLAN types in any combination. Note that the default VLAN will always be present. (For more on the default VLAN, refer to "VLAN Support and the Default VLAN" on page 2-45.)

Table 2-2. VLAN Environments

VLAN Environment	Elements
The default VLAN (port-based; VID of "1") Only	In the default VLAN configuration, all ports belong to VLAN 1 as untagged members. VLAN 1 is a port-based VLAN, for IPv4 traffic.
Multiple VLAN Environment	In addition to the default VLAN, the configuration can include one or more other port-based VLANs and one or more protocol VLANs. (The switches covered in this guide allow up to 256 VLANs of all types.) Using VLAN tagging, ports can belong to multiple VLANs of all types. Enabling routing on the switch enables the switch to route IPv4 traffic between port-based VLANs and between port-based VLANs and IPv4 protocol VLANs. Routing other types of traffic between VLANs requires an external router capable of processing the appropriate protocol(s).

VLAN Operation

The Default VLAN. In figure 2-1, all ports belong to the default VLAN, and devices connected to these ports are in the same broadcast domain. Except for an IP address and subnet, no configuration steps are needed.
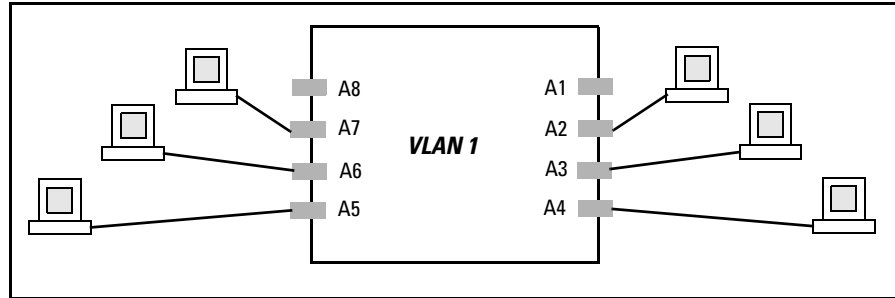
Figure 2-1. Example of a Switch in the Default VLAN Configuration

Multiple Port-Based VLANs. In figure 2-2, routing within the switch is disabled (the default). This means that communication between any routable VLANs on the switch must go through the external router. In this case, VLANs "W" and "X" can exchange traffic through the external router, but traffic in VLANs "Y" and "Z" is restricted to the respective VLANs. Note that VLAN 1, the default VLAN, is also present, but not shown. (The default VLAN cannot be deleted from the switch. However, ports assigned to other VLANs can be removed from the default VLAN, if desired.) If internal (IP) routing is enabled on the switch, then the external router is not needed for traffic to move between port-based VLANs.
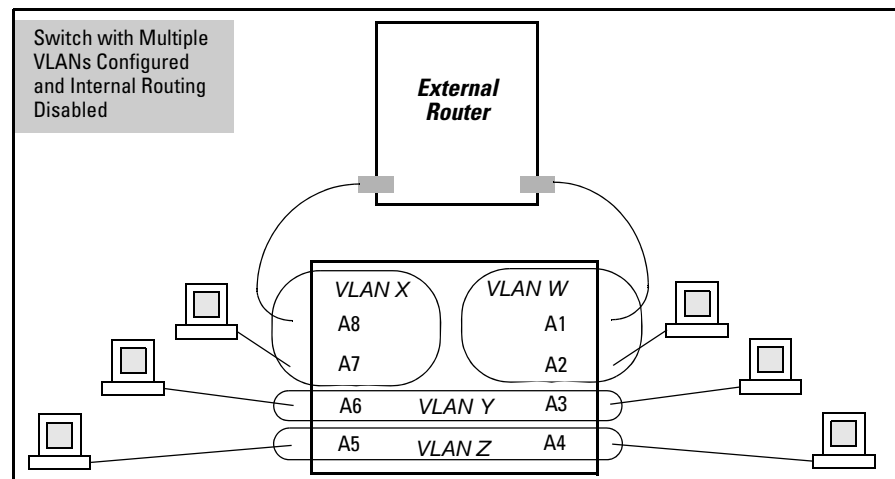
Figure 2-2. Example of Multiple VLANs on the Switch

Protocol VLAN Environment. Figure 2-2 can also be applied to a protocol VLAN environment. In this case, VLANs "W" and "X" represent routable protocol VLANs. VLANs "Y" and "Z" can be any protocol VLAN. As noted for the discussion of multiple port-based VLANs, VLAN 1 is not shown. Enabling internal (IP) routing on the switch allows IP traffic to move between VLANs on the switch. However, routable, non-IP traffic always requires an external router.

Routing Options for VLANs

Table 2-3. Options for Routing Between VLAN Types in the Switch

	Port-Based	IPX	IPv4	IPv6	ARP	Apple-Talk	SNA[2,3]	DEClat[2,3]	Netbeui[2]
Port-Based	Yes	—	Yes	—	—	—	—	—	—
Protocol									
IPX	—	Yes[1]	—	—	—	—	—	—	—
IPv4	Yes	—	Yes	—	—	—	—	—	—
IPv6	—	—	—	Yes[1]	—	—	—	—	—
ARP	—	—	—	—	Yes[1]	—	—	—	—
AppleTalk	—	—	—	—	—	Yes[1]	—	—	—
SNA[2,3]	—	—	—	—	—	—	—	—	—
DEClat[2,3]	—	—	—	—	—	—	—	—	—
NETbeui[2]	—	—	—	—	—	—	—	—	—

[1]Requires an external router to route between VLANs.

[2]Not a routable protocol type. End stations intended to receive traffic in these protocols must be attached to the same physical network.

[3] Protocol VLAN type not supported on the Series 3400cl and 6400cl switches.

Overlapping (Tagged) VLANs

A port can be a member of more than one VLAN of the same type if the device to which the port connects complies with the 802.1Q VLAN standard. For example, a port connected to a central server using a network interface card (NIC) that complies with the 802.1Q standard can be a member of multiple VLANs, allowing members of multiple VLANs to use the server. Although these VLANs cannot communicate with each other through the server, they can all access the server over the same connection from the switch. Where VLANs

overlap in this way, VLAN "tags" are used in the individual packets to distinguish between traffic from different VLANs. A VLAN tag includes the particular VLAN I.D. (VID) of the VLAN on which the packet was generated.
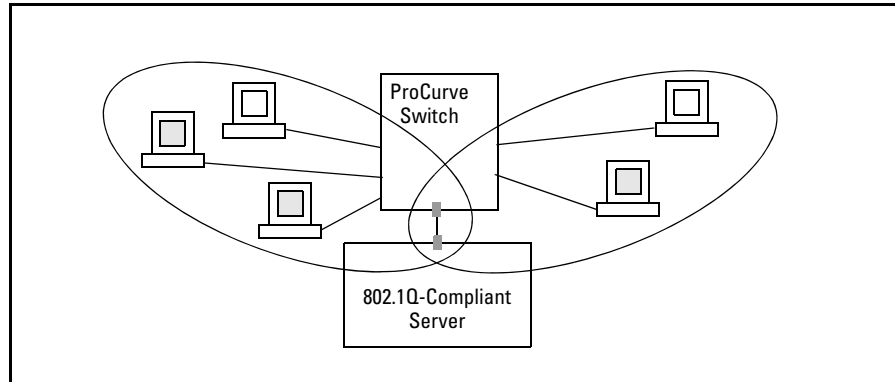
Figure 2-3. Example of Overlapping VLANs Using the Same Server

Similarly, using 802.1Q-compliant switches, you can connect multiple VLANs through a single switch-to-switch link.
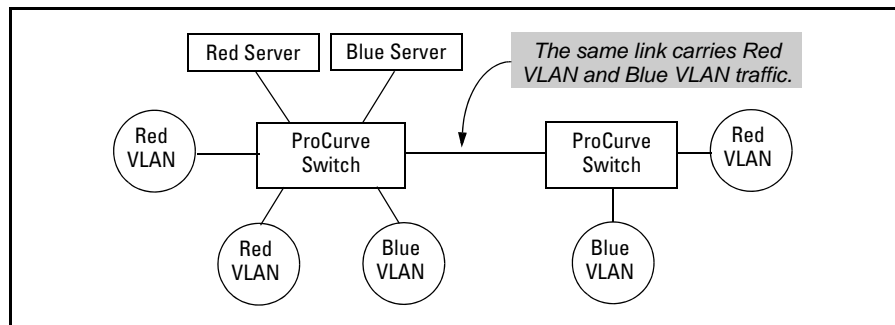
Figure 2-4. Example of Connecting Multiple VLANs Through the Same Link

Introducing Tagged VLAN Technology into Networks Running Legacy (Untagged) VLANs. You can introduce 802.1Q-compliant devices into networks that have built untagged VLANs based on earlier VLAN technology. The fundamental rule is that legacy/untagged VLANs require a separate link for each VLAN, while 802.1Q, or tagged VLANs can combine several VLANs in one link. This means that on the 802.1Q-compliant device, separate ports (configured as untagged) must be used to connect separate VLANs to non-802.1Q devices.

The legacy (non-802.1Q compliant) switch requires a separate link for each VLAN.

VLAN tagging enables the Link to carry Red VLAN and Blue VLAN Traffic
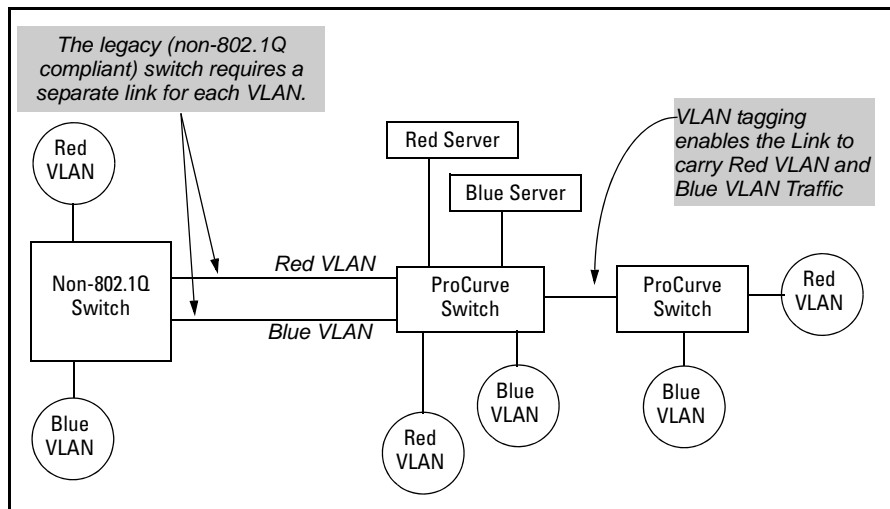
Figure 2-5. Example of Tagged and Untagged VLAN Technology in the Same Network

For more information on VLANs, refer to:

■ "Overview of Using VLANs" (page 2-45)

■ "Menu: Configuring VLAN Parameters (page 2-22)

■ "CLI: Configuring VLAN Parameters" (page 2-22)

■ "Web: Viewing and Configuring VLAN Parameters" (page 2-39)

■ "VLAN Tagging Information" (page 2-40)

■ "Effect of VLANs on Other Switch Features" (page 2-53)

■ "VLAN Restrictions" (page 2-55)

Per-Port Static VLAN Configuration Options

The following figure and table show the options you can use to assign individual ports to a static VLAN. Note that GVRP, if configured, affects these options and VLAN behavior on the switch. The display below shows the per-port VLAN configuration options. Table 2-4 briefly describes these options.

**Example of Per-Port
VLAN Configuration
with GVRP Disabled
(the default)**

**Example of Per-Port
VLAN Configuration
with GVRP Enabled**

```
Port    DEFAULT_VLAN    VLAN-22        Port    DEFAULT_VLAN    VLAN-22
---- + ------------   ---------        ---- + ------------   ---------
A1   | Untagged        Forbid          A1   | Untagged        Forbid
A2   | No              Tagged          A2   | Auto            Tagged
A3   | No              Tagged          A3   | Auto            Tagged
A4   | Forbid          Tagged          A4   | Forbid          Tagged
A5   | Untagged        No              A5   | Untagged        Auto
```

Enabling GVRP causes "No" to display as "Auto".

Figure 2-6. Comparing Per-Port VLAN Options With and Without GVRP
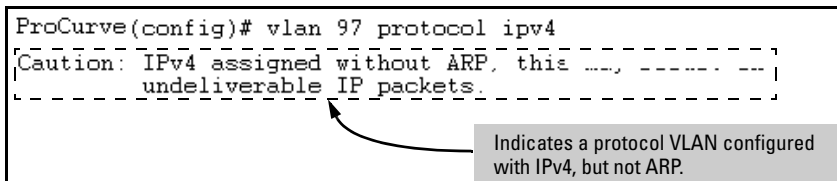
Table 2-4. Per-Port VLAN Configuration Options

Parameter	Effect on Port Participation in Designated VLAN
Tagged	Allows the port to join multiple VLANs.
Untagged	Allows VLAN connection to a device that is configured for an untagged VLAN instead of a tagged VLAN. A port can be an untagged member of only one port-based VLAN. A port can also be an untagged member of only one protocol-based VLAN for any given protocol type. For example, if the switch is configured with the default VLAN plus three protocol-based VLANs that include IPX, then port 1 can be an untagged member of the default VLAN and one of the protocol-based VLANS.
No - or - **Auto**	**No**: Appears when the switch is not GVRP-enabled; prevents the port from joining that VLAN. **Auto**: Appears when GVRP is enabled on the switch; allows the port to dynamically join any advertised VLAN that has the same VID
Forbid	Prevents the port from joining the VLAN, even if GVRP is enabled on the switch.

VLAN Operating Rules

■ **DHCP/Bootp:** If you are using DHCP/Bootp to acquire the switch's configuration, packet time-to-live, and TimeP information, you must designate the VLAN on which DHCP is configured for this purpose as the Primary VLAN. (In the factory-default configuration, the DEFAULT_VLAN is the Primary VLAN.)

■ **Per-VLAN Features:** IGMP and some other features operate on a "per VLAN" basis. This means you must configure such features separately for each VLAN in which you want them to operate.

■ **Default VLAN:** You can rename the default VLAN, but you cannot change its VID (1) or delete it from the switch.

■ **VLAN Port Assignments:** Any ports *not* specifically removed from the default VLAN remain in the DEFAULT_VLAN, regardless of other port assignments. Also, a port must always be a tagged or untagged member of at least one port-based VLAN.

■ **Voice-Over-IP (VoIP):** VoIP operates only over static, port-based VLANs.

■ **Multiple VLAN Types Configured on the Same Port:** A port can simultaneously belong to both port-based and protocol-based VLANs.

■ **Protocol Capacity:** A protocol-based VLAN can include up to three protocol types. In protocol VLANs using the IPv4 protocol, ARP must be one of these protocol types (to support normal IP network operation). Otherwise, IP traffic on the VLAN is disabled. If you configure an IPv4 protocol VLAN that does not already include the ARP VLAN protocol, the switch displays this message:

```
ProCurve(config)# vlan 97 protocol ipv4
Caution: IPv4 assigned without ARP, this ...., ...... ...
         undeliverable IP packets.
```

Indicates a protocol VLAN configured with IPv4, but not ARP.

■ **Deleting Static VLANs:** On the 3400cl and 6400cl switches, and on 5300xl switches running a software release earlier than E.09.xx, if one or more ports are assigned to a non-default VLAN, you cannot delete that VLAN from the switch configuration until you first remove the port(s) from the VLAN configuration. On 5300xl switches running software release E.09.*xx* or greater and on 4200vl switches, you can delete a VLAN regardless of whether there are currently any ports belonging to that VLAN. (The ports are moved to the default VLAN.)

■   **Adding or Deleting VLANs:** Changing the number of VLANs supported on the switch requires a reboot. (From the CLI, you must perform a **write memory** command before rebooting.) Other VLAN configuration changes are dynamic.

■   **Inbound Tagged Packets:** If a tagged packet arrives on a port that is not a tagged member of the VLAN indicated by the packet's VID, the switch drops the packet. Similarly, the switch will drop an inbound, tagged packet if the receiving port is an *untagged* member of the VLAN indicated by the packet's VID.

■   **Untagged Packet Forwarding:** To enable an inbound port to forward an untagged packet, the port must be an untagged member of either a protocol VLAN matching the packet's protocol or an untagged member of a port-based VLAN. That is, when a port receives an incoming, untagged packet, it processes the packet according to the following ordered criteria:

   a.   If the port has no untagged VLAN memberships, the switch drops the packet.

   b.   If the port has an untagged VLAN membership in a protocol VLAN that matches the protocol type of the incoming packet, then the switch forwards the packet on that VLAN.

   c.   If the port is a member of an untagged, port-based VLAN, the switch forwards the packet to that VLAN. Otherwise, the switch drops the packet.

```
┌─────────────────────────────────────────────────────────────────┐
│                      ┌──────────────────┐                         │
│                      │  Port "X" receives │                        │
│                      │  an inbound,       │                        │
│                      │  untagged Packet.  │                        │
│                      └──────────────────┘                         │
│                                                                    │
│                         ╱╲                                         │
│                        ╱  ╲          ┌──────────────┐              │
│                       ╱ Is the ╲  No │   Drop the    │             │
│                      ╱ port an untagged ╲──│   packet.     │        │
│                      ╲ member of any ╱     └──────────────┘         │
│                       ╲  VLANs?  ╱                                  │
│                        ╲      ╱                                     │
│                         ╲    ╱                                      │
│                          Yes                                       │
│                                                                    │
│                         ╱╲                                         │
│                        ╱  ╲                                        │
│                       ╱ Does the ╲                                 │
│                      ╱ packet's protocol ╲  Yes  ┌──────────────┐   │
│                     ╱ match the protocol of ╲────│  Forward the  │   │
│                     ╲ an untagged VLAN ╱         │  packet on that│  │
│                      ╲ membership on ╱           │ protocol VLAN. │  │
│                       ╲ the port? ╱              └──────────────┘   │
│                        ╲      ╱                                     │
│                          No                                        │
│                                                                    │
│                         ╱╲                                         │
│                        ╱  ╲                                        │
│                       ╱ Is the ╲        No  ┌──────────────┐       │
│                      ╱ port a member ╲──────│   Drop the    │      │
│                      ╲ of an untagged, ╱     │   packet.     │      │
│                       ╲ port-based ╱         └──────────────┘       │
│                        ╲ VLAN? ╱                                   │
│                          Yes                                       │
│                      ┌──────────────┐                             │
│                      │  Forward the  │                            │
│                      │  packet on the │                           │
│                      │ port-based VLAN.│                          │
│                      └──────────────┘                             │
└─────────────────────────────────────────────────────────────────┘
```
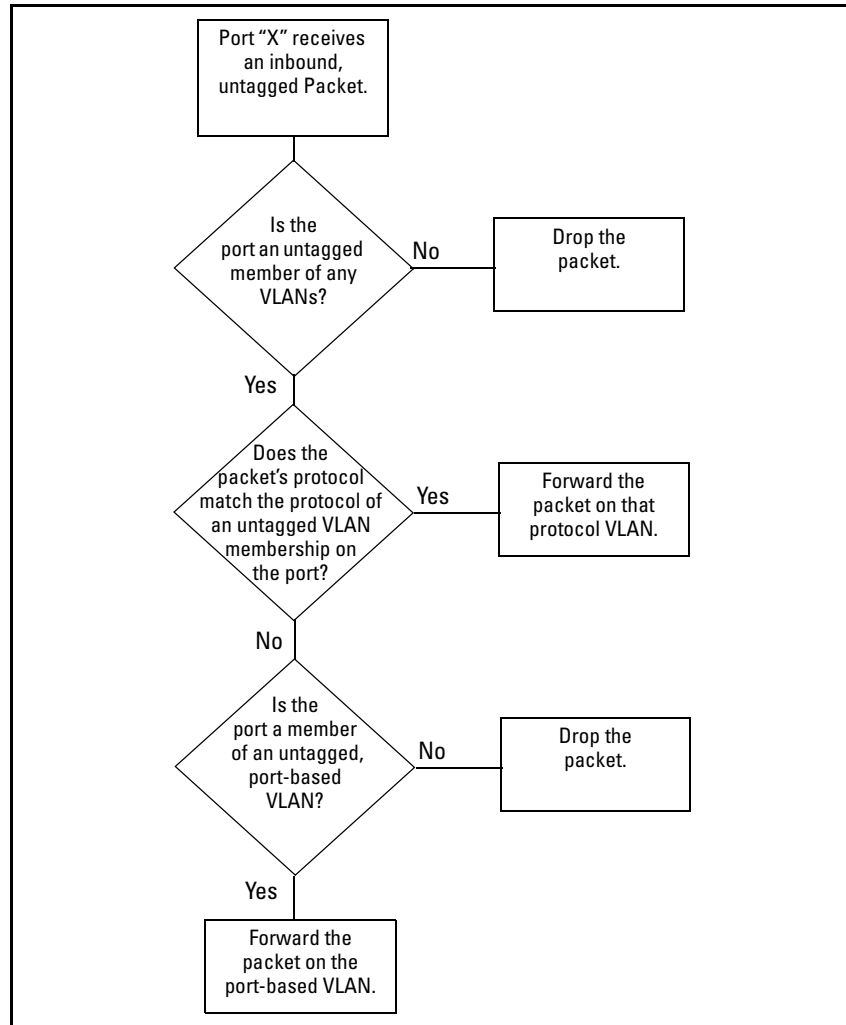
Figure 2-7. Untagged VLAN Operation

■ **Tagged Packet Forwarding:** If a port is a tagged member of the same VLAN as an inbound, tagged packet received on that port, then the switch forwards the packet to an outbound port on that VLAN. (To enable the forwarding of tagged packets, any VLAN to which the port belongs as a

tagged member must have the same VID as that carried by the inbound, tagged packets generated on that VLAN.)
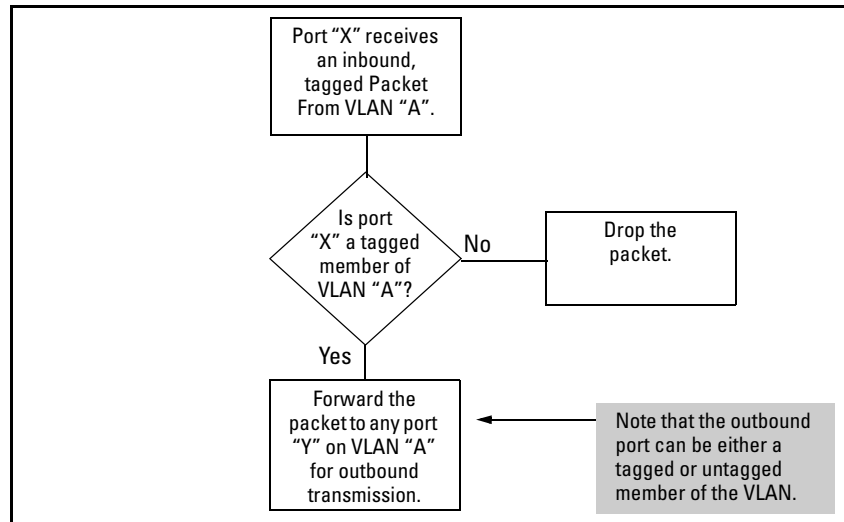
Figure 2-8. Tagged VLAN Operation

See also "Multiple VLAN Considerations" on page 2-18.

General Steps for Using VLANs

1. Plan your VLAN strategy and create a map of the logical topology that will result from configuring VLANs. Include consideration for the interaction between VLANs and other features such as Spanning Tree Protocol, port trunking, and IGMP. (Refer to "Effect of VLANs on Other Switch Features" on page 2-53.) If you plan on using dynamic VLANs, include the port configuration planning necessary to support this feature. (Refer to chapter 3, "GVRP" .)

   By default, VLAN support is enabled and the switch is configured for eight VLANs.

2. Configure at least one VLAN in addition to the default VLAN.

3. Assign the desired switch ports to the new VLAN(s).

4. If you are managing VLANs with SNMP in an IP network, the VLAN through which you are managing the switch must have an IP address. Refer to chapter 7, "Configuring IP Addressing", in the *Management and Configuration Guide* for your switch.

Multiple VLAN Considerations

Switches use a *forwarding database* to maintain awareness of which external devices are located on which VLANs. Some switches, such as those covered by this guide, have a *multiple forwarding database*, which means the switch allows multiple database entries of the same MAC address, with each entry showing the (different) source VLAN and source port. Other switch models have a *single forwarding database*, which means they allow only one database entry of a unique MAC address, along with the source VLAN and source port on which it is found. All VLANs on a switch covered by this guide use the same MAC address. Thus, connecting a Series 5300XL, 4200vl, 3400cl, or 6400cl (multiple forwarding database) switch to a single forwarding database switch where multiple VLANs exist imposes some cabling and port VLAN assignment restrictions. Table 2-5 illustrates the functional difference between the two database types.

Table 2-5. Example of Forwarding Database Content

Multiple Forwarding Database			Single Forwarding Database		
MAC Address	Destination VLAN ID	Destination Port	MAC Address	Destination VLAN ID	Destination Port
0004ea-84d9f4	1	A5	0004ea-84d9f4	100	A9
0004ea-84d9f4	22	A12	0060b0-880af9	105	A10
0004ea-84d9f4	44	A20	0060b0-880a81	107	A17
0060b0-880a81	33	A20			

This database allows multiple destinations for the same MAC address. If the switch detects a new destination for an existing MAC entry, it just **adds** a new instance of that MAC to the table.

This database allows only one destination for a MAC address. If the switch detects a new destination for an existing MAC entry, it **replaces** the existing MAC instance with a new instance showing the new destination.

Table 2-6 lists the database structure of current ProCurve switch models.

Table 2-6. Forwarding Database Structure for Managed ProCurve Switches

Multiple Forwarding Databases*	Single Forwarding Database*
Series 6400cl switches	Switch 1600M/2400M/2424M
Series 6200yl switches	Switch 4000M/8000M
Switch 6108	Series 2500 switches
Series 5400zl switches	Switch 800T
Series 5300xl switches	Switch 2000
Series 4200vl switches	
Series 4100gl switches	
Series 3500yl switches	
Series 3400cl switches	
Series 2900yl switches	
Switch 2810	
Series 2800 switches	
Series 2600/2600-PWR switches	
Series 2510 switches	
Series 1800 switches	

*To determine whether other vendors' devices use single-forwarding or multiple-forwarding database architectures, refer to the documentation provided for those devices.

Single Forwarding Database Operation

When a packet arrives with a destination MAC address that matches a MAC address in the switch's forwarding table, the switch tries to send the packet to the port listed for that MAC address. But, if the destination port is in a different VLAN than the VLAN on which the packet was received, the switch drops the packet. This is not a problem for a switch with a multiple forwarding database (refer to table 2-6, above) because the switch allows multiple instances of a given MAC address; one for each valid destination. However, a switch with a single forwarding database allows only one instance of a given MAC address. If (1) you connect the two types of switches through multiple ports or trunks belonging to different VLANs, and (2) enable routing on the switch having the multiple forwarding database; then, on the switch having the single forwarding database, the port and VLAN record it maintains for the connected multiple-forwarding-database switch can frequently change. This causes poor performance and the appearance of an intermittent or broken connection.

Example of an Unsupported Configuration and How To Correct It

The Problem. In figure 2-9, the MAC address table for Switch 8000M will sometimes record the 5300xl, 4200vl, 3400cl, or 6400cl as accessed on port A1 (VLAN 1), and other times as accessed on port B1 (VLAN 2):
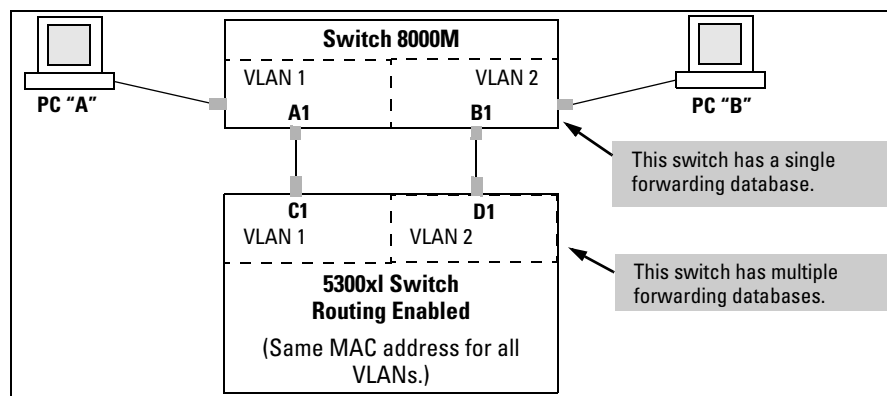
Figure 2-9. Example of Invalid Configuration for Single-Forwarding to Multiple-Forwarding Database Devices in a Multiple VLAN Environment

In figure 2-9, PC "A" sends an IP packet to PC "B".

1. The packet enters VLAN 1 in the Switch 8000 with the 5300xl's MAC address in the destination field. Because the 8000M has not yet learned this MAC address, it does not find the address in its address table, and floods the packet out all ports, including the VLAN 1 link (port "A1") to the 5300xl. The 5300xl then routes the packet through the VLAN 2 link to the 8000M, which forwards the packet on to PC "B". Because the 8000M received the packet from the 5300xl on VLAN 2 (port "B1"), the 8000M's single forwarding database records the 5300xl as being on port "B1" (VLAN 2).

2. PC "A" now sends a second packet to PC "B". The packet again enters VLAN 1 in the Switch 8000 with the 5300xl's MAC address in the destination field. However, this time the Switch 8000M's single forwarding database indicates that the 5300xl is on port B1 (VLAN 2), and the 8000M drops the packet instead of forwarding it.

3. Later, the 5300xl transmits a packet to the 8000M through the VLAN 1 link, and the 8000M updates its address table to indicate that the 5300xl is on port A1 (VLAN 1) instead of port B1 (VLAN 2). Thus, the 8000M's information on the location of the 5300xl changes over time. For this reason,

the 8000M discards some packets directed through it for the 5300xl, resulting in poor performance and the appearance of an intermittent or broken link.

The Solution. To avoid the preceding problem, use only one cable or port trunk between the single-forwarding and multiple-forwarding database devices, and configure the link with multiple, tagged VLANs.
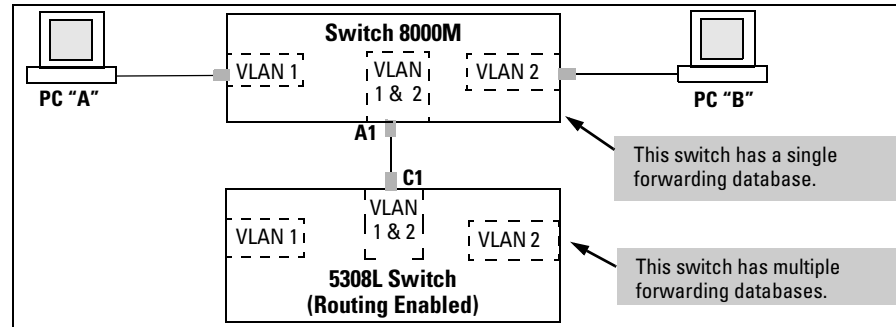
Figure 2-10. Example of a Solution for Single-Forwarding to Multiple-Forwarding Database Devices in a Multiple VLAN Environment

Now, the 8000M forwarding database always lists the 5300xl MAC address on port A1, and the 8000M will send traffic to either VLAN on the 5300xl.

To increase the network bandwidth of the connection between the devices, you can use a trunk of multiple physical links rather than a single physical link.

Multiple Forwarding Database Operation

If you want to connect a switch covered by this guide to another switch that has a multiple forwarding database, you can use either or both of the following connection options:

■ A separate port or port trunk interface for each VLAN. This results in a forwarding database having multiple instances of the same MAC address with different VLAN IDs and port numbers. (See table 2-5.) The fact that the switches covered by this guide use the same MAC address on all VLAN interfaces causes no problems.

■ The same port or port trunk interface for multiple (tagged) VLANs. This results in a forwarding database having multiple instances of the same MAC address with different VLAN IDs, but the same port number.

Allowing multiple entries of the same MAC address on different VLANs enables topologies such as the following:
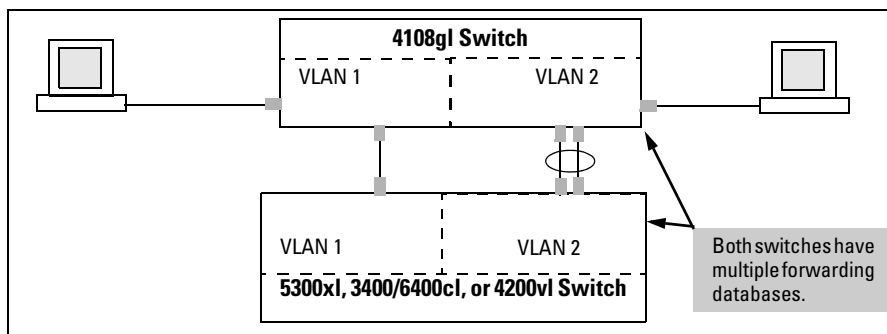
Figure 2-11. Example of a Valid Topology for Devices Having Multiple Forwarding Databases in a Multiple VLAN Environment

Configuring VLANs

Menu: Configuring Port-Based VLAN Parameters

The Menu interface enables you to configure and view port-based VLANs.

N o t e   The Menu interface configures and displays only port-based VLANs. The CLI configures and displays port-based *and* protocol-based VLANs (page 2-28).

In the factory default state, support is enabled for up to eight VLANs. (You can reconfigure the switch to support up to 256 VLANs.) Also, in the default configuration, all ports on the switch belong to the default VLAN and are in the same broadcast/multicast domain. (The default VLAN is also the default Primary VLAN—refer to "The Primary VLAN" on page 2-45.) In addition to the default VLAN, you can configure additional static VLANs by adding new VLAN names and VIDs, and then assigning one or more ports to each VLAN. (The maximum of 256 VLANs includes the default VLAN, all additional static VLANs you configure, and any dynamic VLANs the switch creates if you enable GVRP—page 3-1.) Note that each port can be assigned to multiple VLANs by using VLAN tagging. (See "802.1Q VLAN Tagging" on page 2-40.)

To Change VLAN Support Settings

This section describes:

- Changing the maximum number of VLANs to support
- Changing the Primary VLAN selection (See "Changing the Primary VLAN" on page 2-34.)
- Enabling or disabling dynamic VLANs (Refer to chapter 3, "GVRP" .)

1. From the Main Menu select:

   2. Switch Configuration

       8. VLAN Menu …

           1. VLAN Support

   You will then see the following screen:

```
=========================- CONSOLE - MANAGER MODE -=============================
                    Switch Configuration - VLAN - VLAN Support

  Maximum VLANs to support [8] : 8
  Primary VLAN : DEFAULT_VLAN
  GVRP Enabled [No] : No


  Actions->   Cancel     Edit     Save     Help

Cancel changes and return to previous screen.
Use arrow keys to change action selection and <Enter> to execute action.
```

Figure 2-12. The Default VLAN Support Screen

2. Press **[E]** (for **E**dit), then do one or more of the following:

   - To change the maximum number of VLANs, type the new number (1 - 256 allowed; default 8).
   - To designate a different VLAN as the Primary VLAN, select the **Primary VLAN** field and use the space bar to select from the existing options. (Note that the Primary VLAN must be a static, port-based VLAN.)
   - To enable or disable dynamic VLANs, select the **GVRP Enabled** field and use the Space bar to toggle between options. (For GVRP information, refer to chapter 3, "GVRP" .)

N o t e   For optimal switch memory utilization, set the number of VLANs at the number you will likely be using or a few more. If you need more VLANs later, you can increase this number, but a switch reboot will be required at that time.

3. Press **[Enter]** and then **[S]** to save the VLAN support configuration and return to the VLAN Menu screen.

If you changed the value for **Maximum VLANs to support**, you will see an asterisk next to the **VLAN Support** option (see below).

An asterisk indicates you must reboot the switch to implement the new Maximum VLANs setting.

```
=========================- CONSOLE - MANAGER MODE -=========================
                      Switch Configuration - VLAN Menu

 *1. VLAN Support
  2. VLAN Names
  3. VLAN Port Assignment
  4. Return to Previous Menu...
  0. Return to Main Menu...


Displays the menu to activate and configure, or deactivate VLAN support.
To select menu item, press item number, or highlight item and press <Enter>.
(*Needs reboot to activate changes.)
```

Figure 2-13. VLAN Menu Screen Indicating the Need To Reboot the Switch

- If you changed the VLAN Support option, you must reboot the switch before the Maximum VLANs change can take effect. You can go on to configure other VLAN parameters first, but remember to reboot the switch when you are finished.

- If you did not change the VLAN Support option, a reboot is not necessary.

4. Press **[0]** to return to the Main Menu.

Adding or Editing VLAN Names

Use this procedure to add a new VLAN or to edit the name of an existing VLAN.

1. From the Main Menu select:

   2. Switch Configuration
       8. VLAN Menu …
           2. VLAN Names

   If multiple VLANs are not yet configured you will see a screen similar to figure 2-14:

```
==========================- CONSOLE - MANAGER MODE -==========================
                    Switch Configuration - VLAN - VLAN Names

    802.1Q VLAN ID      Name                        Default VLAN
    --------------    ------------                  and VLAN ID
    1                 DEFAULT_VLAN


  Actions->   Back     Add     Edit     Delete     Help
 Delete highlighted record.
 Use up/down arrow keys to change record selection, left/right arrow keys to
 change action selection, and <Enter> to execute action.
```

Figure 2-14. The Default VLAN Names Screen

2. Press **[A]** (for **Add**). You will then be prompted for a new VLAN name and VLAN ID:

   802.1Q VLAN ID : 1
   Name : _

3. Type in a VID (VLAN ID number). This can be any number from 2 to 4094 that is not already being used by another VLAN. (The switch reserves "1" for the default VLAN.)

   Remember that a VLAN *must* have the same VID in every switch in which you configure that same VLAN. (GVRP dynamically extends VLANs with correct VID numbering to other switches. Refer to chapter 3, "GVRP" .)

4. Press ⬇ to move the cursor to the **Name** line and type the VLAN name (up to 12 characters, with no spaces) of a new VLAN that you want to add, then press **[Enter]**.
   (Avoid these characters in VLAN names: **2, #, $, ^, &, *, (**, and **)**.)

5. Press **[S]** (for **Save**). You will then see the VLAN Names screen with the new VLAN listed.

```
==========================- CONSOLE - MANAGER MODE -==========================
                  Switch Configuration - VLAN - VLAN Names

    802.1Q VLAN ID     Name
    --------------     -----------
    1                  DEFAULT VLAN
    22                 VLAN-22            Example of a New
                                         VLAN and ID

    Actions->   Back    Add     Edit     Delete     Help

 Add a new record.
 Use up/down arrow keys to change record selection, left/right arrow keys to
 change action selection, and <Enter> to execute action.
```
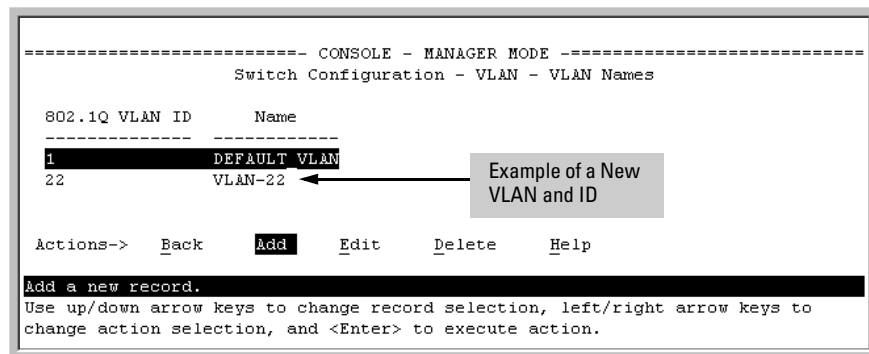
Figure 2-15. Example of VLAN Names Screen with a New VLAN Added

6. Repeat steps 2 through 5 to add more VLANs.

    Remember that you can add VLANs until you reach the number specified
    in the **Maximum VLANs to support** field on the VLAN Support screen (see
    figure 2-12 on page 2-23). This includes any VLANs added dynamically due
    to GVRP operation.

7. Return to the VLAN Menu to assign ports to the new VLAN(s) as described
    in the next section, "Adding or Changing a VLAN Port Assignment".

Adding or Changing a VLAN Port Assignment

Use this procedure to add ports to a VLAN or to change the VLAN assign-
ment(s) for any port. (Ports not specifically assigned to a VLAN are automat-
ically in the default VLAN.)

1. From the Main Menu select:

    2. Switch Configuration

        8. VLAN Menu …

            3. VLAN Port Assignment

    You will then see a VLAN Port Assignment screen similar to the following:

N o t e   The "VLAN Port Assignment" screen displays up to 32 static, port-based
VLANs in ascending order, by VID. If the switch configuration includes more
than 32 such VLANs, use the CLI **show vlans [VID | ports < *port-list* >]** command
to list data on VLANs having VIDs numbered sequentially higher than the first
32.

A port can be assigned to several VLANs, but only one of those assignments can be "Untagged".

```
=========================- CONSOLE - MANAGER MODE -=============================
                 Switch Configuration - VLAN - VLAN Port Assignment

   Port   DEFAULT_VLAN     VLAN-22      |   Port   DEFAULT_VLAN     VLAN-22
   ---- + ------------   ------------   |   ---- + ------------   ------------
   A1   | Untagged        No            |   A8   | Untagged        No
   A2   | Tagged          No            |   A9   | Untagged        No
   A3   | Untagged        No            |   A10  | Untagged        No
   A4   | Untagged        No            |   A11  | Untagged        No
   A5   | Untagged        No            |   A12  | Untagged        No
   A6   | Untagged        No            |   A13  | Untagged        No
   A7   | Untagged        No            |   A14  | Untagged        No


   Actions->   Cancel      Edit     Save      Help

 Cancel changes and return to previous screen.
 Use arrow keys to change action selection and <Enter> to execute action.
```

Figure 2-16. Example of the Port-Based VLAN Port Assignment Screen in the Menu Interface

2. To change a port's VLAN assignment(s):

    a. Press **[E]** (for **Edit**).

    b. Use the arrow keys to select a VLAN assignment you want to change.

    c. Press the Space bar to make your assignment selection (**No**, **Tagged**, **Untagged**, or **Forbid**).

N o t e

For GVRP Operation: If you enable GVRP on the switch, "**No**" converts to "**Auto**", which allows the VLAN to dynamically join an advertised VLAN that has the same VID. See "Per-Port Options for Dynamic VLAN Advertising and Joining" on page 3-9.

Untagged VLANs: Only one untagged VLAN is allowed per port. Also, there must be at least one VLAN assigned to each port. In the factory default configuration, all ports are assigned to the default VLAN (DEFAULT_VLAN).

For example, if you want ports A4 and A5 to belong to both DEFAULT_VLAN and VLAN-22, and ports A6 and A7 to belong only to VLAN-22, you would use the settings in figure page 2-28. (This example assumes the default GVRP setting—disabled—and that you do not plan to enable GVRP later.)

```
========================- CONSOLE - MANAGER MODE -===========================
               Switch Configuration - VLAN - VLAN Port Assignment

     Port    DEFAULT_VLAN     VLAN-22     |   Port    DEFAULT_VLAN     VLAN-22
     ---- + ------------    ------------  |   ---- + ------------    ------------
     A1   | Untagged        No            |   A8   | Untagged        No
     A2   | Untagged        No            |   A9   | Untagged        No
     A3   | Untagged        No            |   A10  | Untagged        No
     A4   | Untagged        Tagged        |   A11  | Untagged        No
     A5   | Untagged        Tagged        |   A12  | Untagged        No
     A6   | No              Untagged      |   A13  | Untagged        No
     A7   | No              Untagged      |   A14  | Untagged        No


     Actions->   Cancel       Edit      Save      Help

     Select the tagging mode for the port/VLAN combination.
     Use arrow keys to change field selection, <Space> to toggle field choices,
     and <Enter> to go to Actions.
```

Ports A4 and A5 are assigned to both VLANs.

Ports A6 and A7 are assigned only to VLAN-22.

All other ports are assigned only to the Default VLAN.

Figure 2-17. Example of Port-Based VLAN Assignments for Specific Ports

For information on VLAN tags ("Untagged" and "Tagged"), refer to "802.1Q VLAN Tagging" on page 2-40.

d. If you are finished assigning ports to VLANs, press **[Enter]** and then **[S]** (for **Save**) to activate the changes you've made and to return to the Configuration menu. (The console then returns to the VLAN menu.)

3. Return to the Main menu.

CLI: Configuring Port-Based and Protocol-Based VLAN Parameters

In the factory default state, all ports on the switch belong to the (port-based) default VLAN (DEFAULT_VLAN; VID = 1) and are in the same broadcast/ multicast domain. (The default VLAN is also the Primary VLAN. For more on this topic, refer to "The Primary VLAN" on page 2-45.) You can configure up to 255 additional static VLANs by adding new VLAN names, and then assigning one or more ports to each VLAN. (The switch accepts a maximum of 256 VLANs, including the default VLAN and any dynamic VLANs the switch creates if you enable GVRP. Refer to chapter 3, "GVRP" .) Note that each port can be assigned to multiple VLANs by using VLAN tagging. (See "802.1Q VLAN Tagging" on page 2-40.)

VLAN Commands	Page
show vlans	below
show vlans < *vid* >	2-31
show vlans ports <*port-list*>	
max-vlans <1-256>	2-33
primary-vlan < *vid* >	2-34
[no] vlan < *vid* >	2-35
auto < *port-list* >	2-37 (Available if GVRP enabled.)
forbid	2-37
name < *vlan-name* >	2-37
protocol < *protocol-list* >	2-35
tagged < *port-list* >	2-37
untagged < *port-list* >	2-37
voice	2-51
static-vlan < *vlan-id* >	2-37 (Available if GVRP enabled.)

Displaying the Switch's VLAN Configuration. The **show vlans** command lists the VLANs currently running in the switch, with VID, VLAN name, and VLAN status. Dynamic VLANs appear only if the switch is running with GVRP enabled and one or more ports has dynamically joined an advertised VLAN. (In the default configuration, GVRP is disabled. (Refer to chapter 3, "GVRP" .)

Syntax: show vlans

> **Maximum VLANs to support:** *Shows the number of VLANs the switch can currently support. (Default: 8; Maximum: 256)*
>
> **Primary VLAN:** *Refer to "The Primary VLAN" on page 2-45.*
>
> **Management VLAN:** *Refer to "The Secure Management VLAN" on page 2-46.*
>
> **802.1Q VLAN ID:** *The VLAN identification number, or VID. Refer to "Terminology" on page 2-6.*

Name: *The default or specified name assigned to the VLAN. For a static VLAN, the default name consists of* **VLAN-x** *where "x" matches the VID assigned to that VLAN. For a dynamic VLAN, the name consists of* **GVRP_x** *where "x" matches the applicable VID.*

Status:

> **Port-Based***: Port-Based, static VLAN*
>
> **Protocol:** *Protocol-Based, static VLAN*
>
> **Dynamic:** *Port-Based, temporary VLAN learned through GVRP (Refer to chapter 3, "GVRP" .)*

Voice: *Indicates whether a (port-based) VLAN is configured as a voice VLAN. Refer to "Voice VLANs" on page 2-51.*

Jumbo: *Indicates whether a VLAN is configured for Jumbo packets. For more on jumbos, refer to the chapter titled "Port Traffic Controls" in the* Management and Configuration Guide *for your switch.*

For example:

```
ProCurve # show vlans

 Status and Counters - VLAN Information

  Maximum VLANs to support : 8
  Primary VLAN : DEFAULT_VLAN
  Management VLAN :

  802.1Q VLAN ID Name       | Status      Voice  Jumbo
  -------------- ------------+ ----------  -----  -----
  1              DEFAULT_VLAN | Port-based No     No
  10             VLAN_10      | Port-based Yes    Yes
  15             VLAN_15      | Port-based No     No
  20             VLAN_20      | Protocol   No     No
  33             GVRP_33      | Dynamic    No     No
```

> When GVRP is disabled (the default), Dynamic VLANs do not exist on the switch and do not appear in this listing. (Refer to chapter 3, "GVRP" .)

Figure 2-18. Example of "Show VLAN" Listing (GVRP Enabled)

Displaying the VLAN Membership of One or More Ports.

This command shows to which VLAN a port belongs.

Syntax: show vlan ports < port-list >

> **802.1Q VLAN ID:** *The VLAN identification number, or VID. Refer to "Terminology" on page 2-6.*

> **Name:** *The default or specified name assigned to the VLAN. For a static VLAN, the default name consists of **VLAN-x** where "**x**" matches the VID assigned to that VLAN. For a dynamic VLAN, the name consists of **GVRP_x** where "**x**" matches the applicable VID.*

> **Status:**

>> **Port-Based***: Port-Based, static VLAN*

>> **Protocol:** *Protocol-Based, static VLAN*

>> **Dynamic:** *Port-Based, temporary VLAN learned through GVRP (Refer to chapter 3, "GVRP".)*

> **Voice:** *Indicates whether a (port-based) VLAN is configured as a voice VLAN. Refer to "Voice VLANs" on page 2-51.*

For example:

```
ProCurve Switch 4204vl# show vlan ports a1-a33

Status and Counters - VLAN Information  - for ports
a1-a33

 802.1Q VLAN ID Name          | Status      Voice
 -------------- ------------- + ---------- -----
 1              DEFAULT_VLAN  | Port-based No
 10             VLAN_10       | Port-based Yes
 15             VLAN 15       | Port-based No
 20             VLAN_20       | Protocol   No
 33             GVRP_33       | Dynamic    No
```

Figure 2-19. Example of "Show VLAN Ports" listing

Displaying the Configuration for a Particular VLAN . This command uses the VID to identify and display the data for a specific static or dynamic VLAN.

Syntax: show vlans < *vlan-id* >

> **802.1Q VLAN ID:** *The VLAN identification number, or VID. Refer to "Terminology" on page 2-6.*
>
> **Name:** *The default or specified name assigned to the VLAN. For a static VLAN, the default name consists of* **VLAN-*x*** *where "*x*" matches the VID assigned to that VLAN. For a dynamic VLAN, the name consists of* **GVRP_*x*** *where "*x*" matches the applicable VID.*
>
> **Status:**
>
> > **Port-Based***: Port-Based, static VLAN*
> >
> > **Protocol:** *Protocol-Based, static VLAN*
> >
> > **Dynamic:** *Port-Based, temporary VLAN learned through GVRP (Refer to the chapter titled "GVRP" in the Advanced Traffic Management Guide for your switch.)*
>
> **Voice:** *Indicates whether a (port-based) VLAN is configured as a voice VLAN. Refer to "Voice VLANs" on page 2-51.*
>
> **Jumbo:** *Indicates whether a VLAN is configured for Jumbo packets. For more on jumbo packets, refer to the chapter titled "Port Traffic Controls" in the* Management and Configuration Guide *for your switch.*
>
> **Port Information:** *Lists the ports configured as members of the VLAN.*
>
> **DEFAULT:** *Shows whether a port is a tagged or untagged member of the listed VLAN.*
>
> **Unknown VLAN:** *Shows whether the port can become a dynamic member of an unknown VLAN for which it receives an advertisement. GVRP must be enabled to allow dynamic joining to occur. Refer to table 3-1 on page 3-8.*
>
> **Status:** *Shows whether the port is participating in an active link.*

```
ProCurve(config)# show vlans 22
 Status and Counters - VLAN Information - Ports - VLAN 22
  802.1Q VLAN ID : 22
  Name : VLAN22
  Status : Port-based
  Voice : Yes
  Jumbo : No

  Port Information Mode      Unknown VLAN Status
  ---------------- --------  ------------ ----------
  A12              Untagged  Learn        Up
  A13              Untagged  Learn        Up
  A14              Untagged  Learn        Up
  A15              Untagged  Learn        Down
  A16              Untagged  Learn        Up
  A17              Untagged  Learn        Up
  A18              Untagged  Learn        Up
```

Figure 2-20. Example of "Show VLAN" for a Specific Static VLAN

Show VLAN lists this data when GVRP is enabled and at least one port on the switch has dynamically joined the designated VLAN.

```
ProCurve# show vlans 33
 Status and Counters - VLAN Information - Ports - VLAN 33

  802.1Q VLAN ID : 33
  Name : GVRP_33
  Status : Dynamic
  Voice : No
  Jumbo : No

  Port Information DEFAULT  Unknown VLAN Status
  ---------------- --------  ------------ ----------
  A6               Auto      Learn        Up
```

Figure 2-21. Example of "Show VLAN" for a Specific Dynamic VLAN

Changing the Number of VLANs Allowed on the Switch. In the default VLAN configuration, the switch allows a maximum of 8 VLANs. You can specify any value from 1 to 256.

Syntax: max-vlans < 1-256 >

> *Specifies the maximum number of VLANs to allow. (If GVRP is enabled, this setting includes any dynamic VLANs on the switch.) As part of implementing a new setting, you must execute a* **write memory** *command (to save the new value to the startup-config file) and then reboot the switch.*

> *Note: If multiple VLANs exist on the switch, you cannot reset the maximum number of VLANs to a value smaller than the current number of VLANs.*

For example, to reconfigure the switch to allow 10 VLANs:

Note that you can
execute these
three steps at
another time.

```
ProCurve(config)# max-vlans 10
Command will take effect after saving configuration and reboot.
ProCurve(config)# write memory
ProCurve(config)# boot
Device will be rebooted, do you want to continue [y/n]? y
```

Figure 2-22. Example of Command Sequence for Changing the Number of VLANs

Changing the Primary VLAN. In the default VLAN configuration, the port-
based default VLAN (**DEFAULT_VLAN**) is the Primary VLAN. However, you can
reassign the Primary VLAN to any port-based, static VLAN on the switch. (For
more on the Primary VLAN, refer to "The Primary VLAN" on page 2-45.) To
identify the current Primary VLAN and list the available VLANs and their
respective VIDs, use **show vlans**.

Syntax: primary-vlan < *vid* / *ascii-name-string* >

> *Reassigns the Primary VLAN function. Re-assignment must be*
> *to an existing, port-based, static VLAN. (The switch will not*
> *reassign the Primary VLAN function to a protocol VLAN.) If you*
> *re-assign the Primary VLAN to a non-default VLAN, you cannot*
> *later delete that VLAN from the switch until you again re-assign*
> *the Primary VLAN to another port-based, static VLAN.*

For example, if you wanted to reassign the Primary VLAN to VLAN 22 and
rename the VLAN with "22-Primary" and display the result:

Reassigns the
Primary VLAN to
VLAN 22.

Renames VLAN 22 to
"22-Primary".

```
ProCurve(config)# primary-vlan 22
ProCurve(config)# vlan 22 name 22-Primary
ProCurve(config)# show vlans

 Status and Counters - VLAN Information

  Maximum VLANs to support : 8
  Primary VLAN : 22-Primary
  Management VLAN :

  802.1Q VLAN ID Name           Status      Voice  Jumbo
  -------------- -------------- ---------- ----- -----
  1              DEFAULT_VLAN   Static      No     No
  22             22-Primary     Static      No     No
```

Figure 2-23. Example of Reassigning Primary VLAN and Changing the VLAN Name

Creating a New Static VLAN (Port-Based or Protocol-Based)
Changing the VLAN Context Level. The **vlan** *< vid >* command operates in
the global configuration context to either configure a static VLAN and/or take
the CLI to the specified VLAN's context.

Syntax: vlan < *vid* / *ascii-name-string* >

[no] vlan < *vid* >

> *If < vid > does not exist in the switch, this command creates a*
> *port-based VLAN with the specified < vid >. If the command*
> *does not include options, the CLI moves to the newly created*
> *VLAN context. If you do not specify an optional name, the*
> *switch assigns a name in the default format:* **VLANn** *where* **n**
> *is the < vid > assigned to the VLAN. If the VLAN already exists*
> *and you enter either the* **vid** *or the* **ascii-name-string***, the CLI*
> *moves to the specified VLAN's context.*
> *The* **[no]** *form of the command deletes the VLAN as follows:*
>
> • ***3400cl and 6400cl Switches:*** *If no ports are members or*
>   *if the member ports also belong to another VLAN, this*
>   *command deletes the VLAN. If one or more ports belong only*
>   *to this VLAN, then the CLI prompts you to remove the ports*
>   *from the VLAN before deleting it.*
>
> • ***5300xl Switches with Pre-E.09.xx Software:*** *Same as*
>   *for the 3400cl and 6400cl switches.*
>
> • ***5300xl Switches with E.09.xx or Greater Software and***
>   ***4200vl Switches:*** *If one or more ports belong only to the*
>   *VLAN to be deleted, the CLI notifies you that these ports will*
>   *be moved to the default VLAN and prompts you to continue*
>   *the deletion. For member ports that also belong to another*
>   *VLAN, there is no "move" prompt.*

[protocol < ipx | ipv4 | ipv6 | arp | appletalk | sna | declat | netbeui >]

> *Configures a static, protocol VLAN of the specified type. If*
> *multiple protocols are configured in the VLAN, then the* **[no]**
> *form removes the specified protocol from the VLAN. If a proto-*
> *col VLAN is configured with only one protocol type and you*
> *use the* **[no]** *form of this command to remove that protocol, the*
> *switch changes the protocol VLAN to a port-based VLAN if the*
> *VLAN does not have an untagged member port. (If an untagged*
> *member port exists on the protocol VLAN, you must either con-*
> *vert the port to a tagged member or remove the port from the*
> *VLAN before removing the last protocol type from the VLAN.)*

> *— Continued —*

— Continued from the Previous Page —

Note: If you create an IPv4 protocol VLAN, you must also assign the ARP protocol option to the VLAN to provide IP address resolution. Otherwise, IP packets are not deliverable. A "Caution" message appears in the CLI if you configure IPv4 in protocol VLAN that does not already include the arp protocol option. The same message appears if you add or delete another protocol in the same VLAN.
Note: SNA and DEClat protocol VLANs are not supported on the Series 3400cl and Series 6400cl switches.

name < *ascii-name-string* >

*When included in a **vlan** command for creating a new static VLAN, specifies a non-default VLAN name. Also used to change the current name of an existing VLAN. (Avoid spaces and the following characters in the <**ascii-name-string**> entry: **@**, **#**, **$**, **^**, **&**, *****, **(**, and **).** To include a blank space in a VLAN name, enclose the name in single or double quotes ('…' or "…").*

[voice]

Designates a VLAN for VoIP use. For more on this topic, refer to "Voice VLANs" on page 2-51.

For example, to create a new, port-based, static VLAN with a VID of 100:

```
ProCurve(config)# vlan 100                          Creates the new VLAN.
ProCurve(vlan-100)# show vlans
                                                    Shows the VLANs
 Status and Counters - VLAN Information             currently configured in
                                                    the switch.
  Maximum VLANs to support : 8
  Primary VLAN : DEFAULT_VLAN
  Management VLAN :

  802.1Q VLAN ID Name       | Status     Voice  Jumbo
  -------------- ---------- + ---------- -----  -----
  1              DEFAULT_VLAN | Port-based No    No
  100            VLAN100      | Port-based No    No
                                 If this field is empty, a Secure Management VLAN
                                 is not configured in the switch. Refer to "The
                                 Secure Management VLAN" on page 2-46
```

Figure 2-24. Example of Creating a New, Port-Based, Static VLAN

To go to a different VLAN context level, such as to the default VLAN:

```
ProCurve(vlan-100)# vlan default_vlan
ProCurve(vlan-1) _
```

**Deleting a VLAN (5300xl Running Software Release E.09.*xx* or Greater
and 4200vl).** If ports B1-B5 belong to both VLAN 2 and VLAN 3, and ports
B6-B10 belong to VLAN 3 only, then deleting VLAN 3 causes the CLI to prompt
you to approve moving ports B6 - B10 to VLAN 1 (the default VLAN). (Ports
B1-B5 are not moved because they still belong to another VLAN.)

```
ProCurve 5304XL(config)# no vlan 3
The following ports will be moved to the default VLAN:
B6-B10
Do you want to continue? [y/n] y
ProCurve Switch 5304XL(config)#
```

Converting a Dynamic VLAN to a Static VLAN. Use this feature if you
want to convert a dynamic, port-based VLAN membership to a static, port-
based VLAN membership. This is necessary if you want to make the VLAN
permanent on the switch.

Syntax: static-vlan < *vlan-id* >

> *Converts a dynamic, port-based VLAN membership to a static,
> port-based VLAN membership. (Allows port-based VLANs
> only). For this command, < vlan-id > refers to the VID of the
> dynamic VLAN membership. (Use* **show vlan** *to help identify the
> VID you need to use.) This command requires that GVRP is
> running on the switch and a port is currently a dynamic
> member of the selected VLAN. After you convert a dynamic
> VLAN to static, you must configure the switch's per-port
> participation in the VLAN in the same way that you would for
> any static VLAN. (For GVRP and dynamic VLAN operation,
> refer to chapter 3, "GVRP" .)*

For example, suppose a dynamic VLAN with a VID of 125 exists on the switch.
The following command converts the VLAN to a port-based, static VLAN.

```
ProCurve(config)# static-vlan 125
```

Configuring Static VLAN Per-Port Settings. The **vlan** *<vlan-id>* com-
mand, used with the options listed below, changes the name of an existing
static VLAN and changes the per-port VLAN membership settings.

N o t e

You can use these options from the configuration level by beginning the
command with **vlan <** *vid* **>**, or from the context level of the specific VLAN by
just typing the command option.

Syntax: [no] vlan < *vid* >

    tagged < *port-list* >

        Configures the indicated port(s) as **Tagged** *for the specified VLAN. The "***no***" version sets the port(s) to either* **No** *or (if GVRP is enabled) to* **Auto***.*

    untagged < *port-list* >

        Configures the indicated port(s) as **Untagged** *for the specified VLAN. The "***no***" version sets the port(s) to either* **No** *or (if GVRP is enabled) to* **Auto***.*

    forbid < *port-list* >

        Used in port-based VLANs to configures **< *port-list* >** *as "forbidden" to become a member of the specified VLAN, as well as other actions. Does not operate with protocol VLANs. The "***no***" version sets the port(s) to either* **No** *or (if GVRP is enabled) to* **Auto***. Refer to the chapter titled "GVRP" in the Advanced Traffic Management Guide for your switch.*

    auto < *port-list* >

        Available if GVRP is enabled on the switch. Returns the per-port settings for the specified VLAN to **Auto** *operation. Note that* **Auto** *is the default per-port setting for a static VLAN if GVRP is running on the switch.* (For information on dynamic VLAN and GVRP operation, refer to the chapter titled "GVRP" in the *Advanced Traffic Management Guide* for your switch.)

For example, suppose you have a VLAN named VLAN100 with a VID of 100, and all ports are set to **No** for this VLAN. To change the VLAN name to "**Blue_Team**" and set ports A1 - A5 to **Tagged**, you would use these commands:

```
ProCurve(config)# vlan 100 name Blue_Team
ProCurve(config)# vlan 100 tagged a1-a5
```

To move to the vlan 100 context level and execute the same commands:

```
ProCurve(config)# vlan 100
ProCurve(vlan-100)# name Blue_Team
ProCurve(vlan-100)# tagged a1-a5
```

Similarly, to change the tagged ports in the above examples to **No** (or **Auto**, if GVRP is enabled), you could use either of the following commands.

At the global config level, use:

```
ProCurve(config)# no vlan 100 tagged a1-a5
```

- or -

At the VLAN 100 context level, use:

```
ProCurve(vlan-100)# no tagged a1-a5
```

N o t e   You cannot use these commands with dynamic VLANs. Attempting to do so results in the message "**VLAN already exists.**" and no change occurs.

Web: Viewing and Configuring VLAN Parameters

In the web browser interface you can do the following:

- Add VLANs
- Rename VLANs
- Remove VLANs
- Configure VLAN tagging mode per-port
- Configure GVRP mode
- Select a new Primary VLAN

To configure other static VLAN port parameters, you will need to use either the CLI or the menu interface (available by Telnet from the web browser interface).

1. Click on the Configuration tab.

2. Click on **[Vlan Configuration]**.

3. Click on **[Add/Remove VLANs]**.

For web-based Help on how to use the web browser interface screen, click on the **[?]** button provided on the web browser screen.

802.1Q VLAN Tagging

General Applications:

■   The switch requires VLAN tagging on a given port if more than one VLAN of the same type uses the port. When a port belongs to two or more VLANs of the same type, they remain as separate broadcast domains and cannot receive traffic from each other without routing. (If multiple, *non-routable* VLANs exist in the switch—such as NETbeui protocol VLANs— then they cannot receive traffic from each other under any circumstances.)

■   The switch requires VLAN tagging on a given port if the port will be receiving inbound, tagged VLAN traffic that should be forwarded. Even if the port belongs to only one VLAN, it forwards inbound tagged traffic only if it is a tagged member of that VLAN.

■   If the only authorized, inbound VLAN traffic on a port arrives untagged, then the port must be an untagged member of that VLAN. This is the case where the port is connected to a non 802.1Q-compliant device or is assigned to only one VLAN.

For example, if port 7 on an 802.1Q-compliant switch is assigned to only the Red VLAN, the assignment can remain "untagged" because the port will forward traffic only for the Red VLAN. However, if both the Red and Green VLANs are assigned to port 7, then at least one of those VLAN assignments must be "tagged" so that Red VLAN traffic can be distinguished from Green VLAN traffic. Figure 2-25 shows this concept:
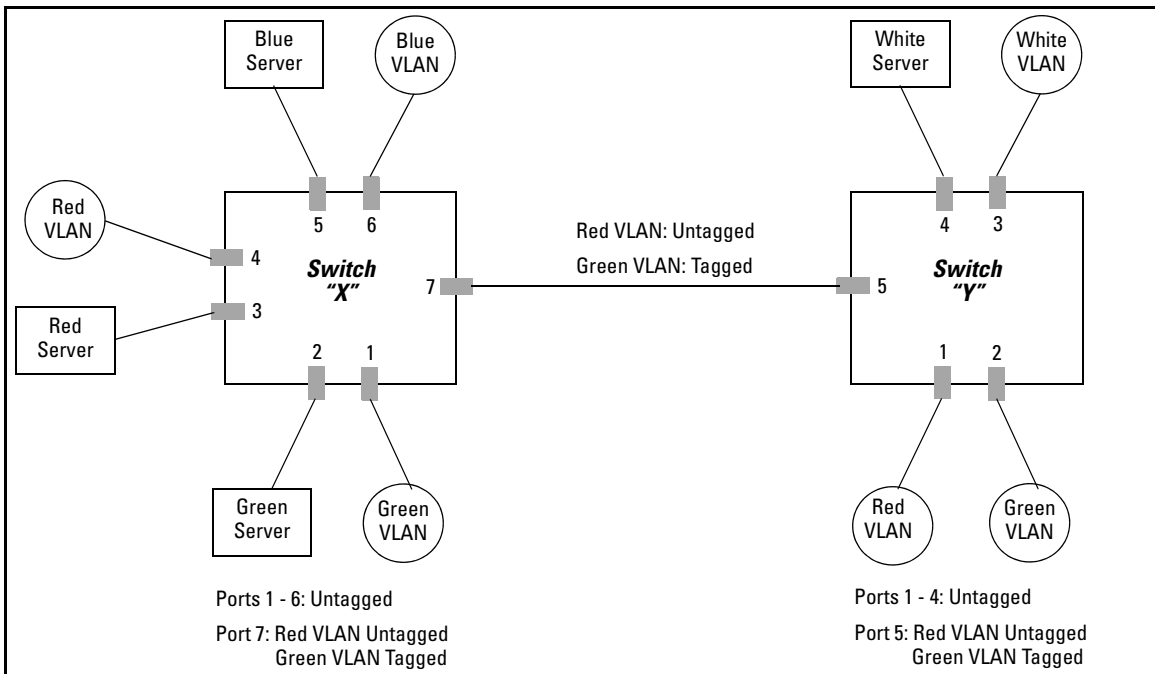
Figure 2-25. Example of Tagged and Untagged VLAN Port Assignments

■ In switch X:

• VLANs assigned to ports X1 - X6 can all be untagged because there is only one VLAN assignment per port. Red VLAN traffic will go out only the Red ports; Green VLAN traffic will go out only the Green ports, and so on. Devices connected to these ports do not have to be 802.1Q-compliant.

• However, because both the Red VLAN and the Green VLAN are assigned to port X7, at least one of the VLANs must be tagged for this port.

■ In switch Y:

• VLANs assigned to ports Y1 - Y4 can all be untagged because there is only one VLAN assignment per port. Devices connected to these ports do not have to be 802.1Q-compliant.

• Because both the Red VLAN and the Green VLAN are assigned to port Y5, at least one of the VLANs must be tagged for this port.

■ In both switches: The ports on the link between the two switches must be configured the same. As shown in figure 2-25 (above), the Red VLAN must be untagged on port X7 and Y5 and the Green VLAN must be tagged on port X7 and Y5, or vice-versa.

N o t e   Each 802.1Q-compliant VLAN must have its own unique VID number, and that VLAN *must* be given the same VID in every device in which it is configured. That is, if the Red VLAN has a VID of 10 in switch X, then 10 must also be used for the Red VID in switch Y.

```
==========================- CONSOLE - MANAGER MODE -=============================
                    Switch Configuration - VLAN - VLAN Names

    802.1Q VLAN ID      Name
    --------------    ------------
    1                 DEFAULT_VLAN
    10                Red_VLAN
    20                Blue_VLAN



   Actions->    Back       Add      Edit      Delete      Help

 Return to previous screen.
 Use up/down arrow keys to change record selection, left/right arrow keys to
 change action selection, and <Enter> to execute action.
```
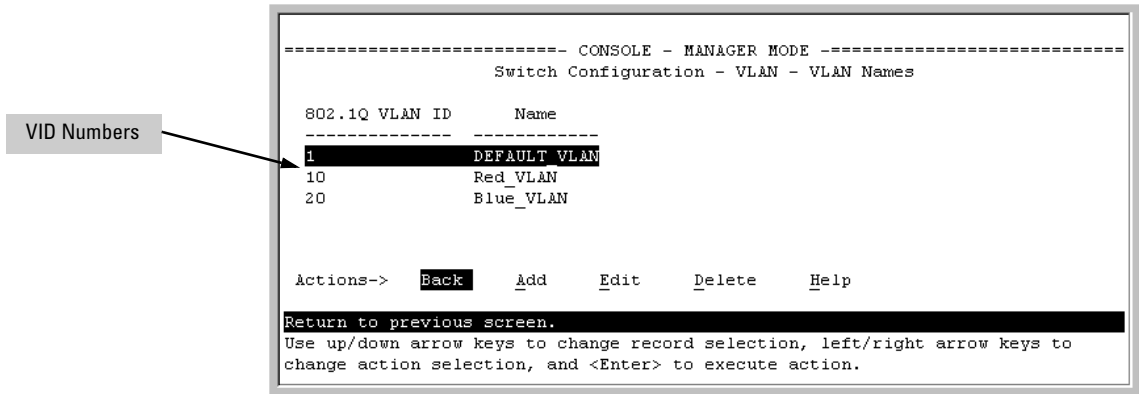
VID Numbers →

Figure 2-26. Example of VLAN ID Numbers Assigned in the VLAN Names Screen

VLAN tagging gives you several options:

■ Since the purpose of VLAN tagging is to allow multiple VLANs on the same port, any port that has only one VLAN assigned to it can be configured as "Untagged" (the default) if the authorized inbound traffic for that port arrives untagged.

■ Any port with two or more VLANs of the same type can have one such VLAN assigned as "Untagged". All other VLANs of the same type must be configured as "Tagged". That is:

Port-Based VLANs	Protocol VLANs
A port can be a member of one untagged, port-based VLAN. All other port-based VLAN assignments for that port must be tagged.	A port can be an untagged member of one protocol-based VLAN of each protocol type. When assigning a port to multiple, protocol-based VLANs sharing the same type, the port can be an untagged member of only one such VLAN.
A port can be a tagged member of any port-based VLAN. See above.	A port can be a tagged member of any protocol-based VLAN. See above.
Note: A given VLAN *must* have the same VID on all 802.1Q-compliant devices in which the VLAN occurs. Also, the ports connecting two 802.1Q devices should have identical VLAN configurations.	

■   If all end nodes on a port comply with the 802.1Q standard and are configured to use the correct VID, then, you can configure all VLAN assignments on a port as "Tagged" if doing so either makes it easier to manage your VLAN assignments, or if the authorized, inbound traffic for all VLANs on the port will be tagged.

For a summary and flowcharts of untagged and tagged VLAN operation on inbound traffic, refer to the following under "VLAN Operating Rules" on pages 2-14 through 2-17:

•   "Inbound Tagged Packets"
•   "Untagged Packet Forwarding" and figure 2-7
•   "Tagged Packet Forwarding" and figure 2-8

Example.   In the following network, switches X and Y and servers S1, S2, and the AppleTalk server are 802.1Q-compliant. (Server S3 could also be 802.1Q-compliant, but it makes no difference for this example.) This network includes both protocol-based (AppleTalk) VLANs and port-based VLANs.
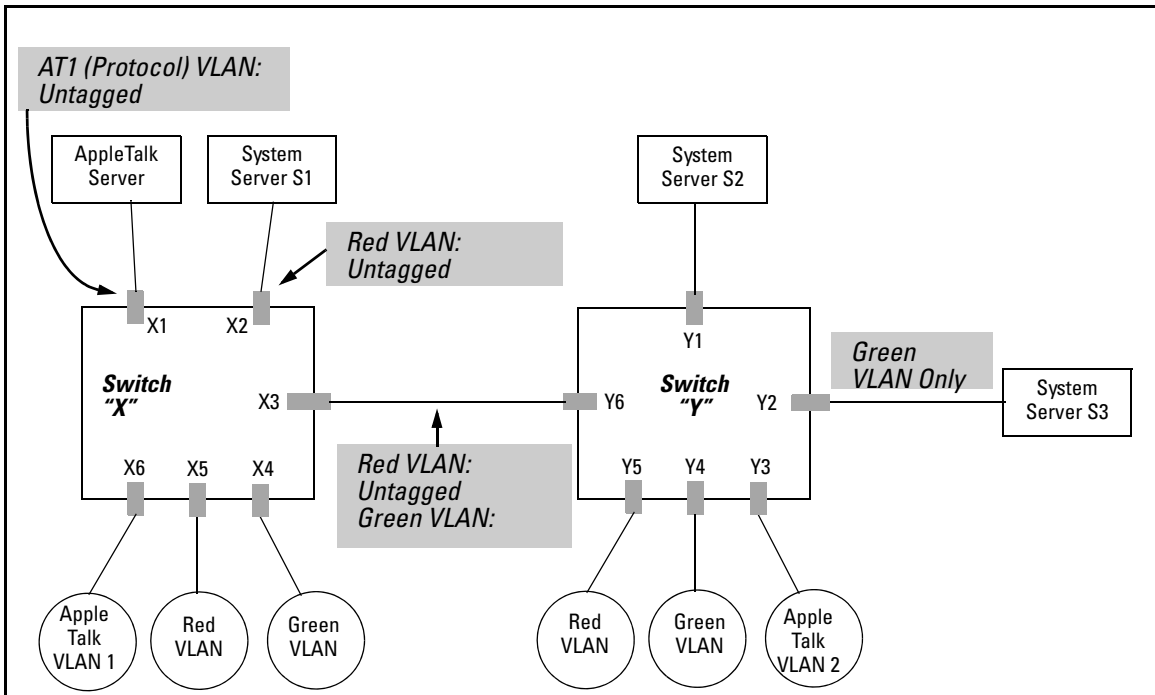
Figure 2-27.   Example of Networked 802.1Q-Compliant Devices with Multiple VLANs on Some Ports

- ■ The VLANs assigned to ports X4 - X6, Y2 - Y5 can all be untagged because there is only one VLAN assigned per port.
- ■ Port X1 has two AppleTalk VLANs assigned, which means that one VLAN assigned to this port can be untagged and the other must be tagged.
- ■ Ports X2 and Y1 have two port-based VLANs assigned, so one can be untagged and the other must be tagged on both ports.
- ■ Ports X3 and Y6 have two port-based VLANs and one protocol-based VLAN assigned. Thus, one port-based VLAN assigned to this port can be untagged and the other must be tagged. Also, since these two ports share the same link, their VLAN configurations must match.

	Switch X					Switch Y			
Port	AT-1 VLAN	AT-2 VLAN	Red VLAN	Green VLAN	Port	AT-1 VLAN	AT-2 VLAN	Red VLAN	Green VLAN
X1	Untagged	Tagged	No*	No*	Y1	No*	No*	Untagged	Tagged
X2	No*	No*	Untagged	Tagged	Y2	No*	No*	No*	Untagged
X3	No*	Untagged	Untagged	Tagged	Y3	No*	Untagged	No*	No*
X4	No*	No*	No*	Untagged	Y4	No*	No*	No*	Untagged
X5	No*	No*	Untagged	No*	Y5	No*	No*	Untagged	No*
X6	Untagged	No*	No*	No*	Y6	No	Untagged	Untagged	Tagged

*"No" means the port is not a member of that VLAN. For example, port X3 is not a member of the Red VLAN and does not carry Red VLAN traffic. Also, if GVRP were enabled (port-based only), "Auto" would appear instead of "No".

N o t e   VLAN configurations on ports connected by the same link must match. Because ports X2 and Y5 are opposite ends of the same point-to-point connection, both ports must have the same VLAN configuration; that is, both ports configure the Red VLAN as "Untagged" and the Green VLAN as "Tagged".

Special VLAN Types

VLAN Support and the Default VLAN

In the factory default configuration, VLAN support is enabled and all ports on the switch belong to the port-based, default VLAN (named DEFAULT_VLAN). This places all ports in the switch into one physical broadcast domain. In the factory-default state, the default VLAN is also the *Primary* VLAN.

You can partition the switch into multiple virtual broadcast domains by configuring one or more additional VLANs and moving ports from the default VLAN to the new VLANs. (The switch supports up to 256 static and dynamic VLANs.) You can change the name of the default VLAN, but you cannot change the default VLAN's VID (which is always "1"). Although you can remove all ports from the default VLAN (by placing them in another port-based VLAN), this VLAN is always present; that is, you cannot delete it from the switch.

For details on port VLAN settings, refer to "Configuring Static VLAN Per-Port Settings" on page 2-37

The Primary VLAN

Because certain features and management functions run on only one VLAN in the switch, and because DHCP and Bootp can run per-VLAN, there is a need for a dedicated VLAN to manage these features and ensure that multiple instances of DHCP or Bootp on different VLANs do not result in conflicting configuration values for the switch. The *Primary* VLAN is the VLAN the switch uses to run and manage these features and data. In the factory-default configuration, the switch designates the default VLAN (DEFAULT_VLAN; VID = 1) as the Primary VLAN. However, to provide more control in your network, you can designate another static, port-based VLAN as primary. To summarize, *designating a non-default VLAN as primary* means that:

■ The switch reads DHCP responses on the Primary VLAN instead of on the default VLAN. (This includes such DHCP-resolved parameters as the TimeP server address, Default TTL, and IP addressing—including the Gateway IP address—when the switch configuration specifies DHCP as the source for these values.)

■ The default VLAN continues to operate as a standard VLAN (except, as noted above, you cannot delete it or change its VID).

■ Any ports not specifically assigned to another VLAN will remain assigned to the Default VLAN, regardless of whether it is the Primary VLAN.

Candidates for Primary VLAN include any static, port-based VLAN currently configured on the switch. (Protocol-Based VLANs and dynamic—GVRP-learned—VLANs that have not been converted to a static VLAN cannot be the Primary VLAN.) To display the current Primary VLAN, use the CLI **show vlan** command.

N o t e   If you configure a non-default VLAN as the Primary VLAN, you cannot delete that VLAN unless you first select a different VLAN to serve as primary.

If you manually configure a gateway on the switch, it ignores any gateway address received via DHCP or Bootp.

To change the Primary VLAN configuration, refer to "Changing the Primary VLAN" on page 2-34.

The Secure Management VLAN

Configuring a secure Management VLAN creates an isolated network for managing the ProCurve switches that support this feature. (As of September, 2006, the Secure Management VLAN feature is available on these ProCurve switches:

- Series 6400cl switches
- Switch 6108
- Series 5400 switches
- Series 5300xl switches
- Series 4200vl switches
- Series 4100gl switches

- 2900yl switches
- Series 3400cl switches
- 2810 switches
- Series 2800 switches
- Series 2600 switches
- 2510 switches

If you configure a Secure Management VLAN, access to the VLAN and to the switch's management functions (Menu, CLI, and web browser interface) is available only through ports configured as members.

- Multiple ports on the switch can belong to the Management VLAN. This allows connections for multiple management stations you want to have access to the Management VLAN, while at the same time allowing Management VLAN links between switches configured for the same Management VLAN.

- Only traffic from the Management VLAN can manage the switch, which means that only the workstations and PCs connected to ports belonging to the Management VLAN can manage and reconfigure the switch.

Figure 2-28 illustrates use of the Management VLAN feature to support management access by a group of management workstations.

N o t e   The Secure Management VLAN must be a static, port-based VLAN with a manually configured IP address and subnet mask. (The switch does not allow the Management VLAN to acquire IP addressing through DHCP/Bootp.)
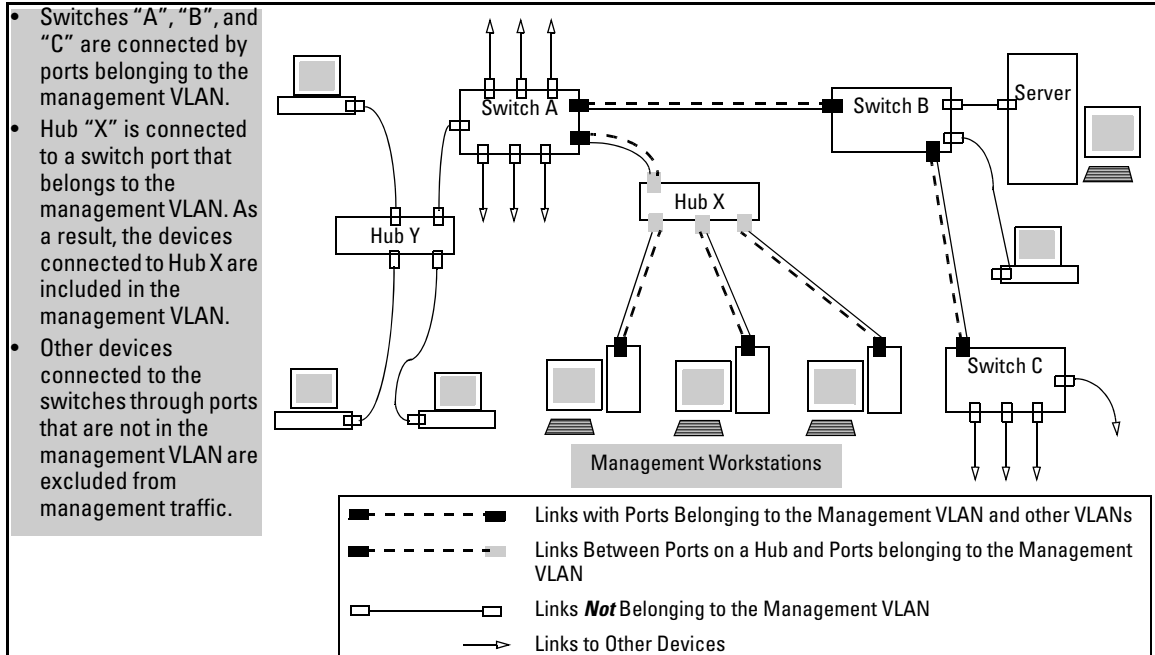
Figure 2-28. Example of Potential Security Breaches

In figure 2-29, Workstation 1 has management access to all three switches through the Management VLAN, while the PCs do not. This is because configuring a switch to recognize a Management VLAN automatically excludes attempts to send management traffic from any other VLAN.
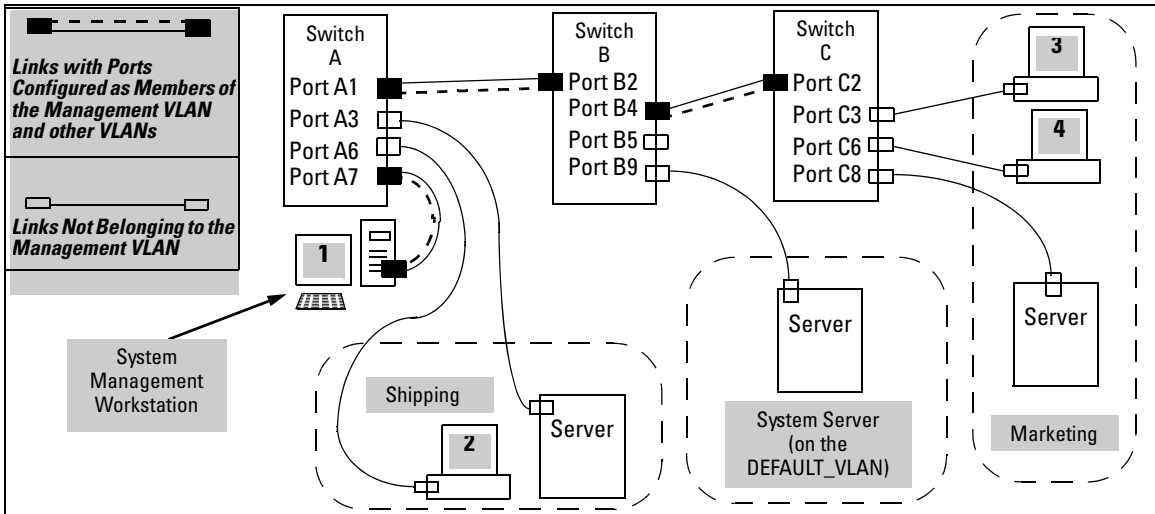
Figure 2-29. Example of Management VLAN Control in a LAN

Table 2-7. VLAN Membership in Figure 2-29

Switch	A1	A3	A6	A7	B2	B4	B5	B9	C2	C3	C6	C8
Management VLAN (VID = 7)	**Y**	N	N	**Y**	**Y**	**Y**	N	N	**Y**	N	N	N
Marketing VLAN (VID = 12)	N	N	N	N	N	N	N	N	N	**Y**	**Y**	**Y**
Shipping Dept. VLAN (VID = 20)	N	**Y**	**Y**	N	N	N	N	N	N	N	N	N
DEFAULT-VLAN (VID = 1)	**Y**	**Y**	**Y**	**Y**	**Y**	**Y**	**Y**	**Y**	**Y**	**Y**	**Y**	**Y**

Preparation

1. Determine a VID and VLAN name suitable for your Management VLAN.

    (You must manually configure the IP addressing for the Management
    VLAN. The switch does not allow the Management VLAN to acquire an IP
    address through DHCP/Bootp.)

2. Plan your Management VLAN topology to use ProCurve switches that
    support this feature. (Refer to page 2-46.) The ports belonging to the
    Management VLAN should be only the following:

    •   Ports to which you will connect authorized management stations
        (such as Port A7 in figure 2-29.)

    •   Ports on one switch that you will use to extend the Management VLAN
        to ports on other ProCurve switches (such as ports A1 and B2 or B4
        and C2 in figure 2-29 on page 2-48.).

Hubs dedicated to connecting management stations to the Management VLAN can also be included in the above topology. Note that any device connected to a hub in the Management VLAN will also have Management VLAN access.

3. Configure the Management VLAN on the selected switch ports.

4. Test the management VLAN from all of the management stations authorized to use the Management VLAN, including any SNMP-based network management stations. Ensure that you include testing any Management VLAN links between switches.

N o t e

If you configure a Management VLAN on a switch by using a Telnet connection through a port that is not in the Management VLAN, then you will lose management contact with the switch if you log off your Telnet connection or execute **write memory** and **reboot** the switch.

Configuration

Syntax: [no] management-vlan < *vlan-id* | *vlan-name* >

Configures an existing VLAN as the management VLAN. The **no** *form disables the management VLAN and returns the switch to its default management operation. Default: Disabled. In this case, the VLAN returns to standard VLAN operation.*

For example, suppose you have already configured a VLAN named **My_VLAN** with a VID of 100. Now you want to configure the switch to do the following:

■ Use **My_VLAN** as a Management VLAN (tagged, in this case) to connect port A1 on switch "A" to a management station. (The management station includes a network interface card with 802.1Q tagged VLAN capability.)

■ Use port A2 to extend the Management VLAN to port B1 (which is already configured as a tagged member of **My_VLAN**) on an adjacent Procurve switch that supports the Management VLAN feature.
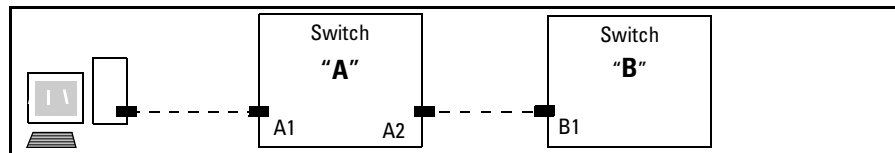
Figure 2-30. Illustration of Configuration Example

```
ProCurve (config)# management-vlan 100
ProCurve (config)# vlan 100 tagged a1
ProCurve (config)# vlan 100 tagged a2
```

Deleting the Management VLAN

You can disable the Secure Management feature without deleting the VLAN itself. For example, either of the following commands disables the Secure Management feature in the above example:

```
ProCurve (config)# no management-vlan 100
ProCurve (config)# no management-vlan my_vlan
```

Operating Notes for Management VLANs

- Use only a static, port-based VLAN for the Management VLAN.

- The Management VLAN does not support IGMP operation.

- On switches covered by this manual, with routing enabled, routing between the Management VLAN and other VLANs is not allowed.

- If there are more than 25 VLANs configured on the switch, reboot the switch after configuring the management VLAN. (*ProCurve Series 5300xl and Series 4200vl switches only.*)

- If you implement a Management VLAN in a switch mesh environment, all meshed ports on the switch will be members of the Management VLAN.

- Only one Management-VLAN can be active in the switch. If one Management-VLAN VID is saved in the startup-config file and you configure a different VID in the running-config file, the switch uses the running-config version until you either use the **write-memory** command or reboot the switch.

- During a Telnet session to the switch, if you configure the Management-VLAN to a VID that excludes the port through which you are connected to the switch, you will continue to have access only until you terminate the session by logging out or rebooting the switch.

- During a web browser session to the switch, if you configure the Management-VLAN to a VID that excludes the port through which you are connected to the switch, you will continue to have access only until you close the browser session or rebooting the switch.

N o t e   The Management-VLAN feature does not control management access through a direct connection to the switch's serial port.

- Enabling Spanning Tree where there are multiple links using separate VLANs, including the Management VLAN, between a pair of switches, Spanning Tree will force the blocking of one or more links. This may include the link carrying the Management VLAN, which will cause loss of management access to some devices. This can also occur where meshing is configured and the Management VLAN is configured on a separate link.
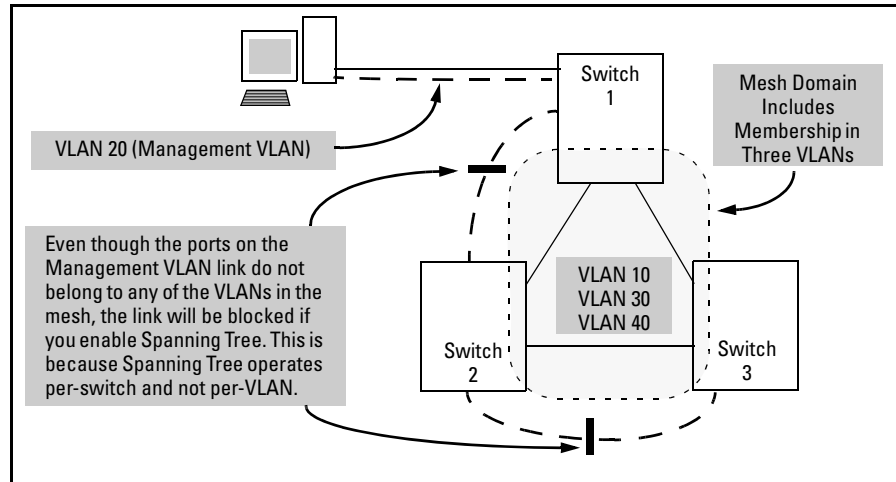
Figure 2-31. Example of Inadvertently Blocking a Management VLAN Link by Implementing Spanning Tree

Voice VLANs

Configuring voice VLANs separates voice traffic from data traffic and shields your voice traffic from broadcast storms. This section describes how to configure the switch for voice VLAN operation.

Operating Rules for Voice VLANs

■   You must statically configure voice VLANs. GVRP and dynamic VLANs do not support voice VLAN operation.

■   Configure all ports in a voice VLAN as tagged members of the VLAN. This ensures retention of the QoS (Quality of Service) priority included in voice VLAN traffic moving through your network.

■   If a telephone connected to a voice VLAN includes a data port used for connecting other networked devices (such as PCs) to the network, then you must configure the port as a tagged member of the voice VLAN and a tagged or untagged member of the data VLAN you want the other networked device to use.

Components of Voice VLAN Operation

■ **Voice VLAN(s):** Configure one or more voice VLANs on the switch. Some reasons for having multiple voice VLANs include:

• Employing telephones with different VLAN requirements

• Better control of bandwidth usage

• Segregating telephone groups used for different, exclusive purposes

Where multiple voice VLANs exist on the switch, you can use routing to communicate between telephones on different voice VLANs. .

■ **Tagged/Untagged VLAN Membership:** If the appliances using a voice VLAN transmit tagged VLAN packets, then configure the member ports as tagged members of the VLAN. Otherwise, configure the ports as untagged members.

Voice VLAN QoS Prioritizing (Optional)

Without configuring the switch to prioritize voice VLAN traffic, one of the following conditions applies:

■ If the ports in a voice VLAN are not tagged members, then the switch forwards all traffic on that VLAN at "normal" priority.

■ If the ports in a voice VLAN are tagged members, then the switch forwards all traffic on that VLAN at whatever priority the traffic has when received inbound on the switch.

Using the switch's QoS VLAN-ID (VID) Priority option, you can change the priority of voice VLAN traffic moving through the switch. If all port memberships on the voice VLAN are tagged, the priority level you set for voice VLAN traffic is carried to the next device. With all ports on the voice VLAN configured as tagged members, you can enforce a QoS priority policy moving through the switch and through your network. To set a priority on a voice VLAN, use the following command:

Syntax: vlan < *vid* > qos priority < 0 - 7 >

The qos priority default setting is 0 (normal), with 1 as the lowest priority and 7 as the highest priority.

For example, if you configured a voice VLAN with a VID of 10, and wanted the highest priority for all traffic on this VLAN, you would execute the following command:

```
ProCurve(config) # vlan 10 qos priority 7
ProCurve (config) # write memory
```

Note that you also have the option of resetting the DSCP (DiffServe Code-point) on tagged voice VLAN traffic moving through the switch. For more on this and other QoS topics, refer to the chapter titled "Quality of Service (QoS): Managing Bandwidth More Effectively".

Voice VLAN Access Security

You can use port security configured on an individual port or group of ports in a voice VLAN. That is, you can allow or deny access to a phone having a particular MAC address. Refer to chapter titled "Configuring and Monitoring Port Security" in the Access Security Guide (p/n 5990-6052, February 2004 or a later version).

Note   MAC authentication is not recommended in voice VLAN applications.

Effect of VLANs on Other Switch Features

Spanning Tree Operation with VLANs

Depending on the spanning-tree option configured on the switch, the spanning-tree feature may operate as a single instance across all ports on the switch (regardless of VLAN assignments) or multiple instance on a per-VLAN basis. For single-instance operation, this means that if redundant physical links exist between the switch and another 802.1Q device, all but one link will be blocked, regardless of whether the redundant links are in separate VLANs. In this case you can use port trunking to prevent Spanning Tree from unnecessarily blocking ports (and to improve overall network performance). For multiple-instance operation, physically redundant links belonging to different VLANs can remain open. Refer to chapter 6, "Spanning-Tree Operation" .

Note that Spanning Tree operates differently in different devices. For example, in the (obsolete, non-802.1Q) ProCurve Switch 2000 and the ProCurve Switch 800T, Spanning Tree operates on a per-VLAN basis, allowing redundant physical links as long as they are in separate VLANs.

IP Interfaces

There is a one-to-one relationship between a VLAN and an IP network interface. Since the VLAN is defined by a group of ports, the state (up/down) of those ports determines the state of the IP network interface associated with that VLAN. When a port-based VLAN or an IPv4 or IPv6 protocol-based VLAN comes up because one or more of its ports is up, the IP interface for that VLAN is also activated. Likewise, when a VLAN is deactivated because all of its ports are down, the corresponding IP interface is also deactivated.

VLAN MAC Address

The switches covered by this guide has one unique MAC address for all of its VLAN interfaces. You can send an 802.2 test packet to this MAC address to verify connectivity to the switch. Likewise, you can assign an IP address to the VLAN interface, and when you Ping that address, ARP will resolve the IP address to this single MAC address. In a topology where a switch covered by this guide has multiple VLANs and must be connected to a device having a single forwarding database, such as the Switch 4000M, some cabling restrictions apply. For more on this topic, refer to "Multiple VLAN Considerations" on page 2-18.

Port Trunks

When assigning a port trunk to a VLAN, all ports in the trunk are automatically assigned to the same VLAN. You cannot split trunk members across multiple VLANs. Also, a port trunk is tagged, untagged, or excluded from a VLAN in the same way as for individual, untrunked ports.

Port Monitoring

If you designate a port on the switch for network monitoring, this port will appear in the Port VLAN Assignment screen and can be configured as a member of any VLAN. For information on how broadcast, multicast, and unicast packets are tagged inside and outside of the VLAN to which the monitor port is assigned, refer to the section titled "VLAN-Related Problems" in the "Troubleshooting" chapter of the *Management and Configuration Guide* for your switch.

Jumbo Packet Support on the Series 3400cl and Series 6400cl Switches

Jumbo packet support for the 3400cl and 6400cl switches is enabled per-VLAN and applies to all ports belonging to the VLAN. For more information, refer to the chapter titled "Port Traffic Controls" in the *Management and Configuration Guide* for your switch. (Jumbo packet support is not available on the Series 5300xl switches or Series 4200vl switches.)

VLAN Restrictions

■   A port must be a member of at least one VLAN. In the factory default configuration, all ports are assigned to the default VLAN (DEFAULT_VLAN; VID = 1).

■   A port can be a member of one untagged, port-based VLAN. All other port-based VLAN assignments for that port must be tagged. (The "Untagged" designation enables VLAN operation with non 802.1Q-compliant devices.)

■   A port can be an untagged member of one protocol-based VLAN of each protocol type. When assigning a port to multiple, protocol-based VLANs sharing the same type, the port can be an untagged member of only one such VLAN.

■   With routing enabled on the switch, the switch can route traffic between:

•   Multiple, port-based VLANs

•   A port-based VLAN and an IPv4 protocol-based VLAN

•   A port-based VLAN and an IPv6 protocol-based VLAN

•   An IPv4 protocol-based VLAN and an IPv6 protocol VLAN.

Other, routable, protocol-based VLANs must use an external router to move traffic between VLANs. With routing disabled, all routing between VLANs must be through an external router.

■   On the 3400cl and 6400cl switches, and on 5300xl switches running a software version earlier than E.09.xx, prior to deleting a static VLAN, you must first re-assign all ports in the VLAN to another VLAN. On 5300xl switches running software version E.09.*xx* or greater and on 4200vl switches, you can use the **no vlan < *vid* >** command to delete a static VLAN. For more information, refer to "Creating a New Static VLAN (Port-Based or Protocol-Based) Changing the VLAN Context Level" on page 2-35.

—This page is intentionally unused —

3

GVRP

Contents

Overview

This chapter describes GVRP and how to configure it with the switch's built-in interfaces, and assumes an understanding of VLANs, which are described in chapter 2, "Static Virtual LANs (VLANs)" .

For general information on how to use the switch's built-in interfaces, refer to these chapters in the *Management and Configuration Guide* for your switch:

■   Chapter 3, "Using the Menu Interface"

■   Chapter 4, "Using the Command Line Interface (CLI)"

■   Chapter 5, "Using the Web Browser Interface

■   Chapter 6, "Switch Memory and Configuration"

Introduction

Feature	Default	Menu	CLI	Web
view GVRP configuration	n/a	page 3-13	page 3-14	page 3-18
list static and dynamic VLANs on a GVRP-enabled switch	n/a	—	page 3-16	page 3-18
enable or disable GVRP	disabled	page 3-13	page 3-15	page 3-18
enable or disable GVRP on individual ports	enabled	page 3-13	page 3-15	—
control how individual ports handle advertisements for new VLANs	Learn	page 3-13	page 3-15	page 3-18
convert a dynamic VLAN to a static VLAN	n/a	—	page 3-17	—
configure static VLANs	DEFAULT_VLAN (VID = 1)	page 2-22	page 2-28	page 2-39

GVRP—GARP VLAN Registration Protocol—is an application of the Generic Attribute Registration Protocol—GARP. GVRP is defined in the IEEE 802.1Q standard, and GARP is defined in the IEEE 802.1D-1998 standard.

N o t e   To understand and use GVRP you must have a working knowledge of 802.1Q VLAN tagging. (Refer to chapter 2, "Static Virtual LANs (VLANs)" .)

GVRP uses "GVRP Bridge Protocol Data Units" ("GVRP BPDUs") to "advertise" static VLANs. In this manual, a GVRP BPDU is termed an *advertisement*. Advertisements are sent outbound from ports on a switch to the devices directly connected to those ports.

While GVRP is enabled on the switch, you cannot apply any ACLs to VLANs configured on the same switch.

GVRP enables the switch to dynamically create 802.1Q-compliant VLANs on links with other devices running GVRP. This enables the switch to automatically create VLAN links between GVRP-aware devices. (A GVRP link can include intermediate devices that are not GVRP-aware.) This operation reduces the chances for errors in VLAN configuration by automatically providing VLAN ID (VID) consistency across the network. That is, you can use GVRP to propagate VLANs to other GVRP-aware devices instead of manually

having to set up VLANs across your network. After the switch creates a dynamic VLAN, you can optionally use the CLI **static** *<vlan-id>* command to convert it to a static VLAN or allow it to continue as a dynamic VLAN for as long as needed. You can also use GVRP to dynamically enable port membership in static VLANs configured on a switch.

General Operation

When GVRP is enabled on a switch, the VID for any static VLANs configured on the switch is *advertised* (using BPDUs—Bridge Protocol Data Units) out all ports, regardless of whether a port is up or assigned to any particular VLAN. A GVRP-aware port on another device that receives the advertisements over a link can dynamically join the advertised VLAN.

A dynamic VLAN (that is, a VLAN learned through GVRP) is tagged on the port on which it was learned. Also, a GVRP-enabled port can forward an advertisement for a VLAN it learned about from other ports on the same switch (internal source), but the forwarding port will not itself join that VLAN until an advertisement for that VLAN is received through a link from another device (external source) on that specific port

Operating Note: When a GVRP-aware port on a switch learns a VID through GVRP from another device, the switch begins advertising that VID out all of its ports except the port on which the VID was learned.

Core switch with static VLANs (VID= 1, 2, & 3). Port 2 is a member of VIDs 1, 2, & 3.

1. Port 2 advertises VIDs 1, 2, & 3.

2. Port 1 receives advertisement of VIDs 1, 2, & 3 AND becomes a member of VIDs 1, 2, & 3.

3. Port 3 advertises VIDs 1, 2, & 3, but port 3 is NOT a member of VIDs 1, 2, & 3 at this point.

4. Port 4 receives advertisement of VIDs 1, 2, & 3 AND becomes a member of VIDs 1, 2, & 3.

5. Port 5 advertises VIDs 1, 2, & 3, but port 5 is NOT a member of VIDs 1, 2, & 3 at this point.

Port 6 is statically configured to be a member of VID 3.

Switch 1 GVRP On	Switch 2 GVRP On	Switch 3 GVRP On	Static VLAN configured End Device (NIC or switch) with GVRP On
2	1 3	4 5	6

11. Port 2 receives advertisement of VID 3. (Port 2 is already statically configured for VID 3.)

9. Port 3 receives advertisement of VID 3 AND becomes a member of VID 3. (Still not a member of VIDs 1 & 2.)

10. Port 1 advertises VID 3.

7. Port 5 receives advertisement of VID 3 AND becomes a member of VID 3. (Still not a member of VIDs 1 & 2.)

8. Port 4 advertises VID 3.

6. Port 6 advertises VID 3.

Figure 3-1. .Example of Forwarding Advertisements and Dynamic Joining

Note that if a static VLAN is configured on at least one port of a switch, and that port has established a link with another device, then all other ports of that switch will send advertisements for that VLAN.

For example, in the following figure, Tagged VLAN ports on switch "A" and switch "C" advertise VLANs 22 and 33 to ports on other GVRP-enabled switches that can dynamically join the VLANs.
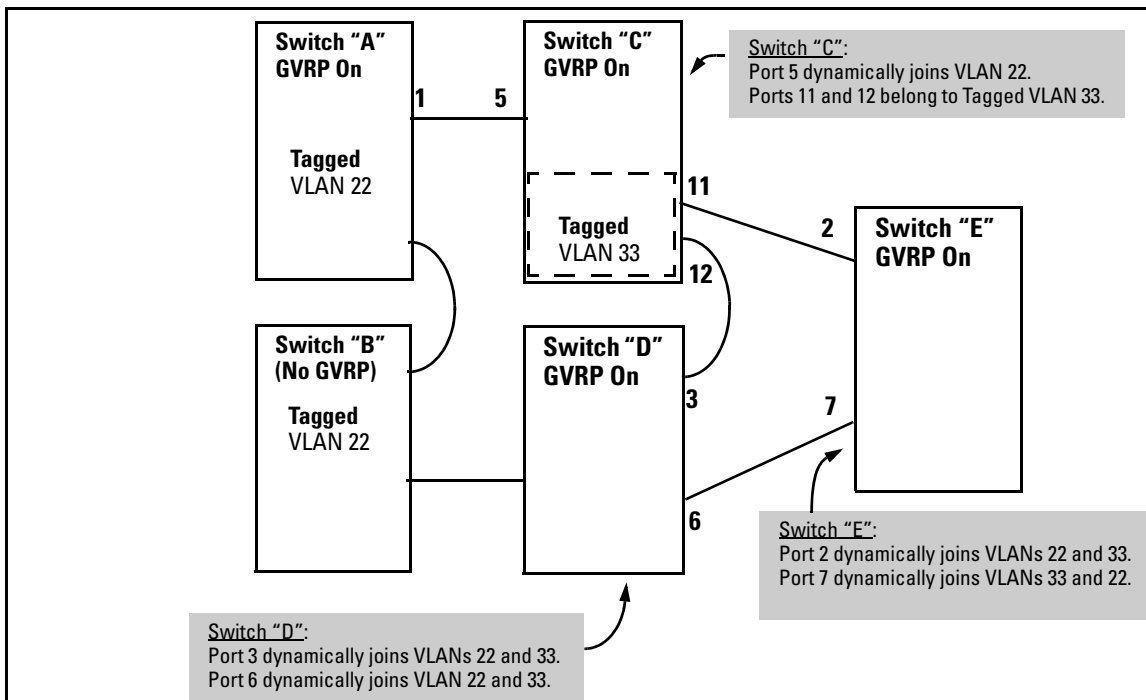
Figure 3-2. Example of GVRP Operation

N o t e

A port can learn of a dynamic VLAN through devices that are not aware of GVRP (Switch "B", above). VLANs must be disabled in GVRP-unaware devices to allow tagged packets to pass through.

A GVRP-aware port receiving advertisements has these options:

■ If there is not already a static VLAN with the advertised VID on the receiving port, then dynamically create the VLAN and become a member.

■ If the switch already has a static VLAN assignment with the same VID as in the advertisement, and the port is configured to **Auto** for that VLAN, then the port will dynamically join the VLAN and begin moving that VLAN's traffic. (For more detail on **Auto**, see "Per-Port Options for Dynamic VLAN Advertising and Joining" on page 3-9.)

■ Ignore the advertisement for that VID.

■ Don't participate in that VLAN.

Note also that a port belonging to a Tagged or Untagged static VLAN has these configurable options:

■ Send VLAN advertisements, and also receive advertisements for VLANs on other ports and dynamically join those VLANs.

■ Send VLAN advertisements, but ignore advertisements received from other ports.

■ Avoid GVRP participation by not sending advertisements and dropping any advertisements received from other devices.

IP Addressing. A dynamic VLAN does not have an IP address, and moves traffic on the basis of port membership in VLANs. However, after GVRP creates a dynamic VLAN, you can convert it to a static VLAN. Note that it is then necessary to assign ports to the VLAN in the same way that you would for a static VLAN that you created manually.   In the static state you can configure IP addressing on the VLAN and access it in the same way that you would any other static (manually created) VLAN.

Per-Port Options for Handling GVRP "Unknown VLANs"

An "unknown VLAN" is a VLAN that the switch learns of by receiving an advertisement for that VLAN on a port that is not already a member of that VLAN. If the port is configured to learn unknown VLANs, then the VLAN is dynamically created and the port becomes a tagged member of the VLAN. For example, suppose that in figure 3-2 (page 3-6), port 1 on switch "A" is con-nected to port 5 on switch "C". Because switch "A" has VLAN 22 statically configured, while switch "C" does not have this VLAN statically configured (and does not "Forbid" VLAN 22 on port 5), VLAN 22 is handled as an "Unknown VLAN" on port 5 in switch "C". Conversely, if VLAN 22 was statically configured on switch C, but port 5 was not a member, port 5 would become a member when advertisements for VLAN 22 were received from switch "A".

When you enable GVRP on a switch, you have the per-port join-request options listed in table 3-1:

Table 3-1. Options for Handling "Unknown VLAN" Advertisements:

Unknown VLAN Mode	Operation
Learn (the Default)	Enables the port to become a member of any unknown VLAN for which it receives an advertisement. Allows the port to advertise other VLANs that have at least one other port on the same switch as a member.
Block	Prevents the port from joining any new dynamic VLANs for which it receives an advertisement. Allows the port to advertise other VLANs that have at least one other port as a member.
Disable	Causes the port to ignore and drop all GVRP advertisements it receives and also prevents the port from sending any GVRP advertisements.

The CLI **show gvrp** command and the menu interface VLAN Support screen show a switch's current GVRP configuration, including the Unknown VLAN settings.

```
ProCurve# show gvrp
 GVRP support
  Maximum VLANs to support : 8
  GVRP Enabled : Yes          ◄────────  GVRP Enabled
                                         (Required for Unknown
  Port Type      | Unknown VLAN          VLAN operation.)
  ---- --------- + ------------
   A1   10/100TX |  Learn
   A2   10/100TX |  Learn
   A3   10/100TX |  Block                Unknown VLAN Settings
   A4   10/100TX |  Block                Default: Learn
   A5   10/100TX |  Learn
   A6   10/100TX |  Disable
   A7   10/100TX |  Learn
   A8   10/100TX |  Learn
    •       •          •
    •       •          •
    •       •          •
```

Figure 3-3. Example of GVRP Unknown VLAN Settings

Per-Port Options for Dynamic VLAN Advertising and Joining

Initiating Advertisements. As described in the preceding section, to enable dynamic joins, GVRP must be enabled and a port must be configured to Learn (the default). However, to send advertisements in your network, one or more static (**Tagged**, **Untagged**, or **Auto**) VLANs must be configured on one or more switches (with GVRP enabled), depending on your topology.

Enabling a Port for Dynamic Joins. You can configure a port to dynamically join a static VLAN. The join will then occur if that port subsequently receives an advertisement for the static VLAN. (This is done by using the **Auto** and **Learn** options described in table 3-2, on the next page.

Parameters for Controlling VLAN Propagation Behavior. You can configure an individual port to actively or passively participate in dynamic VLAN propagation or to ignore dynamic VLAN (GVRP) operation. These options are controlled by the GVRP "Unknown VLAN" and the static VLAN configuration parameters, as described in the following table:

Table 3-2. Controlling VLAN Behavior on Ports with Static VLANs

Per-Port "Unknown VLAN" (GVRP) Configuration	Static VLAN Options—Per VLAN Specified on Each Port [1]		
	Port Activity: Tagged or Untagged (Per VLAN)[2]	Port Activity: Auto[2] (Per VLAN)	Port Activity: Forbid (Per VLAN)[2]
Learn (the Default)	The port: • Belongs to specified VLAN. • Advertises specified VLAN. • Can become a member of dynamic VLANs for which it receives advertisements. • Advertises dynamic VLANs that have at least one other port (on the same switch) as a member.	The port: • Will become a member of specified VLAN if it receives advertisements for specified VLAN from another device. • Will advertise specified VLAN. • Can become a member of other, dynamic VLANs for which it receives advertisements. • Will advertise a dynamic VLAN that has at least one other port (on the same switch) as a member.	The port: 1. Will not become a member of the specified VLAN. 2. Will not advertise specified VLAN. 3. Can become a member of other dynamic VLANs for which it receives advertisements. 4. Will advertise a dynamic VLAN that has at least one other port on the same switch as a member.
Block	The port: • Belongs to the specified VLAN. • Advertises this VLAN. • Will not become a member of new dynamic VLANs for which it receives advertisements. • Will advertise dynamic VLANs that have at least one other port as a member.	The port: • Will become a member of specified VLAN if it receives advertisements for this VLAN. • Will advertise this VLAN. • Will not become a member of new dynamic VLANs for which it receives advertisements. • Will advertise dynamic VLANs that have at least one other port (on the same switch) as a member.	The port: • Will not become a member of the specified VLAN. • Will not advertise this VLAN. • Will not become a member of dynamic VLANs for which it receives advertisements. • Will advertise dynamic VLANs that have at least one other port (on the same switch) as a member.
Disable	The port: • Is a member of the specified VLAN. • Will ignore GVRP PDUs. • Will not join any advertised VLANs. • Will not advertise VLANs.	The port: • Will not become a member of the specified VLAN. • Will ignore GVRP PDUs. • Will not join any dynamic VLANs. • Will not advertise VLANs.	The port: • Will not become a member of this VLAN. • Will ignore GVRP PDUs. • Will not join any dynamic VLANs. • Will not advertise VLANs.

[1] Each port of the switch must be a Tagged or Untagged member of at least one VLAN. Thus, any port configured for GVRP to Learn or Block will generate and forward advertisements for static VLAN(s) configured on the switch and also for dynamic VLANs the switch learns on other ports.

[2] To configure tagging, **Auto**, or **Forbid**, see "Configuring Static VLAN Per-Port Settings" on page 2-37 (for the CLI) or "Adding or Changing a VLAN Port Assignment" on page 2-26 (for the menu).

As the preceding table indicates, when you enable GVRP, a port that has a Tagged or Untagged static VLAN has the option for both generating advertisements and dynamically joining other VLANs.

Note

In table 3-2, above, the Unknown VLAN parameters are configured on a per-port basis using the CLI. The Tagged, Untagged, Auto, and Forbid options are configured per static VLAN on every port, using either the menu interface or the CLI.

Because dynamic VLANs operate as Tagged VLANs, and because a tagged port on one device cannot communicate with an untagged port on another device, HP recommends that you use Tagged VLANs for the static VLANs you will use to generate advertisements.

GVRP and VLAN Access Control

When you enable GVRP on a switch, the default GVRP parameter settings allow all of the switch's ports to transmit and receive dynamic VLAN advertisements (GVRP advertisements) and to dynamically join VLANs. The two preceding sections describe the per-port features you can use to control and limit VLAN propagation. To summarize, you can:

■ Allow a port to advertise and/or join dynamic VLANs (Learn mode—the default).

■ Allow a port to send VLAN advertisements, but not receive them from other devices; that is, the port cannot dynamically join a VLAN but other devices can dynamically join the VLANs it advertises (Block mode).

■ Prevent a port from participating in GVRP operation (Disable mode).

Port-Leave From a Dynamic VLAN

A dynamic VLAN continues to exist on a port for as long as the port continues to receive advertisements of that VLAN from another device connected to that port or until you:

■ Convert the VLAN to a static VLAN (See "Converting a Dynamic VLAN to a Static VLAN" on page 3-17.)

■ Reconfigure the port to **Block** or **Disable**

■ Disable GVRP

■ Reboot the switch

The time-to-live for dynamic VLANs is 10 seconds. That is, if a port has not received an advertisement for an existing dynamic VLAN during the last 10 seconds, the port removes itself from that dynamic VLAN.

Planning for GVRP Operation

These steps outline the procedure for setting up dynamic VLANs for a segment.

1. Determine the VLAN topology you want for each segment (broadcast domain) on your network.

2. Determine the VLANs that must be static and the VLANs that can be dynamically propagated.

3. Determine the device or devices on which you must manually create static VLANs in order to propagate VLANs throughout the segment.

4. Determine security boundaries and how the individual ports in the segment will handle dynamic VLAN advertisements. (See table 3-1 on page 3-8 and table 3-2 on page 3-10.)

5. Enable GVRP on all devices you want to use with dynamic VLANs and configure the appropriate "Unknown VLAN" parameter (**Learn**, **Block**, or **Disable**) for each port.

6. Configure the static VLANs on the switch(es) where they are needed, along with the per-VLAN parameters (**Tagged**, **Untagged**, **Auto**, and **Forbid**— see table 3-2 on page 3-10) on each port.

7. Dynamic VLANs will then appear automatically, according to the configuration options you have chosen.

8. Convert dynamic VLANs to static VLANs where you want dynamic VLANs to become permanent.

Configuring GVRP On a Switch

The procedures in this section describe how to:

■   View the GVRP configuration on a switch

■   Enable and disable GVRP on a switch

■   Specify how individual ports will handle advertisements

To view or configure static VLANs for GVRP operation, refer to "Per-Port Static VLAN Configuration Options" on page 2-12.

Menu: Viewing and Configuring GVRP

1.   From the Main Menu, select:

2. Switch Configuration …
     8. VLAN Menu …
          1. VLAN Support

```
==========================- CONSOLE - MANAGER MODE -============================
                   Switch Configuration - VLAN - VLAN Support

  Maximum VLANs to support [8] : 8
  Primary VLAN : DEFAULT_VLAN
  GVRP Enabled [No] : No



  Actions->   Cancel     Edit      Save      Help
Cancel changes and return to previous screen.
Use arrow keys to change action selection and <Enter> to execute action.
```
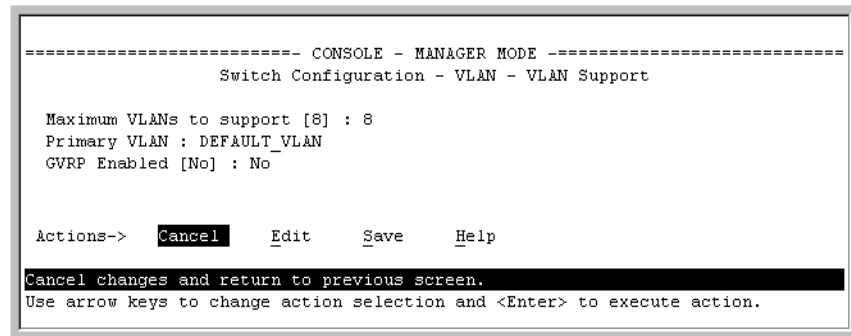
Figure 3-4. The VLAN Support Screen (Default Configuration)

2.   Do the following to enable GVRP and display the Unknown VLAN fields:

   a.   Press **[E]** (for **Edit**).

   b.   Use ↓ to move the cursor to the **GVRP Enabled** field.

   c.   Press the Space bar to select **Yes**.

   d.   Press ↓ again to display the **Unknown VLAN** fields.

The Unknown VLAN fields enable you to configure each port to:
- Learn - Dynamically join any advertised VLAN and advertise all VLANs learned through other ports.
- Block - Do not dynamically join any VLAN, but still advertise all VLANs learned through other ports.
- Disable - Ignore and drop all incoming advertisements and do not transmit any advertisements.

```
==========================- CONSOLE - MANAGER MODE -==========================
                       Switch Configuration - VLAN - VLAN Support
 Maximum VLANs to support [8] : 8
 Primary VLAN : DEFAULT_VLAN
 GVRP Enabled [No] : Yes

 Port    Type        Unknown VLAN  |  Port    Type        Unknown VLAN
 ----  ---------  + ------------   |  ----  ---------  + ------------
 A1    10/100TX   | Learn          |  A8    10/100TX   | Learn
 A2    10/100TX   | Learn          |  A9    10/100TX   | Learn
 A3    10/100TX   | Learn          |  A10   10/100TX   | Learn
 A4    10/100TX   | Learn          |  A11   10/100TX   | Learn
 A5    10/100TX   | Learn          |  A12   10/100TX   | Learn
 A6    10/100TX   | Learn          |  A13   10/100TX   | Learn
 A7    10/100TX   | Learn          |  A14   10/100TX   | Learn

 Actions->   Cancel      Edit      Save      Help


 Use arrow keys to change field selection, <Space> to toggle field choices,
 and <Enter> to go to Actions.
```

Figure 3-5. Example Showing Default Settings for Handling Advertisements

3. Use the arrow keys to select the port you want, and the Space bar to select Unknown VLAN option for any ports you want to change.

4. When you finish making configuration changes, press **[Enter]**, then **[S]** (for **S**ave) to save your changes to the Startup-Config file.

CLI: Viewing and Configuring GVRP

GVRP Commands Used in This Section

show gvrp	below
gvrp	page 3-15
unknown-vlans	page 3-15

Displaying the Switch's Current GVRP Configuration. This command shows whether GVRP is disabled, along with the current settings for the maximum number of VLANs and the current Primary VLAN. (For more on the last two parameters, see chapter 2, "Static Virtual LANs (VLANs)" .)

Syntax: show gvrp   *Shows the current settings.*

```
ProCurve> show gvrp
 GVRP support
  Maximum VLANs to support : 8
  Primary VLAN : DEFAULT_VLAN
  GVRP Enabled : No
```

Figure 3-6. Example of "Show GVRP" Listing with GVRP Disabled

```
ProCurve> show gvrp
 GVRP support
  Maximum VLANs to support : 8
  Primary VLAN : DEFAULT_VLAN
  GVRP Enabled : Yes

  Port Type       | Unknown VLAN
  ---- ---------- + ------------
  A1   10/100TX   | Learn
  A2   10/100TX   | Learn
  A3   10/100TX   | Block
  A4   10/100TX   | Disable
  A5   10/100TX   | Disable
  A6   10/100TX   | Learn
  A7   10/100TX   | Learn
   .        .     |    .
   .        .     |    .
   .        .     |    .
```

This example includes non-default settings for the Unknown VLAN field for some ports.

Figure 3-7. Example of Show GVRP Listing with GVRP Enabled

Enabling and Disabling GVRP on the Switch. This command enables GVRP on the switch.

Syntax: gvrp

This example enables GVRP:

```
ProCurve(config)# gvrp
```

This example disables GVRP operation on the switch:

```
ProCurve(config)# no gvrp
```

Enabling and Disabling GVRP On Individual Ports. When GVRP is enabled on the switch, use the **unknown-vlans** command to change the Unknown VLAN field for one or more ports. You can use this command at either the Manager level or the interface context level for the desired port(s).

Syntax: interface < *port-list* > unknown-vlans < learn | block | disable >

Changes the Unknown VLAN field setting for the specified port(s).

For example, to change and view the configuration for ports A1-A2 to **Block**:

```
ProCurve(config)interface a1-a2 unknown-vlans block

ProCurve(config) show gvrp
GVRP support
 Maximum VLANs to support : 8
 Primary VLAN : DEFAULT_VLAN
 GVRP Enabled : Yes

Port Type      | Unknown VLAN
---- --------- + ------------
1    10/100TX  | Block
2    10/100TX  | Block
3    10/100TX  | Learn
4    10/100TX  | Learn
 •      •           •
 •      •           •
 •      •           •
```

Figure 3-8. Displaying the Static and Dynamic VLANs Active on the Switch

Syntax: show vlans

The **show vlans** *command lists all VLANs present in the switch.*

For example, in the following illustration, switch "B" has one static VLAN (the default VLAN), with GVRP enabled and port 1 configured to **Learn** for Unknown VLANs. Switch "A" has GVRP enabled and has three static VLANs: the default VLAN, VLAN-222, and VLAN-333. In this scenario, switch B will dynamically join VLAN-222 and VLAN-333:

```
┌──────────────────────┐                              ┌──────────────────────┐
│ Switch "A"           │                              │ Switch "B"           │
│ GVRP enabled.        │           ┌─────────────┐    │ GVRP enabled.        │
│ 3 Static VLANs:      │           │ Port 1: Set to│   │ 1 Static VLANs:      │
│  − DEFAULT_VLAN      │           │ "Learn" Mode │    │  − DEFAULT_VLAN      │
│  − VLAN-222          │           └─────────────┘    │                      │
│  − VLAN-333          │                              │                      │
└──────────────────────┘                              └──────────────────────┘
```
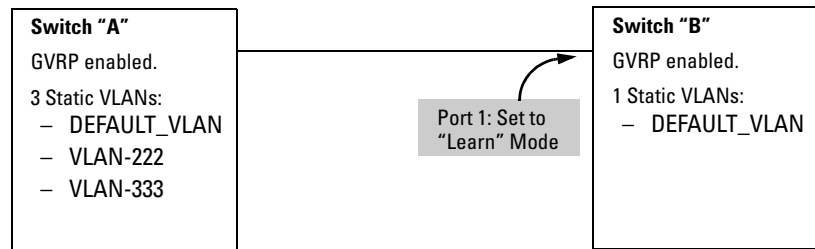
The **show vlans** command lists the dynamic (and static) VLANs in switch "B" after it has learned and joined VLAN-222 and VLAN-333.

```
Switch-B> show vlans

   Status and Counters - VLAN Information

   VLAN support : Yes
   Maximum VLANs to support : 8                   Dynamic VLANs
   Primary VLAN : DEFAULT_VLAN                     Learned from
                                                   Switch "A"
                                                   through Port 1
   802.1Q VLAN ID Name            Status
   -------------- ------------- ----------
   1              DEFAULT_VLAN  Static
   222            GVRP_222      Dynamic
   333            GVRP_333      Dynamic
```

Figure 3-9. Example of Listing Showing Dynamic VLANs

Converting a Dynamic VLAN to a Static VLAN. If a port on the switch has joined a dynamic VLAN, you can use the following command to convert that dynamic VLAN to a static VLAN:

Syntax: static < *dynamic-vlan-id* >

> *Converts the a dynamic VLAN to a static VLAN.*

For example, to convert dynamic VLAN 333 (from the previous example) to a static VLAN:

```
ProCurve(config)# static 333
```

When you convert a dynamic VLAN to a static VLAN, all ports on the switch are assigned to the VLAN in Auto mode.

Web: Viewing and Configuring GVRP

To view, enable, disable, or reconfigure GVRP:

1. Click on the **Configuration** tab.

2. Click on **[VLAN Configuration]** and do the following:
   - To enable or disable GVRP, click on **GVRP Enabled**.
   - To change the Unknown VLAN field for any port:
     i. Click on **[GVRP Security]** and make the desired changes.
     ii. Click on **[Apply]** to save and implement your changes to the Unknown VLAN fields.

For web-based Help on how to use the web browser interface screen, click on the **[?]** button provided on the web browser screen.

GVRP Operating Notes

- A dynamic VLAN must be converted to a static VLAN before it can have an IP address.

- The total number of VLANs on the switch (static and dynamic combined) cannot exceed the current Maximum VLANs setting. For example, in the factory default state, the switch supports eight VLANs. Thus, in a case where four static VLANs are configured on the switch, the switch can accept up to four additional VLANs in any combination of static and dynamic. Any additional VLANs advertised to the switch will not be added unless you first increase the Maximum VLANs setting. In the Menu interface, click on **2. Switch Configuration …** | **8. VLAN Menu** | **1. VLAN Support**. In the global config level of the CLI, use **max-vlans**.

- Converting a dynamic VLAN to a static VLAN and then executing the **write memory** command saves the VLAN in the startup-config file and makes it a permanent part of the switch's VLAN configuration.

- Within the same broadcast domain, a dynamic VLAN can pass through a device that is not GVRP-aware. This is because a hub or a switch that is not GVRP-ware will flood the GVRP (multicast) advertisement packets out all ports.

- GVRP assigns dynamic VLANs as Tagged VLANs. To configure the VLAN as Untagged, you must first convert it to a static VLAN.

■   Rebooting a switch on which a dynamic VLAN exists deletes that VLAN. However, the dynamic VLAN re-appears after the reboot if GVRP is enabled and the switch again receives advertisements for that VLAN through a port configured to add dynamic VLANs.

■   By receiving advertisements from other devices running GVRP, the switch learns of static VLANs on those other devices and dynamically (automatically) creates tagged VLANs on the links to the advertising devices. Similarly, the switch advertises its static VLANs to other GVRP-aware devices, as well as the dynamic VLANs the switch has learned.

■   A GVRP-enabled switch does not advertise any GVRP-learned VLANs out of the port(s) on which it originally learned of those VLANs.

■   While GVRP is enabled on the switch, you cannot apply any ACLs to VLANs configured on the same switch.

■   A VLAN enabled for jumbo traffic cannot be used to create a dynamic VLAN. A port belonging to a statically configured, jumbo-enabled VLAN cannot join a dynamic VLAN.

— This page intentionally unused. —

4

Multimedia Traffic Control with IP Multicast (IGMP)

Contents

Overview

This chapter describes multimedia traffic control with IP multicast (IGMP) to reduce unnecessary bandwidth usage on a per-port basis, and how to configure it with the switch's built-in interfaces:

For general information on how to use the switch's built-in interfaces, refer to these chapters in the *Management and Configuration Guide* for your switch:

■ Chapter 3, "Using the Menu Interface"

■ Chapter 4, "Using the Command Line Interface (CLI)"

■ Chapter 5, "Using the Web Browser Interface

■ Chapter 6, "Switch Memory and Configuration"

IGMP General Operation and Features

IGMP Features

Feature	Default	Menu	CLI
view igmp configuration	n/a	—	page 4-6
show igmp status for multicast groups used by the selected VLAN	n/a	—	Yes
enabling or disabling IGMP (Requires VLAN ID Context)	disabled	—	page 4-8
per-port packet control	auto	—	page 4-9
IGMP traffic priority	normal	—	page 4-10
querier	enabled	—	page 4-10
fast-leave	disabled	—	page 4-13

In a network where IP multicast traffic is transmitted for various multimedia applications, you can use the switch to reduce unnecessary bandwidth usage on a per-port basis by configuring IGMP (Internet Group Management Protocol controls). In the factory default state (IGMP disabled), the switch simply floods all IP multicast traffic it receives on a given VLAN through all ports on that VLAN (except the port on which it received the traffic). This can result in significant and unnecessary bandwidth usage in networks where IP multicast traffic is a factor. Enabling IGMP allows the ports to detect IGMP queries and report packets and manage IP multicast traffic through the switch.

IGMP is useful in multimedia applications such as LAN TV, desktop conferencing, and collaborative computing, where there is multipoint communication; that is, communication from one to many hosts, or communication originating from many hosts and destined for many other hosts. In such multipoint applications, IGMP will be configured on the hosts, and multicast traffic will be generated by one or more servers (inside or outside of the local network). Switches in the network (that support IGMP) can then be configured to direct the multicast traffic to only the ports where needed. If multiple VLANs are configured, you can configure IGMP on a per-VLAN basis.

Enabling IGMP allows detection of IGMP queries and report packets in order to manage IP multicast traffic through the switch. If no other querier is detected, the switch will then also function as the querier. (If you need to disable the querier feature, you can do so through the IGMP configuration MIB. Refer to "Changing the Querier Configuration Setting" on page 4-10.)

| **N o t e** | IGMP configuration on the Series 5300xl switches and 4200vl switches oper-ates at the VLAN context level. If you are not using VLANs, then configure IGMP in VLAN 1 (the default VLAN) context. |

IGMP Terms

■ **IGMP Device:** A switch or router running IGMP traffic control features.

■ **IGMP Host:** An end-node device running an IGMP (multipoint, or multicast communication) application.

■ **Querier:** A required IGMP device that facilitates the IGMP protocol and traffic flow on a given LAN. This device tracks which ports are connected to devices (IGMP clients) that belong to specific multicast groups, and triggers updates of this information. A querier uses data received from the queries to determine whether to forward or block multicast traffic on specific ports. When the switch has an IP address on a given VLAN, it automatically operates as a Querier for that VLAN if it does not detect a multicast router or another switch functioning as a Querier. When enabled (the default state), the switch's querier function eliminates the need for a multicast router. In most cases, HP recommends that you leave this parameter in the default "enabled" state even if you have a multicast router performing the querier function in your multicast group. For more information, see "How IGMP Operates" on page 4-11.

IGMP Operating Features

Basic Operation

In the factory default configuration, IGMP is disabled. To enable IGMP

- If multiple VLANs are not configured, you configure IGMP on the default VLAN (DEFAULT_VLAN; VID = 1).

- If multiple VLANs are configured, you configure IGMP on a per-VLAN basis for every VLAN where this feature is to be used.

Enhancements

With the CLI, you can configure these additional options:

- **Forward with High Priority.** Disabling this parameter (the default) causes the switch or VLAN to process IP multicast traffic, along with other traffic, in the order received (usually, normal priority). Enabling this parameter causes the switch or VLAN to give a higher priority to IP multicast traffic than to other traffic.

- **Auto/Blocked/Forward:** You can use the console to configure individual ports to any of the following states:
    - **Auto** (the default): Causes the switch to interpret IGMP packets and to filter IP multicast traffic based on the IGMP packet information for ports belonging to a multicast group. This means that IGMP traffic will be forwarded on a specific port only if an IGMP host or multicast router is connected to the port.
    - **Blocked:** Drop all inbound IGMP protocol packets on the specified ports. This has the effect of preventing IGMP traffic from moving through specific ports.
    - **Forward:** Causes the switch to forward all IGMP and IP multicast transmissions through the port.

- **Operation With or Without IP Addressing:** This feature helps to conserve IP addresses by enabling IGMP to run on VLANs that do not have an IP address. See "Operation With or Without IP Addressing" on page 4-12.

- **Querier Capability:** The switch performs this function for IGMP on VLANs having an IP address when there is no other device in the VLAN acting as querier. See "Using the Switch as Querier" on page 4-18.

N o t e s   Whenever IGMP is enabled, the switch generates an Event Log message
                     indicating whether querier functionality is enabled.

                     IP multicast traffic groups are identified by IP addresses in the range of
                     224.0.0.0 to 239.255.255.255. Also, incoming IGMP packets intended for
                     reserved, or "well-known" multicast addresses automatically flood through all
                     ports (except the port on which the packets entered the switch). For more on
                     this topic, see "Excluding Well-Known or Reserved Multicast Addresses from
                     IP Multicast Filtering" on page 4-19.

                     For more information, refer to "How IGMP Operates" on page 4-11.

CLI: Configuring and Displaying IGMP

IGMP Commands Used in This Section

show ip igmp configuration	page 4-7
ip igmp	page 4-8
high-priority-forward	page 4-10
auto <[ethernet] ***<port-list>***	page 4-9
blocked <[ethernet] ***<port-list>***	page 4-9
forward <[ethernet] ***<port-list>***	page 4-9
querier	page 4-10
show ip igmp	Refer to the section titled "Internet Group Management Protocol (IGMP) Status" in appendix B of the *Management and Configuration Guide* for your switch.

Note for 3400cl and   IGMP and ACLs share resources. If ACLs use up all allotted masks, then IGMP
6400cl Switches   cannot be configured on the switch. Conversely, configuring IGMP on any
                             VLAN uses one mask resource on all switch ports, which reduces by 1 the
                             number of masks available for ACL configuration.

Viewing the Current IGMP Configuration. This command lists the IGMP configuration for all VLANs configured on the switch or for a specific VLAN.

Syntax: show ip igmp config

Displays IGMP configuration for all VLANs on the switch.

show ip igmp vlan < *vid* > config

Displays IGMP configuration for a specific VLAN on the switch, including per-port data.

(For IGMP operating status, refer to the section titled "Internet Group Management Protocol (IGMP) Status" in appendix B, "Monitoring and Analyzing Switch Operation" of the Management and Configuration Guide for you switch.)

For example, suppose you have the following VLAN and IGMP configurations on the switch:

VLAN ID	VLAN Name	IGMP Enabled	Forward with High Priority	Querier
1	DEFAULT_VLAN	Yes	No	No
22	VLAN-2	Yes	Yes	Yes
33	VLAN-3	No	No	No

You could use the CLI to display this data as follows:

```
ProCurve> show ip igmp config
 IGMP Service
  VLAN ID       VLAN NAME     IGMP Enabled Forward with High Priority Querier
  ------------ ------------ ------------ -------------------------- -------
    1            DEFAULT_VLAN Yes          No                         No
   22            VLAN-2       Yes          Yes                        Yes
   33            VLAN-3       No           No                         Yes
```

Figure 4-1. Example Listing of IGMP Configuration for All VLANs in the Switch

The following version of the **show ip igmp** command includes the VLAN ID (*vid*) designation, and combines the above data with the IGMP per-port configuration:

```
                          ProCurve(config)# show ip igmp 1 config
IGMP Configuration         IGMP Service
for the Selected          ╱ ‾ ‾ ‾ ‾ ‾ ‾ ‾ ‾ ‾ ‾ ‾ ‾ ‾ ‾ ‾ ╲
VLAN                      ╱   VLAN ID : 1                      ╲
                         │    VLAN NAME    : DEFAULT_VLAN       │
                    ───▶ │    IGMP Enabled : Yes                │
                         ╲    Forward with High Priority : No   ╱
                          ╲ _ Querier Allowed : Yes _ _ _ _ _ ╱

                          ╱ ‾ ‾ ‾ ‾ ‾ ‾ ‾ ‾ ‾ ‾ ‾ ‾ ‾ ‾ ╲
                         ╱  Port Type      | IP Mcast       ╲
                         │  ---- --------- + --------        │
IGMP Configuration       │  A1   100/1000T | Auto            │
On the Individual        │  A2   100/1000T | Auto            │
Ports in the VLAN        │  A3   100/1000T | Forward          │
                    ───▶ │  A4   100/1000T | Forward          │
                         │  A5   100/1000T | Blocked          │
                         │  A6   100/1000T | Blocked          │
                         │   •      •          •             │
                         ╲   •      •          •            ╱
                          ╲ _•_ _ _ •_ _ _ _ _ •_ _ _ _ _ ╱
```

Figure 4-2. Example Listing of IGMP Configuration for A Specific VLAN

Enabling or Disabling IGMP on a VLAN. You can enable IGMP on a VLAN, along with the last-saved or default IGMP configuration (whichever was most recently set), or you can disable IGMP on a selected VLAN.

Syntax: [no] ip igmp

> *Enables IGMP on a VLAN. Note that this command must be executed in a VLAN context.*

For example, here are methods to enable and disable IGMP on the default VLAN (VID = 1).

```
ProCurve(config)# vlan 1 ip igmp
```
> *Enables IGMP on VLAN 1.*
```
ProCurve(vlan-1)# ip igmp
```
> *Same as above.*
```
ProCurve(config)# no vlan 1 ip igmp
```
> *Disables IGMP on vlan 1.*

N o t e

If you disable IGMP on a VLAN and then later re-enable IGMP on that VLAN, the switch restores the last-saved IGMP configuration for that VLAN. For more on how switch memory operates, refer to the chapter titled "Switch Memory and Configuration" in the *Management and Configuration Guide* for your switch.

You can also combine the ip igmp command with other IGMP-related commands, as described in the following sections.

Configuring Per-Port IGMP Traffic Filters.

Syntax: vlan < *vid* > ip igmp [auto < *port-list* > | blocked < *port-list* > | forward < *port-list* >]

> *Used in the VLAN context, this command specifies how each port should handle IGMP traffic. (Default:* **auto**.*)*
> *Note: Where a static multicast filter is configured on a port, and an IGMP filter created by this command applies to the same port, the IGMP filter overrides the static multicast filter for any inbound multicast traffic carrying the same multicast address as is configured in the static filter. (Refer to the section titled "Filter Types and Operation" in the "Port Traffic Controls" chapter of the Management and Configuration Guide for your switch.*

For example, suppose you wanted to configure IGMP as follows for VLAN 1 on the 100/1000T ports on a module in slot 1:

Ports A1-A2	auto	Filter multicast traffic. Forward IGMP traffic to hosts on these ports that belong to the multicast group for which the traffic is intended. (Also forward any multicast traffic through any of these ports that is connected to a multicast router.)
Ports A3-A4	forward	Forward all multicast traffic through this port.
Ports A5-A6	blocked	Drop all inbound IGMP protocol packets on the specified ports.

Depending on the privilege level, you could use one of the following commands to configure IGMP on VLAN 1 with the above settings:

```
ProCurve(config)# vlan 1 ip igmp auto a1,a2 forward a3,a4
blocked a5,a6

ProCurve(config)# ip igmp auto a1,a2 forward a3,a4 blocked
a5,a6
```

The following command displays the VLAN and per-port configuration result-ing from the above commands.

```
ProCurve> show igmp vlan 1 config
```

Configuring IGMP Traffic Priority.

Syntax: vlan < *vid* > ip igmp high-priority-forward

> *This command assigns "high" priority to IGMP traffic or returns a high-priority setting to "normal" priority. (The traffic will be serviced at its inbound priority.) (Default:* **norma***l.)*

```
ProCurve(config)# vlan 1 ip igmp high-priority-forward
```

> *Configures high priority for IGMP traffic on VLAN 1.*

```
ProCurve(vlan-1)# ip igmp high-priority-forward
```

> *Same as above command, but in the VLAN 1 context level.*

```
ProCurve(vlan 1)# no ip igmp high-priority-forward
```

> *Returns IGMP traffic to "normal" priority.*

```
ProCurve> show ip igmp config
```

> *Show command to display results of above high-priority commands.*

Configuring the Querier Function.

Syntax: [no] vlan *<vid>* ip igmp querier

> *This command disables or re-enables the ability for the switch to become querier if necessary. The* **no** *version of the command disables the querier function on the switch. The* **show ip igmp config** *command displays the current querier command. (Default Querier Capability: Enabled.)*

How IGMP Operates

The Internet Group Management Protocol (IGMP) is an internal protocol of the Internet Protocol (IP) suite. IP manages multicast traffic by using switches, multicast routers, and hosts that support IGMP. (In Hewlett-Packard's implementation of IGMP, a multicast router is not necessary as long as a switch is configured to support IGMP with the **querier** feature enabled.) A set of hosts, routers, and/or switches that send or receive multicast data streams to or from the same source(s) is termed a *multicast group*, and all devices in the group use the same multicast group address. The multicast group running version 2 of IGMP uses three fundamental types of messages to communicate:

■   **Query:** A message sent from the querier (multicast router or switch) asking for a response from each host belonging to the multicast group. If a multicast router supporting IGMP is not present, then the switch must assume this function in order to elicit group membership information from the hosts on the network. (If you need to disable the querier feature, you can do so through the CLI, using the IGMP configuration MIB. See "Configuring the Querier Function" on page 4-10.)

■   **Report (Join):** A message sent by a host to the querier to indicate that the host wants to be or is a member of a given group indicated in the report message.

■   **Leave Group:** A message sent by a host to the querier to indicate that the host has ceased to be a member of a specific multicast group.

Note on IGMP version 3 support

When an IGMPv3 Join is received by the switch, it accepts the host request and begins to forward the IGMP traffic. This means that ports which have not joined the group and are not connected to routers or the IGMP Querier will not receive the group's multicast traffic.

The switch does not support the IGMPv3 "Exclude Source" or "Include Source" options in the Join Reports. Rather, the group is simply joined from all sources.

The switch does not support becoming a version 3 Querier. It will become a version 2 Querier in the absence of any other Querier on the network.

An IP multicast packet includes the multicast group (address) to which the packet belongs. When an IGMP client connected to a switch port needs to receive multicast traffic from a specific group, it joins the group by sending an IGMP report (join request) to the network. (The multicast group specified

in the join request is determined by the requesting application running on the IGMP client.) When a networking device with IGMP enabled receives the join request for a specific group, it forwards any IP multicast traffic it receives for that group through the port on which the join request was received. When the client is ready to leave the multicast group, it sends a Leave Group message to the network and ceases to be a group member. When the leave request is detected, the appropriate IGMP device will cease transmitting traffic for the designated multicast group through the port on which the leave request was received (as long as there are no other current members of that group on the affected port).

Thus, IGMP identifies members of a multicast group (within a subnet) and allows IGMP-configured hosts (and routers) to join or leave multicast groups.

IGMP Data. To display data showing active group addresses, reports, queries, querier access port, and active group address data (port, type, and access), refer to the section titled "Internet Group Management Protocol (IGMP) Status" in appendix B, "Monitoring and Analyzing Switch Operation" of the *Management and Configuration Guide* for you switch.).

Operation With or Without IP Addressing

You can configure IGMP on VLANs that do not have IP addressing. The benefit of IGMP without IP addressing is a reduction in the number of IP addresses you have to use and configure. This can be significant in a network with a large number of VLANs. The limitation on IGMP without IP addressing is that the switch cannot become Querier on any VLANs for which it has no IP address—so the network administrator must ensure that another IGMP device will act as Querier. It is also advisable to have an additional IGMP device available as a backup Querier. See the following table.

Table 4-1.Comparison of IGMP Operation With and Without IP Addressing

IGMP Function Available With IP Addressing Configured on the VLAN	Available *Without* IP Addressing?	Operating Differences Without an IP Address
Forward multicast group traffic to any port on the VLAN that has received a join request for that multicast group.	Yes	None
Forward join requests (reports) to the Querier.	Yes	None
Configure individual ports in the VLAN to **Auto** (the default)/**Blocked**, or **Forward**.	Yes	None

IGMP Function Available With IP Addressing Configured on the VLAN	Available *Without* IP Addressing?	Operating Differences Without an IP Address
Configure IGMP traffic forwarding to normal or high-priority forwarding.	Yes	None
Age-Out IGMP group addresses when the last IGMP client on a port in the VLAN leaves the group.	Yes	Requires that another IGMP device in the VLAN has an IP address and can operate as Querier. This can be a multicast router or another switch configured for IGMP operation. (HP recommends that the VLAN also include a device operating as a backup Querier in case the device operating as the primary Querier fails for any reason.
Support Fast-Leave IGMP and Forced Fast-Leave IGMP (below).	Yes	
Support automatic Querier election.	No	Querier operation not available.
Operate as the Querier.	No	Querier operation not available.
Available as a backup Querier.	No	Querier operation not available.

Automatic Fast-Leave IGMP

Fast-Leave IGMP. Depending on the switch model, Fast-Leave is enabled or disabled in the default configuration.

Switch Model or Series	Data-Driven IGMP Included?	IGMP Fast-Leave Setting	Default IGMP Behavior
Switch 6400cl Switch 5300xl Switch 4200vl Switch 3400cl Switch 2500	Yes	Always Enabled	Drops unjoined mulitcast traffic except for always-fowarded traffic toward the Querier or multicast routers, and out of IGMP-forward ports. Selectively forwards joined multicast traffic.
Switch 2600 Switch 2600-PWR Switch 4100gl Switch 6108	No	Disabled in the Default Configuration	IGMP Fast-Leave disabled in the default configuration. Floods unjoined multicast traffic to all ports. Selectively forwards joined multicast traffic.

On switches that do not support Data-Driven IGMP, unregistered multicast groups are flooded to the VLAN rather than pruned. In this scenario, Fast-Leave IGMP can actually increase the problem of multicast flooding by removing the IGMP group filter before the Querier has recognized the IGMP

leave. The Querier will continue to transmit the multicast group during this short time, and because the group is no longer registered the switch will then flood the multicast group to all ports.

On ProCurve switches that do support Data-Driven IGMP ("Smart" IGMP), when unregistered multicasts are received the switch automatically filters (drops) them. Thus, the sooner the IGMP Leave is processed, the sooner this multicast traffic stops flowing.

Because of the multicast flooding problem mentioned above, the IGMP Fast-Leave feature is disabled by default on all ProCurve switches that do not support Data-Driven IGMP. (See the table above.) The feature can be enabled on these switches via an SNMP set of this object:

    hpSwitchIgmpPortForceLeaveState.<vid>.<port number>

However, this is not recommended as this will increase the amount of multi-cast flooding during the period between the client's IGMP Leave and the Querier's processing of that Leave. For more information on this tope refer to "Forced Fast-Leave IGMP" on page page 4-16.

Automatic Fast-Leave Operation. If a switch port has the following char-acteristics, then the Fast-Leave operation will apply:

1. Connected to only one end node

2. The end node currently belongs to a multicast group; i.e. is an IGMP client

3. The end node subsequently leaves the multicast group

Then the switch does not need to wait for the Querier status update interval, but instead immediately removes the IGMP client from its IGMP table and ceases transmitting IGMP traffic to the client. (If the switch detects multiple end nodes on the port, automatic Fast-Leave does not activate—regardless of whether one or more of these end nodes are IGMP clients.)

In the next figure, automatic Fast-Leave operates on the switch ports for IGMP clients "3A" and "5A", but not on the switch port for IGMP clients "7A" and 7B, Server "7C", and printer "7D".

Fast-Leave IGMP automatically operates on the ports connected to IGMP clients 3A and 5A, but does not operate on the port connected to Switch 7X because the Series 5300xl switch detects multiple end nodes on that port.

Fast-Leave IGMP activates on these two ports.

Fast-Leave IGMP does not activate on this port.
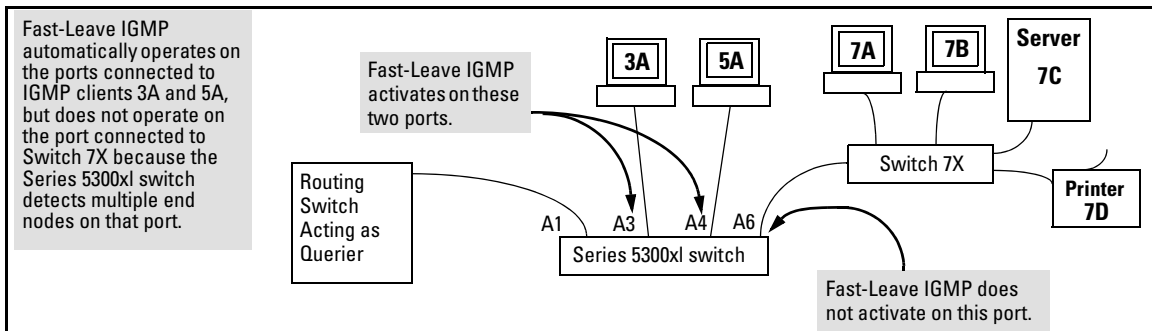
Figure 4-3. Example of Automatic Fast-Leave IGMP Criteria

When client "3A" running IGMP is ready to leave the multicast group, it transmits a Leave Group message. Because the switch knows that there is only one end node on port A3, it removes the client from its IGMP table and halts multicast traffic (for that group) to port A3. If the switch is not the Querier, it does not wait for the actual Querier to verify that there are no other group members on port A3. If the switch itself is the Querier, it does not query port A3 for the presence of other group members.

Note that Fast-Leave operation does not distinguish between end nodes on the same port that belong to different VLANs. Thus, for example, even if all of the devices on port A6 in figure 4-3 belong to different VLANs, Fast-Leave does not operate on port A6.

Default (Enabled) IGMP Operation Solves the "Delayed Leave" Problem. Fast-leave IGMP is enabled by default. When Fast-leave is disabled and multiple IGMP clients are connected to the same port on an IGMP device (switch or router), if only one IGMP client joins a given multicast group, then later sends a Leave Group message and ceases to belong to that group, the switch automatically retains that IGMP client in its IGMP table and continues forwarding IGMP traffic to the IGMP client until the Querier triggers confirmation that no other group members exist on the same port. This delayed leave operation means that the switch continues to transmit unnecessary multicast traffic through the port until the Querier renews multicast group status.

Configuring Fast-Leave IGMP

Syntax: [no] ip igmp fastleave < *port-list* >

Enables IGMP fast-leaves on the specified ports in the selected VLAN. The **no** *form of the command disables IGMP fast-leave on the specified ports in the selected VLAN. Use* **show running** *to display the ports per-VLAN on which Fast-Leave is disabled.*

Forced Fast-Leave IGMP

When enabled, Forced Fast-Leave IGMP speeds up the process of blocking unnecessary IGMP traffic to a switch port that is connected to multiple end nodes. (This feature does not activate on ports where the switch detects only one end node). For example, in figure 4-3, even if you configured Forced Fast-Leave on all ports in the switch, the feature would activate only on port A6 (which has multiple end nodes) when a Leave Group request arrived on that port.

When a port having multiple end nodes receives a Leave Group request from one end node for a given multicast group "X", Forced Fast-Leave activates and waits a small amount of time to receive a join request from any other group "X" member on that port. If the port does not receive a join request for that group within the forced-leave interval, the switch then blocks any further group "X" traffic to the port.

Configuring Forced Fast-Leave IGMP

Syntax: [no] vlan < *vid* > ip igmp forcedfastleave <port-list>

Enables IGMP Forced Fast-Leave on the specified ports in the selected VLAN, even if they are cascaded. (Default: Disabled.) The **no** *form of the command disables Forced Fast-Leave on the specified ports in the selected VLAN. Use* **show running** *to display the ports per-VLAN on which Forced Fast-Leave is enabled.*

To view a non-default IGMP forced fast-leave configuration on a VLAN, use the **show running-config** command. (The **show running-config** output does not include forced fast-leave if it is set to the default of **0**.)

Forced fast-leave can be used when there are multiple devices attached to a port.

Configuring Delayed Group Flush

When enabled, this feature continues to filter IGMP groups for a specified additional period of time after IGMP leaves have been sent. The delay in flushing the group filter prevents unregistered traffic from being forwarded by the server during the delay period. In practice, this is rarely necessary on 5300xl or 3400cl switches, which support data-driven IGMP. (Data-Driven IGMP, which is enabled by default, prunes off any unregistered IGMP streams detected on the switch.)

Syntax: igmp delayed-flush < *time-period* >

> *Where leaves have been sent for IGMP groups, enables the switch to continue to flush the groups for a specified period of time. This command is applied globally to all IGMP-configured VLANs on the switch. Range:* **0 - 255**; *Default: Disabled (**0**).*

Syntax: show igmp delayed-flush

> *Displays the current* **igmp delayed-flush** *setting.*

Using the Switch as Querier

The function of the IGMP Querier is to poll other IGMP-enabled devices in an IGMP-enabled VLAN to elicit group membership information. The switch performs this function if there is no other device in the VLAN, such as a multicast router, to act as Querier. Although the switch automatically ceases Querier operation in an IGMP-enabled VLAN if it detects another Querier on the VLAN, you can also use the switch's CLI to disable the Querier capability for that VLAN.

N o t e   A Querier is required for proper IGMP operation. For this reason, if you disable the Querier function on a switch, ensure that there is an IGMP Querier (and, preferably, a backup Querier) available on the same VLAN.

If the switch becomes the Querier for a particular VLAN (for example, the DEFAULT_VLAN), then subsequently detects queries transmitted from another device on the same VLAN, the switch ceases to operate as the Querier for that VLAN. If this occurs, the switch Event Log lists a pair of messages similar to these:

```
I 01/15/01 09:01:13 igmp: DEFAULT_VLAN: Other Querier detected
I 01/15/01 09:01:13 igmp: DEFAULT_VLAN: This switch is no longer Querie
```

In the above scenario, if the other device ceases to operate as a Querier on the default VLAN, then the switch detects this change and can become the Querier as long as it is not pre-empted by some other IGMP Querier on the VLAN. In this case, the switch Event Log lists messages similar to the following to indicate that the switch has become the Querier on the VLAN:

```
I 01/15/01 09:21:55 igmp: DEFAULT_VLAN: Querier Election in process
I 01/15/01 09:22:00 igmp: DEFAULT_VLAN: This switch has been elected
```

Excluding Well-Known or Reserved Multicast Addresses from IP Multicast Filtering

Each multicast host group is identified by a single IP address in the range of 224.0.0.0 through 239.255.255.255. Specific groups of consecutive addresses in this range are termed "well-known" addresses and are reserved for pre-defined host groups. IGMP does not filter these addresses, so any packets the switch receives for such addresses are flooded out all ports assigned to the VLAN on which they were received (except the port on which the packets entered the VLAN).

The following table lists the 32 well-known address groups (8192 total addresses) that IGMP does not filter on for all switches covered by this guide.

Table 4-2.IP Multicast Address Groups Excluded from IGMP Filtering

Groups of Consecutive Addresses in the Range of 224.0.0.*X* to 239.0.0.*X**		Groups of Consecutive Addresses in the Range of 224.128.0.*X* to 239.128.0.*X**	
224.0.0.*x*	232.0.0.*x*	224.128.0.*x*	232.128.0.*x*
225.0.0.*x*	233.0.0.*x*	225.128.0.*x*	233.128.0.*x*
226.0.0.*x*	234.0.0.*x*	226.128.0.*x*	234.128.0.*x*
227.0.0.*x*	235.0.0.*x*	227.128.0.*x*	235.128.0.*x*
228.0.0.*x*	236.0.0.*x*	228.128.0.*x*	236.128.0.*x*
229.0.0.*x*	237.0.0.*x*	229.128.0.*x*	237.128.0.*x*
230.0.0.*x*	238.0.0.*x*	230.128.0.*x*	238.128.0.*x*
231.0.0.*x*	239.0.0.*x*	231.128.0.*x*	239.128.0.*x*
***X** is any value from 0 to 255.*			

N o t e s

IP Multicast Filters. *This operation applies to the Procurve Series 5300xl switches, as well as on the 1600M, 2400M, 2424M, 4000M, and 8000M, but not to the Series 2500, 2650, Series 4100gl, Series 4200vl, or 6108 switches (which do not have static traffic/security filters).*

IP multicast addresses occur in the range from 224.0.0.0 through 239.255.255.255 (which corresponds to the Ethernet multicast address range of 01005e-000000 through 01005e-7fffff). Where a switch has a static Traffic/ Security filter configured with a "Multicast" filter type and a "Multicast Address" in this range, the switch will use the static filter unless IGMP learns of a multicast group destination in this range. In this case, IGMP dynamically takes over the filtering function for the multicast destination address(es) for as long as the IGMP group is active. If the IGMP group subsequently deactivates, the switch returns filtering control to the static filter.

Reserved Addresses Excluded from IP Multicast (IGMP) Filtering. Traffic to IP multicast groups in the IP address range of 224.0.0.0 to 224.0.0.255 will always be flooded because addresses in this range are "well known" or "reserved" addresses. Thus, if IP Multicast is enabled and there is an IP multicast group within the reserved address range, traffic to that group will be flooded instead of filtered by the switch.

Number of IP Multicast Addresses Allowed

5300xl Switches. The total of IGMP filters (addresses) and static multicast filters together can range from 389 to 420, depending on the current **max-vlans** configuration. If multiple VLANs are configured, then each filter is counted once per VLAN in which it is used.

3400cl and 6400cl Switches. These switches supports up to 252 IGMP filters (addresses). If multiple VLANs are configured, then each filter is counted once per VLAN in which it is used.

5

PIM-DM (Dense Mode) on the 5300xl Switches

Contents

Overview

This chapter describes protocol-independent multicast routing operation on the ProCurve Series 5300xl switches and how to configure it with the switch's built-in interfaces, and assumes an understanding of multimedia traffic control with IP multicast (IGMP), which is described in chapter 4, "Multimedia Traffic Control with IP Multicast (IGMP)".

For general information on how to use the switch's built-in interfaces, refer to these chapters in the *Management and Configuration Guide* for your switch:

- Chapter 3, "Using the Menu Interface"
- Chapter 4, "Using the Command Line Interface (CLI)"
- Chapter 5, "Using the Web Browser Interface
- Chapter 6, "Switch Memory and Configuration"

Introduction

This feature operates only on the Series 5300xl switches.

Feature	Default	Menu	CLI	Web
Configure PIM Global	n/a	—	5-12	—
Configure PIM VLAN Interface	n/a	—	5-15	—
Display PIM Route Data	Disabled	—	5-23	—
Display PIM Status	0 (Forward All)	—	5-27	—

In a network where IP multicast traffic is transmitted for multimedia applications, multicast traffic is blocked at routed interface (VLAN) boundaries unless a multicast routing protocol is running. PIM-DM (Protocol Independent Multicasting-Dense Mode, draft version 3) enables and controls multicast traffic routing on the Switch Series 5300XL devices.

PIM-DM is used in networks where, at any given time, multicast group members exist in relatively large numbers and are present in most subnets. PIM-DM operates with any unicast IPv4 routing protocol available on the switch. However, note that PIM-DM uses "flooding" to initially propagate a multicast group to a network, then prunes back the branches that have no hosts requiring membership. For this reason, ample bandwidth is a requirement in PIM-DM applications.

IGMP provides the communication link between a host and the multicast router running PIM. Where hosts are connected directly to the routing device, such as the 5300XL, both PIM and IGMP must run on the switch.

Feature Overview

PIM-DM on the Switch Series 5300XL devices includes:

- **Routing Protocol Support:** PIM uses whichever unicast routing proto-
col is running on the routing switch. These can include:
  - RIP
  - OSPF
  - Static routes
  - Directly connected interfaces

- **Interface Support:** PIM-DM supports up to 127 outbound VLANs (and 1
inbound VLAN) in its multicast routing table (MRT) at any given time,
meaning that the sum of all outbound VLANs across all current flows on
a routing switch may not exceed 63. (A single flow may span one inbound
VLAN and up to 63 outbound VLANs, depending on the VLAN member-
ships of the hosts actively belonging to the flow.)

- **IGMP Compatibility:** PIM-DM is compatible with IGMP versions 1 - 3,
and is fully interoperable with IGMP for determining multicast flows.

- **XRRP:** PIM-DM is fully interoperable with XRRP to quickly transition
multicast routes in the event of a failover.

- **MIB Support:** With some exceptions, PIM-DM supports the parts of the
Multicast Routing MIB applicable to PIM-DM operation. (Refer to "Excep-
tions to Support for RFC 2932 - Multicast Routing MIB" on page 5-41.)

- **PIM Draft Specifications:** Compatible with PIM-DM draft specification,
versions 1 and 2.

PIM-DM Operation

PIM-DM operates at the router level to direct traffic for a particular multicast
group along the most efficient path to the VLANs having hosts that have joined
that group. A unicast source address and a multicast group address comprise
a given source/group (S/G) pair. Multicast traffic moving from a source to a
multicast group address creates a *flow* to the area(s) of the network requiring
the traffic. That is, the flow destination is the multicast group address, and not
a specific host or VLAN. Thus, a single multicast flow has one source and one

multicast group address (destination), but may reach many hosts in different subnets, depending on which hosts have issued joins for the same multicast group.

PIM routes the multicast traffic for a particular S/G pair on paths between the source unicast address and the VLANs where it is requested (by joins from hosts connected to those VLANs). Physical destinations for a particular multicast group can be hosts in different VLANs or networks. Individual hosts use IGMP configured per-VLAN to send joins requesting membership in a particular multicast group. All hosts that have joined a given multicast group (defined by a multicast address) remain in that group as long as they continue to issue periodic joins.

On the Switch Series 5300XL devices, PIM-DM interoperates with IGMP and the switch's routing protocols. (Note that PIM-DM operates independently of the routing protocol you choose to run on your switches, meaning you can use PIM-DM with RIP, OSPF, or static routes configured.) PIM-DM utilizes a unicast routing table to find the path to the originator of the multicast traffic and sets up multicast "trees" for distributing multicast traffic. (This method is termed *reverse path forwarding*, or *RPF*).

For the flow of a given multicast group, PIM-DM creates a "tree" structure between the source and the VLANs where hosts have joined the group. The tree structure consists of:

■ Extended branches to VLANs with hosts that currently belong to the group

■ Pruned branches to VLANs with no hosts that belong to the group
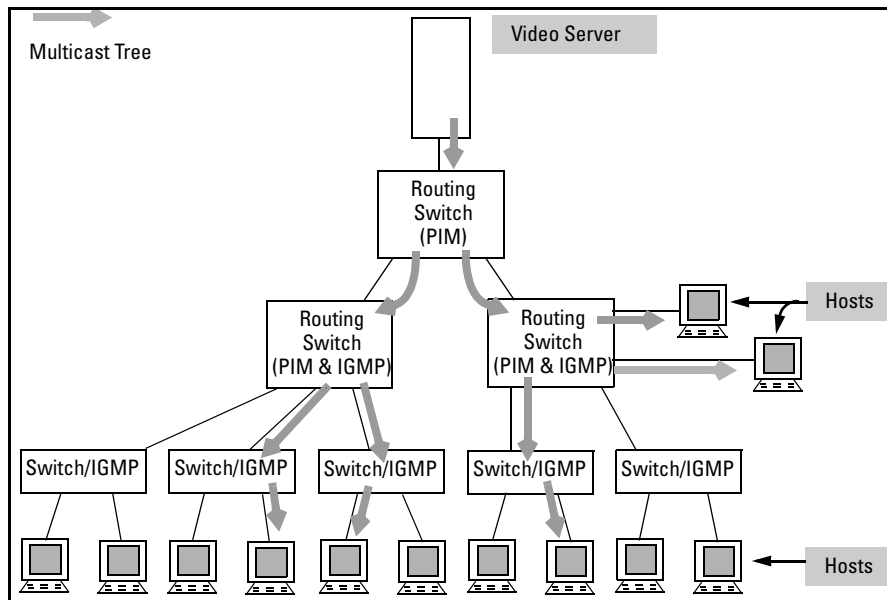
Figure 5-1. Example of Multicast "Tree" for a Given Flow

When the routing switch detects a new multicast flow, it initially floods the traffic throughout the PIM-DM domain, and then prunes the traffic on the branches (network paths) where joins have not been received from individual hosts. This creates the 'tree' structure shown above. The routing switch maintains individual branches in the multicast tree as long as there is at least one host maintaining a membership in the multicast group. When all of the hosts in a particular VLAN drop out of the group, PIM-DM prunes that VLAN from the multicast tree. Similarly, if the routing switch detects a join from a host in a pruned VLAN, it adds that branch back into the tree.

N o t e   Where the multicast routers in a network use one or more multinetted VLANs, there must be at least one subnet common to all routers on the VLAN. This is necessary to provide a continuous forwarding path for the multicast traffic on the VLAN. Refer to the **[all | source-ip-address]** option under "PIM VLAN (Interface) Configuration Context" on page 5-15.

Multicast Flow Management

This section provides details on how the routing switch manages forwarding and pruned flows. This information is useful when planning topologies to include multicast support and when viewing and interpreting the "show" command output for PIM-DM features.

Initial Flood and Prune. As mentioned earlier, when a router running PIM-DM receives a new multicast flow, it initially floods the traffic to all downstream multicast routers. PIM-DM then prunes the traffic on paths to VLANs that have no host joins for that multicast address. (Note that PIM-DM does not re-forward traffic back to its source VLAN.)

Maintaining the Prune State. For a multicast group "X" on a given VLAN, when the last host belonging to group "X" leaves the group, PIM places that VLAN in a prune state, meaning the group "X" multicast traffic is blocked to that VLAN. The prune state remains until a host on the same VLAN issues a join for group "X", in which case the router cancels the prune state and changes the flow to the forwarding state.

State Refresh Packets and Bandwidth Conservation. A 5300XL multicast router, if directly connected to a multicast source such as a video conferencing application, periodically transmits *state refresh* packets to downstream multicast routers. On routers that have pruned the multicast flow, the state refresh packets keep the pruned state alive. On routers that have been added to the network after the initial flooding and pruning of a multicast group, the state refresh packets inform the newly added router of the current state of that branch. This means that if all multicast routers in a network support the state refresh packet, then the multicast router directly connected to the multicast source performs only one flood-prune cycle to the edge of the network when a new flow (multicast group) is introduced, and preserves bandwidth for other uses. Note, however, that some vendors' multicast routers do not offer the state refresh feature. In this case, PIM-DM must periodically advertise an active multicast group to these devices by repeating the flood/prune cycle on the paths to such routers.. For better traffic management in multicast-intensive networks where some multicast routers do not offer the state refresh feature, you may want to group such routers where the increased bandwidth usage will have the least effect on overall network performance.

These ProCurve 5300xl multicast routers support the state refresh feature but must handle periodic flood-prune cycles for the downstream routers that lack this

Video Server

Video Server

Video Server

5304XL #4

5308XL #1

Other Multicast Router

Other Multicast Router

These multicast routers do not have the state refresh feature and thus require periodic flood-prune cycles to advertise active multicast group. In this case it may be better to group these routers on the same multicast tree to avoid the additional flood/ prune cycles on the routers that do support state refresh.

These ProCurve 5300xl multicast routers support the state refresh feature and do not require periodic flood-prune cycles for a given multicast group, which frees up bandwidth for other uses.

5304XL #2

ProCurve 5304XL #3

Indicates Paths Requiring Periodic Flood-Prune Cycles for a Given Multicast Group

Figure 5-2. Example of Bandwidth Conservation in 5300XL Routing Switches with PIM-DM State Refresh

General Configuration Elements

The configured elements PIM-DM requires are:

1. IP routing enabled on all routing switches you want to carry routed multicast traffic.

2. Configure the routing method(s) needed to reach the interfaces (VLANs) on which you want multicast traffic available for hosts in your network:

    *   Enable RIP or OSPF at both the global and VLAN levels on the routers where there are connected hosts that may issue multicast joins.
    *   Configure static routes to and from the destination subnets.

3. Enable IP multicast routing.

4. For each VLAN on which there are hosts that you want to join multicast groups, enable IGMP on that VLAN. Repeat this action on every switch and router belonging to the VLAN.

5. Enable PIM-DM at the global level on the routing switch and on the VLANs where you want to allow routed multicast traffic.

N o t e   When you initially enable PIM-DM, HP recommends that you leave the PIM-DM configuration parameters at their default settings. You can then assess performance and make configuration changes where a need appears.

Terminology

Flow: Multicast traffic moving between a unicast source and a multicast group. One S/G pair is counted as a single flow, regardless of the number of hosts belonging to the related multicast group.

Host: A client device that requests multicast traffic by transmitting IGMP "joins" for a specific multicast group, such as a video conferencing application.

MRT (Multicast Routing Table). The routing switch creates this table internally to maintain data on each multicast group it supports. The "Show" commands described later in this chapter display MRT data managed in this table.

Multicast Address: In IP multicast traffic on the switch, this is a single IP address that can be used by a group of related or unrelated clients wanting the same data. A single S/G pair consists of unicast source address and a multicast group address. Sometimes termed a "multicast group address". See also "Source" and "S/G Pair".

Multicast Routing: A method for transmitting multicast datagrams from a source in one IP network to a multicast address in one or more other IP networks.

PIM Neighbor: On a routing switch configured for PIM operation, a PIM neighbor is another PIM-configured routing switch or router that is either directly connected to the first routing switch or connected through networked switches and/or hubs.

Prune: To eliminate branches of a multicast tree that have no hosts sending joins to request or maintain membership in that particular multicast group.

S/G Pair: The unicast address of the server transmitting the multicast traffic and the multicast address to which the server is transmitting the traffic.

Source (S): In IP multicast traffic on the switch, the source (S) is the unicast address of the server transmitting the multicast traffic. A single S/G pair consists of unicast source address and a multicast group address. See also "S/G Pair".

PIM-DM Operating Rules

■ The routing switch supports 1022 multicast flows in hardware and up to 978 additional flows in software. (For more on this topic, refer to "Flow Capacity" on page 5-35.)

■ The multicast routing table (MRT) that PIM-DM creates allows up to 127 outbound VLANs, meaning that at any given time, PIM-DM supports multicast routing across 127 VLANs.

■ The routing switch allows one instance of PIM per VLAN. Thus, in networks using multinetted VLANs, all routers on a given VLAN intended to route multicast packets must have a least one common subnet on that VLAN. Thus, in the case of multinetting, you must select one subnet on the multinetted VLAN to use for multicast routing. To facilitate this, the routing switch provides a command for specifying which IP address PIM will use on each VLAN.

Configuring PIM-DM on the Series 5300xl Switches

PIM-DM requires configuration on both the global level and on the VLAN (interface) level. The recommended configuration order is:

1. Enable IGMP on all VLANs where hosts may join a multicast group.

2. Enable the following at the global level on the Switch Series 5300XL device.
   - IP routing
   - IP multicast routing
   - Router PIM and any non-default, global PIM settings you want to apply
   - Router RIP, Router OSPF, and/or a static route

3. If you selected RIP or OSPF in step step 2, then on each VLAN where you want multicast routing to operate, enable the same option.

4. Enable the following in each VLAN context where you want multicast routing to operate:
   - IP RIP or IP OSPF
   - IP PIM
   - Any non-default, VLAN-level IP PIM settings you want to apply

PIM Global Configuration Context

Note

PIM-DM operation requires a routing protocol enabled on the routing switch. You can use RIP, OSPF, and/or static routing. The examples in this section use RIP. For more on these topics, refer to chapter 11, "IP Routing Features", in this guide.

Syntax: [no] ip multicast-routing

> *Enables or disables IP multicast routing on the routing switch. IP routing must be enabled. (**Default:** Disabled.)*

Syntax: [no] router pim

> *Enables or disables PIM at the global level. IP routing must be enabled first. (Default: Disabled.)*

Syntax: router pim [state-refresh < 10 - 300 >]

> *Sets the interval in seconds between successive State Refresh messages originated by the routing switch. Note that only the routing switch connected directly to the unicast source initiates state-refresh packets. All other PIM routers in the network only propagate these state-refresh packets. (Default: 60 seconds)*

Syntax: [no] router pim trap < all | neighbor-loss | hardware-mrt-full | software-mrt-full>

> *Enables and disables these PIM SNMP traps:*
> **all** — *Enable/Disable all PIM notification traps.*
> **neighbor-loss** — *Enable/Disable the notification trap sent when the timer for a multicast router neighbor expires and the switch has no other multicast router neighbors on the same VLAN with a lower IP address. (Default: Disabled.)*
> **hardware-mrt-full** — *Enable/Disable notification trap when the hardware multicast routing table (MRT) is full (1023 active flows). In this state, any additional flows are handled by the software MRT, which increases processing time for the affected flows. (Default: Disabled.)*
> **software-mrt-full** — *Enable/Disable notification trap when the routing switch's software multicast routing table is full (that is, when routing resources for active flows are exhausted). (Default: Disabled.) Note that in this state, the routing switch does not accept any additional flows.*

Example of Configuring PIM at the Global Level. In figure 5-1 on page 5-6, the "5308XL #1" routing switch is directly connected to the multicast sources for the network. In this case, suppose that you want to do the following:

■ Reduce the state-refresh time from the default 60 seconds to 30 seconds. Note that the routing switch transmits state-refresh packets only if it is directly connected to the multicast source.

■ Configure an SNMP trap to notify your network management station if the routing switch's hardware multicast routing table becomes filled to the maximum of 1023 active flows.

To configure global-level PIM operation for the "5308XL #1" routing switch, you would use the commands shown in figure 5-3, below.

```
ProCurve(config)# ip routing                              Enables IP routing.
ProCurve(config)# ip multicast-routing                    Enables multicast routing.
ProCurve(config)# router pim                              Enables PIM.
ProCurve(pim)# router rip                                 Enables RIP.
ProCurve(pim)# state-refresh 45                           Configures a non-default State
                                                          Refresh timer.
ProCurve(pim)# trap hardware-mrt-full                     Sets an SNMP trap to notify an
                                                          SNMP management station if the
ProCurve(pim)# write mem                                  hardware multicast routing table
ProCurve(pim)# exit                                       fills with active flows.


          Using show config displays the configuration changes
          resulting from the above commands.


ProCurve(config)# show config
Startup configuration:
; J4850A Configuration Editor; Created on release #E.08.01
hostname "HPswitch"
module 1 type J4820A
ip routing
snmp-server community "public" Unrestricted
snmp-server host 15.29.38.205 "public" Not-INFO
snmp-server host 15.255.124.84 "public" Not-INFO
vlan 1
    name "DEFAULT_VLAN"
    untagged A1-A5,A16,A19
    ip address dhcp-bootp
    no untagged A6-A15,A17-A18,A20-A24
    exit
vlan 29
    name "VLAN29"
    .
    .
    .
    exit
vlan 25
    name "VLAN25"
    untagged A20-A24
    ip address 25.38.10.1
    exit
ip multicast-routing
ip ssh filetransfer
router rip
    exit
router pim
    trap hardware-mrt-full
    state-refresh 45
    exit
```

Figure 5-3.   Example of Configuring PIM-DM on a Routing Switch at the Global Level

After configuring the global-level PIM operation on a routing switch, go to the device's VLAN context level for each VLAN you want to include in your multicast routing domain. (Refer to "PIM VLAN (Interface) Configuration Context", below.

PIM VLAN (Interface) Configuration Context

Syntax: [no] ip pim
[no] vlan < *vid* > ip pim

> *Enables multicast routing on the VLAN interface to which the CLI is currently set. The* **no** *form disables PIM on the VLAN. Default: Disabled.*

Syntax: [no] ip pim [all | < *source-ip-address* >]
[no] vlan < *vid* > ip pim [all | < *source-ip-address* >]

> *In networks using multinetted VLANs, all routers on a given VLAN intended to route multicast packets must have a least one common subnet on that VLAN. Use this command when the VLAN is configured with multiple IP addresses (multinetting) to specify the IP address to use as the source address for PIM protocol packets outbound on the VLAN. Use* < *ip-address* > *to designate a single subnet in cases where multicast routers on the same multinetted VLAN are not configured with identical sets of subnet IP addresses . Use* < all > *if the multinetted VLAN is configured with the same set of subnet addresses. (**Default:** The Primary VLAN.)*

Syntax: ip pim [hello-interval < 5 - 30 >]
vlan < *vid* > ip pim [hello-interval < 5 - 30 >]

> *Changes the frequency at which the routing switch transmits PIM "Hello" messages on the current VLAN. The routing switch uses "Hello" packets to inform neighboring routers of its presence. The routing switch also uses this setting to compute the* **Hello Hold Time**, *which is included in Hello packets sent to neighbor routers.* **Hello Hold Time** *tells neighbor routers how long to wait for the next Hello packet from the routing switch. If another packet does not arrive within that time, the router removes the neighbor adjacency on that VLAN from the routing table, which removes any flows running on that interface. Shortening the Hello interval reduces the Hello Hold Time. This has the effect of changing how quickly other routers will stop sending traffic to the routing switch if they do not receive a new Hello packet when expected.*

*For example, if multiple routers are connected to the same VLAN and the routing switch requests multicast traffic, all routers on the VLAN receive that traffic. (Those which have pruned the traffic will drop it when they receive it.) If the upstream router loses contact with the routing switch receiving the multicast traffic (that is, fails to receive a Hello packet when expected), then the shorter Hello Interval causes it to stop transmitting multicast traffic onto the VLAN sooner, resulting in less unnecessary bandwidth usage. Not used with the **no** form of the **ip pim** command.*

Syntax: ip pim [hello-delay < 0 - 5 >]
               vlan < *vid* > ip pim [hello-delay < 0 - 5 >]

Changes the maximum time in seconds before the routing switch actually transmits the initial PIM Hello message on the current VLAN. In cases where a new VLAN activates with connections to multiple routers, if all of the connected routers sent Hello packets at the same time, then the receiving router could become momentarily overloaded. This value randomizes the transmission delay to a time between **0** *and the* **hello delay** *setting. Using "***0***" means no delay. After the routing switch sends the initial Hello Packet to a newly detected VLAN interface, it sends subsequent Hello packets according to the current* **Hello Interval** *setting. Not used with the* **no** *form of the* **ip pim** *command. Default: 5 seconds.*

Syntax: ip pim [graft-retry-interval < 1-10 >]
               vlan < *vid* > ip pim [graft-retry-interval < 1-10 >]

Graft packets result when a downstream router transmits a request to join a flow. The upstream router responds with a graft acknowledgment packet. If the Graft Ack is not received within the time period of the **graft-retry-interval***, it resends the graft packet. This command changes the interval (in seconds) the routing switch waits for the Graft Ack (acknowledgement) from another router before resending the Graft request. Not used with the* **no** *form of the* **ip pim** *command. (Default: 3 seconds.)*

Syntax: ip pim [max-graft-retries < 1 - 10 >
vlan < *vid* > ip pim [max-graft-retries < 1 - 10 >

> *Changes the number of times the routing switch will retry sending the same graft packet to join a flow. If a Graft Ack response is not received after the specified number of retries, the routing switch ceases trying to join the flow. In this case the flow is removed until either a state refresh from upstream re-initiates the flow or an upstream router floods the flow. Increasing this value helps to improve multicast reliability. Not used with the* **no** *form of the* **ip pim** *command. (Default: 3 attempts.)*

Syntax: ip pim [lan-prune-delay]
vlan < *vid* > ip pim [lan-prune-delay]

> *Enables the LAN Prune Delay option on the current VLAN. With* **lan-prune-delay** *enabled, the routing switch informs downstream neighbors how long it will wait before pruning a flow after receiving a prune request. Other, downstream routers on the same VLAN must send a Join to override the prune before the* **lan-prune-delay** *time if they want the flow to continue. This prompts any downstream neighbors with hosts continuing to belong to the flow to reply with a Join. If no joins are received after the* **lan-prune-delay** *period, the routing switch prunes the flow. The* **propagation-delay** *and* **override-interval** *settings (below) determine the* **lan-prune-delay** *setting.*
>
> *Uses the* **no** *form of the* **ip pim** *command to disable the LAN Prune Delay option. (Default: Enabled.)*

Syntax: ip pim [propagation-delay < 250-2000 >]
vlan < *vid* > ip pim [propagation-delay < 250-2000 >]

ip pim [override-interval < 500 - 6000 >]
vlan < *vid* > ip pim [override-interval < 500 - 6000 >]

> *A routing switch sharing a VLAN with other multicast
> routers uses these two values to compute the* **lan-prune-delay**
> *setting (above) for how long to wait for a PIM-DM join after
> receiving a prune packet from downstream for a particular
> multicast group. For example, a network may have multiple
> routing switches sharing VLAN "X". When an upstream
> routing switch initially floods traffic from multicast group
> "X" to VLAN "Y", if one of the routing switches on VLAN "Y"
> does not want this traffic it issues a prune response to the
> upstream neighbor. The upstream neighbor then goes into a
> "prune pending" state for group "X" on VLAN "Y". (During
> this period, the upstream neighbor continues to forward the
> traffic.) During the "pending" period, another routing
> switch on VLAN "Y" can send a group "X" Join to the
> upstream neighbor. If this happens, the upstream neighbor
> drops the "prune pending" state and continues forwarding*
>
> *the traffic. But if no routers on the VLAN send a Join, then
> the upstream router prunes group "X" from VLAN "Y" when
> the* **lan-prune-delay** *timer expires. (Defaults:* **propagation-
> delay** *= 500 milliseconds;* **override-interval** *= 2500
> milliseconds.)*

Syntax: ip pim [ttl-threshold < 0 - 255 >]
            vlan < *vid* > ip pim

> *Sets the multicast datagram time-to-live (router hop-count)
> threshold for the VLAN. Any IP multicast datagrams or state
> refresh packets with a TTL less than this threshold will not
> be forwarded out the interface. The default value of 0 means
> all multicast packets are forwarded out the interface.*
>
> *This parameter provides a method for containing multicast
> traffic within a network, or even within specific areas of a
> network. Initially, the multicast traffic source sets a TTL
> value in the packets it transmits. Each time one of these
> packets passes through a multicast routing device, the TTL
> setting decrements by 1. On a Switch Series 5300XL device,
> if the packet arrives with a TTL lower than the* **mroute ttl-
> threshold**, *the routing switch does not forward the packet.
> Changing this parameter on a routing switch requires
> knowledge of the TTL setting of incoming multicast packets.
> A value that is too high can allow multicast traffic to go
> beyond your internal network. A value that is too low may
> prevent some intended hosts from receiving the desired
> multicast traffic. (Default: 0 — forwards multicast traffic
> regardless of packet TTL setting.)*

Example of Configuring PIM-DM Operation at the VLAN Level. The
network in figure 5-4 uses VLAN 25 for multicast traffic. However, this VLAN
is multinetted and there is only one subnet (25.38.10.x) in VLAN 25 that is
common to all three routing switches. Thus, when configuring VLAN 25 on
these routing switches to perform multicast routing, it is necessary to use **ip
pim < *source-ip-address* >** to designate the common subnet as the source
address for outbound multicast traffic on VLAN 25. (If only identical subnets
were present in the multinetted VLAN 25 configuration on all three devices,
then the **ip pim all** command would be used instead.) Note that the other VLANs
in the network are not multinetted and therefore do not require the **ip pim <
all | *source-ip-address* >** option.

For this example, assume that the VLANs and IP addressing are already
configured on the routing switch.

On the three routing switches, VLAN 25 is multinetted with subnets that match in only one instance. Since subnet 25.38.10.x exists on VLAN 25 in all routing switches, it serves as the source IP address for multicast traffic outbound on VLAN 25 for the network.

The remaining VLANs (27, 28, 29, and 30) in the network are not multinetted on the routing switches and it is not necessary to configure a source address for multicast routing on these other VLANs.

In this example, the multicast source transmits packets with a TTL (time-to-live) of 192. To prevent these packets from moving beyond routers 2 and 3, you would configure the TTL in the downstream routers (below routers 2 and 3) at 190. (It is not necessary to configure the TTL on routers 1 - 3.)

Figure 5-4.   Example of a Multicast Network with a Multinetted VLAN

Figure 5-5 illustrates the configuration steps for configuring multicast routing at the VLAN level for the 5300XL #1 routing switch shown in figure 5-4.

```
ProCurve(config)# vlan 25
ProCurve(vlan-25)# ip igmp
ProCurve(vlan-25)# ip rip
ProCurve(vlan-25)# ip pim 25.38.10.1
ProCurve(vlan-25)# vlan 27

ProCurve(vlan-27)# ip igmp
ProCurve(vlan-27)# ip rip
ProCurve(vlan-27)# ip pim

ProCurve(vlan-27)# vlan 29
ProCurve(vlan-29)# ip igmp
ProCurve(vlan-29)# ip rip
ProCurve(vlan-29)# ip pim

ProCurve(vlan-29)# write mem
ProCurve(vlan-29)# exit
```

Figure 5-5.   VLAN-Level Configuration Steps To Support PIM-DM on the 5308XL #1 Device

```
ProCurve(config)# show config
Startup configuration:
; J4850A Configuration Editor; Created on release #E.08.XX

hostname 'ProCurve"
module 1 type J4820A
ip routing                                          Enables IP routing;
.                                                   required for multicast
.                                                   routing.
.
vlan 29
   name "VLAN29"
   untagged A11-A15,A17
   ip address 29.29.30.1 255.255.248.0
   ip igmp
   exit
vlan 27
   name "VLAN27"
   untagged A6-A10,A18
   ip address 27.27.30.1 255.255.248.0
   ip igmp
   exit                                             Multinetting and IGMP
vlan 25                                             enabled in VLAN 25.
   name "VLAN25"
   untagged A20-A24
   ip address 25.38.10.1 255.255.255.0
   ip address 25.38.11.1 255.255.255.0
   ip address 25.38.12.1 255.255.255.0
   ip igmp
   exit
ip multicast-routing                                Multicast Routing Configuration
router rip                                          for Global Level..
   exit
router pim
   trap hardware-mrt-full
   state-refresh 45                                 Indicates the source-IP-address
   exit                                             for multicast packets forwarded on
vlan 25                                             this VLAN.
   ip rip
   ip pim 25.38.10.1                                Multicast Routing
   ip pim ttl-threshold 2                           Configuration for VLAN 25.
   exit
vlan 27
   ip rip                                           Multicast Routing
   ip pim all                                       Configurations for VLANs
   ip pim ttl-threshold 2                           27 and 29.
   exit
vlan 29
   ip rip
   ip pim all                                       Dashed lines indicate configuration
   ip pim ttl-threshold 2                           settings affecting multicast routing.
   exit
```

**Figure 5-6.   The Configuration Supporting Multicast Routing on the 5300XL #1
Routing Switch Shown in Figure 5-4 (Page 5-20)**

Displaying PIM Data and Configuration Settings on the Series 5300xl Switches

Displaying PIM Route Data

Syntax: show ip mroute

> *Without parameters, lists all VLANs actively forwarding routed, multicast traffic.*
> **Group Address:** *The multicast address of the specific multicast group (flow).*
> **Source Address:** *The unicast address of the multicast group source.*
> **Neighbor:** *The IP address of the upstream multicast router interface (VLAN) from which the multicast traffic is coming. A blank field for a given multicast group indicates that the multicast server is directly connected to the routing switch.*
> **VLAN:** *The interface on which the multicast traffic is moving.*

For example, the next figure displays the show ip route output on the "5300XL #2" routing switch in figure 5-4 on page 5-20. This case illustrates two multicast groups from the same multicast server source.

```
ProCurve(config)# show ip mroute        Indicates the upstream multicast
                                        router interface (VLAN) from which the
 IP Multicast Route Entries             multicast traffic is coming.

  Total number of entries : 2
                                                  ↙
  Group Address    Source Address   Neighbor         VLAN
  --------------   --------------   --------------   ----
  239.255.255.1    27.27.30.2       29.29.30.1        29
  239.255.255.5    27.27.30.2       29.29.30.1        29
```

Figure 5-7.   Example Showing the Route Entry Data on the "5300XL #2" Routing Switch in Figure 5-4 on Page 5-20

Syntax: show ip mroute [interface < *vid* >]

Lists these settings:

VLAN: *The VID specified in the command.*

Protocol Identity: *PIM-DM only.*

TTL: *The time-to-live threshold for packets forwarded through this VLAN. When configured, the routing switch drops multicast packets having a TTL lower than this value. (When a packet arrives, the routing switch decrements it's TTL by 1, then compares the decremented packet TTL to the value set by this command.) A* **TTL Threshold** *setting of* **0** *(the default) means all multicast packets are forwarded regardless of the TTL value they carry. A multicast packet must have a TTL greater than 1 when it arrives at the routing switch. Otherwise the routing switch drops the packet instead of forwarding it on the VLAN.*

```
ProCurve(config)# show ip mroute interface 29

 IP Multicast Interface

  VLAN      : 29
  Protocol  : PIM-DM

  TTL Threshold : 0
```

Figure 5-8.   Example of the Above Command on the "5300XL #2" Routing Switch in Figure 5-4 on Page 5-20

Syntax: show ip mroute [< *multicast-ip-addr* > < *source-ip-addr* >]

> *Lists the following data for the specified flow (multicast group):*

> **Group Address***: The multicast group IP address for the current group.*
> **Source Address***: The multicast source address < source-ip-addr > for the current group.*
> **Source Mask***: The subnet mask applied to the multicast source address < source-ip-addr >.*
> **Neighbor***: Lists the IP address of the upstream next-hop router running PIM-DM; that is, the router from which the routing switch is receiving datagrams for the current multicast group. This value is* **0.0.0.0** *if the routing switch has not detected the upstream next-hop router's IP address. This field is empty if the multicast server is directly connected to the routing switch.*
> **VLAN***: Lists the VLAN ID (VID) on which the routing switch received the specified multicast flow.*

> **Up Time (Sec)***: The elapsed time in seconds since the routing switch learned the information for the current instance of the indicated multicast flow.*

> **Expiry Time (Sec)***: Indicates the remaining time in seconds before the routing switch ages-out the current flow (group membership). This value decrements until:*

> • *Reset by a state refresh packet originating from the upstream multicast router. (The upstream multicast router issues state refresh packets for the current group as long as it either continues to receive traffic for the current flow or receives state refresh packets for the current flow from another upstream multicast router.)*
> • *Reset by a new flow for the current multicast group on the VLAN.*
> • *The timer expires (reaches* **0***). In this case the switch has not received either a state refresh packet or new traffic for the current multicast group, and ages-out (drops) the group entry.*

Multicast Routing Protocol: *Identifies the multicast routing protocol through which the current flow was learned.*

Unicast Routing Protocol: *Identifies the routing protocol through which the routing switch learned the upstream interface for the current multicast flow. The listed protocol will be either* **RIP**, **OSPF**, *or* **Static Route**.

Downstream Interfaces:

VLAN: *Lists the VID of the VLAN that the routing switch is using to send the outbound packets of the current multicast flow to the next-hop router.*
State: *Indicates whether the outbound VLAN and next-hop router for the current multicast flow are receiving datagrams.*

– **Pruned**: *The routing switch has not detected any joins from the current multicast flow and is not currently forwarding datagrams in the current VLAN.*
– **Forwarding**: *The routing switch has received a join for the current multicast flow and is forwarding datagrams in the current VLAN.*

Up Time (Sec): *Indicates the elapsed time in seconds since the routing switch learned the displayed information about the current multicast flow.*

Expiry Time: *Shows the remaining time in seconds until the Next-Hop routing switch ages-out the current flow (group membership) on the indicated VLAN. Includes the date calculated for the age-out event. This value decrements until:*

• *Reset by a state refresh packet originating from the upstream multicast router. (The upstream multicast router issues state refresh packets for the current group as long as it either continues to receive traffic for the current flow or receives state refresh packets for the current flow from another upstream multicast router.*
• *Reset by a new flow for the current multicast group on the VLAN.*
• *The timer expires (reaches* **0**). *In this case the switch has not received either a state refresh packet or new traffic for the current multicast group, and ages-out (drops) the group entry.*

Note that the "Next-Hop routing switch" is the next multicast routing switch in the path from the current multicast routing switch to the source for the displayed multicast flow.

```
ProCurve# show ip mroute 239.255.255.5 27.27.30.2

 IP Multicast Route Entry

  Group Address   : 239.255.255.5
  Source Address  : 27.27.30.2
  Source Mask     : 255.255.248.0
  Neighbor        : 13.30.229.1
  VLAN            : 27
  Up Time (sec)     : 408
  Expiry Time (sec) : 150

  Multicast Routing Protocol : PIM-DM
  Unicast Routing Protocol   : rip

 Downstream Interfaces

  VLAN State      Up Time (sec)       Expiry Time (sec)
  ---- ---------- ------------------- -------------------
  28   pruned     408                 98
```

A blank **Neighbor** field indicates that the multicast server is directly connected to the routing switch.

Figure 5-9.   Example Output for "5300XL #1" Routing Switch in Figure 5-4 on Page 5-20

Displaying PIM Status

Syntax: show ip pim

Displays PIM status and global parameters.
PIM Status: *Shows either* **enabled** *or* **disabled.**
State Refresh Interval (sec): *A PIM routing switch originates state refresh messages to inform its neighbors of the active flows it is currently routing. This updates the current flow data on PIM routers that join or rejoin a multicast network after the initial flood and prune. This enables hosts on such routers to join a multicast group without having to wait for a "flood and prune" cycle. PIM routers having the state refresh capability can eliminate all but an initial flood and prune cycle. PIM routers without this capability periodically trigger a flood and prune cycle on the path between the PIM router and the multicast source. (Range: 10 - 300 seconds; Default: 60 seconds.)*
Traps: Enables the following SNMP traps:

– **neighbor-loss:** *Sends a trap if a neighbor router is lost.*
– **hardware-mrt-full:** *Sends a trap if the hardware multicast router (MRT) table is full (511 active flows).*
– **software-mrt-full:** *Sends a trap if the software multicast router (MRT) table is full (511 active flows). This can occur only if the hardware MRT is also full.*
– **all:** *Enables all of the above traps.*

```
ProCurve# show ip pim

 PIM Global Parameters

  PIM Status                    : enabled
  State Refresh Interval (sec) : 45
  Traps                         : hardware-mrt-full
```

Figure 5-10. Example Output for the "5304XL #1" Routing Switch in Figure 5-4 on Page 5-20

Syntax: show ip pim [interface]

> *Lists the PIM interfaces (VLANs) currently configured in the routing switch.*
> **VLAN:** *Lists the VID of each VLAN configured on the switch to support PIM-DM.*
> **IP Address:** *Lists the IP addresses of the PIM interfaces (VLANs).*
> **Mode:** *Shows* **dense** *only.*

```
ProCurve# show ip pim interface

 PIM Interfaces

  VLAN IP Address       Mode
  ---- --------------- -----------
   25   25.38.10.1       dense
   27   27.27.30.1       dense
   29   29.29.30.1       dense
```

Figure 5-11. Example Output for the "5304xl #1" Routing Switch in Figure 5-4 on Page 5-20

Syntax: show ip pim [interface [< *vid* >]]

> *Displays the current configuration for the specified VLAN*
> *(PIM interface). Refer to table 5-1, below.*

```
ProCurve# show ip pim interface 29

 PIM Interface

  VLAN         : 29
  IP Address : 29.29.30.1
  Mode         : dense

  Designated Router :

  Hello Interval (sec) : 30
  Hello Delay (sec)      : 5

  Graft Retry Interval(sec) : 3
  Max Graft Retries          : 2
  Override Interval (msec)  : 2500      Lan Prune Delay         : Yes
  Propagation Delay (msec)  : 500       Lan Delay Enabled       : Yes
  SR TTL Threshold           : 0         State Refresh Capable   : Yes
```

Figure 5-12. Example Output for the "5304xl #1" Routing Switch in Figure 5-4 on Page 5-20

Table 5-1. PIM Interface Configuration Settings

Field	Default	Control Command
VLAN	n/a	vlan < *vid* > ip pim
IP	n/a	vlan < *vid* > ip pim < all | *ip-addr* >
Mode	dense	n/a; PIM Dense only
Hello Interval (sec)	30	ip pim hello interval < 5 - 30 >
Hello Hold Time	105	The routing switch computes this value from the current "Hello Interval" and includes it in the "Hello" packets the routing switch sends to neighbor routers. Neighbor routers use this value to determine how long to wait for another Hello packet from the routing switch. Refer to the description of the Hello Interval on page 5-15.
Hello Delay	5	vlan < *vid* > ip pim hello delay < 0 - 5 >
Graft Retry Interval (sec)	3	vlan < *vid* > ip pim graft-retry-interval < 1 - 10 >

PIM-DM (Dense Mode) on the 5300xl Switches
Displaying PIM Data and Configuration Settings on the Series 5300xl Switches

Field	Default	Control Command
Max Graft Retries	2	vlan < *vid* > ip pim graft-retries < 1 - 10 >
Override Interval (msec)	2500	vlan < *vid* > ip pim override-interval < 500 - 6000 >
Propagation Delay (msec)	500	vlan < *vid* > ip pim propagation-delay < 250-2000 >
SR TTL Threshold (router hops)	0	vlan < *vid* > ip pim ttl-threshold < 0 - 255 >
LAN Prune Delay	Yes	vlan < *vid* > ip pim lan-prune-delay
LAN Delay Enabled	No	Shows **Yes** if all multicast routers on the current VLAN interface enabled LAN-prune-delay. Otherwise shows **No**.
State Refresh Capable	n/a	Indicates whether the VLAN responds to state refresh packets. The VLAN connected to the multicast source does not receive state refresh packets and thus is not state-refresh capable. Downstream VLANs in Switch Series 5300XL devices are state-refresh capable.

Syntax: show ip pim [mroute]

> *Shows PIM-specific information from the IP multicast routing table (IP MRT). When invoked without parameters, lists all PIM entries currently in the routing switch's IP MRT.*
> **Group Address:** *Lists the multicast group addresses currently active on the routing switch.*
> **Source Address:** *Lists the multicast source address for each* **Group Address**.
> **Metric:** *Indicates the path cost upstream to the multicast source. Used when multiple multicast routers contend to determine the best path to the multicast source. The lower the value, the better the path. This value is set to* **0** *(zero) for directly connected routes.*
> **Metric Pref:** *Used when multiple multicast routers contend to determine the path to the multicast source. When this value differs between routers, PIM selects the router with the lowest value. If* **Metric Pref** *is the same between contending multicast routers, then PIM selects the router with the lowest* **Metric** *value to provide the path for the specified multicast traffic. This value is set to* **0** *(zero) for directly connected routes. (* **Metric Pref** *is based on the routing protocol in use: RIP, OSPF, or static routing. Also, different vendors may assign different values for this setting.)*

5-30

This output shows the routing switch is receiving two multicast groups from an upstream device at 27.27.30.2. The "**0**" metric shows that the routing switch is directly connected to the multicast source.

```
HPswitch# show ip pim mroute

 PIM Route Entries

  Group Address     Source Address   Metric      Metric Pref
  _____   _____  _____  _____
   239.255.255.1    27.27.30.2          0              0
   239.255.255.5    27.27.30.2          0              0
```

Figure 5-13. Example Showing a Routing Switch Detecting two Multicast Groups from a Directly Connected Multicast Server

Syntax: show ip pim [mroute [< *multicast-group-address* >
            < *multicast-source-address* >]]

> *Displays the PIM route entry information for the specified multicast group (flow):*
> **Group Address:** *Lists the specified multicast group address.*
> **Source Address:** *Lists the specified multicast source address.*
> **Source Mask:** *Lists the network mask for the multicast source address.*
> *Metric: Lists the number of multicast router hops to the source address.*
> **Metric:** *Indicates the path cost upstream to the multicast source. Used when multiple multicast routers contend to determine the best path to the multicast source. The lower the value, the better the path.*
> **Metric Pref:** *Used when multiple multicast routers contend to determine the path to the multicast source. When this value differs between routers, PIM selects the router with the lowest value. If Metric Pref is the same between contending multicast routers, then PIM selects the router with the lowest* **Metric** *value to provide the path for the specified multicast traffic. (Different vendors assign differing values for this setting.)*
> **Assert Timer:** *The time remaining until the routing switch ceases to wait for a response from another multicast router to negotiate the best path back to the multicast source. If this timer expires without a response from any contending multicast routers, then the routing switch assumes it is the best path, and the specified multicast group traffic will flow through the routing switch.*

DownStream Interfaces:
– **VLAN:** Lists the VID of the destination VLAN on the next-hop multicast router.
– **Prune Reason:** *Identifies the reason for pruning the flow to the indicated VLAN:*
  • **Prune:** *A neighbor multicast router has sent a prune request.*
  • **Assert:** *Another multicast router connected to the same VLAN has been elected to provide the path for the specified multicast group traffic.*
  • **Other:** *Used where the VLAN is in the pruned state for any reason other than the above two reasons (such as no neighbors exist and no directly connected hosts have done joins).*

```
ProCurve# show ip pim mroute 239.255.255.1 27.27.30.2

 PIM Route Entry

  Group Address  : 239.255.255.1
  Source Address : 27.27.30.2
  Source Mask    : 255.255.248.0

  Metric       : 3
  Metric Pref  : 120
  Assert Timer : 0

DownStream Interfaces

  VLAN Prune Reason
  ---- ------------
   28   prune
```

This example displays the MRT data on the first of the two multicast groups shown in figure 5-13 on page 5-31.

Figure 5-14. Example From the "5304xl #1" Routing Switch in Figure 5-4 on Page 5-20 Showing a Multicast Group from a Directly Connected Source

Syntax: show ip pim [neighbor]

> *Lists PIM neighbor information for all PIM neighbors connected to the routing switch:*
>
> **IP Address:** *Lists the IP address of a neighbor multicast router.*
> **VLAN:** *Lists the VLAN through which the routing switch connects to the indicated neighbor.*
> **Up Time:** *Shows the elapsed time during which the neighbor has maintained a PIM route to the routing switch.*
> **Expiry Time:** *Indicates how long before the routing switch ages-out the current flow (group membership). This value decrements until:*
>
> - *Reset by a state refresh packet originating from the upstream multicast router. (The upstream multicast router issues state refresh packets for the current group as long as it either continues to receive traffic for the current flow or receives state refresh packets for the current flow from another upstream multicast router.)*
> - *Reset by a new flow for the current multicast group on the VLAN.*
>
> *The timer expires (reaches* **0***). In this case the switch has not received either a state refresh packet or new traffic for the current multicast group, and ages-out (drops) the group entry.*
> *If the IP-ADDR is specified then detailed information for the specified neighbor is shown.*

```
  This example simulates output from the "5304XL #1" Routing
  Switch in Figure 5-4 on Page 5-20. The data identifies the first
  downstream neighbor ("5300XL #2").

ProCurve # show ip pim neighbor

 PIM Neighbors

  IP Address      VLAN Up Time (sec)      Expiry Time (sec)
  --------------- ---- ------------------- -------------------
  29.29.30.2       29   196                89
```

Figure 5-15. Example of PIM Neighbor Output

Syntax: show ip pim [neighbor [< *ip-address* >]]

> *Lists the same information as* **show ip pim neighbor**
> *(page 5-33) for the specified PIM neighbor:*

```
This example simulates output from the "5304XL #1" Routing Switch in Figure 5-4 on Page 5-20. The
data is from the first downstream neighbor ("5300XL #2").

ProCurve# show ip pim neighbor 29.29.30.2

 PIM Neighbor

  IP Address   : 29.29.30.2
  VLAN         : 29

  Up Time (sec)      : 26
  Expiry Time (sec) : 79
```

**Figure 5-16. Example From the "5304xl #1" Routing Switch in Figure 5-4 on Page
5-20 Showing a Specific Neighbor ("5300xl #2")**

Operating Notes

PIM Routers without State Refresh Messaging Capability. A PIM
router without a state refresh messaging capability learns of currently active
flows in a multicast network through periodic flood and prune cycles on the
path back to the source. The Switch Series 5300XL devices sense downstream
multicast routers that do not have the state refresh capability and will period-
ically flood active multicast groups to these devices. This periodic flooding is
not necessary if all of the downstream multicast routers are ProCurve 5300XL
devices. (The ProCurve Routing Switch Series 9300 and the routers offered by
some other vendors do not offer the state refresh capability.)

Flow Capacity. The routing switch provides an ample multicast environment, supporting 1022 multicast flows in hardware across a maximum of 64 VLANs. (A flow comprises a unicast source address and a multicast group address, regardless of the number of active hosts belonging to the multicast group at any given time.) While the typical multicast environment should not normally exceed 1022 flows, the routing switch can support up to 978 additional flows in software, depending on available system resources. (Because the switch processes flows in hardware much faster than in software, you may notice slower processing times for flows occurring in software.) Also, while the routing switch can support up to 2,000 flows, the total demand on system resources from the combined use of more than 1,022 simultaneous flows, a high number of VLANs supporting multicast routing, and/or other, resource-intensive features can oversubscribe memory resources, which reduces the number of flows the routing switch can support in software. That is, the switch does not route flows in software that oversubscribe current memory resources. If the routing switch regularly exceeds the hardware limit of 1022 flows and begins routing flows in software, you may want to move some hosts that create multicast demand to another routing switch, or reduce the number of VLANs on the routing switch by moving some VLANs to another routing switch. Note that the routing switch generates a log message if it either routes a flow in software or drops a flow intended for software routing because memory is oversubscribed. (Refer to "Messages Related to PIM Operation" on page 5-37.)

IGMP Traffic High-Priority Disabled. Enabling IP multicast routing to support PIM-DM operation has the effect of disabling IGMP traffic high-priority, if configured. (Refer to "Configuring IGMP Traffic Priority" on page 4-10.)

ACLs and PIM. The switch allows ACL filtering on unicast addresses, but not on multicast addresses. Also, an ACL does not take effect on a flow if the flow began before the ACL was configured.

When To Enable IGMP on a VLAN. When PIM is enabled on a VLAN, it is not necessary to also enable IGMP unless there may be Joins occurring on that VLAN. But if IGMP is enabled on a VLAN, you must also enable PIM if you want that VLAN to participate in multicast routing.

IP Address Removed. If you remove the IP address for a VLAN, the switch automatically removes the PIM configuration for that VLAN.

Troubleshooting

Symptom: Noticeable slowdown in some multicast traffic. If the switch is supporting more than 1022 active flows. This generates the message `Unable to learn HW IP multicast groups, table FULL` in the Event Log because there is no room in the hardware Multicast Routing Table to add another Multicast Group. Software will route any multicast packets sent to multicast groups that are not in the hardware Multicast Routing Table, but it will be slower and packets may be dropped if the data rate is greater than 3000 packets per second. Refer to "Flow Capacity" on page 5-35.

Note that the PIM protocol uses one MRT entry for every IP multicast source/group pair that it is routing. An entry is not used if the multicast flow is bridged and not routed. Entries in this table are automatically aged out if they are unused for a period of time.

Heavy Memory Usage. Heavy use of PIM (many S/G flows over many VLANs) combined with other memory-intensive features, can oversubscribe memory resources and impact overall performance. If available memory is exceeded, the switch drops any new multicast flows and generates appropriate Event Log messages. Corrective actions can include reducing the number of VLANs on the 5300xl switch by moving some VLANs to another device, free up system resources by disabling another, non-PIM feature, and/or moving some hosts to another device. For more information, refer to "Operating Notes" on page 5-34 and "Messages Related to PIM Operation" on page 5-37.

IPv4 Table Operation. The IPv4 table, which contains the active IP multicast addresses the switch is currently supporting, has 128k entries. However, the IPv4 table also contains IP host entries for every IP source or destination that the switch has learned, as well as ACL flow entries. Entries in this table are generally aged out if they are unused for 5 minutes or more.

Messages Related to PIM Operation

These messages appear in the Event Log and, if Syslog Debug is configured, in the designated Debug destinations.

N o t e   The <*counter*> value displayed at the end of each PIM Event Log message (and SNMP trap messages, if trap receivers are configured) indicates the number of times the switch has detected a recurring event since the last reboot. For more information, refer to "Using the Event Log To Identify Problem Sources" in the "Troubleshooting" appendix of the February, 2004 (or later) version of the *Management and Configuration Guide* for your switch.

Message	Meaning
< *alpha-string* > pkt, src IP < *ip-addr* > vid < *vlan-id* > (not a nbr) (<*counter*>)	A PIM packet arrived from another router for which no neighbor was found. May indicate a misconfiguration between the sending and receiving router. May also occur if a connected router is disconnected, then reconnected.
Bad TTL in State Refresh pkt from IP < *source-ip-addr* > (<*counter*>)	The switch detected a TTL of 0 (zero) in the PIM portion of a state refresh packet. (Note that this is not the IP TTL.)
Failed alloc of HW < *alpha-str* > for flow < *multicast-address* >, < *source-address* > (<*dup-msg-cnt*>)	There are more than 1022 active flows. The switch routes the excess through software, which processes traffic at a slower rate. If this will be an ongoing or chronic condition, transfer some of the flows to another router.
Failed to alloc a PIM < *data-type* > pkt (<*counter*>)	The router was unable to allocate memory for a PIM control packet. Router memory is oversubscribed. Reduce the number of VLANs or increase the hello delay and/or override interval to reduce the number of simultaneous packet transmissions. Note that if the number of flows exceeds 1022, the excess flows are routed in software, which reduces the number of packet transmissions. In this case, reducing the number of flows by moving some clients to other routers can help.
Failed to initialize < *text-str* > as a call back routine (<*counter*>)	Indicates an internal error. Report the incident to your HP customer care center and re-install the router software.
I/F configured with IP < *ip-address* > on vid < *vlan-id* > (<*counter*>)	Indicates that the interface (VLAN) has been configured with the indicated IP address. At boot-up or when an IP address is changed, the switch generates this message for each PIM-configured VLAN.

Message	Meaning
`I/F removal with IP <`*ip-addr*`> on vid <`*vlan-id*`>` (<*counter*>)	Indicates that a PIM interface (VLAN) has been removed from the router as a result of an IP address change or removal.
`MCAST flow <`*multicast-address*`> <`*source-address*`> not rteing (rsc low)` (<*counter*>)	The indicated multicast flow is not routing. The routing switch is low on memory resources as a result of too many flows for the number of configured VLANs. Remedies include one or more of the following: • Reduce the number of configured VLANs by moving some VLANs to another router. • Free up system resources by disabling another feature, such as one of the spanning-tree protocols or either the RIP or the OSPF routing protocol. (Unless you are using static routes, you will need to retain a minimum of one unicast routing protocol.) Another option that may help is to reduce the number of configured QoS filters. • Move some hosts that create multicast demand to another router.
`MCAST MAC add for <`*mac-address*`> failed` (<*counter*>)	Indicates a hardware problem. Check the cabling and router ports.
`Multicast Hardware Failed to Initialize` (<*counter*>)	Indicates a hardware failure that halts hardware processing of PIM traffic. The software will continue to process PIM traffic at a slower rate. Contact your HP customer care center.
`No IP address configured on VID <`*vlan-id*`>` (<*dup-msg-cnt*>)	PIM has detected a VLAN without an IP address. Configure an IP address on the indicated VLAN.
`Pkt dropped from <`*ip-address*`>, (<`*cause*`>) vid <`*vlan-id*`>` (<*counter*>)	A PIM packet from <*ip-address*> was dropped due to one of the following causes: • No PIM interface on the VLAN • Bad packet length • Bad IP header length • Bad IP total length
`Pkt rcvd with a cksum error from <`*ip-addr*`>` (<*counter*>)	A packet having a checksum error was received from <*ip-address*>. Check the cabling and ports on the local and the remote routers.
`Rcvd incorrect hello from <`*ip-addr*`>` (<*counter*>)	Indicates receipt of a malformed hello packet. (That is, the packet does not match the current specification.) Ensure that compatible versions of PIM-DM are being used.
`Rcvd <`*text-str*`> pkt with bad len from <`*ip-addr*`>` (<*counter*>)	A peer router may be sending incorrectly formatted PIM packets.
`Rcvd hello from <`*ip-address*`> on vid <`*vlan-id*`>` (<*counter*>)	Indicates a misconfiguration where two routers are directly connected with different subnets on the same connected interface.

Message	Meaning
`Rcvd pkt from rtr` *< ip-address >*`, unkwn pkt type` *< value >* `(`*<counter>*`)`	A packet received from the router at *< ip-address >* is an unknown PIM packet type. (The *< value >* variable is the numeric value received in the packet.)
`Rcvd pkt ver#` *< ver-num >*`, from` *< ip-address >*`, expected` *< ver-num >* `(`*<counter>*`)`	The versions of PIM-DM on the sending and receiving routers do not match. Differing versions will typically be compatible, but features not supported in both versions will not be available.
`Rcvd unkwn addr fmly` *< addr-type >* `in` *< text-str >* `pkt from` *< ip-addr >* `(`*<counter>*`)`	The router received a PIM packet with an unrecognized encoding. As of February, 2004, the router recognizes IPv4 encoding.
`Rcvd unkwn opt` *< opt-nbr >* `in` *< text-string >* `pkt from` *< ip-addr >* `(`*<counter>*`)`	The router received a PIM packet carrying an unknown PIM option. The packet may have been generated by a newer version of PIM-DM, or is corrupt. In most cases, normal PIM-DM operation will continue.
`Send error(`*< failure-type >*`) on` *< packet-type >* `pkt on VID` *< vid >* `(`*<counter>*`)`	Indicates a send error on a packet. This can occur if a VLAN went down right after the packet was sent. The message indicates the failure type, the packet type, and the VLAN ID on which the packet was sent.
`Unable to alloc`*< text-str >* `table (`*<counter>*`)`	The router was not able to create some tables PIM-DM uses. Indicates that the router is low on memory resources. Remedies include one or more of the following: • Reduce the number of configured VLANs by moving some VLANs to another router. • Free up system resources by disabling another feature, such as one of the spanning-tree protocols or either the RIP or the OSPF routing protocol. (Unless you are using static routes, you will need to retain a minimum of one unicast routing protocol.) Another option that may help is to reduce the number of configured QoS filters. • Move some hosts that create multicast demand to another router.
`Unable to alloc a buf of size` *< bytes >* `for` *< data-flow >* `(`*<counter>*`)`	Multicast routing is unable to acquire memory for a flow. Router memory is oversubscribed. Reduce the number of VLANs or the number of features in use. Remedies include one or more of the following: • Reduce the number of configured VLANs by moving some VLANs to another router. • Free up system resources by disabling another feature, such as one of the spanning-tree protocols or either the RIP or the OSPF routing protocol. (Unless you are using static routes, you will need to retain a minimum of one unicast routing protocol.) Another option that may help is to reduce the number of configured QoS filters. • Move some hosts that create multicast demand to another router.

Message	Meaning
`Unable to alloc a msg buffer for` _< text-message >_ `(`_< counter >_`)`	Multicast routing is unable to acquire memory for a flow. Router memory is oversubscribed. Reduce the number of VLANs or the number of features in use. Remedies include one or more of the following: • Reduce the number of configured VLANs by moving some VLANs to another router. • Free up system resources by disabling another feature, such as one of the spanning-tree protocols or either the RIP or the OSPF routing protocol. (Unless you are using static routes, you will need to retain a minimum of one unicast routing protocol.) Another option that may help is to reduce the number of configured QoS filters. • Move some hosts that create multicast demand to another router.

Applicable RFCs

PIM on the Switch Series 5300XL devices is compatible with these RFCs:

- RFC 3376 - Internet Group Management Protocol, Version 3

- RFC 2365 - Administratively Scoped IP Multicast

- RFC 2932 - Multicast Routing MIB, _with exceptions (Refer to "Exceptions to Support for RFC 2932 - Multicast Routing MIB"._)

- RFC 2933 - IGMP MIB

- RFC 2934 - Protocol Independent Multicast MIB for IPv4

- draft-ietf-ssm-arch-01.txt - Source-Specific Multicast for IP (draft specification, expires May 2003

Exceptions to Support for RFC 2932 - Multicast Routing MIB

These MIB objects are not supported in the 5300XL routing switch.

ipMRouteInterfaceRateLimit

ipMRouteInterfaceInMcastOctets

ipMRouteInterfaceOutMcastOctets

ipMRouteInterfaceHCInMcastOctets

ipMRouteInterfaceHCOutMcastOctets

ipMRouteBoundaryTable

ipMRouteBoundaryEntry

ipMRouteBoundaryIfIndex

ipMRouteBoundaryAddress

ipMRouteBoundaryAddressMask

ipMRouteBoundaryStatus OBJECT-TYPE

ipMRouteScopeNameTable

ipMRouteScopeNameEntry

ipMRouteScopeNameAddress

ipMRouteScopeNameAddressMask

ipMRouteScopeNameLanguage

ipMRouteScopeNameString

ipMRouteScopeNameDefault

ipMRouteScopeNameStatus

— This page is intentionally unused. —

6

Spanning-Tree Operation

Contents

Overview

N o t e *The Series 4200vl switches support MSTP only.*

STP Features

802.1D Spanning Tree Protocol *	Default	Menu	CLI
Viewing the STP Configuration	n/a	page 6-22	page 6-14
Enable/Disable STP	Disabled	page 6-22	page 6-26
Reconfiguring General Operation	priority: 32768 max age: 20 s hello time: 2 s fwd. delay: 15 s	page 6-22	page 6-27
Reconfiguring Per-Port STP	path cost: var priority: 128 mode: norm	page 6-22	page 6-28
802.1w Spanning Tree Protocol*	**Default**	**Menu**	**CLI**
Viewing the RSTP/STP Configuration	n/a	page 6-20	page 6-14
Enable/Disable RSTP/STP (RSTP is selected as the default protocol.)	Disabled	page 6-20	page 6-15
Reconfiguring Whole-Switch Values	Protocol Version: RSTP Force Version: RSTP-operation Switch Priority: 8 Hello Time: 2 s Max Age: 20 s Forward Delay: 15 s	page 6-20	page 6-16

Reconfiguring Per-Port Values	Path Cost:
	   Depends on port
	   type
	Priority: 8
	Edge Port: Yes
	Point-to-point:
	   Force-true
	MCheck: Yes
		page 6-20	page 6-18

802.1s Spanning Tree Protocol	Default	Menu	CLI
Viewing the MSTP Status and Configuration	n/a	—	page 6-74
Enable/Disable MSTP and Configure Global Parameters	Disabled	—	page 6-59
Configuring Basic Port Connectivity Parameters	edge-port: No mcheck: Yes hello-time: 2 path-cost: auto point-to-point MAC: Force-True priority: 128 (multiplier: 8)	—	page 6-63 and following
Configuring MSTP Instance Parameters	instance (MSTPI): none priority: 32768 (multiplier: 8)	—	page 6-66
Configuring MSTP Instance Per-Port Parameters	Auto	—	page 6-69
Enabling/Disabling MSTP Spanning Tree Operation	Disabled	—	page 6-72
Enabling an Entire MST Region at Once	n/a	—	page 6-72
* Available on the 3400cl, 6400cl and 5300xl switches			

Without spanning tree, having more than one active path between a pair of nodes causes loops in the network, which can result in duplication of messages, leading to a "broadcast storm" that can bring down the network.

Single-Instance spanning tree operation (802.1D STP and 802.1w RSTP) ensures that only one active path at a time exists between any two nodes in a physical network. In networks where there is more than one physical, active path between any two nodes, enabling single-instance spanning tree ensures one active path between such nodes by blocking all redundant paths.

Multiple-Instance spanning tree operation (802.1s) ensures that only one active path exists between any two nodes in a spanning-tree *instance*. A spanning-tree instance comprises a unique set of VLANs, and belongs to a specific spanning-tree *region*. A region can comprise multiple spanning-tree instances (each with a different set of VLANs), and allows one active path among regions in a network. Applying VLAN tagging to the ports in a multiple-instance spanning-tree network enables blocking of redundant links in one instance while allowing forwarding over the same links for non-redundant use by another instance. For example, suppose you have three switches in a region

configured with VLANs grouped into two instances, as follows:

VLANs	Instance 1	Instance 2
10, 11, 12	Yes	No
20, 21, 22	No	Yes

The logical and physical topologies resulting from these VLAN/Instance groupings result in blocking on different links for different VLANs:
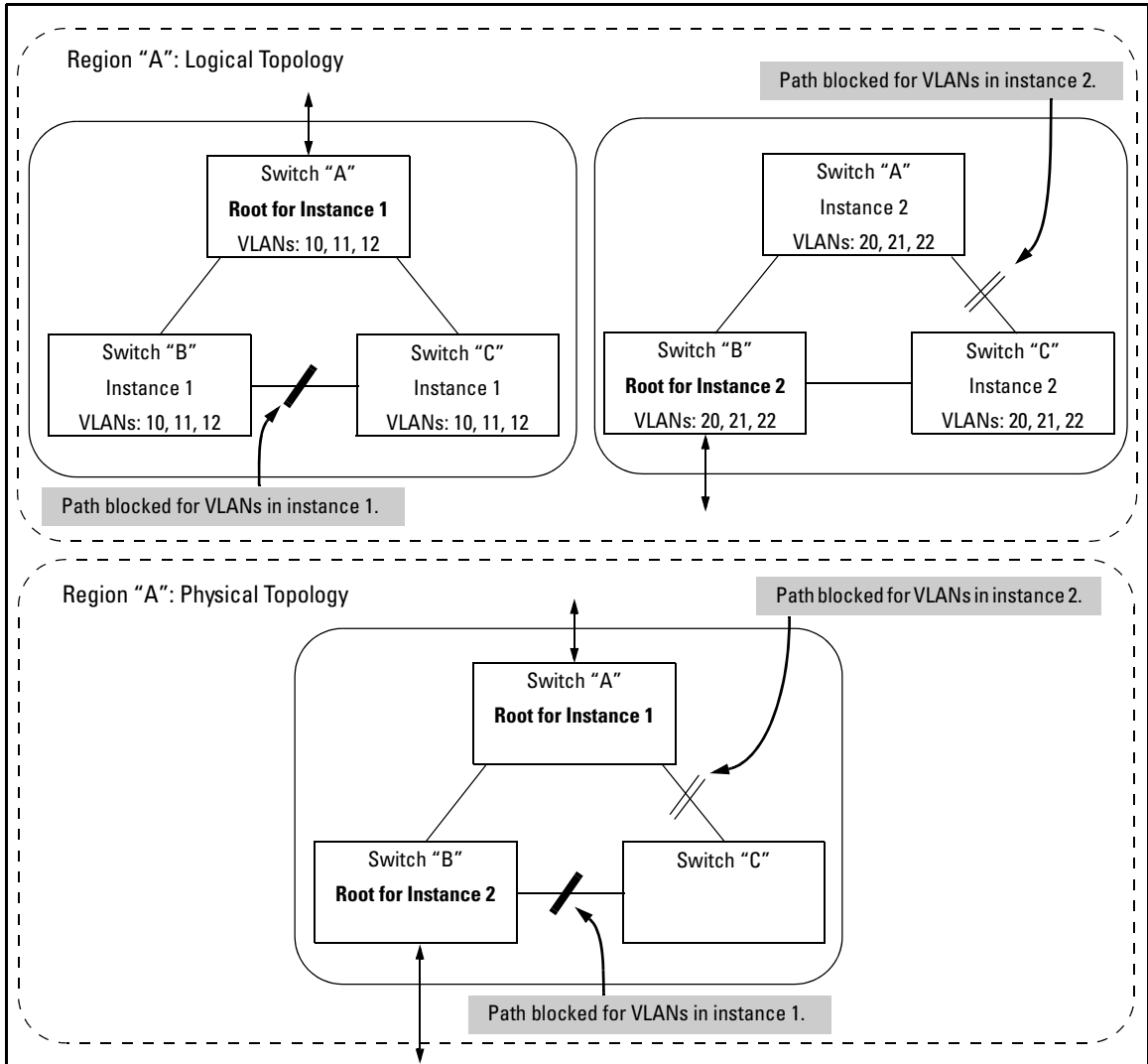
Figure 6-1. Example of a Multiple Spanning-Tree Application

Note for 802.1D and 802.1w Spanning-Tree Operation for the Series 5300xl and Series 3400/6400cl switches

You should enable spanning tree operation in any switch that is part of a redundant physical link (loop topology). (HP recommends that you do so on all switches belonging to a loop topology.) This topic is covered in more detail under "How STP and RSTP Operate on the 5300xl, 3400cl and 6400cl Switches" on page 6-8.

As recommended in the IEEE 802.1Q VLAN standard, the switches covered by this guide use **single-instance STP** for 802.1D and 802.1w spanning-tree operation. (In this case, the switch generates untagged Bridge Protocol Data Units—BPDUs.) This implementation creates a single spanning tree to make sure there are no network loops associated with any of the connections to the switch, regardless of whether multiple VLANs are configured on the switch. Thus, when using 802.1D or 802.1w spanning tree, these switches do not distinguish between VLANs when identifying redundant physical links. In this case, if VLANs are configured on the switch, see "RSTP and STP Operation with 802.1Q VLANs" on page 6-9.

The RSTP (802.1w) and STP (802.1D) Spanning Tree Options (5300xl, 3400/ 6400cl switches)

Caution

Spanning tree interprets a switch mesh as a single link. Because the switch automatically gives faster links a higher priority, the default STP or RSTP parameter settings are usually adequate for spanning tree operation. Also, because incorrect STP or RSTP settings can adversely affect network performance, you should not make changes unless you have a strong understanding of how spanning tree operates.

In a mesh environment, the default RSTP timer settings (**Hello Time** and **Forward Delay**) are usually adequate for RSTP operation. Because a packet crossing a mesh may traverse several links within the mesh, using smaller-than-default settings for the RSTP **Hello Time** and **Forward Delay** timers can cause unnecessary topology changes and end-node connectivity problems.

For more on STP and RSTP, see the IEEE 802.1D and 802.1w standards.

RSTP (802.1w)

RSTP is supported on the Series 5300xl, 3400cl and 6400cl switches.

The IEEE 802.1D version of spanning tree (STP) can take a fairly long time to resolve all the possible paths and to select the most efficient path through the network. The IEEE 802.1w Rapid Reconfiguration Spanning Tree (RSTP) significantly reduces the amount of time it takes to establish the network path. The result is reduced network downtime and improved network robustness.

In addition to faster network reconfiguration, RSTP also implements greater ranges for port path costs to accommodate the higher and higher connection speeds that are being implemented.

RSTP is designed to be compatible with IEEE 802.1D STP, and HP recommends that you employ it in your network. For more information, refer to "Transitioning from STP to RSTP" on page 6-12.

STP (802.1D)

STP is supported on the Series 5300xl, 3400cl and 6400cl switches.

The IEEE 802.1D version of spanning tree has been in wide use and can coexist in a network in which RSTP (802.1w) has been introduced. If your network currently uses 802.1D STP and you are not yet ready to implement RSTP, you can apply STP to the switch until such time as you are ready to move ahead with RSTP. STP on the switches covered by this guide offers the full range of STP features found in earlier product releases, including:

- **STP Fast Mode for Overcoming Server Access Failures:** If an end node is configured to automatically access a server, the duration of the STP startup sequence can result in a "server access failure". On ports where this is a problem, configuring STP Fast Mode can eliminate the failure. For more information, see "STP Fast Mode" on page 6-29. The next sections describe how to configure STP on the switch. For more information on STP operation, see "How STP and RSTP Operate on the 5300xl, 3400cl and 6400cl Switches" on page 6-8.

- **Fast-Uplink STP for Improving the Recovery (Convergence) Time in Wiring Closet Switches with Redundant Uplinks:** This means that a switch having redundant links toward the root device can decrease the convergence time to a new uplink port to as little as ten seconds. For more information, refer to "Fast-Uplink Spanning Tree Protocol (STP)" on page 6-30.

How STP and RSTP Operate on the 5300xl, 3400cl and 6400cl Switches

The switch automatically senses port identity and type, and automatically defines spanning-tree parameters for each type, as well as parameters that apply across the switch. You can use the default values for these parameters, or adjust them as needed.

While allowing only one active path through a network at any time, spanning tree retains any redundant physical path to serve as a backup (blocked) path in case the existing active path fails. Thus, if an active path fails, spanning tree automatically activates (unblocks) an available backup to serve as the new active path for as long as the original active path is down. For example:

- Active path from node A to node B: 1—> 3
- Backup (redundant) path from node A to node B: 4 —> 2 —> 3
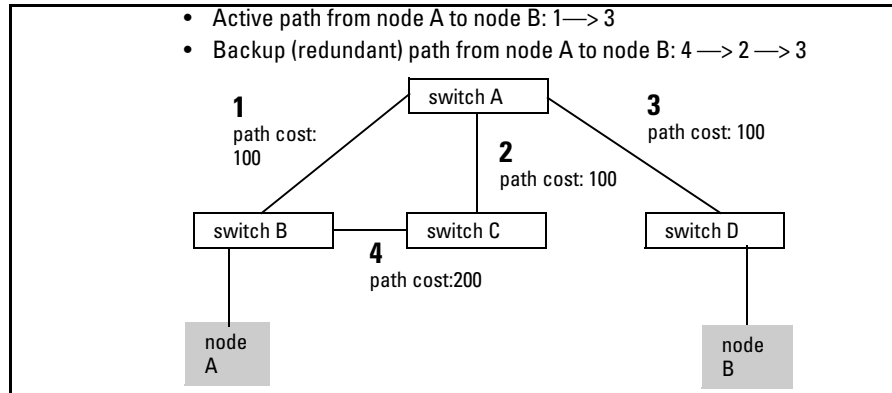
Figure 6-2. General Example of Redundant Paths Between Two Nodes

In the factory default configuration, spanning tree operation is off. If a redundant link (loop) exists between nodes in your network, you should enable the spanning tree operation of your choice.

<table>
<tr><td>

N o t e
</td><td>

Spanning tree retains its current parameter settings when disabled. Thus, if you disable spanning tree, then later re-enable it, the parameter settings will be the same as before spanning tree was disabled.
</td></tr>
</table>

RSTP and STP Operation with 802.1Q VLANs. As recommended in the IEEE 802.1Q VLAN standard, when 802.1D or 802.1w spanning tree is enabled on the switch, a single spanning tree is configured for all ports across the switch, including those in separate VLANs. This means that if redundant physical links exist in separate VLANs, spanning tree will block all but one of those links. However, if you need to use spanning tree on the switch in a VLAN environment with redundant physical links, you can prevent blocked redundant links by using a port trunk. The following example shows how you can use a port trunk with 802.1Q (tagged) VLANs and spanning tree without unnecessarily blocking any links or losing any bandwidth.
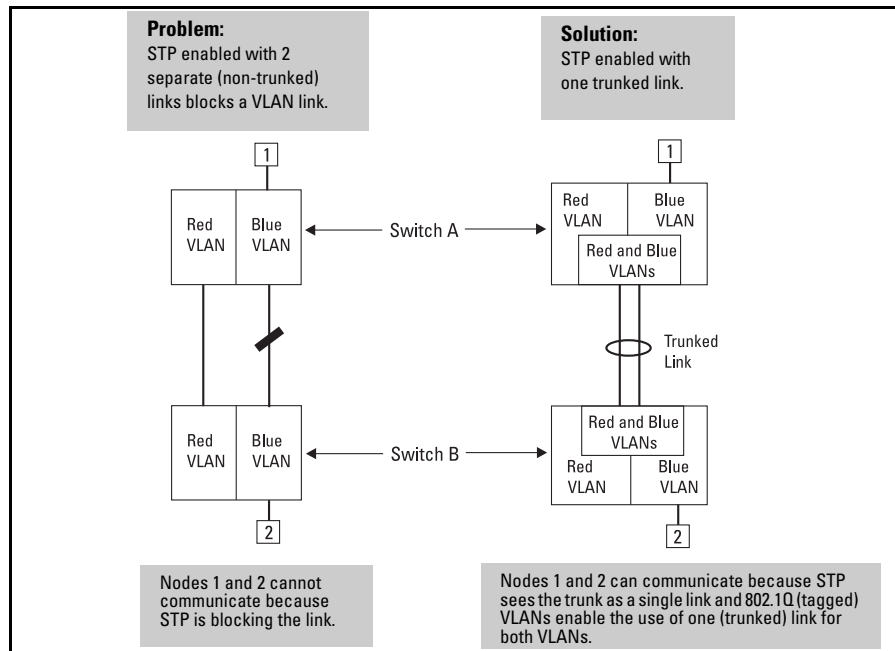
Figure 6-3.   Example of Using a Trunked Link with STP and VLANs

Configuring Rapid Reconfiguration Spanning Tree (RSTP)

RSTP is supported on the Series 5300xl, 3400cl and 6400cl switches.

This section describes the operation of the IEEE 802.1w Rapid Spanning Tree Protocol (RSTP).

Overview

RSTP Feature		Default	Menu	CLI
Viewing the RSTP/STP configuration		*n/a*	page 6-20	page 6-14
enable/disable RSTP/STP (RSTP is selected as the default protocol)		disabled	page 6-20	6-15
reconfiguring whole-switch values	Protocol Version:	RSTP	page 6-20	page 6-16
	Force Version:	RSTP-operation		
	Switch Priority:	8		
	Hello Time:	2 s		
	Max Age:	20 s		
	Forward Delay:	15 s		
reconfiguring per-port values	Path Cost:	*depends on port type*	page 6-20	page 6-18
	Priority:	8		
	Edge Port:	Yes		
	Point-to-point:	Force-true		
	MCheck:	Yes		

As indicated in the manual, the spanning tree protocol is used to ensure that only one active path at a time exists between any two end nodes in the network in which your switch is installed. Multiple paths cause a loop in the network over which broadcast and multicast messages are repeated continuously, which floods the network with traffic creating a broadcast storm.

In networks where there is more than one physical path between any two nodes, enabling spanning tree ensures a single active path between two such nodes by selecting the one most efficient path and blocking the other redun-

dant paths. If a switch or bridge in the path becomes disabled, spanning tree activates the necessary blocked segments to create the next most efficient path.

Transitioning from STP to RSTP

IEEE 802.1w RSTP is designed to be compatible with IEEE 802.1D STP. Even if all the other devices in your network are using STP, you can enable RSTP on your switch, and even using the default configuration values, your switch will interoperate effectively with the STP devices. If any of the switch ports are connected to switches or bridges on your network that do not support RSTP, RSTP can still be used on this switch. RSTP automatically detects when the switch ports are connected to non-RSTP devices in the spanning tree and communicates with those devices using 802.1D STP BPDU packets.

Because RSTP is so much more efficient at establishing the network path, it is highly recommended that all your network devices be updated to support RSTP. RSTP offers convergence times of less than one second under optimal circumstances. To make the best use of RSTP and achieve the fastest possible convergence times there are some changes that you should make to the RSTP default configuration. See "Optimizing the RSTP Configuration" below for more information on these changes.

N o t e   Under some circumstances, it is possible for the rapid state transitions employed by RSTP to result in an increase in the rates of frame duplication and misordering in the switched LAN. In order to allow RSTP switches to support applications and protocols that may be sensitive to frame duplication and misordering, setting the Force Protocol Version parameter to **stp-compat-ible** allows RSTP to be operated with the rapid transitions disabled. The value of this parameter applies to all ports on the switch. See the information on **Force Version** on page 6-16.

As indicated above, one of the benefits of RSTP is the implementation of a larger range of port path costs, which accommodates higher network speeds. New default values have also been implemented for the path costs associated with the different network speeds. This can create some incompatibility between devices running the older 802.1D STP and your switch running RSTP. Please see the "Note on Path Cost" on page 6-19 for more information on adjusting to this incompatibility.

Configuring RSTP

The default switch configuration has spanning tree disabled with RSTP as the selected protocol. That is, when spanning tree is enabled, RSTP is the version of spanning tree that is enabled, by default.

Optimizing the RSTP Configuration

To optimize the RSTP configuration on your switch, follow these steps (note that for the **Menu** method, all of these steps can be performed at the same time by making all the necessary edits on the "Spanning Tree Operation" screen and then saving the configuration changes):

1. Set the switch to support RSTP (RSTP is the default):

    CLI: spanning-tree protocol-version rstp

    Menu: Main Menu —> 2. Switch Configuration —> 4. Spanning Tree Operation —> select Protocol Version: RSTP

2. Set the "point-to-point-mac" value to false on all ports that are connected to shared LAN segments (that is, to connections to hubs):

    CLI: spanning-tree [ethernet] < ***port-list*** > point-to-point-mac force-false

    Menu: Main Menu —> 2. Switch Configuration —> 4. Spanning Tree Operation —> for each appropriate port, select Point-to-Point: Force-False

3. Set the "edge-port" value to false for all ports connected to other switches, bridges, and hubs:

    CLI: no spanning-tree [ethernet] < ***port-list*** > edge-port

    Menu: Main Menu —> 2. Switch Configuration —> 4. Spanning Tree Operation —> for each appropriate port, select Edge: No

4. Set the "mcheck" value to false for all ports that are connected to devices that are known to be running IEEE 802.1D spanning tree:

    CLI: no spanning-tree [ethernet] < ***port-list*** > mcheck

    Menu: Main Menu —> 2. Switch Configuration —> 4. Spanning Tree Operation —> for each appropriate port, select MCheck: No

5. Enable RSTP Spanning Tree:

    CLI: spanning-tree

    Menu: Main Menu —> 2. Switch Configuration —> 4. Spanning Tree Operation —> select
    STP Enabled: Yes

CLI: Configuring RSTP

Spanning Tree Commands in This Section	STP	RSTP	Page for RSTP Use
show spanning-tree config	Y	Y	Below on this page
spanning-tree	Y	Y	page 6-15
protocol-version <rstp	stp>	Y	Y	page 6-16
force-version <rstp-operation	stp-compatible>	N	Y	page 6-16
forward-delay <4 - 30>	Y	Y	page 6-16
hello-time <1 - 10>	Y	Y	page 6-16
maximum-age <6 - 40>	Y	Y	page 6-16
priority <0 - 15	0 - 65535>	Y	Y	page 6-16
<[ethernet] *port-list*>	Y	Y	page 6-18
path-cost <1 - 200 000 000>	Y	Y	page 6-18
priority <0 - 15	0 - 65535>	Y	Y	page 6-19
edge-port	N	Y	page 6-18
point-to-point-mac	N	Y	page 6-18
mcheck	N	Y	page 6-18
mode <norm	fast>	Y	N	Refer to "802.1D Spanning-Tree Protocol (STP) on 5300xl, 3400cl and 6400cl Switches" on page 6-22.
show spanning-tree			This command lists additional RSTP/STP/MSTP monitoring data that is not covered in this section. Refer to the section titled "Spanning Tree Protocol Information" in the "Monitoring and Analyzing *Switch Operation*" appendix of the *Management and Configuration Guide* for your switch.

Viewing the Current Spanning Tree Configuration. Use this command to display the current spanning tree configuration.

Syntax: show spanning-tree config

> *Lists the switch's full spanning tree configuration, including whole-switch and per-port settings, regardless of whether spanning tree is disabled.*
>
> *(**Default:** n/a; Abbreviated Command:* **sho span config***)*

In the default configuration, the output from this command appears similar to the following:

```
ProCurve(config)# show spanning-tree config

 Rapid Spanning Tree Configuration

  STP Enabled [No] : No
  Force Version [RSTP-operation] : RSTP-operation
  Switch Priority [8] : 8                   Hello Time [2] : 2
  Max Age [20] : 20                         Forward Delay [15] : 15

  Port Type      | Cost      Priority Edge Point-to-Point MCheck
  ---- --------- + --------- -------- ---- -------------- ------
  1    100/1000T | 20000     8        Yes  Force-True     Yes
  2    100/1000T | 20000     8        Yes  Force-True     Yes
  3    100/1000T | 20000     8        Yes  Force-True     Yes
  4    100/1000T | 20000     8        Yes  Force-True     Yes
  5    100/1000T | 20000     8        Yes  Force-True     Yes
  6    100/1000T | 20000     8        Yes  Force-True     Yes
  7    100/1000T | 20000     8        Yes  Force-True     Yes
  8    100/1000T | 20000     8        Yes  Force-True     Yes
  9    100/1000T | 20000     8        Yes  Force-True     Yes
  .    .         | .         .        .    .              .
  .    .         | .         .        .    .              .
  .    .         | .         .        .    .              .
```

Figure 6-4.   Example of the Spanning Tree Configuration Display (ProCurve Series 3400cl Switch)

Enabling or Disabling RSTP. Issuing the command to enable spanning tree on the switch implements, by default, the RSTP version of spanning tree for all physical ports on the switch. Disabling spanning tree removes protection against redundant network paths.

Syntax: [no] spanning-tree

> *Abbreviation:* [no] span

This command enables spanning tree with the current parameter settings or disables spanning tree, using the "no" option, without losing the most-recently configured parameter settings.

Enabling STP Instead of RSTP. If you decide, for whatever reason, that you would prefer to run the IEEE 802.1D (STP) version of spanning tree, then issue the following command:

Syntax: spanning-tree protocol-version stp

> *Abbreviation:* span prot stp

For the STP version of spanning tree, the rest of the information in this section does not apply. Refer to "802.1D Spanning-Tree Protocol (STP) on 5300xl, 3400cl and 6400cl Switches" on page 6-22 for more information on the STP version and its parameters.

Reconfiguring Whole-Switch Spanning Tree Values. You can configure one or more of the following parameters, which affect the spanning tree operation of the whole switch:

Table 6-1.   Whole-Switch RSTP Parameters

Parameter	Default	Description
protocol-version	RSTP	Identifies which of the spanning tree protocols will be used when spanning tree is enabled on the switch.
force-version	rstp-operation	Sets the spanning tree compatibility mode. Even if **rstp-operation** is selected though, if the switch detects STP BPDU packets on a port, it will communicate to the attached device using STP BPDU packets.
		If errors are encountered, as described in the Note on page 6-12, the Force-Version value can be set to **stp-compatible**, which forces the switch to communicate out all ports using operations that are compatible with IEEE 802.1D STP.
priority	32768 (8 as a step value)	Specifies the protocol value used along with the switch MAC address to determine which device in the spanning tree is the root. The lower the priority value, the higher the priority.
		The value you enter has changed from the STP value. The range is 0 - 61440, but for RSTP the value is entered as a multiple (a step) of 4096. You enter a value in the range 0 - 15. The default value of 32768 is derived by the default setting of 8.
		Displaying the RSTP configuration (**show spanning-tree config**) shows 8, but displaying the RSTP operation (**show spanning-tree**) shows 32768.
*maximum-age	20 seconds	Sets the maximum age of received spanning tree information before it is discarded. The range is 6 to 40 seconds.
*hello-time	2 seconds	Sets the time between transmission of spanning tree messages. Used only when this switch is the root. The range is 1 to 10 seconds.
*forward-delay	15 seconds	Sets the time the switch waits between transitioning ports from listening to learning and from learning to forwarding states. The range is 4 to 30 seconds.

*These parameters are the same for RSTP as they are for STP. The switch uses its own maximum-age, hello-time, and forward-delay settings only if it is operating as the root device in the spanning tree. If another device is the root device, then the switch uses the other device's settings for these parameters.

N o t e   Executing the **spanning-tree** command alone enables spanning tree. Executing the command with one or more of the whole-switch RSTP parameters shown in the table on the previous page, or with any of the per-port RSTP parameters shown in the table on page 6-18, does not enable spanning tree. It only configures the spanning tree parameters, regardless of whether spanning tree is actually running (enabled) on the switch.

Using this facility, you can completely configure spanning tree the way you want and then enable it. This method minimizes the impact on the network operation.

Syntax: *Abbreviations:*

spanning-tree   span
   protocol-version <rstp | stp>   prot <rstp | stp>
   force-version <rstp-operation | stp-compatible>   forc <rstp | stp>
   priority <0 - 15>   pri <0 - 15>
   maximum-age <6 - 40 seconds>   max <6 - 40>
   hello-time <1- 10 seconds>   hello <1 - 10>
   forward-delay <4 - 30 seconds>   forw <4 - 30>

Defaults: See the table on the previous page.

Multiple parameters can be included on the same command line. For example, to configure a maximum-age of 30 seconds and a hello-time of 3 seconds, you would issue the following command:

```
ProCurve (config)# span max 30 hello 3
```

Reconfiguring Per-Port Spanning Tree Values. You can configure one or more of the following parameters, which affect the spanning tree operation of the specified ports only:

Table 6-2.   Per-Port RSTP Parameters

Parameter	Default	Description
edge-port	Yes	Identifies ports that are connected to end nodes. During spanning tree establishment, these ports transition immediately to the Forwarding state.
		In this way, the ports operate very similarly to ports that are configured in "fast mode" under the STP implementation in previous ProCurve switch software.
		Disable this feature on all switch ports that are connected to another switch, or bridge, or hub. Use the "**no**" option on the spanning tree command to disable edge-port. This option is available only with RSTP or MSTP operation. (Note that when MSTP is enabled, the **edge-port** default setting is disabled.)
		An optional feature of edge-port functionality for RSTP is that it will automatically switch from an Edge port to a non-edge port upon receipt of a BPDU. If a bridge or switch is detected on the segment, the port will automatically operate as if **Edge = No** has been set.
mcheck	Yes	Ports with mcheck set to true are forced to send out RSTP BPDUs for 3 seconds. This allows for switches that are running RSTP to establish their connection quickly and for switches running 802.1D STP to be identified.
		If the whole-switch parameter Force-Version is set to "stp-compatible", the mcheck setting is ignored and STP BPDUs are sent out all ports.
		Disable this feature on all ports that are known to be connected to devices that are running 802.1D STP. Use the "no" option on the spanning tree command to disable mcheck. This option is available only with RSTP or MSTP operation.
path-cost	10 Mbps – 2 000 000 100 Mbps – 200 000 1 Gbps – 20 000	Assigns an individual port cost that the switch uses to determine which ports are the forwarding ports. The range is 1 to 200,000,000 or auto.
		By default, this parameter is automatically determined by the port type, as shown by the different default values. If you have previously configured a specific value for this parameter, you can issue the command with the **auto** option to restore the automatic setting feature.
		Please see the Note on Path Cost on page 6-19 for information on compatibility with devices running 802.1D STP for the path cost values.
point-to-point-mac	force-true	This parameter is used to tell the port if it is connected to a point-to-point link, such as to another switch or bridge or to an end node (**force-true**).
		This parameter should be set to **force-false** for all ports that are connected to a hub, which is a shared LAN segment.
		You can also set this parameter to **auto** and the switch will automatically set the force-false value on all ports that it detects are not running at full duplex. All connections to hubs are not full duplex. This command is available only with RSTP operation.

Parameter	Default	Description
priority	128 (8 as a step value)	This parameter is used by RSTP to determine the port(s) to use for forwarding. The port with the lowest number has the highest priority. The range is 0 to 240, but you configure the value by entering a multiple of 16. You enter a value in the range 0 - 15. The default value of 128 is derived by the default setting of 8. Displaying the RSTP configuration (**show spanning-tree config**) shows 8, but displaying the RSTP operation (**show spanning-tree**) shows 128.

Syntax:	Abbreviations:

spanning-tree [ethernet] < ***port-list*** >
    path-cost < 1 - 200000000 >
    point-to-point-mac < force-true | force-false | auto >
    priority < 0 - 15 >
	span < *port-list* >
	    path <1 - 200000000>
	    force < force-t | force-f | auto >
	    pri <0 - 15>

[no] spanning-tree [ethernet] < ***port-list*** >
    edge-port
    mcheck
	[no] span < *port-list* >
	    edge
	    mch

Defaults: see the table on the previous page.

Note on Path Cost   RSTP and MSTP implement a greater range of path costs and new default path cost values to account for higher network speeds. These values are different than the values defined by 802.1D STP as shown below.

Port Type	802.1D STP Path Cost	RSTP and MSTP Path Cost
10 Mbps	100	2 000 000
100 Mbps	10	200 000
1 Gbps	5	20 000

Because the maximum value for the path cost allowed by 802.1D STP is 65535, devices running that version of spanning tree cannot be configured to match the values defined by RSTP and MSTP, at least for 10 Mbps and 100 Mbps ports. In LANs where there is a mix of devices running 802.1D STP, RSTP, and/ or MSTP, you should reconfigure the devices so the path costs match for ports with the same network speeds.

Menu: Configuring RSTP

1. From the console CLI prompt, enter the menu command.

   ProCurve # **menu**

2. From the switch console Main Menu, select

   2. Switch Configuration …

       4. Spanning Tree Operation

3. Press **[E]** (for **Edit**) to highlight the **Protocol Version** parameter field.

4. Press the Space bar to select the version of spanning tree you wish to run: **RSTP** or **STP**.

   Note: If you change the protocol version, you will have to reboot the switch for the change to take effect. See step 9 and step 10.

5. Press the **[Tab]** or down arrow key to go to the **STP Enabled** field. Note that when you do this, the remaining fields on the screen will then be appropriate for the version of spanning tree that was selected in step 3. The screen image below is for RSTP.

6. Press the Space bar to select **Yes** to enable spanning tree.

```
============================- TELNET - MANAGER MODE -============================
                 Switch Configuration - Spanning Tree Operation

  Protocol Version : RSTP
  STP Enabled [No] : No
  Force Version [RSTP-operation] : RSTP-operation
  Switch Priority [8] : 8                    Hello Time [2] : 2
  Max Age [20] : 20                          Forward Delay [15] : 15

  Port    Type          Cost       Priority   Edge   Point-to-Point   MCheck
  ----    --------- + ---------   --------   ----   --------------   ------
  A1      10/100TX  | 200000      8          Yes    Force-True       Yes
  A2      10/100TX  | 200000      8          Yes    Force-True       Yes
  A3      10/100TX  | 200000      8          Yes    Force-True       Yes
  A4      10/100TX  | 200000      8          Yes    Force-True       Yes
  A5      10/100TX  | 200000      8          Yes    Force-True       Yes
  A6      10/100TX  | 200000      8          Yes    Force-True       Yes

  Actions->    Cancel      Edit      Save      Help

Cancel changes and return to previous screen.
Use arrow keys to change action selection and <Enter> to execute action.
```

Figure 6-5. Example of the RSTP Configuration Screen

7. Press the **[Tab]** key or use the arrow keys to go to the next parameter you want to change, then type in the new value or press the Space bar to select a value. (To get help on this screen, press **[Enter]** to select the **Actions —>** line, then press **[H]**, for **Help**, to display the online help.)

8. Repeat step 6 for each additional parameter you want to change.

   Please see "Optimizing the RSTP Configuration" on page 6-13 for recommendations on configuring RSTP to make it operate the most efficiently.

9. When you are finished editing parameters, press **[Enter]** to return to the **Actions —>** line and press **[S]** to save the currently displayed spanning tree settings and return to the Main Menu.

10. If you have changed the Protocol Version, in step 1, reboot the switch now by selecting

    6. Reboot Switch

802.1D Spanning-Tree Protocol (STP) on 5300xl, 3400cl and 6400cl Switches

Menu: Configuring 802.1D STP

1. From the Main Menu, select:

   2. Switch Configuration …

       4. Spanning Tree Operation

```
=========================== CONSOLE - MANAGER MODE -============================
                 Switch Configuration - Spanning Tree Operation
    Protocol Version : RSTP ◄────── Use this field to select the 802.1D version of STP.
    STP Enabled [No] : No
    Force Version [RSTP-operation] : RSTP-operation
    Switch Priority [8] : 8              Hello Time [2] : 2
    Max Age [20] : 20                    Forward Delay [15] : 15

    Port   Type         Cost       Priority   Edge  Point-to-Point   MCheck
    ----   ---------- + ---------  --------   ----  --------------   ------
    A1     10/100TX   | 200000     8          Yes   Force-True       Yes
    A2     10/100TX   | 200000     8          Yes   Force-True       Yes
    A3     10/100TX   | 200000     8          Yes   Force-True       Yes
    A4     10/100TX   | 200000     8          Yes   Force-True       Yes
    A5     10/100TX   | 200000     8          Yes   Force-True       Yes
    A6     10/100TX   | 200000     8          Yes   Force-True       Yes

    Actions->  Cancel      Edit      Save      Help

    Cancel changes and return to previous screen.
    Use arrow keys to change action selection and <Enter> to execute action.
```

Figure 6-6. The Default "Spanning Tree Operation" Screen

2. Press **[E]** (for **Edit**) to highlight the **Protocol Version** field. In the default configuration this field is set to **RSTP**.

3. Press the Space bar once to change the field to STP. This changes the Protocol Version selection to the 802.1D Spanning Tree Protocol.

4. Press ⬇ to highlight the **STP Enabled** field.

5. Press the Space bar to select **Yes**. (**Yes** in this field means to enable spanning-tree operation.)

```
=========================— CONSOLE — MANAGER MODE —=========================
                 Switch Configuration - Spanning Tree Operation

   Protocol Version : STP
   STP Enabled [No] : Yes             Use this field to enable spanning tree.
   Switch Priority [32768] : 32768       Hello Time [2] : 2
   Max Age [20] : 20                     Forward Delay [15] : 15

   Port   Type        Cost       Priority    Mode
   ----   ---------  + --------   --------    ----
   A1     10/100TX   | 10         128         Norm
   A2     10/100TX   | 10         128         Norm
   A3     10/100TX   | 10         128         Norm
   A4     10/100TX   | 10         128         Norm
   A5     10/100TX   | 10         128         Norm
   A6     10/100TX   | 10         128         Norm
   A7     10/100TX   | 10         128         Norm

   Actions->   Cancel      Edit      Save      Help

 Select whether to enable Spanning Tree operation for the switch.
 Use arrow keys to change field selection, <Space> to toggle field choices,
 and <Enter> to go to Actions.
```

Read-Only Fields

Figure 6-7.   Enabling Spanning-Tree Operation

6.   If the remaining STP parameter settings are adequate for your network, go to step 10.

7.   Use **[Tab]** or the arrow keys to select the next parameter you want to change, then type in the new value or press the Space Bar to select a value. (If you need information on STP parameters, press **[Enter]** to select the **Actions** line, then press **H** to get help.)

8.   Repeat step 7 for each additional parameter you want to change.

   Note: For information on the **Mode** parameter, see "STP Fast Mode" on page 6-29.

9.   When you are finished editing parameters, press **[Enter]** to return to the **Actions** line.

10. Press **[S]** to save the currently displayed STP parameter settings. You will then see the "Switch Configuration Menu" with an asterisk (*) at the **Spanning Tree Operation** line, indicating that you must reboot the switch before the Protocol Version change (step 5) takes effect.

The Spanning Tree Operation menu is not present for the Series 4200vl switches

```
========================- CONSOLE - MANAGER MODE -========================
                            Switch Configuration Menu

      1. System Information
      2. Port/Trunk Settings
      3. Network Monitoring Port
     *4. Spanning Tree Operation
      5. IP Configuration
      6. SNMP Community Names
      7. IP Authorized Managers
      8. VLAN Menu...
      0. Return to Main Menu...

 Configures the switch and port Spanning Tree parameters.
 To select menu item, press item number, or highlight item and press <Enter>.
 (*Needs reboot to activate changes.)
```

**Figure 6-8.   The Configuration Menu Indicating a Reboot Is Needed to Implement a
               Configuration Change**

11. Press **[0]** to return to the Main menu.

```
========================- CONSOLE - MANAGER MODE -========================
                                   Main Menu

       1. Status and Counters...
      *2. Switch Configuration...
       3. Console Passwords...
       4. Event Log
       5. Command Line (CLI)
       6. Reboot Switch
       7. Download OS
       8. Run Setup
       9. Stacking...
       0. Logout

 Displays the menu for customizing the switch configuration.
 To select menu item, press item number, or highlight item and press <Enter>.
 (*Needs reboot to activate changes.)
```

**Figure 6-9.   The Main Menu Indicating a Reboot Is Needed To Implement a
               Configuration Change**

12. Press **[6]** to reboot the switch. This implements the Protocol Version
     change (steps 2 and 3 on page 6-22).

CLI: Configuring 802.1D STP

STP Commands Used in This Section

show spanning-tree config	Below
spanning-tree	
protocol-version	page 6-26
forward-delay **< 4 - 30 >**	page 6-27
hello-time **< 1 - 10 >**	page 6-27
maximum-age **< 6 - 40 >**	page 6-27
priority **< 0 - 65535>**	page 6-27
ethernet *< port-list >*	page 6-28
path-cost **< 1 - 65535 >**	page 6-28
priority **< 0 - 255 >**	page 6-28
mode **< norm	fast >**	page 6-28

Viewing the Current STP Configuration.

Syntax: show spanning-tree config

> *Regardless of whether STP is disabled (the default), this command lists the switch's full STP configuration, including general settings and port settings.*

When the switch is configured for 802.1D STP, this command displays information similar to the following:

```
ProCurve(config)# show spanning-tree config
 Spanning Tree Operation

  Protocol Version : STP
  STP Enabled [No] : No
  Switch Priority [32768] : 32768          Hello Time [2] : 2
  Max Age [20] : 20                        Forward Delay [15] : 15

  Port Type       | Cost       Priority Mode
  ---- ---------- + ---------- -------- ----
  A1   10/100TX   | 10         128      Norm
  A2   10/100TX   | 10         128      Norm
  A3   10/100TX   | 10         128      Norm
  A4   10/100TX   | 10         128      Norm
  A5   10/100TX   | 10         128      Norm
  .    .          | .          .        .
  .    .          | .          .        .
  .    .          | .          .        .
```

Command Listing when **STP** is the Protocol Version (See also page 6-14)

Figure 6-10. Example of the Default STP Configuration Listing with 802.1D STP Configured at the Protocol Version

Configuring the Switch To Use the 802.1D Spanning Tree Protocol (STP). In the default configuration, the switch is set to **RSTP** (that is, 802.1w Rapid Spanning Tree), and spanning tree operation is disabled. To reconfigure the switch to 802.1D spanning tree, you must:

1. Change the spanning tree protocol version to **stp**.

2. Use **write memory** to save the change to the startup-configuration.

3. Reboot the switch.

4. If you have not previously enabled spanning-tree operation on the switch, use the **spanning-tree** command again to enable STP operation.

Syntax: spanning-tree protocol-version stp
write memory
boot

For example:

```
ProCurve(config)# spanning-tree protocol-version stp
STP version was changed. To activate the change you must
save the configuration to flash and reboot the device.

ProCurve(config)# write memory

ProCurve(config)# boot
Device will be rebooted, do you want to continue [y/n]? y

Rebooting the System
```

Figure 6-11. Steps for Changing Spanning-Tree Operation to the 802.1D Protocol

Enabling (or Disabling) Spanning Tree Operation on the Switch.

Syntax: [no] spanning-tree

> *This command enables (or disables) spanning tree operation for either spanning tree version—STP/802.1D or RSTP/802.1w (the default). (Default: Disabled.)*

> *Before using this command, ensure that the version of spanning tree you want to use is active on the switch. (See the preceding topic, "Configuring the Switch To Use the 802.1D Spanning Tree Protocol (STP)" on page 6-26.)*

For example:

```
ProCurve(config)# spanning-tree
```

Enabling STP implements the spanning tree protocol for all physical ports on the switch, regardless of whether multiple VLANs are configured. Disabling STP removes protection against redundant loops that can significantly slow or halt a network.

This command enables STP with the current parameter settings or disables STP without losing the most-recently configured parameter settings. (To learn how the switch handles parameter changes, how to test changes without losing the previous settings, and how to replace previous settings with new settings, refer to the chapter titled "Switch Memory and Configuration" in the Management and Configuration Guide for your switch.) When enabling STP, you can also include the STP general and per-port parameters described in the next two sections. When you use the "no" form of the command, you can do so only to disable STP. (STP parameter settings are not changed when you disable STP.)

C a u t i o n   Because incorrect STP settings can adversely affect network performance, HP recommends that you use the default STP parameter settings. You should not change these settings unless you have a strong understanding of how STP operates. For more on STP, see the IEEE 802.1D standard.

These additional commands are available in the Series 3400cl switches,

Syntax: show spanning-tree detail

> *Displays the 802.1D (STP) or 802.1w (RSTP) status and counters for all ports on the switch, depending on which spanning-tree option is enabled.*

show spanning-tree < port-list > [config | detail]

> **config:** *Displays the 802.1D (STP) or 802.1w (RSTP) spanning-tree configuration for the specified ports, depending on which spanning-tree option is enabled.*

> **detail:** *Displays the 802.1D (STP) or 802.1w (RSTP) status and counters for the specified ports, depending on which spanning-tree option is enabled.*

Reconfiguring General STP Operation on the Switch. You can configure one or more of the following parameters:

Table 6-3.   General STP Operating Parameters

Name	Default	Range	Function
priority	32768	0 - 65535	Specifies the priority value used along with the switch MAC address to determine which device is root. The lower a priority value, the higher the priority.
*maximum-age	20 seconds	6 - 40 seconds	Maximum received message age the switch allows for STP information before discarding the message.
*hello-time	2 seconds	1 - 10	Time between messages transmitted when the switch is the root.
*forward-delay	15 seconds	4 - 30 seconds	Time the switch waits before transitioning from the listening to the learning state, and between the learning state to the forwarding state.

*The switch uses its own maximum-age, hello-time, and forward-delay settings only if it is operating as the root device. If another device is operating as the root device, then the switch uses the other device's settings for these parameters.

N o t e

Executing **spanning-tree** alone enables STP. Executing spanning-tree with one or more of the above "STP Operating Parameters" does not enable STP. It only configures the STP parameters (regardless of whether STP is actually running (enabled) on the switch).

Syntax: spanning-tree
        priority < 0 - 65355 >
        maximum-age < 6 - 40 seconds >
        hello-time < 1 - 10 seconds >
        forward-delay < 4 - 30 seconds >

Default: Refer to table 6-3, above.

For example, to configure a **maximum-age** of 30 seconds and a **hello-time** of 3 seconds for STP:

```
ProCurve(config)# spanning-tree maximum-age 30 hello-time 3
```

Reconfiguring Per-Port STP Operation on the Switch.

Syntax: spanning-tree < *port-list* > path-cost **<** 1 - 65535 > priority **<** 0 - 255 **>** mode **<** norm | fast >

Enables STP (if not already enabled) and configures the per-port parameters listed in table 6-4.

Table 6-4.   Per-Port STP Parameters

Name	Default	Range	Function
path-cost	Ethernet: 100 10/100Tx: 10 100 Fx: 10 Gigabit: 5	1 - 65535	Assigns an individual port cost that the switch uses to determine which ports are the forwarding ports.
priority	128	0 - 255	Used by STP to determine the port(s) to use for forwarding. The port with the lowest number has the highest priority.
mode	norm	norm *- or -* fast *- or -* uplink	Specifies whether a port progresses through the listening, learning, and forwarding (or blocking) states ("norm" mode) or transitions directly to the forwarding state ("fast" mode). • For information on when to use Fast mode, see "STP Fast Mode" on page 6-29.) • For information on Uplink mode, see "Fast-Uplink Spanning Tree Protocol (STP)" on page 6-30

You can also include STP general parameters in this command. See "Recon-figuring General STP Operation on the Switch" on page 6-27.

For example, the following configures ports C5 and C6 to a path cost of **15**, a priority of **100**, and **fast** mode:

```
ProCurve(config)# spanning-tree c5-c6 path-cost 15 priority 100 mode fas
```

STP Fast Mode

For standard STP operation, when a network connection is established on a device that is running STP, the port used for the connection goes through a sequence of states (Listening and Learning) before getting to its final state (Forwarding or Blocking, as determined by the STP negotiation). This sequence takes two times the forward delay value configured for the switch. The default is 15 seconds on ProCurve switches, per the IEEE 802.1D standard recommendation, resulting in a total STP negotiation time of 30 seconds. Each switch port goes through this start-up sequence whenever the network con-nection is established on the port. This includes, for example, when the switch or connected device is powered up, or the network cable is connected.

A problem can arise from this long STP start-up sequence because some end nodes are configured to automatically try to access a network server when-ever the end node detects a network connection. Typical server access includes to Novell servers, DHCP servers, and X terminal servers. If the server access is attempted during the time that the switch port is negotiating its STP

state, the server access will fail. To provide support for this end node behavior, the switches covered by this manual offer a configuration mode, called "Fast Mode", that causes the switch port to skip the standard STP start-up sequence and put the port directly into the "Forwarding" state, thus allowing the server access request to be forwarded when the end node needs it.

If you encounter end nodes that repeatedly indicate server access failure when attempting to bring up their network connection, and you have enabled STP on the switch, try changing the configuration of the switch ports associated with those end nodes to STP Fast Mode.

C a u t i o n

The Fast Mode configuration should be used only on switch ports connected to end nodes. Changing the Mode to Fast on ports connected to hubs, switches, or routers may cause loops in your network that STP may not be able to immediately detect, in all cases. This will cause temporary loops in your network. After the fast start-up sequence, though, the switch ports operate according to the STP standard, and will adjust their state to eliminate continuing network loops.

To Enable or Disable Fast Mode for a Switch Port: You can use either the CLI or the menu interface to toggle between STP Fast mode and STP Normal mode. (To use the menu interface, see "Menu: Configuring 802.1D STP" on page 6-22.)

Syntax: spanning-tree < *port-list* > mode <fast | norm>

For example, to configure Fast mode for ports C1-C3 and C5:

```
ProCurve(config)# spanning-tree c1-c3,c5 mode fast
```

Fast-Uplink Spanning Tree Protocol (STP)

Fast-Uplink STP is an option added to the switch's 802.1D STP to improve the recovery (convergence) time in wiring closet switches with redundant uplinks. Specifically, a switch having redundant links toward the root device can decrease the convergence time (or failover) to a new uplink (STP root) port to as little as ten seconds. To realize this performance, the switch must be:

■ Used as a wiring closet switch (also termed an *edge switch* or a *leaf switch*).

■ Configured for fast-uplink STP mode on two or more ports intended for redundancy in the direction of the root switch, so that at any time only one of the redundant ports is expected to be in the forwarding state.

N o t e   Fast-Uplink STP operates only with 802.1D STP and is not available with the Rapid STP (802.1w) feature (6-11).

C a u t i o n   In general, fast-uplink spanning tree on the switch is useful when running STP in a tiered topology that has well-defined edge switches. Also, ensure that an interior switch is used for the root switch and for any logical backup root switches. You can accomplish this by using the Spanning Tree Priority (some-times termed bridge priority) settings that define the primary STP root switch and at least one failover root switch (in the event that the primary root switch fails). Inappropriate use of Fast-Uplink STP can cause intermittent loops in a network topology. For this reason, the Fast-Uplink STP feature should be used only by experienced network administrators who have a strong understanding of the IEEE 802.1D standard and STP interactions and operation. If you want to learn more about STP operation, you may find it helpful to refer to publications such as:

Perlman, Radia, *Interconnections: Bridges, Routers, Switches, and Internetworking Protocols* (second edition), Addison-Wesley Profes-sional Computing Series, October 1999

N o t e   When properly implemented, fast-uplink STP offers a method for achieving faster failover times than standard STP, and is intended for this purpose for instances where 802.1D STP has been chosen over 802.1w RSTP.

To use fast-uplink STP, configure fast-uplink (**Mode** = **Uplink**) only on the switch's upstream ports; (that is, two or more ports forming a group of redundant links in the direction of the STP root switch). If the active link in this group goes down, fast-uplink STP selects a different upstream port as the root port and resumes moving traffic in as little as ten seconds. The device(s) on the other end of the links must be running STP. However, because fast uplink should be configured only on the switch's uplink ports, the device(s)

on the other end of the links can be either ProCurve devices or another vendor's devices, regardless of whether they support fast uplink. For example:
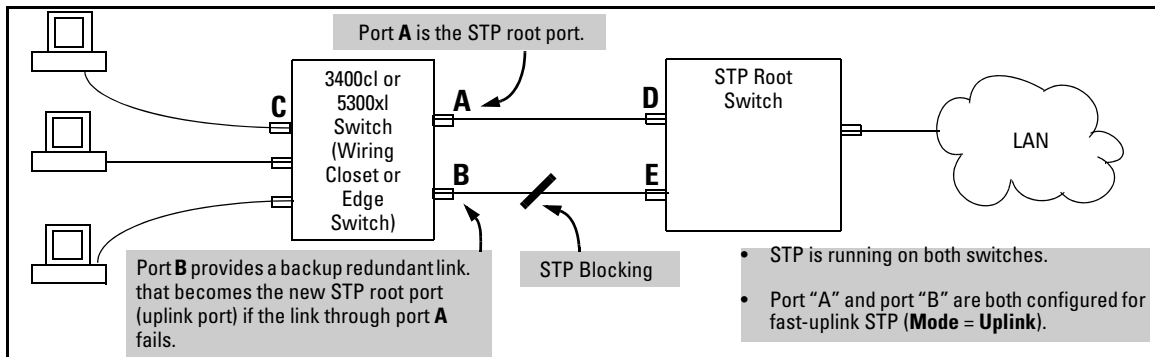
Figure 6-12. Example of How To Implement Fast-Uplink STP

Terminology

Term	Definition
downlink port (downstream port)	A switch port that is linked to a port on another switch (or to an end node) that is sequentially further away from the STP root device. For example, port "C" in figure 6-12, above, is a downlink port.
edge switch	For the purposes of fast-uplink STP, this is a switch that has no other switches connected to its downlink ports. An edge switch is sequentially further from the root device than other switches to which it is connected. Also termed *wiring closet switch* or *leaf switch*. For example, switch "4" in figure 6-13 (page 6-33) is an edge switch.
interior switch	In an STP environment, a switch that is sequentially closer to the STP root device than one or more other switches to which it is connected. For example, switches "1", "2", and "3" in figure 6-13 (page 6-33) are interior switches.
single-instance spanning tree	A single spanning-tree ensuring that there are no logical network loops associated with any of the connections to the switch, regardless of whether there are any VLANs configured on the switch. For more information, see "Spanning Tree Protocol (STP)" in chapter 9, "Configuring Advanced Features", in the Management and Configuration Guide for your switch.
uplink port (upstream port)	A switch port linked to a port on another switch that is sequentially closer to the STP root device. For example, ports "A" and "B" in figure 6-12 on page 6-32 are uplink ports.
wiring closet switch	Another term for an "edge" or "leaf" switch.

When single-instance spanning tree (STP) is running in a network and a forwarding port goes down, a blocked port typically requires a period of

(2 x (*forward delay*) + link down detection)

to transition to forwarding. In a normal spanning tree environment, this transition is usually 30 seconds (with the **Forward Delay** parameter set to its default of 15 seconds). However, by using the fast-uplink spanning tree feature, a port on a switch used as an *edge switch* can make this transition in as little as ten seconds. (In an STP environment, an *edge switch* is a switch that is connected only to switches that are closer to the STP root switch than the edge switch itself, as shown by switch "4" in figure 6-13, below.)
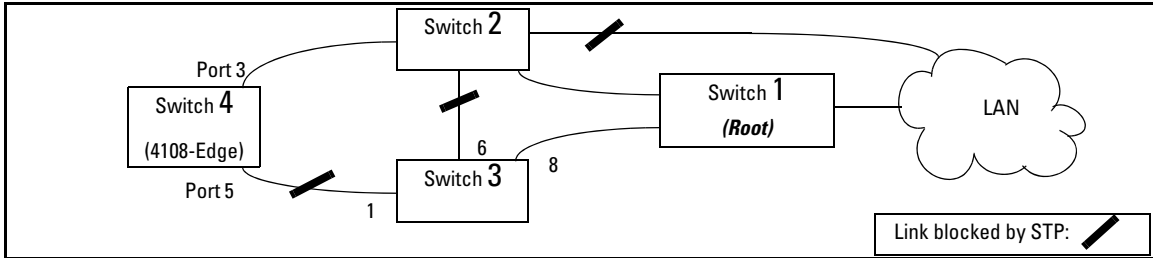
Figure 6-13. Example of an Edge Switch in a Topology Configured for STP Fast Uplink

In figure 6-13, STP is enabled and in its default configuration on all switches, unless otherwise indicated in table 6-5, below:

Table 6-5.   STP Parameter Settings for Figure 6-13

STP Parameter	Switch "1"	Switch "2"	Switch "3"	Switch "4"
Switch Priority	0[1]	1[2]	32,768 (default)	32,768 (default)
(Fast) Uplink	No	No	No	Ports 3 & 5

[1]This setting ensures that Switch "1" will be the primary root switch for STP in figure 6-13.
[2]This setting ensures that Switch "2" will be the backup root switch for STP in figure 6-13.

With the above-indicated topology and configuration:

- **Scenario 1:** If the link between switches "4" and "2" goes down, then the link between switches "4" and "3" will begin forwarding in as little as ten seconds.

- **Scenario 2:** If Switch "1" fails, then:
  - Switch "2" becomes the root switch.
  - The link between Switch "3" and Switch "2" begins forwarding.
  - The link between Switch "2" and the LAN begins forwarding.

Operating Rules for Fast Uplink

- A switch with ports configured for fast uplink must be an edge switch and not either an interior switch or the STP root switch.

Configure fast-uplink on only the edge switch ports used for providing redundant STP uplink connections in a network. (Configuring Fast-Uplink STP on ports in interior switches can create network performance problems.) That is, a port configured for STP uplink should not be connected to a switch that is sequentially further away from the STP root device. For example, switch "4" in figure 6-13 (page 6-33) is an edge switch.

■ Configure fast uplink on a group (two or more) of redundant edge-switch uplink ports where only one port in the group is expected to be in the forwarding state at any given time.

■ Edge switches cannot be directly linked together using fast-uplink ports. For example, the connection between switches 4 and 5 in figure 6-14 is not allowed for fast-uplink operation.
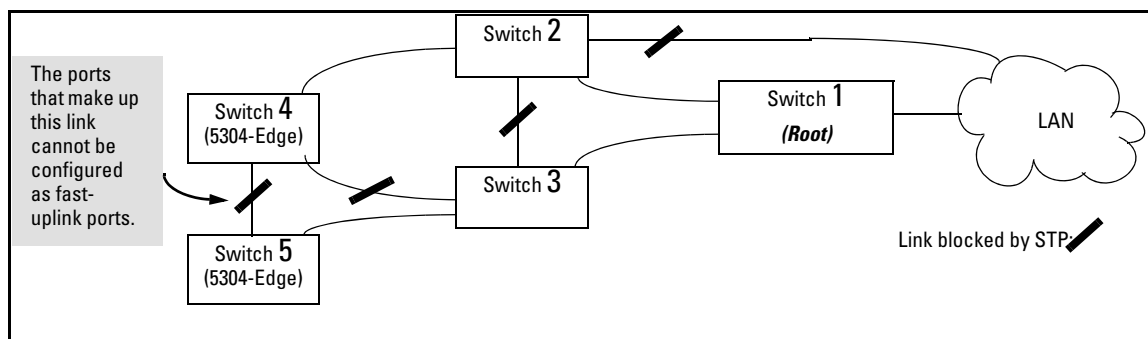
Figure 6-14. Example of a Disallowed Connection Between Edge Switches
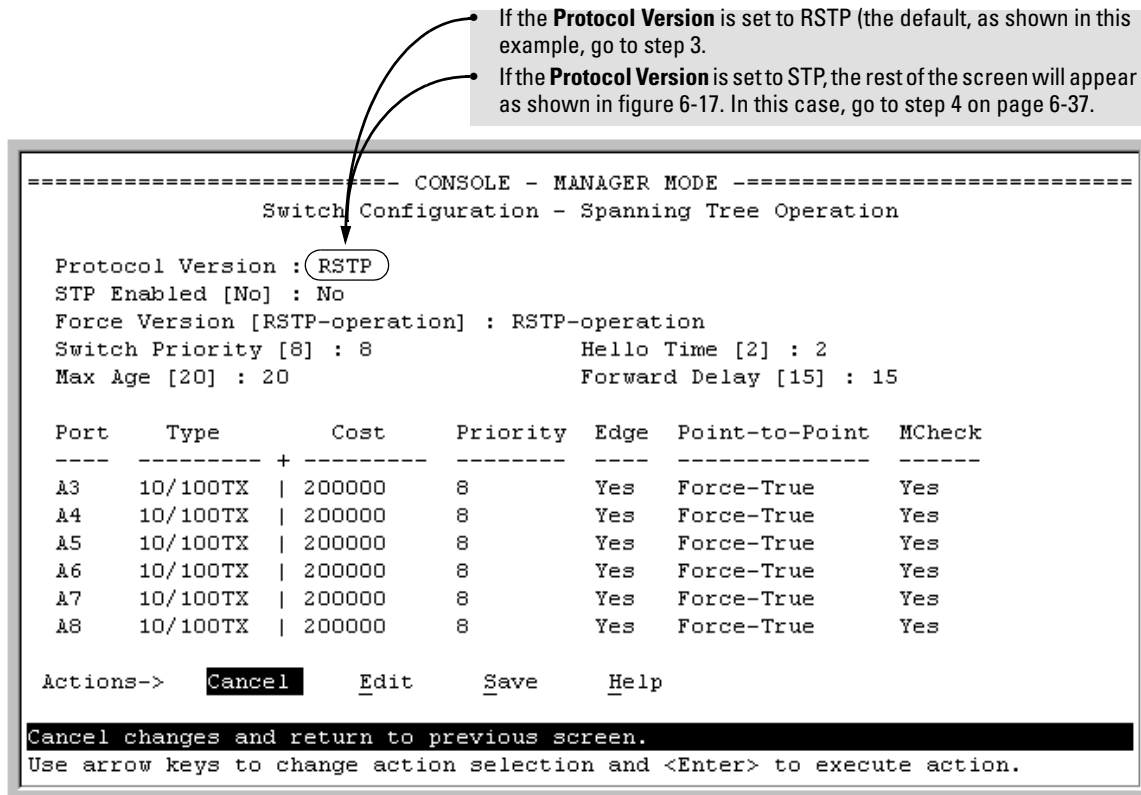
■ Apply fast-uplink only on the uplink ports of an edge switch. For example, on switch "4" (an edge switch) in figure 6-14 above, only the ports connecting switch "4" to switches "2" and "3" are upstream ports that would use fast uplink. Note also that fast uplink should *not* be configured on both ends of a point-to-point link, but only on the uplink port of an edge switch.

■ Ensure that the switch you intend as a backup root device will in fact become the root if the primary root fails, and that no ports on the backup root device are configured for fast-uplink operation. For example, if the **STP Priority** is the same on all switches—default: 32768—then the switch with the lowest MAC address will become the root switch. If that switch fails, then the switch with the next-lowest MAC address will become the root switch. Thus, you can use **STP Priority** to control which switch STP selects as the root switch and which switch will become the root if the first switch fails.

■ Fast-Uplink STP requires a minimum of two uplink ports.

Menu: Viewing and Configuring Fast-Uplink STP

You can use the menu to quickly display the entire STP configuration and to make any STP configuration changes.

To View and/or Configure Fast-Uplink STP. This procedure uses the Spanning Tree Operation screen to enable STP and to set the Mode for fast-uplink STP operation.

1. From the Main Menu select:

   2. Switch Configuration …
         4. Spanning Tree Operation

2. In the default STP configuration, RSTP is the selected protocol version. If this is the case on your switch, you must change the Protocol Version to STP in order to use Fast-Uplink STP:

If the **Protocol Version** is set to RSTP (the default, as shown in this example, go to step 3.

If the **Protocol Version** is set to STP, the rest of the screen will appear as shown in figure 6-17. In this case, go to step 4 on page 6-37.

```
=========================- CONSOLE - MANAGER MODE -===============================
                  Switch Configuration - Spanning Tree Operation

   Protocol Version : (RSTP)
   STP Enabled [No] : No
   Force Version [RSTP-operation] : RSTP-operation
   Switch Priority [8] : 8                  Hello Time [2] : 2
   Max Age [20] : 20                        Forward Delay [15] : 15

   Port    Type         Cost      Priority   Edge  Point-to-Point   MCheck
   ----   ---------  + ---------  --------   ----  --------------   ------
   A3     10/100TX   | 200000     8          Yes   Force-True       Yes
   A4     10/100TX   | 200000     8          Yes   Force-True       Yes
   A5     10/100TX   | 200000     8          Yes   Force-True       Yes
   A6     10/100TX   | 200000     8          Yes   Force-True       Yes
   A7     10/100TX   | 200000     8          Yes   Force-True       Yes
   A8     10/100TX   | 200000     8          Yes   Force-True       Yes

   Actions->    Cancel      Edit      Save      Help

 Cancel changes and return to previous screen.
 Use arrow keys to change action selection and <Enter> to execute action.
```

Figure 6-15. The Default STP Screen With the Protocol Version Field Set to "RSTP"

3. If the Protocol Version is set to RSTP (as shown in figure 6-15), do the following:

   a. Press **[E]** (**Edit**) to move the cursor to the **Protocol Version** field.

   b. Press the Space bar once to change the **Protocol Version** field to STP.

   c. Press **[Enter]** to return to the command line.

   d. Press **[S]** (for **Save**) to save the change and exit from the Spanning Tree Operation screen. you will then see a screen with the following:

The asterisk indicates that you must reboot the switch to implement the configuration change from RSTP to STP.

The Spanning Tree Operation menu item is not present for Series 4200vl switches.

```
=========================- CONSOLE - MANAGER MODE -===
                              Switch Configuration Menu

  1. System Information
  2. Port/Trunk Settings
  3. Network Monitoring Port
*4. Spanning Tree Operation
  5. IP Configuration
  6. SNMP Community Names
  7. IP Authorized Managers
  8. VLAN Menu...
  0. Return to Main Menu...
```

Figure 6-16. Changing from RSTP to STP Requires a System Reboot

   e. Press **[0]** (zero) to return to the Main Menu, then **[6]** to reboot the switch.

   f. After you reboot the switch, enter the menu command at the CLI to return to the Main Menu, then select:

> **2. Switch Configuration …**
> > **4. Spanning Tree Operation**

You will then see the Spanning Tree screen with **STP** (802.1D) selected in the **Protocol Version** field (figure 6-17).

```
==========================- CONSOLE - MANAGER MODE -=============================
                Switch Configuration - Spanning Tree Operation
  Protocol Version : STP
  STP Enabled [No] : No
  Switch Priority [32768] : 32768        Hello Time [2] : 2
  Max Age [20] : 20                      Forward Delay [15] : 15

  Port    Type          Cost      Priority    Mode
  ----    ---------  +  ----   ---   --------   ----
  A1      10/100TX   |  100          128        Norm
  A4      10/100TX   |  100          128        Norm
  A5      10/100TX   |  100          128        Norm
  A6      10/100TX   |  100          128        Norm
  A7      10/100TX   |  100          128        Norm
  A3      10/100TX   |  100          128        Norm
  A9      10/100TX   |  100          128        Norm

  Actions->   Cancel      Edit      Save      Help

Cancel changes and return to previous screen.
Use arrow keys to change action selection and <Enter> to execute action.
```

In this example, ports A2 and A3 have already been configured as a port trunk (**Trk1**), which appears at the end of the port listing.

All ports (and the trunk) are in their default STP configuration.

Note: In the actual menu screen, you must scroll the cursor down the port list to view the trunk configuration (ports A2 and A3).

Figure 6-17. The Spanning Tree Operation Screen

4. On the ports and/or trunks you want to use for redundant fast uplink connections, change the mode to **Uplink**. In this example, port A1 and Trk1 (using ports A2 and A3) provide the redundant uplinks for STP:

   a. Press **[E]** (for **Edit**), then enable STP on the switch by using the Space bar to select **Yes** in the Spanning Tree Enabled field.

   b. Use **[Tab]** to move to the Mode field for port A1.

   c. Use the Space bar to select **Uplink** as the mode for port A1.

   d. Use ↓ to move to the Mode field for Trk1.

   e. Use the Space bar to select **Uplink** as the Mode for Trk1.

   f. Press **[Enter]** to return the cursor to the Actions line.
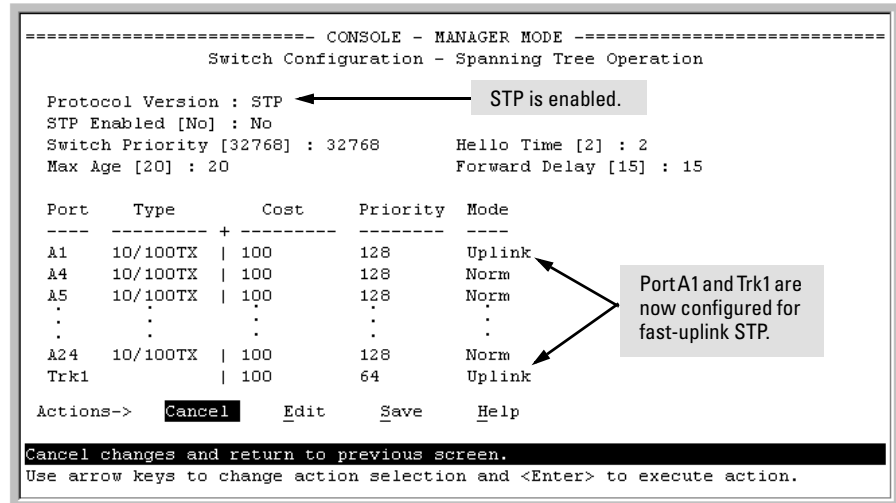
```
==========================- CONSOLE - MANAGER MODE -=============================
                  Switch Configuration - Spanning Tree Operation

   Protocol Version : STP  ◄─────────────────── STP is enabled.
   STP Enabled [No] : No
   Switch Priority [32768] : 32768        Hello Time [2] : 2
   Max Age [20] : 20                      Forward Delay [15] : 15

   Port    Type        Cost       Priority  Mode
   ----  --------- + --------- -------- ----
   A1    10/100TX  | 100          128      Uplink
   A4    10/100TX  | 100          128      Norm              Port A1 and Trk1 are
   A5    10/100TX  | 100          128      Norm              now configured for
   .         .         .            .        .               fast-uplink STP.
   .         .         .            .        .
   A24   10/100TX  | 100          128      Norm
   Trk1            | 100          64       Uplink

   Actions->   Cancel      Edit      Save      Help

 Cancel changes and return to previous screen.
 Use arrow keys to change action selection and <Enter> to execute action.
```

Figure 6-18. Example of STP Enabled with Two Redundant Links Configured for Fast-Uplink STP

5. Press **[S]** (for **Save**) to save the configuration changes to flash (non-volatile) memory.

To View Fast-Uplink STP Status. Continuing from figures 6-17 and 6-18 in the preceding procedure, this task uses the same screen that you would use to view STP status for other operating modes.

1. From the Main Menu, select:

    1. Status and Counters …
        7. Spanning Tree Information

```
==========================- CONSOLE - MANAGER MODE -====================
               Status and Counters - Spanning Tree Information
    STP Enabled               : Yes
    Switch Priority           : 32,768
    Hello Time                : 2
    Max Age                   : 20
    Forward Delay             : 15

    Topology Change Count     : 2
    Time Since Last Change : 15 mins

    Root MAC Address          : 0060b0-889e00
    Root Path Cost            : 20
    Root Port                 : Trk1
    Root Priority             : 16000

    Actions->    Back      Show ports      Help

Return to previous screen.
Use arrow keys to change action selection and <Enter> to execute action.
```

Indicates which uplink is the active path to the STP root device.

Note: A switch using fast-uplink STP must never be the STP root device.

Figure 6-19. Example of STP Status with Trk1 (Trunk 1) as the Path to the STP Root Device

2. Press **[S]** (for **Show ports**) to display the status of individual ports.

```
==========================- CONSOLE - MANAGER MODE -==========================
          Status and Counters - Spanning Tree - Port Information

   Port      Type     Cost   Priority    State      Designated Bridge
   ------   --------- -----  --------   ----------   -----------------
   A1       10/100TX   10      128      Blocking     0030c1-7fcc40
   A4       10/100TX   10      128      Disabled
   A5       10/100TX   10      128      Forwarding   00306e-d61880
   A6       10/100TX   10      128      Forwarding   00306e-d61880
   :          :         :       :          :            :
   A24      10/100TX   10      128      Forwarding   00306e-d61880
   Trk1     Trunk      10       64      Forwarding   0030c1-7fcc40

   Actions->    Back    Help

Return to previous screen.
Use up/down arrow keys to scroll to other entries, left/right arrow keys to
change action selection, and <Enter> to execute action.
```

Redundant STP Link in (Fast) Uplink Mode

Links to PC or Workstation End Nodes

Redundant STP Link in (Fast) Uplink Mode

Figure 6-20. Example of STP Port Status with Two Redundant STP Links

In figure 6-20:

- Port A1 and Trk1 (trunk 1; formed from ports A2 and A3) are redundant fast-uplink STP links, with trunk 1 forwarding (the active link) and port A1 blocking (the backup link). (To view the configuration for port A1 and Trk1, see figure 6-18 on page 6-38.)

- If the link provided by trunk 1 fails (on both ports), then port A1 begins forwarding in fast-uplink STP mode.

- Ports A5, A6, and A24 are connected to end nodes and do not form redundant links.

CLI: Viewing and Configuring Fast-Uplink STP

Using the CLI to View Fast-Uplink STP. You can view fast-uplink STP using the same **show** commands that you would use for standard STP operation:

Syntax: show spanning-tree

Lists STP status.

Syntax: show spanning-tree config

Lists STP configuration for the switch and for individual ports.

For example, figures 6-21 and 6-22 illustrate a possible topology, STP status listing, and STP configuration for a switch with:

- STP enabled and the switch operating as an Edge switch
- Port A1 and trunk 1 (Trk1) configured for fast-uplink STP operation
- Several other ports connected to PC or workstation end nodes
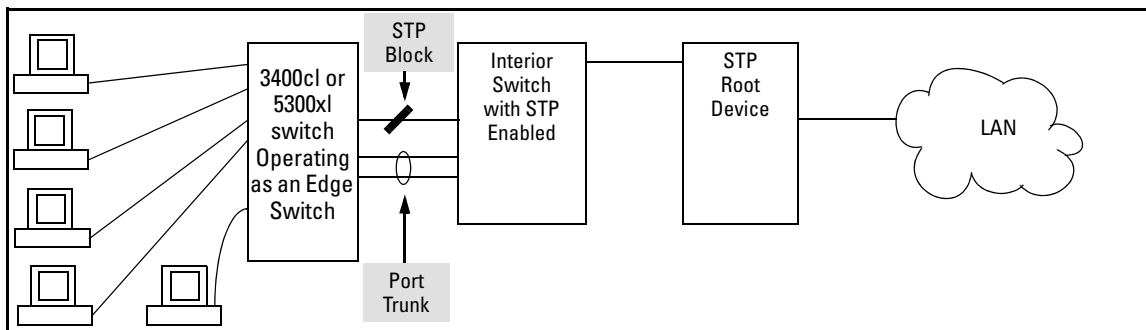
Figure 6-21. Example Topology for the Listing Shown in Figure 6-22

```
ProCurve (config)# show spanning-tree

Status and Counters - Spanning Tree Information

 STP Enabled           : Yes
 Switch Priority       : 32,768
 Hello Time            : 2
 Max Age               : 20
 Forward Delay         : 15

 Topology Change Count : 25
 Time Since Last Change : 13 mins

 Root MAC Address      : 0001e7-a09900
 Root Path Cost        : 20
 Root Port             : Trk1
 Root Priority         : 16768

 Port   Type      Cost  Priority State    | Designated Bridge
 ------ --------- ----- -------- --------- + -----------------
 A1     10/100TX  10    128      Blocking  | 0030c1-a9c800
 A4     10/100TX  10    128      Disabled  |
 A5     10/100TX  10    128      Forwarding| 00306e-d61880
 A6     10/100TX  10    128      Forwarding| 00306e-d61880
 -  MORE  --
 A7     10/100TX  10    128      Forwarding| 00306e-d61880
 A8     10/100TX  10    128      Disabled  |
 A9     10/100TX  10    128      Forwarding| 00306e-d61880
 A10    10/100TX  10    128      Forwarding| 00306e-d61880
 A11    10/100TX  10    128      Disabled  |
 A12    10/100TX  10    128      Disabled  |
 Trk1             10    64       Forwarding| 0030c1-a9c800
```

Indicates that Trk1 (Trunk 1) provides the currently active path to the STP root device.

Redundant STP link in the Blocking state.

Links to PC or Workstation End Nodes

Redundant STP link in the Forwarding state. (See the "Root Port field, above. This is the currently active path to the STP root device.)
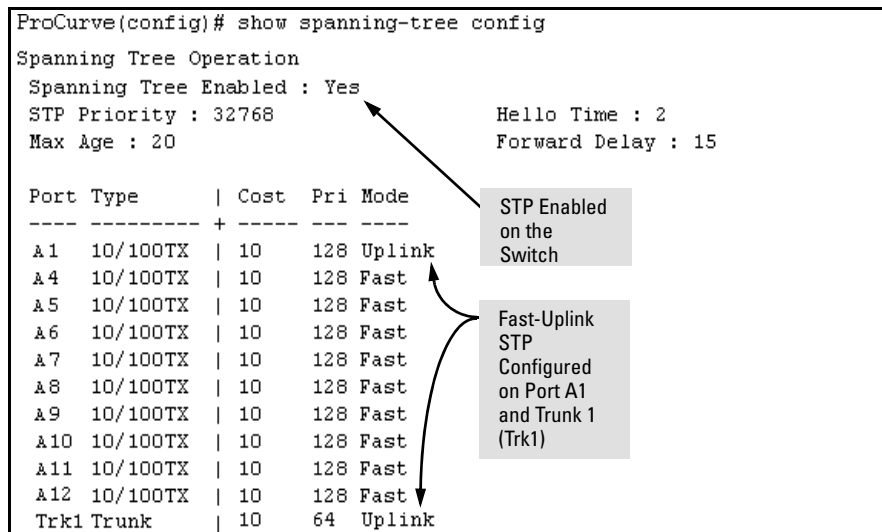
Figure 6-22. Example of a Show Spanning-Tree Listing for the Topology Shown in Figure 6-21

```
ProCurve(config)# show spanning-tree config

Spanning Tree Operation
 Spanning Tree Enabled : Yes
 STP Priority : 32768                    Hello Time : 2
 Max Age : 20                            Forward Delay : 15

 Port Type      | Cost  Pri Mode
 ---- --------- + ----- --- ----
 A1   10/100TX  | 10    128 Uplink
 A4   10/100TX  | 10    128 Fast
 A5   10/100TX  | 10    128 Fast
 A6   10/100TX  | 10    128 Fast
 A7   10/100TX  | 10    128 Fast
 A8   10/100TX  | 10    128 Fast
 A9   10/100TX  | 10    128 Fast
 A10  10/100TX  | 10    128 Fast
 A11  10/100TX  | 10    128 Fast
 A12  10/100TX  | 10    128 Fast
 Trk1 Trunk     | 10    64  Uplink
```

STP Enabled on the Switch

Fast-Uplink STP Configured on Port A1 and Trunk 1 (Trk1)

Figure 6-23. Example of a Configuration Supporting the STP Topology Shown in Figure 6-21

Using the CLI To Configure Fast-Uplink STP. This example uses the CLI to configure the switch for the fast-uplink operation shown in figures 6-21, 6-22, and 6-23. (The example assumes that ports A2 and A3 are already configured as members of the port trunk—Trk1, and all other STP parameters are left in their default state.)

Note that the default STP Protocol Version is RSTP (Rapid STP, or 802.1w). Thus, if the switch is set to the STP default, you must change it to the STP (802.1D) Protocol Version before you can configure Fast-Uplink. For example:

```
ProCurve(config)# show spanning-tree                            Lists STP
 Status and Counters - Spanning Tree Information                configuration.
  Protocol Version : RSTP                                       Shows the default
  STP Enabled : No                                              STP protocol

  Port Type       Cost      Priority State    | Designated Bridge
  ---- --------- --------- -------- ---------- + -----------------

                                                              1. Changes the Spanning-Tree
ProCurve(config)# spanning-tree protocol-version stp             protocol to STP (required for
STP version was changed. To activate the change you must         Fast-Uplink).
save the configuration to flash and reboot the device.        2. Saves the change to the
ProCurve(config)# write mem                                      startup-configuration
ProCurve(config)# boot                                        3. Reboots the switch. (Required
Device will be rebooted, do you want to continue [y/n]?  y       for this configuration change.)
Boot from primary flash
```

Figure 6-24. Example of Changing the STP Configuration from the Default RSTP (802.1w) to STP (802.1D)

Syntax: spanning-tree < *port/trunk-list* > mode uplink

> *Enables STP on the switch and configures fast-uplink STP on the designated interfaces (port or trunk).*

For example:

```
ProCurve(config)# spanning-tree e A1,trk1 mode uplink
```

Operating Notes

Effect of Reboots on Fast-Uplink STP Operation. When configured, fast-uplink STP operates on the designated ports in a running switch. However, if the switch experiences a reboot, the fast-uplink ports (Mode = **Uplink**) use the longer forwarding delay used by ports on standard 802.1D STP (non fast-uplink). This prevents temporary loops that could otherwise result while the switch is determining the STP status for all ports. That is, on ports configured for fast-uplink STP, the first STP state transition after a reboot takes the same amount of time as for redundant ports that are not configured for fast-uplink STP.

Using Fast Uplink with Port Trunks. To use a port trunk for fast-uplink STP, configure it in the same way that you would an individual port for the same purpose. A port trunk configured for fast uplink operates in the same way as an individual, non-trunked port operates; that is, as a logical port.

N o t e   When you add a port to a trunk, the port takes on the STP mode configured for the trunk, regardless of which STP mode was configured on the port before it was added to the trunk. Thus, all ports belonging to a trunk configured with **Uplink** in the STP **Mode** field will operate in the fast-uplink mode. (If you remove a port from a trunk, the port reverts to the STP Mode setting it had before you added the port to the trunk.)

To use fast uplink over a trunk, you must:

1. Create the trunk.

2. Configure the trunk for fast uplink in the same way that you would configure an individual port for fast uplink.

When you first create a port trunk, its STP Mode setting will be **Norm**, regardless of whether one or more ports in the trunk are set to fast uplink (Mode = **Uplink**). You must still specifically configure the trunk Mode setting to **Uplink**. Similarly, if you eliminate a trunk, the Mode setting on the individual ports in the trunk will return to their previous settings.

For Troubleshooting Information on Fast Uplink. Refer to the section titled "Spanning-Tree Protocol (STP) and Fast-Uplink Problems" in appendix C, "Troubleshooting" in the Management and Configuration Guide for your switch.

802.1s Multiple Spanning Tree Protocol (MSTP)

The 802.1D and 802.1w spanning tree protocols (5300xl, 3400cl and 6400cl switches) operate without regard to a network's VLAN configuration, and maintain one common spanning tree throughout a bridged network. Thus, these protocols map one loop-free, logical topology on a given physical topology. The 802.1s Multiple Spanning Tree protocol (MSTP) uses VLANs to create multiple spanning trees in a network, which significantly improves network resource utilization while maintaining a loop-free environment.

While the per-VLAN spanning tree approach adopted by some vendors over-comes the network utilization problems inherent in using STP or RSTP, using a per-VLAN technology with multiple VLANs can overload the switch's CPU. MSTP on the switches covered by this guide complies with the IEEE 802.1s standard, and extends STP and RSTP functionality to map multiple independent spanning tree instances onto a physical topology. With MSTP, each spanning tree instance can include one or more VLANs and applies a separate, per-instance forwarding topology. Thus, where a port belongs to multiple VLANs, it may be dynamically blocked in one spanning tree instance, but forwarding in another instance. This achieves load-balancing across the network while keeping the switch's CPU load at a moderate level (by aggregating multiple VLANs in a single spanning tree instance). Like RSTP, MSTP provides fault tolerance through rapid, automatic reconfiguration if there is a failure in a network's physical topology.

With MSTP-capable switches, you can create a number of MST regions containing multiple spanning tree instances. This requires the configuration of a number of MSTP-capable switches. However, it is NOT necessary to do this. You can just enable MSTP on an MSTP-capable switch and a spanning tree instance is created automatically. This instance always exists by default when spanning tree is enabled, and is the spanning tree instance that communicates with STP and RSTP environments. The MSTP configuration commands operate exactly like RSTP commands and MSTP is backward-compatible with the RSTP-enabled and STP-enabled switches in your network.

C a u t i o n   Spanning tree interprets a switch mesh as a single link (meshing is supported on the 5300xl, 3400cl and 6400cl switches only). Because the switch automatically gives faster links a higher priority, the default MSTP parameter settings are usually adequate for spanning tree operation. Also, because incorrect MSTP settings can adversely affect network performance, you should not change the MSTP settings from their default values unless you have a strong understanding of how spanning tree operates.

In a mesh environment, the default MSTP timer settings (**Hello Time** and **Forward Delay**) are usually adequate for MSTP operation. Because a packet crossing a mesh may traverse several links within the mesh, using smaller-than-default settings for the MSTP **Hello Time** and **Forward Delay** timers can cause unnecessary topology changes and end-node connectivity problems.

For MSTP information beyond what is provided in this manual, refer to the IEEE 802.1s standard.

MSTP Structure

MSTP maps active, separate paths through separate spanning tree instances and between MST regions. Each MST region comprises one or more MSTP switches. Note that MSTP recognizes an STP or RSTP LAN as a distinct spanning-tree region.
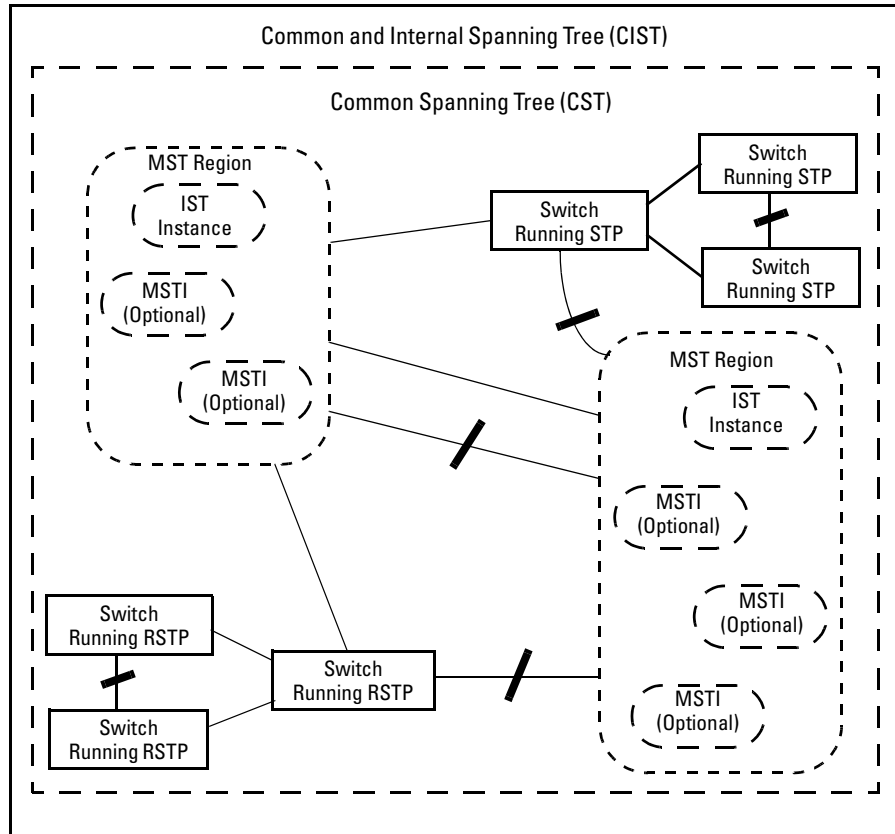
Figure 6-25. Example of MSTP Network with Legacy STP and RSTP Devices Connected

Common and Internal Spanning Tree (CIST): The CIST identifies the regions in a network and administers the CIST root bridge for the network, the root bridge for each region, and the root bridge for each spanning-tree instance in each region.

Common Spanning Tree (CST): The CST administers the connectivity among the MST regions, STP LANs, and RSTP LANs in a bridged network.

MST Region: An MST region comprises the VLANs configured on physically connected MSTP switches. All switches in a given region must be configured with the same VLANs and Multiple Spanning Tree Instances (MSTIs).

Internal Spanning Tree (IST): The IST administers the topology within a given MST region. When you configure a switch for MSTP operation, the switch automatically includes all of the static VLANs configured on the switch

in a single, active spanning tree topology (instance) within the IST. This is termed the "IST instance". Any VLANs you subsequently configure on the switch are added to this IST instance. To create separate forwarding paths within a region, group specific VLANs into different Multiple Spanning Tree Instances (MSTIs). (Refer to "Multiple Spanning Tree Instance", below.)

Types of Multiple Spanning Tree Instances: A multiple spanning tree network comprises separate spanning-tree instances existing in an MST region. (There can be multiple regions in a network.) Each instance defines a single forwarding topology for an exclusive set of VLANs. By contrast, an STP or RSTP network has only one spanning tree instance for the entire network, and includes all VLANs in the network. (An STP or RSTP network operates as a single-instance network.) A region can include two types of STP instances:

■ **Internal Spanning-Tree Instance (IST Instance):** This is the default spanning tree instance in any MST region. It provides the root switch for the region and comprises all VLANs configured on the switches in the region that are not specifically assigned to Multiple Spanning Tree Instances (MSTIs, described below). All VLANs in the IST instance of a region are part of the same, single spanning tree topology, which allows only one forwarding path between any two nodes belonging to any of the VLANs included in the IST instance. All switches in the region must belong to the set of VLANs that comprise the IST instance. Note that the switch automatically places dynamic VLANs (resulting from GVRP operation) in the IST instance. Dynamic VLANs cannot exist in an MSTI (described below).

■ **MSTI (Multiple Spanning Tree Instance):** This type of configurable spanning tree instance comprises all static VLANs you specifically assign to it, and must include at least one VLAN. The VLAN(s) you assign to an MSTI must initially exist in the IST instance of the same MST region. When you assign a static VLAN to an MSTI, the switch removes the VLAN from the IST instance. (Thus, you can assign a VLAN to only one MSTI in a given region.) All VLANs in an MSTI operate as part of the same single spanning tree topology. (The switch does not allow dynamic VLANs in an MSTI.)

C a u t i o n   When you enable MSTP on the switch, the default MSTP spanning tree configuration settings comply with the values recommended in the IEEE 802.1s Multiple Spanning Tree Protocol (MSTP) standard. Note that inappropriate changes to these settings can result in severely degraded network performance. For this reason, *HP strongly recommends that changing these default settings be reserved only for experienced network administrators who have a strong understanding of the IEEE 802.1D/w/s standards and operation.*

How MSTP Operates

In the factory default configuration, spanning tree operation is off. Also, the switch retains its currently configured spanning tree parameter settings when disabled. Thus, if you disable spanning tree, then later re-enable it, the parameter settings will be the same as before spanning tree was disabled. The switch also includes a "Pending" feature that enables you to exchange MSTP configurations with a single command. (Refer to "Enabling an Entire MST Region at Once or Exchanging One Region Configuration for Another" on page 6-72.)

N o t e   The switch automatically senses port identity and type, and automatically defines spanning-tree parameters for each type, as well as parameters that apply across the switch. Although these parameters can be adjusted, *HP strongly recommends leaving these settings in their default configurations unless the proposed changes have been supplied by an experienced network administrator who has a strong understanding of the IEEE 802.1D/w/s standards and operation.*

MST Regions

All MSTP switches in a given region must be configured with the same VLANs. Also, each MSTP switch within the same region must have the same VLAN-to-instance assignments. (A VLAN can belong to only one instance within any region.) Within a region:

■   All of the VLANs belonging to a given instance compose a single, active spanning-tree topology for that instance.

■   Each instance operates independently of other regions.

Between regions there is a single, active spanning-tree topology.

How Separate Instances Affect MSTP Operation. Assigning different groups of VLANs to different instances ensures that those VLAN groups use independent forwarding paths. For example, in figure 6-26 each instance has a different forwarding path.
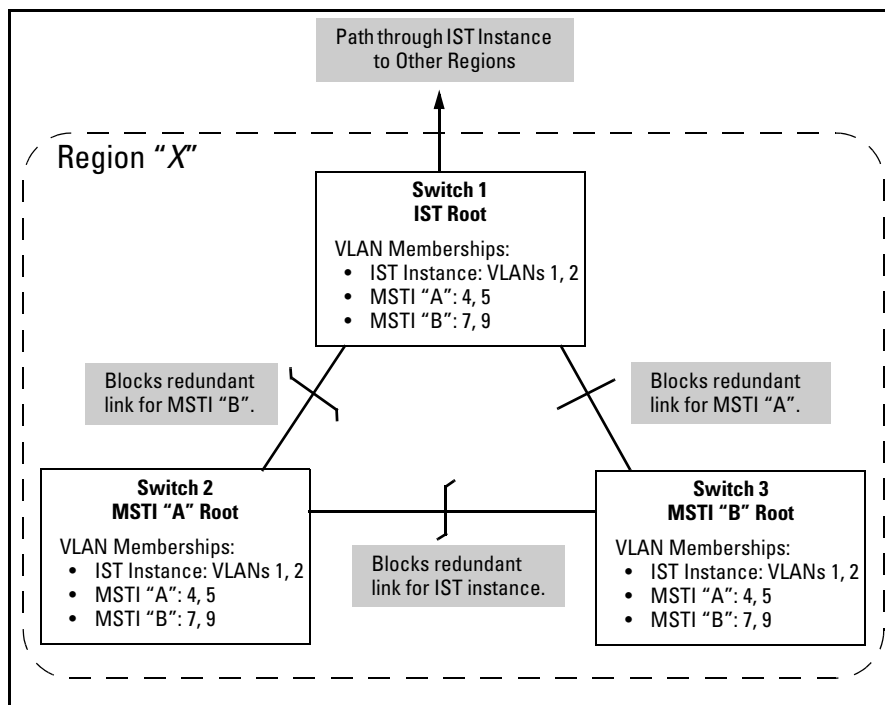
Figure 6-26. Active Topologies Built by Three Independent MST Instances

While allowing only one active path through a given instance, MSTP retains any redundant physical paths in the instance to serve as backups (blocked) paths in case the existing active path fails. Thus, if an active path in an instance fails, MSTP automatically activates (unblocks) an available backup to serve as the new active path through the instance for as long as the original active path is down. Note also that a given port may simultaneously operate in different states (forwarding or blocking) for different spanning-tree instances within the same region. This depends on the VLAN memberships to which the port is assigned. For example, if a port belongs to VLAN 1 in the IST instance of a region and also belongs to VLAN 4 in MSTI "x" in the same region, the port may apply different states to traffic for these two different instances.

Within a region, traffic routed between VLANs in separate instances can take only one physical path. To ensure that traffic in all VLANs within a region can travel between regions, all of the boundary ports for each region should belong to all VLANs configured in the region. Otherwise, traffic from some areas within a region could be blocked from moving to other regions.

All MSTP switches (as well as STP and RSTP switches) in a network use BPDUs (Bridge Protocol Data Units) to exchange information from which to build multiple, active topologies in the individual instances within a region and between regions. From this information:

- The MSTP switches in each LAN segment determine a designated bridge and designated port or trunk for the segment.

- The MSTP switches belonging to a particular instance determine the root bridge and root port or trunk for the instance.

- For the IST instance within a region, the MSTP switches linking that region to other regions (or to STP or RSTP switches) determine the IST root bridge and IST root port or trunk for the region. (For any Multiple Spanning-Tree instance—MSTI—in a region, the regional root may be a different switch that is not necessarily connected to another region.)

- The MSTP switches block redundant links within each LAN segment, across all instances, and between regions, to prevent any traffic loops.

As a result, each individual instance (spanning tree) within a region determines its regional root bridge, designated bridges, and designated ports or trunks.

Regions, Legacy STP and RSTP Switches, and the Common Spanning Tree (CST)

The IST instance and any MST instances in a region exist only within that region. Where a link crosses a boundary between regions (or between a region and a legacy STP or RSTP switch), traffic is forwarded or blocked as determined by the Common Spanning Tree (CST). The CST ensures that there is only one active path between any two regions, or between a region and a switch running STP and RSTP. (Refer to figure 6-25 on page 6-47.)

N o t e   The Series 4200vl switches support MSTP only.

MSTP Operation with 802.1Q VLANs

As indicated in the preceding sections, within a given MST instance, a single spanning tree is configured for all VLANs included in that instance. This means that if redundant physical links exist in separate VLANs within the same instance, MSTP blocks all but one of those links. However, you can prevent the bandwidth loss caused by blocked redundant links for different VLANs in an instance by using a port trunk. The following example shows how you can use a port trunk with 802.1Q (tagged) VLANs and MSTP without unnecessarily blocking any links or losing any bandwidth.
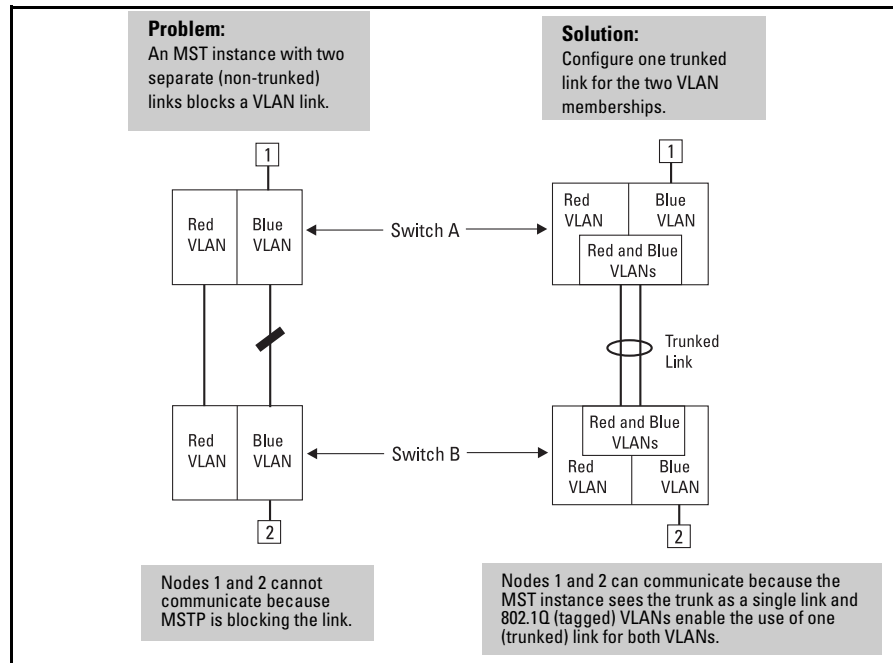
Figure 6-27. Example of Using a Trunked Link To Support Multiple VLAN Connectivity within the Same MST Instance

N o t e   All switches in a region should be configured with the VLANs used in that region, and all ports linking MSTP switches together should be members of all VLANs in the region. Otherwise, the path to the root for a given VLAN will be broken if MSTP selects a spanning tree through a link that does not include that VLAN.

Terminology

Bridge: See "MSTP Bridge".

Common and Internal Spanning Tree (CIST): Comprises all LANs, STP, and RSTP bridges and MSTP regions in a network. The CIST automatically determines the MST regions in a network and defines the root bridge (switch) and designated port for each region. The CIST includes the Common Spanning Tree (CST), the Internal Spanning Tree (IST) within each region, and any multiple spanning-tree instances (MSTIs) in a region.

Common Spanning Tree (CST): Refers to the single forwarding path the switch calculates for STP (802.1D) and RSTP (802.1w) topologies, and for inter-regional paths in MSTP (802.1s) topologies. Note that all three types of spanning tree can interoperate in the same network. Also, the MSTP switch interprets a device running 802.1D STP or 802.1w RSTP as a separate region. (Refer to figure 6-25 on page 6-47.)

Internal Spanning Tree (IST): Comprises all VLANs within a region that are not assigned to a multiple spanning-tree instance configured within the region. All MST switches in a region should belong to the IST. In a given region "X", the IST root switch is the regional root switch and provides information on region "X" to other regions.

MSTP (Multiple Spanning Tree Protocol): A network supporting MSTP allows multiple spanning tree instances within configured regions, and a single spanning tree among regions, STP bridges, and RSTP bridges.

MSTP BPDU (MSTP Bridge Protocol Data Unit): These BPDUs carry region-specific information, such as the region identifier (region name and revision number). If a switch receives an MSTP BPDU with a region identifier that differs from its own, then the port on which that BPDU was received is on the boundary of the region in which the switch resides.

MSTP Bridge: In this manual, an MSTP bridge is a 5300xl, 4200vl, 3400cl, or 6400cl switch (or another 802.1s-compatible device) configured for MSTP operation.

MST Region: An MST region forms a multiple spanning tree domain and is a component of a single spanning-tree domain within a network. For switches internal to the MST region:

- All switches have identical MST configuration identifiers (region name and revision number).
- All switches have identical VLAN assignments to the region's IST and (optional) MST instances.
- One switch functions as the designated bridge (IST root) for the region.
- No switch has a point-to-point connection to a bridging device that cannot process RSTP BPDUs.

Operating Rules

- All switches in a region must be configured with the same set of VLANs, as well as the same MST configuration name and MST configuration number.

- Within a region, a VLAN can be allocated to either a single MSTI or to the region's IST instance.

- All switches in a region must have the same VID-to-MST instance and VID-to-IST instance assignments.

- There is one root MST switch per configured MST instance.

- Within any region, the root switch for the IST instance is also the root switch for the region. Because boundary ports provide the VLAN connectivity between regions, all boundary ports on a region's root switch should be configured as members of all static VLANs defined in the region.

- There is one root switch for the Common and Internal Spanning Tree (CIST). Note that the per-port **hello-time** parameter assignments on the CIST root switch propagate to the ports on downstream switches in the network and override the **hello-time** configured on the downstream switch ports.

- Where multiple MST regions exist in a network, there is only one active, physical communication path between any two regions, or between an MST region and an STP or RSTP switch. MSTP blocks any other physical paths as long as the currently active path remains in service.

- Within a network, an MST region appears as a virtual RSTP bridge to other spanning tree entities (other MST regions, and any switches running 802.1D or 802.1w spanning-tree protocols).

- Within an MSTI, there is one spanning tree (one physical, communication path) between any two nodes. That is, within an MSTI, there is one instance of spanning tree, regardless of how many VLANs belong to the MSTI. Within an IST instance, there is also one spanning tree across all VLANs belonging to the IST instance.

- An MSTI comprises a unique set of VLANs and forms a single spanning-tree instance within the region to which it belongs.

- Communication between MST regions uses a single spanning tree.

- If a port on a switch configured for MSTP receives a legacy (STP/802.1D or RSTP/802.1w) BPDU, it automatically operates as a legacy port. In this case, the MSTP switch interoperates with the connected STP or RSTP switch as a separate MST region.

- Within an MST region, there is one logical forwarding topology per instance, and each instance comprises a unique set of VLANs. Where multiple paths exist between a pair of nodes using VLANs belonging to the same instance, all but one of those paths will be blocked for that instance. However, if there are different paths in different instances, all such paths are available for traffic. Separate forwarding paths exist through separate spanning tree instances.

- A port can have different states (forwarding or blocking) for different instances (which represent different forwarding paths).
- MSTP interprets a switch mesh as a single link.
- A dynamic VLAN learned by GVRP will always be placed in the IST instance and cannot be moved to any configured MST instance.

Transitioning from STP or RSTP to MSTP

N o t e   STP and RSTP are available on the Series 5400xl, 3400cl and 6400cl switches.

IEEE 802.1s MSTP includes RSTP functionality and is designed to be compatible with both IEEE 802.1D and 802.1w spanning-tree protocols. Even if all the other devices in your network are using STP, you can enable MSTP on the switches covered by this guide. Also, using the default configuration values, your 5300xl, 4200vl, and 3400cl/6400cl switches will interoperate effectively with STP and RSTP devices. MSTP automatically detects when the switch ports are connected to non-MSTP devices in the spanning tree and communicates with those devices using 802.1D or 802.1w STP BPDU packets, as appropriate.

Because MSTP is so efficient at establishing the network path, HP highly recommends that you update all of your 5300xl switches to support 802.1s/ MSTP. (All 3400cl/6400cl switch software versions support 802.1s. Also, for switches that do not support 802.1s/MSTP, HP recommends that you update to RSTP to benefit from the convergence times of less than one second under optimal circumstances.) To make the best use of MSTP and achieve the fastest possible convergence times, there are some changes that you should make to the MSTP default configuration.

N o t e   Under some circumstances, it is possible for the rapid state transitions employed by MSTP and RSTP to result in an increase in the rates of frame duplication and misordering in the switched LAN. In order to allow MSTP and RSTP switches to support applications and protocols that may be sensitive to frame duplication and misordering, setting the Force Protocol Version parameter to **STP-compatible** allows MSTP and RSTP to operate with the rapid transitions disabled. The value of this parameter applies to all ports on the switch. See information on **force version** on page 6-16.

As indicated above, one of the benefits of MSTP and RSTP is the implementation of a larger range of port path costs, which accommodates higher network speeds. New default values have also been implemented for the path costs associated with the different network speeds. This can create some

incompatibility between devices running the older 802.1D STP and your switch running MSTP or RSTP. Please see the "Note on Path Cost" on page 6-19 for more information on adjusting to this incompatibility.

Tips for Planning an MSTP Application

■ Ensure that the VLAN configuration in your network supports all of the forwarding paths necessary for the desired connectivity. All ports connecting one switch to another within a region and one switch to another between regions should be configured as members of all VLANs configured in the region.

■ All ports or trunks connecting one switch to another within a region should be configured as members of all VLANs in the region. Otherwise, some VLANs could be blocked from access to the spanning-tree root for an instance or for the region.

■ Plan individual regions based on VLAN groupings. That is, plan on all MSTP switches in a given region supporting the same set of VLANs. Within each region, determine the VLAN membership for each spanning-tree instance. (Each instance represents a single forwarding path for all VLANs in that instance.)

■ There is one logical spanning-tree path through the following:
   • Any inter-regional links
   • Any IST or MST instance within a region
   • Any legacy (802.1D or 802.1w) switch or group of switches. (Where multiple paths exist between an MST region and a legacy switch, expect the CST to block all but one such path.)

■ Determine the root bridge and root port for each instance.

■ Determine the designated bridge and designated port for each LAN segment.

■ Determine which VLANs to assign to each instance, and use port trunks with 802.1Q VLAN tagging where separate links for separate VLANs would result in a blocked link preventing communication between nodes on the same VLAN. (Refer to "MSTP Operation with 802.1Q VLANs" on page 6-51.)

■ Identify the edge ports connected to end nodes and enable the edge-port setting for these ports. Leave the edge-port setting disabled for ports connected to another switch, a bridge, or a hub.

Note on MSTP Rapid State Transitions Under some circumstances the rapid state transitions employed by MSTP (and RSTP) can increase the rates of frame duplication and misordering in the switched LAN. To allow MSTP switches to support applications and protocols that may be sensitive to frame duplication and misordering, setting the Force Protocol Version (**force-version**) parameter to **stp-compatible** allows MSTP to operate with rapid transitions disabled. The value of this parameter applies to all ports on the switch. See the information on **force-version** on page 6-16.

Steps for Configuring MSTP

This section outlines the general steps for configuring MSTP operation in your network, and assumes you have already planned and configured the VLANs you want MSTP to use. The actual MSTP parameter descriptions are in the following sections.

N o t e The switch supports MSTP configuration through the CLI. After you specify MSTP and reboot the switch as described above, the switch removes the **Spanning Tree** option from the Menu interface. If you later reconfigure the switch to use STP or RSTP, the switch returns the **Spanning Tree** option to the Menu interface.

This section assumes that you have already

1. Configured the MSTP operation mode. This specifies MSTP as the spanning tree operating mode. Changing the current MSTP operation mode requires you to save the change and reboot to activate the selection. (The **spanning-tree protocol-version** command is not available for the Series 4200vl switch as this step is not necessary.)

    spanning-tree protocol-version <stp | rstp | mstp>

2. Configure MSTP global parameters. This step involves configuring the following:

    • Required parameters for MST region identity:

        Region Name: **spanning-tree config-name**

        Region Revision Number: **spanning-tree config-revision**

- Optional MSTP parameter changes for region settings:

  HP recommends that you leave these parameters at their default settings for most networks. Refer to the "Caution" on page 6-48.

  – The maximum number of hops before the MSTP BPDU is discarded (default: 20)
     spanning-tree max-hops

  – Force-Version operation
     spanning-tree force-version

  – Forward Delay
     spanning-tree forward-delay

  – Hello Time (used if the switch operates as the root device.)
     spanning-tree hello-time

  – Maximum age to allow for STP packets before discarding
     spanning-tree maximum-age

  – Device spanning-tree priority. Specifies the priority value used along with the switch MAC address to determine which device is root. The lower a priority value, the higher the priority.
     spanning-tree priority

3. Configure MST instances.
   - Configure one instance for each VLAN group that you want to operate as an active topology within the region to which the switch belongs. When you create the instance, you must include a minimum of one VID. You can add more VIDs later if desired.
      spanning-tree instance

      To move a VLAN from one instance to another, first use **no spanning-tree instance < *n* > vlan < *vid* >** to unmap the VLAN from the current instance, then add the VLAN to the other instance. (While the VLAN is unmapped from an MSTI, it is associated with the region's IST instance.)

   - Configure the priority for each instance.
      spanning-tree instance

4. Configure MST instance port parameters. Enable **edge-port** for ports connected to end nodes (page 6-63), but leave it disabled (the default) for connections to another switch, a bridge, or a hub. Set the path cost value for the port(s) used by a specific MST instance. Leaving this setting at the

default auto allows the switch to calculate the path-cost from the link speed.

spanning-tree instance

5. Enable spanning-tree operation on the switch.

spanning-tree

Configuring MSTP Operation Mode and Global Parameters

Command	Page
spanning-tree protocol-version mstp*	6-60
spanning-tree config-name < *ascii-string* >	6-61
spanning-tree config-revision < *revision-number* >	6-61
spanning-tree max-hops < *hop-count* >	6-62
spanning-tree force-version < stp-compatible	rstp-operation	mstp-operation>	6-62
spanning-tree hello-time < 1..10 >	6-63
* Not present on the Series 4200vl switches	

The commands in this section apply on the switch level, and do not affect individual port configurations.

Syntax: spanning-tree protocol-version mstp

Note: This command is not present for the 4200vl switches.

Changes the current spanning-tree protocol on the switch to 802.1s Multiple Spanning Tree. Must be followed by **write mem** *and* **reboot** *to activate the change. After rebooting, the switch is ready to operate as an MSTP bridge. Note that this command does not enable spanning-tree operation. To activate the configured spanning-tree operation on the switch, execute* **spanning-tree**.

Note: When you activate spanning-tree operation or change the spanning-tree configuration while spanning tree is enabled, the switch must recalculate the network paths it uses. To minimize traffic delays while this convergence occurs, HP recommends that you not activate spanning tree operation until you have finished configuring all devices in your network. Refer to "Enabling an Entire MST Region at Once or Exchanging One Region Configuration for Another" on page 6-72.

N o t e   The following commands are available only when the switch is configured for MSTP protocol operation.

Syntax: [no] spanning-tree config-name < *ascii-string* >

>*This command resets the configuration name of the MST region in which the switch resides. This name can include up to 32 nonblank characters and is case-sensitive. On all switches within a given MST region, the configuration names must be identical. Thus, if you want more than one MSTP switch in the same MST region, you must configure the identical region name on all such switches. If you retain the default configuration name on a switch, it cannot exist in the same MST region with another switch. (Default Name: A text string using the hexadecimal representation of the switch's MAC address)*
>
>*The* **no** *form of the command overwrites the currently configured name with the default name.*
>
>>***Note:*** *This option is available only when the switch is configured for MSTP operation. Also, there is no defined limit on the number of regions you can configure.*

Syntax: spanning-tree config-revision < *revision-number* >

>*This command configures the revision number you designate for the MST region in which you want the switch to reside. This setting must be the same for all switches residing in the same region. Use this setting to differentiate between region configurations in situations such as the following:*
>
>- *Changing configuration settings within a region where you want to track the configuration versions you use*
>- *Creating a new region from a subset of switches in a current region and want to maintain the same region name.*
>- *Using the* **pending** *option to maintain two different configuration options for the same physical region.*
>
>*Note that this setting must be the same for all MSTP switches in the same MST region. (Range:* **0 - 65535***; Default:* **0***)*
>
>>***Note:*** *This option is available only when the switch is configured for MSTP operation.*

Syntax: spanning-tree max-hops < *hop-count* >

This command resets the number of hops allowed for BPDUs in an MST region. When an MSTP switch receives a BPDU, it decrements the hop-count setting the BPDU carries. If the hop-count reaches zero, the receiving switch drops the BPDU. Note that the switch does not change the message-age and maximum-age data carried in the BPDU as it moves through the MST region and is propagated to other regions. (Range: 1 - 40; Default: 20)

Syntax: spanning-tree force-version < stp-compatible | rstp-operation | mstp-operation >

Sets the spanning-tree compatibility mode. When the switch is configured with MSTP mode, this command forces the switch to emulate behavior of earlier versions of spanning tree protocol or return to MSTP behavior. The command is useful in test or debug applications, and removes the need to reconfigure the switch for temporary changes in spanning-tree operation.

stp-compatible: *The switch applies 802.1D STP operation on all ports.*

rstp-operation: *The switch applies 802.1w operation on all ports except those ports where it detects a system using 802.1D Spanning Tree.*

mstp-operation: *The switch applies 802.1s MSTP operation on all ports where compatibility with 802.1D or 802.1w spanning tree protocols is not required.*

This command is available when the protocol version is set to **mstp** *(see 'protocol-version' above).*

Note that even when mstp-operation is selected, if the switch detects an 802.1D BPDU or an 802.1w BPDU on a port, it communicates with the device linked to that port using STP or RSTP BPDU packets. Also, if errors are encountered as described in the "Note on MSTP Rapid State Transitions" on page 6-57, setting **force-version** *to* **stp-compatible** *forces the MSTP switch to communicate out all ports using operations that are compatible with IEEE 802.1D STP.*

Syntax: spanning-tree hello-time < 1..10 >

> *If MSTP is running and the switch is operating as the CIST root for your network, this command specifies the time in seconds between transmissions of BPDUs for all ports on the switch configured with the* **Global** *option (the default). This parameter applies in MSTP, RSTP and STP modes. During MSTP operation, you can override this global setting on a per-port basis with this command:* **spanning-tree < *port-list* > hello-time < 1..10 >** (page 6-63). (*Default:* **2**.)

Configuring Basic Port Connectivity Parameters

Command	Page
spanning-tree < *port-list* >	
edge-port	below
mcheck	below
hello-time < global	1..10 >	6-64
spanning-tree path-cost < auto	200000000 >	6-64
spanning-tree point-to-point-mac < force-true	force-false	auto>	6-65
spanning-tree priority <0-15>	6-65

The basic port connectivity parameters affect spanning-tree links at the global level. In most cases, HP recommends that you use the default settings for these parameters and apply changes on a per-port basis only where a nondefault setting is clearly indicated by the circumstances of individual links.

Syntax: [no] spanning-tree < *port-list* > < edge-port | mcheck >

> [edge-port]
>
> > *Enable* **edge-port** *on ports connected to end nodes. During spanning tree establishment, ports with* **edge-port** *enabled transition immediately to the forwarding state. Disable this feature on any switch port that is connected to another switch, bridge, or hub. (Default:* **No** *- disabled)*
> >
> > *The* **no spanning-tree < *port-list* > edge-port** *command disables edge-port operation on the specified ports.*

[mcheck]

Forces a port to send RSTP BPDUs for 3 seconds. This allows for another switch connected to the port and running RSTP to establish its connection quickly and for identifying switches running 802.1D STP. If the whole-switch force-version parameter is set to stp-compatible, the switch ignores the mcheck setting and sends 802.1D STP BPDUs out all ports. Disable this feature on all ports that are known to be connected to devices that are running 802.1D STP. (Default: **Yes** *- enabled)*

The **no spanning-tree <** *port-list* **> mcheck** *command disables mcheck.*

Syntax: spanning-tree < *port-list* > < hello-time | path-cost | point-to-point-mac | priority >

[hello-time < global | 1 - 10 >]

When the switch is the CIST root, this parameter specifies the interval (in seconds) between periodic BPDU transmissions by the designated ports. This interval also applies to all ports in all switches downstream from each port in the < port-list *>. A setting of* **global** *indicates that the ports in <* port-list*> on the CIST root are using the value set by the global spanning-tree* **hello-time** *value (page 6-63). When a given switch "X" is not the CIST root, the per-port* **hello-time** *for all active ports on switch "X" is propagated from the CIST root, and is the same as the* **hello-time** *in use on the CIST root port in the currently active path from switch "X" to the CIST root. (That is, when switch "X" is not the CIST root, then the upstream CIST root's port* **hello-time** *setting overrides the* **hello-time** *setting configured on switch "X". (Default Per-Port setting:* **Use Global***. Default Global Hello-Time:* **2***.)*

[path-cost < auto | 1..200000000 >]

Assigns an individual port cost that the switch uses to determine which ports are forwarding ports in a given spanning tree. In the default configuration (auto) the switch determines a port's path cost by the port's type:

– *10 Mbps:* **2000000**
– *100 Mbps:* **200000**
– *1 Gbps:* **20000**

Refer to "Note on Path Cost" on page 6-19 for information on compatibility with devices running 802.1D STP for the path cost values (Default: Auto.).

[point-to-point-mac < force-true | force-false | auto >]

> *This parameter informs the switch of the type of device to which a specific port connects.*
>
> **Force-True** *(****default****): Indicates a point-to-point link to a device such as a switch, bridge, or end-node.*
>
> **Force-False***: Indicates a connection to a hub (which is a shared LAN segment).*
>
> **Auto:** *Causes the switch to set Force-False on the port if it is not running at full duplex. (Connections to hubs are half-duplex.)*

[priority <priority-multiplier>]

> *MSTP uses this parameter to determine the port(s) to use for forwarding. The port with the lowest priority number has the highest priority. The range is 0 to 240. However, this command specifies the priority as a multiplier (0 - 15) of 16. That is, when you specify a priority multiplier of 0 - 15, the actual priority assigned to the switch is:*
>
> $$(priority\text{-}multiplier)\ x\ 16$$
>
> *For example, if you configure "**2**" as the priority multiplier on a given port, then the actual* **Priority** *setting is 32. Thus, after you specify the port priority multiplier, the switch displays the actual port priority (and not the multiplier) in the* **show spanning-tree** *or* **show spanning-tree < *port-list* >** *displays.*
>
> *You can view the actual multiplier setting for ports by executing* **show running** *and looking for an entry in this format:*
>
> > spanning-tree < *port-list* > priority < *priority-multiplier* >
>
> *For example, configuring port A2 with a priority multiplier of "3" results in this line in the* **show running** *output:*
>
> > ```
> > spanning-tree A2 priority 3
> > ```

Configuring MST Instance Parameters

Command	Page
[no] spanning-tree instance < 1..16 > vlan < *vid*> [*vid..vid*] no spanning-tree instance < 1..16 >	6-66
spanning-tree instance < 1..16 > priority <priority-multiplier >	6-67
spanning-tree priority <priority-multiplier>	6-68

Syntax: [no] spanning-tree instance < 1..16 > vlan < *vid* [*vid..vid*] >
no spanning-tree instance < priority-multiplier >

> *Configuring MSTP on the switch automatically configures the IST instance and places all statically configured VLANs on the switch into the IST instance. This command creates a new MST instance (MSTI) and moves the VLANs you specify from the IST to the MSTI. At least one VLAN must be mapped to a MSTI when you create it. (A VLAN cannot be mapped to more than one instance at a time.) You can create up to 16 MSTIs in a region. Use the no form of the command to remove a VLAN from an MSTI. (Removing a VLAN from an MSTI returns the VLAN to the IST instance, where it can either remain or be re-assigned to another MSTI configured in the region.)*
>
> *The* **no** *form of the command deletes the specified VLAN, or if no VLANs are specified, the* **no** *form of the command deletes the specified MSTI.*

Syntax: spanning-tree instance < 1..16 > priority < 0 .. 15 >

This command sets the switch (bridge) priority for the designated instance. This priority is compared with the priorities of other switches in the same instance to determine the root switch for the instance. The lower the priority value, the higher the priority. (If there is only one switch in the instance, then that switch is the root switch for the instance.) The root bridge in a given instance provides the path to connected instances in other regions that share one or more of the same VLAN(s). (Traffic in VLANs assigned to a numbered STP instance in a given region moves to other regions through the root switch for that instance.)

The priority range for an MSTP switch is 0-61440. However, this command specifies the priority as a multiplier (0 - 15) of 4096. That is, when you specify a priority multiplier value of 0 - 15, the actual priority assigned to the switch for the specified MST instance is:

$$(priority\text{-}multiplier) \text{ x } 4096$$

*For example, if you configure "**5**" as the priority-multiplier for MST Instance 1 on a given MSTP switch, then the **Switch Priority** setting is 20,480 for that instance in that switch.*

***Note:** If multiple switches in the same MST instance have the same priority setting, then the switch with the lowest MAC address becomes the root switch for that instance.*

Syntax: spanning-tree priority < priority-multiplier>

> *Every switch running an instance of MSTP has a Bridge Identifier, which is a unique identifier that helps distinguish this switch from all others. The switch with the lowest Bridge Identifier is elected as the root for the tree.*
>
> *The Bridge Identifier is composed of a configurable Priority component (2 bytes) and the bridge's MAC address (6 bytes). The ability to change the Priority component provides flexibility in determining which switch will be the root for the tree, regardless of its MAC address.*
>
> *This command sets the switch (bridge) priority for the designated region in which the switch resides. The switch compares this priority with the priorities of other switches in the same region to determine the root switch for the region. The lower the priority value, the higher the priority. (If there is only one switch in the region, then that switch is the root switch for the region.) The root switch in a region provides the path to connected regions for the traffic in VLANs assigned to the region's IST instance. (Traffic in VLANs assigned to a numbered STP instance in a given region moves to other regions through the root switch for that instance.)*
>
> *The priority range for an MSTP switch is 0-61440. However, this command specifies the priority as a multiplier (0 - 15) of 4096. That is, when you specify a priority multiplier value of 0 - 15, the actual priority assigned to the switch is:*
>
> $$(priority\text{-}multiplier) \text{ x } 4096$$
>
> *For example, if you configure "**2**" as the priority-multiplier on a given MSTP switch, then the **Switch Priority** setting is 8,192 (2 x 4,092).*
>
> ***Note:*** *If multiple switches in the same MST region have the same priority setting, then the switch with the lowest MAC address becomes the root switch for that region.*

Configuring MST Instance Per-Port Parameters

Command	Page
spanning-tree instance < 1..16 > < port-list > path-cost < auto | 1..200000000 >	6-69
spanning-tree instance < 1..16 > < *port-list* > priority < *priority-multiplier* >	6-70
spanning-tree < *port-list* > priority < *priority-multiplier* >	6-71

Syntax: spanning-tree instance < 1..16 >< *port-list* > path-cost < auto \| 1..200000000 >

> *This command assigns an individual port cost for the specified MST instance. (For a given port, the path cost setting can be different for different MST instances to which the port may belong.) The switch uses the path cost to determine which ports are the forwarding ports in the instance; that is which links to use for the active topology of the instance and which ports to block. The settings are either* **auto** *or in a range from 1 to 200,000,000. With the* **auto** *setting, the switch calculates the path cost from the link speed:*
>
> *10 Mbps — 2000000*
> *100 Mbps — 200000*
> *1 Gbps — 20000*
> *(Default:* **Auto***)*

Syntax: spanning-tree instance < 1..16 >< *port-list* > priority <*priority-multiplier*>

This command sets the priority for the specified port(s) in the specified MST instance. (For a given port, the priority setting can be different for different MST instances to which the port may belong.)

The priority range for a port in a given MST instance is 0-255. However, this command specifies the priority as a multiplier (0 - 15) of 16. That is, when you specify a priority multiplier of 0 - 15, the actual priority assigned to the switch is:

$$(priority\text{-}multiplier) \; x \; 16$$

*For example, if you configure "**2**" as the priority multiplier on a given port in an MST instance, then the actual* **Priority** *setting is 32. Thus, after you specify the port priority multiplier in an instance, the switch displays the actual port priority (and not the multiplier) in the* **show spanning-tree instance < 1..16 >** *or* **show spanning-tree < *port-list* > instance < 1..16 >** *displays.*

You can view the actual multiplier setting for ports in the specified instance by executing **show running** *and looking for an entry in this format:*

spanning-tree instance < 1..15 > < *port-list* > priority < *priority-multiplier* >

For example, configuring port A2 with a priority multiplier of "3" in instance 1, results in this line in the **show running** *output:*

```
spanning-tree instance 1 A2 priority 3
```

Syntax: spanning-tree< *port-list* > priority < *priority-multiplier* >

> *This command sets the priority for the specified port(s) for the IST (that is, Instance 0) of the region in which the switch resides. The "priority" component of the port's "Port Identifier" is set. The Port Identifier is a unique identifier that helps distinguish this switch's ports from all others. It consists of the Priority value with the port number extension— PRIORITY:PORT_NUMBER. A port with a lower value of Port Identifier is more likely to be included in the active topology.*
>
> *This priority is compared with the priorities of other ports in the IST to determine which port is the root port for the IST instance. The lower the priority value, the higher the priority. The IST root port (or trunk) in a region provides the path to connected regions for the traffic in VLANs assigned to the region's IST instance.*
>
> *The priority range for a port in a given MST instance is 0-240. However, this command specifies the priority as a multiplier (0 - 15) of 16. That is, when you specify a priority multiplier of 0 - 15, the actual priority assigned to the switch is:*

$$(\text{priority-multiplier}) \times 16$$

> *For example, configuring "***5***" as the priority multiplier on a given port in the IST instance for a region creates an actual* **Priority** *setting of* **80**. *Thus, after you specify the port priority multiplier for the IST instance, the switch displays the actual port priority (and not the multiplier) in the* **show spanning-tree instance ist** *or* **show spanning-tree < *port-list* > instance ist** *displays. You can view the actual multiplier setting for ports in the IST instance by executing* **show running** *and looking for an entry in this format:*

    spanning-tree < *port-list* > priority < *priority-multiplier* >

> *For example, configuring port A2 with a priority multiplier of "2" in the IST instance, results in this line in the* **show running** *output:*

    spanning-tree A2 priority 2

Enabling or Disabling Spanning Tree Operation

This command enables or disables spanning tree operation for any spanning tree protocol enabled on the switch. Before using this command to enable spanning tree, ensure that the version you want to use is active on the switch.

Syntax: [no] spanning-tree

> *Enabling spanning tree with MSTP configured implements MSTP for all physical ports on the switch, according to the VLAN groupings for the IST instance and any other configured instances. Disabling MSTP removes protection against redundant loops that can significantly slow or halt a network.*
>
> *This command simply turns spanning tree on or off. It does not change the existing spanning tree configuration.*

Enabling an Entire MST Region at Once or Exchanging One Region Configuration for Another

Command	Page
spanning-tree pending < apply	config-name	config-revision	instance	reset >	6-73

This operation exchanges the currently active MSTP configuration with the currently pending MSTP configuration. It enables you to implement a new MSTP configuration with minimal network disruption or to exchange MSTP configurations for testing or troubleshooting purposes.

When you configure or reconfigure MSTP, the switch re-calculates the corresponding network paths. This can have a ripple effect throughout your network as adjacent MSTP switches recalculate network paths to support the configuration changes invoked in a single switch. Although MSTP employs RSTP operation, the convergence time for implementing MSTP changes can be disruptive to your network. However, by using the spanning-tree **pending** feature, you can set up an MSTP on the switch and then invoke all instances of the new configuration at the same time, instead of one at a time.

To Create a Pending MSTP Configuration. This procedure creates a pending MSTP configuration and exchanges it with the active MSTP configuration.

1. Configure the VLANs you want included in any instances in the new region. When you create the pending region, all VLANs configured on the switch will be assigned to the pending IST instance unless assigned to other, pending MST instances.

2. Configure MSTP as the spanning-tree protocol, then execute **write mem** and reboot. (The pending option is available only with MSTP enabled.)

3. Configure the pending region name to assign to the switch.

4. Configure the pending **config-revision** number for the region name.

5. If you want an MST instance other than the IST instance, configure the instance number and assign the appropriate VLANs (VIDs). (The **pending** command creates the region's IST instance automatically.)

6. Repeat step 5 for each additional MST instance you want to configure.

7. Use the **show spanning-tree pending** command to review your pending configuration (page 6-80).

8. Use the **spanning-tree pending apply** command to exchange the currently active MSTP configuration with the pending MSTP configuration.

Syntax: spanning-tree pending < apply | *config-name* | *config-revision* | instance | reset >

    apply

      Exchanges the currently active MSTP configuration with the pending MSTP configuration.

    config-name

      Specifies the pending MST region name. Must be the same for all MSTP switches in the region. (Default: The switch's MAC address.)

    config-revision

      *Specifies the pending MST region configuration revision number. Must be the same for all MSTP switches in the region. (Default: **0**).*

    instance < 1..16 > vlan [< *vid* | *vid-range* >

      Creates the pending instance and assigns one or more VLANs to the instance.

    reset

      Copies the switch's currently active MSTP configuration to the pending configuration. This is useful when you want to experiment with the current MSTP configuration while maintaining an unchanged version.

9. To view the current pending MSTP configuration, use the **show spanning-tree pending** command (page 6-80).

Displaying MSTP Statistics and Configuration

Command	Page
MSTP Statistics:	
show spanning-tree [< *port-list* >]	below
show spanning-tree instance < ist | 1..16 >	6-76
MSTP Configuration	
show spanning-tree [*port-list*] config	6-77
show spanning-tree [port-list] config instance < ist | 1..16 >	6-78
show spanning-tree mst-config	6-79
show spanning-tree pending< < instance | ist > | mst-config >	6-80

Displaying MSTP Statistics

Displaying Switch Statistics for the Common Spanning Tree. This command displays the MSTP statistics for the connections between MST regions in a network.

Syntax: show spanning-tree

> *This command displays the switch's global and regional spanning-tree status, plus the per-port spanning-tree operation at the regional level. Note that values for the following parameters appear only for ports connected to active devices:* **Designated Bridge**, **Hello Time**, **PtP**, *and* **Edge**.

Syntax: show spanning-tree < *port-list* >

> *This command displays the spanning-tree status for the designated port(s). You can list data for a series of ports and port trunks by specifying the first and last port or trunk of any consecutive series of ports and trunks. For example, to display data for port A20-A24 and trk1, you would use this command:* **show spanning-tree a20-a24,trk1**

```
Switch-1(config)# show spanning-tree
 Multiple Spanning Tree (MST) Information

  STP Enabled   : Yes
  Force Version : MSTP-operation
  IST Mapped VLANs : 1,66

  Switch MAC Address : 0004ea-5e2000
  Switch Priority    : 32768
  Max Age  : 20
  Max Hops : 20
  Forward Delay : 15

  Topology Change Count  : 0
  Time Since Last Change : 2 hours

  CST Root MAC Address : 00022d-47367f
  CST Root Priority    : 0
  CST Root Path Cost   : 4000000
  CST Root Port        : A1

  IST Regional Root MAC Address : 000883-028300
  IST Regional Root Priority    : 32768
  IST Regional Root Path Cost   : 200000
  IST Remaining Hops            : 19

                     |             Prio                   | Designated     Hello
   Port  Type        | Cost        rity  State            | Bridge         Time   PtP Edge
   ----  ---------   + ---------   ----- ----------        + -------------  -----  --- ----
   A1    10/100TX    | Auto        128   Forwarding        | 000883-028300  9      Yes No
   A2    10/100TX    | Auto        128   Blocking          | 0001e7-948300  9      Yes No
   A3    10/100TX    | Auto        128   Forwarding        | 000883-02a700  2      Yes No
   A4    10/100TX    | Auto        128   Disabled          |
   A5    10/100TX    | Auto        128   Disabled          |
    .      .         |   .          .      .               |
    .      .         |   .          .      .               |
    .      .         |   .          .      .               |
```

Switch's Spanning Tree Configuration and Identity of VLANs Configured in the Switch for the IST Instance

Identifies the overall spanning-tree root for the network.

Lists the switch's MSTP root data for connectivity with other regions and STP or RSTP devices.

Identifies the spanning-tree root for the IST Instance for the region.

Internal Spanning Tree Data (IST Instance) for the region in which the Switch Operates

Yes means the switch is operating the port as if it is connected to switch, bridge, or end node (but *not* a hub).

For **Edge**, **No** (edge-port operation disabled) indicates the port is configured for connecting to a LAN segment that includes a bridge or switch. **Yes** indicates the port is configured for a host (end node) link. Refer to the **edge-port** description under "Configuring Basic Port Connectivity Parameters" on page 6-63.

Figure 6-28. Example of Common Spanning Tree Status on an MSTP Switch

Displaying Switch Statistics for a Specific MST Instance.

Syntax: show spanning-tree instance < ist | 1..16 >

> *This command displays the MSTP statistics for either the IST instance or a numbered MST instance running on the switch.*

```
Switch-1(config)# show spanning-tree instance 1

 MST Instance Information

  Instance ID : 1
  Mapped VLANs : 11,22

  Switch Priority          : 32768

  Topology Change Count   : 4
  Time Since Last Change  : 6 secs

  Regional Root MAC Address : 0001e7-948300
  Regional Root Priority    : 32768
  Regional Root Path Cost   : 400000
  Regional Root Port        : A1
  Remaining Hops            : 18
                                                       Designated
  Port Type      Cost       Priority Role       State       Bridge
  ---- --------- ---------- -------- ---------- ----------- --------------
  A1   10/100TX  200000     128      Root       Forwarding  000883-028300
  A2   10/100TX  200000     128      Designated Forwarding  000883-02a700
  A3   10/100TX  200000     112      Designated Forwarding  000883-02a700
  A4   10/100TX  Auto       128      Disabled   Disabled
  .       .         .          .          .          .
  .       .         .          .          .          .
  .       .         .          .          .          .
```

Figure 6-29. Example of MSTP Statistics for a Specific Instance on an MSTP Switch

Displaying the MSTP Configuration

Displaying the Global MSTP Configuration. This command displays the switch's basic and MST region spanning-tree configuration, including basic port connectivity settings.

Syntax: show spanning-tree config

The upper part of this output shows the switch's global spanning-tree configuration that applies to the MST region. The port listing shows the spanning-tree port parameter settings for the spanning-tree region operation (configured by the **spanning-tree < port-list >** *command). For information on these parameters, refer to "Configuring Basic Port Connectivity Parameters" on page 6-63.*

Syntax: show spanning-tree < *port-list* > config

This command shows the same data as the above command, but lists the spanning-tree port parameter settings for only the specified port(s) and/or trunk(s). You can list data for a series of ports and port trunks by specifying the first and last port or trunk of any consecutive series of ports and trunks. For example, to display data for port A20-A24 and trk1, use this command: **show spanning-tree a20-a24,trk1 config**

```
Switch-2(config)# show spanning-tree config        Global Priority    Global Hello Time

 Multiple Spanning Tree (MST) Configuration Information

  STP Enabled [No] : Yes
  Force Version [MSTP-operation] : MSTP-operation
                                                                        Per-Port Hello Time
  MST Configuration Name : REGION_1                                     (Overrides Global Hello-
  MST Configuration Revision : 1          Switch Priority : 32768       Time on individual ports.)
  Forward Delay [15] : 15                 Hello Time [2] : 2
  Max Age [20] : 20                       Max Hops [20] : 20

  Port  Type      | Cost       Priority Edge Point-to-Point MCheck Hello Time
  ----  --------- + --------   -------- ---- -------------- ------ ----------
  A3    10/100TX  | Auto       128      Yes  Force-True     Yes    Use Global
  A4    10/100TX  | Auto       128      Yes  Force-True     Yes    Use Global
  :     :           Per-Port Priority   :    :              :      :
  :     :                    :          :    :              :      :
  A20   10/100TX  | Auto       128      Yes  Force-True     Yes    Use Global
  A21   10/100TX  | Auto       128      Yes  Force-True     Yes    Use Global
  A22   10/100TX  | Auto       128      Yes  Force-True     Yes    Use Global
  A23   10/100TX  | Auto       128      Yes  Force-True     Yes    Use Global
  A24   10/100TX  | Auto       128      Yes  Force-True     Yes    Use Global
  Trk1            | Auto       128      Yes  Force-True     Yes    Use Global
```

Figure 6-30. Example of Displaying the Switch's Global Spanning-Tree Configuration

Displaying Per-Instance MSTP Configurations. These commands displays the per-instance port configuration and current state, along with instance identifiers and regional root data.

Syntax: show spanning-tree config instance < ist | 1..16 >

> *The upper part of this output shows the instance data for the specified instance. The lower part of the output lists the spanning-tree port settings for the specified instance.*

Syntax: show spanning-tree < *port-list* > config instance < ist | 1..16 >

> *This command shows the same data as the above command, but lists the spanning-tree port parameter settings for only the specified port(s) and/or trunk(s). You can list data for a series of ports and port trunks by specifying the first and last port or trunk of any consecutive series of ports and trunks. For example, to display data for port A20-A24 and trk1, use this command:*

show spanning-tree a20-a24,trk1 config instance 1

```
Switch-2(config)# show spanning-tree config instance 1

 MST Instance Configuration Information

 Instance ID : 1                          ◄───   Instance-Specific Data
 Switch Priority : 32768
 Mapped VLANs : 11,22

 Port Type        | Cost        Priority
 ----  ---------  + ---------  --------
 A3    10/100TX   | Auto        128
 A4    10/100TX   | Auto        128       ◄───   Port Settings for the
 A5    10/100TX   | Auto        128              specified instance.
  .      .              .           .
  .      .              .           .
  .      .              .           .
 A23   10/100TX   | Auto        128
 A24   10/100TX   | Auto        128
 Trk1             | 100000      128
```

Figure 6-31. Example of the Configuration Listing for a Specific Instance

Displaying the Region-Level Configuration in Brief. This command output is useful for quickly verifying the allocation of VLANs in the switch's MSTP configuration and for viewing the configured region identifiers.

Syntax: show spanning-tree mst-config

This command displays the switch's regional configuration.

*Note: The switch computes the **MSTP Configuration Digest** from the VID to MSTI configuration mappings on the switch itself. As required by the 802.1s standard, all MSTP switches within the same region must have the same VID to MSTI assignments, and any given VID can be assigned to either the IST or one of the MSTIs within the region. Thus, the MSTP Configuration Digest must be identical for all MSTP switches intended to belong to the same region. When comparing two MSTP switches, if their Digest identifiers do not match, then they cannot be members of the same region.*

```
Switch-2(config)# show spanning-tree mst-config

 MST Configuration Identifier Information

  MST Configuration Name : REGION_1
  MST Configuration Revision : 1
  MST Configuration Digest : 0xDAD6A13EC5141980B7EBDA71D8991E7C

  IST Mapped VLANs : 1,66                        Refer to the "Note", above.

  Instance ID Mapped VLANs
  ---------- ------------------------------------------------------------
  1           11,22
  2           33,44,55
```

Figure 6-32. Example of a Region-Level Configuration Display

Displaying the Pending MSTP Configuration. This command displays the MSTP configuration the switch will implement if you execute the spanning-tree pending apply command (Refer to "Enabling an Entire MST Region at Once or Exchanging One Region Configuration for Another" on page 6-72.)

Syntax: show spanning-tree pending < instance | mst-config >

instance < 1..16 | ist >

Lists region, instance I.D. and VLAN information for the specified, pending instance.

mst-config

Lists region, IST instance VLAN(s), numbered instances, and assigned VLAN information for the pending MSTP configuration.

```
ProCurve # show spanning-tree pending instance 1

 Pending MST Instance Configuration Information

  MST Configuration Name : New-Version_01
  MST Configuration Revision : 10
  Instance ID : 1
  Mapped VLANs : 1,22




Switch-1(config)# show spanning-tree pending mst-config

 Pending MST Configuration Identifier Information

  MST Configuration Name : New-Version_01
  MST Configuration Revision : 10

  IST Mapped VLANs : 11,33

  Instance ID Mapped VLANs
  ----------- --------------------------------------------
  1            1,22
```

Figure 6-33. Example of Displaying a Pending Configuration

Operating Notes

SNMP MIB Support for MSTP. MSTP is a superset of the STP/802.1D and RSTP/802.1w protocols and uses the MIB objects defined for these two protocols.

Troubleshooting

Duplicate packets on a VLAN, or packets not arriving on a LAN at all. The allocation of VLANs to MSTIs may not be identical among all switches in a region.

A Switch Intended To Operate Within a Region Does Not Receive Traffic from Other Switches in the Region. An MSTP switch intended for a particular region may not have the same configuration name or region revision number as the other switches intended for the same region. The MSTP Configuration Name and MSTP Configuration Revision number must be identical on all MSTP switches intended for the same region. Another possibility is that the set of VLANs configured on the switch may not match the set of VLANs configured on other switches in the intended region.

—This page left blank intentionally—

7

Switch Meshing

Contents

Introduction

Switch meshing is not available on the Series 4200vl switches.

Switch meshing is a load-balancing technology that enhances reliability and performance in these ways:

■   Provides significantly better bandwidth utilization than either Spanning Tree Protocol (STP) or standard port trunking.

■   Uses redundant links that remain open to carry traffic, removing any single point of failure for disabling the network, and allowing quick responses to individual link failures. This also helps to maximize investments in ports and cabling.

■   Unlike trunked ports, the ports in a switch mesh can be of different types and speeds (10 and 100 Mbps, gigabit, and 10 gigabit). For example, a 10Base-FL port and a 1GB port can be included in the same switch mesh.

The mesh-configured ports in switches 1-4 form a Switch Mesh Domain
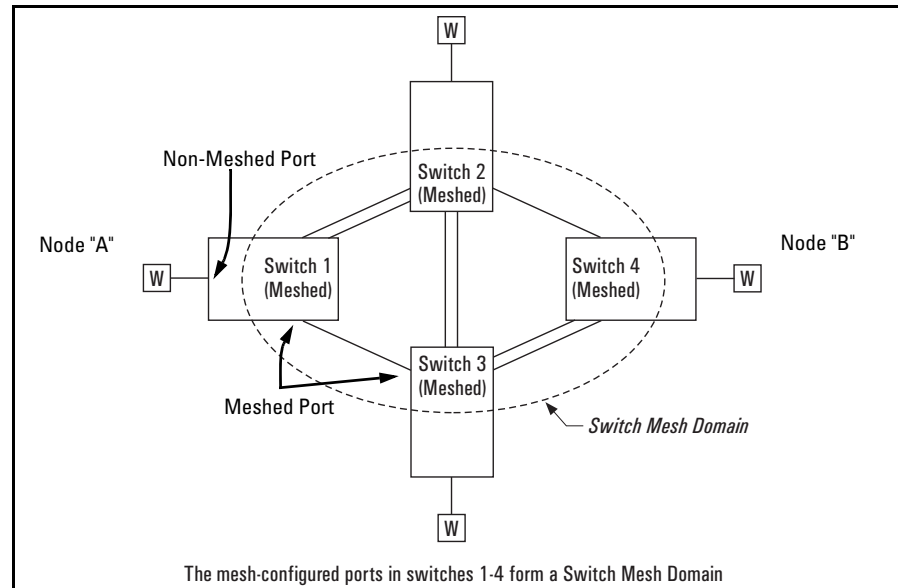
Figure 7-1. Example of Switch Meshing

Finding the Fastest Path.   Using multiple switches redundantly linked together to form a *meshed switch domain*, switch meshing dynamically distributes traffic across load-balanced switch paths by seeking the fastest paths for new traffic between nodes. In actual operation, the switch mesh periodically determines the best (lowest latency) paths, then assigns these paths as the need arises. The path assignment remains until the related MAC address entry times out. The mesh sees later traffic between the same nodes as new traffic, and may assign a different path, depending on conditions at the time. For example, at one time the best path from node A to node B is through switch 2. However, if traffic between node A and node B ceases long enough for the path assignment to age out, then the next time node A has traffic for node B, the assigned path between these nodes may be through switch 3 if network conditions have changed significantly.

N o t e   The **mac-age-time** parameter determines how long an inactive path assignment remains in memory. Refer to "System Information" in the chapter titled "Interface Access, System Information, and Friendly Port Names" in the *Management and Configuration Guide* for your switch.

Because Redundant Paths Are Active, Meshing Adjusts Quickly to Link Failures.   If a link in the mesh fails, the fast convergence time designed into meshing typically has an alternate route selected in less than a second for traffic that was destined for the failed link.

Meshing Allows Scalable Responses to Increasing Bandwidth Demand.   As more bandwidth is needed in a LAN backbone, another switch and another set of links can be added. This means that bandwidth is not limited by the number of trunk ports allowed in a single switch.

Meshing Features

Feature	Default	Menu	CLI	Web
viewing a mesh configuration	n/a	7-11	7-14	n/a
Configuring a Switch Mesh	n/a	7-11	7-17	n/a
Backwards Compatibility Mode	Disabled	n/a	7-17	n/a

Switch Meshing Fundamentals

Terminology

Switch Mesh Domain. This is a group of meshed switch ports exchanging meshing protocol packets. Paths between these ports can have multiple redundant links without creating broadcast storms.
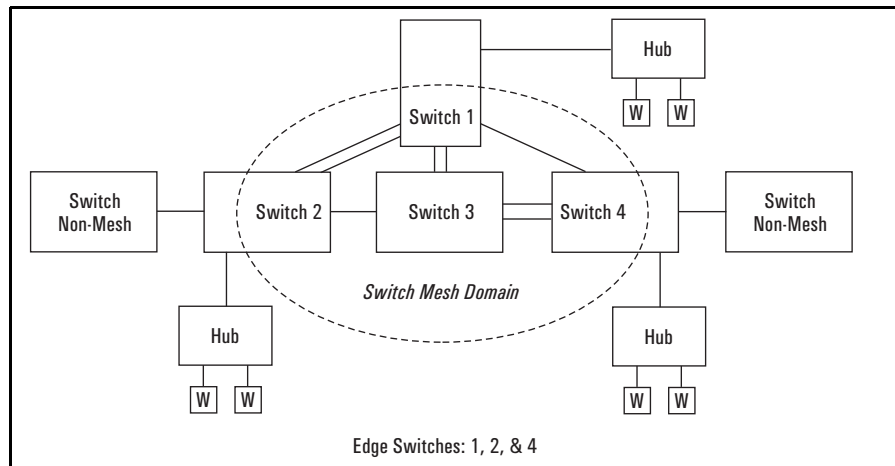
Figure 7-2. Example of a Switch Mesh Domain in a Network

Edge Switch. This is a switch that has some ports in the switch meshing domain and some ports outside of the domain. (See figure 7-2, above.)

Operating Rules

(See also "Mesh Design Optimization" on page 7-24.)

- A meshed switch can have some ports in the meshed domain and other ports outside the meshed domain. That is, ports within the meshed domain must be configured for meshing, while ports outside the meshed domain must not be configured for meshing.
- Meshed links must be point-to-point switch links.
- On any switch, all meshed ports belong to the same mesh domain.
- A switch can have up to 24 meshed ports.
- A mesh domain can include up to 12 switches.
- *On 3400cl and 6400cl switches only,* you must disable Stack Management (stacking) on the switch (**no stack**) before enabling meshing on any switch port. (In the default configuration, stacking is enabled on the 3400cl and 6400cl switches. Stacking is not available on the 5300xl switches.)
- Up to five interswitch, meshed hops are allowed in the path connecting two nodes through a switch mesh domain. A path of six or more meshed hops between two nodes is unusable. However, in most mesh topologies, there would normally be a shorter path available, and paths of five hops or fewer through the same mesh will continue to operate.
- Hub links between meshed switch links are not allowed.
- If the switch has multiple static VLANs and you configure a port for meshing, the port becomes a tagged member of all such VLANs . If you remove a port from meshing, it becomes an untagged member of only the default VLAN.
- A port configured as a member of a *static* trunk (LACP or Trunk) cannot also be configured for meshing.
- If a port belongs to a *dynamic* LACP trunk and you impose meshing on the port, it automatically ceases to be a member of the dynamic trunk.
- Meshing is not supported on ports with 802.1X port access security.
- On a port configured for meshing, if you subsequently remove meshing from the port's configuration and reboot the switch, the port returns to its default configuration. (It *does not* revert to any non-default configuration it had before being configured for meshing).
- In a given mesh domain, switches in the same product family must run the same switch software version. For example, if you update the software version on one Series 5300xl switch, then you must update the software version on any other Series 5300xl in the mesh. HP recommends that you always use the most recent software version available for the switches in your network.

■ If meshing is configured on the switch, the routing features (IP routing, RIP, and OSPF) must be disabled. *That is, the switch's meshing and routing features cannot be enabled at the same time.*

■ The spanning-tree configuration must be the same for all switches in the mesh (enabled or disabled). If spanning tree is enabled in the mesh, it must be the same version on all switches in the mesh: 802.1D, 802.1w, or 802.1s. If there are any 1600M/2400M/2424M/4000M/8000M switches in the mesh, then only 802.1D STP can be used.

■ If a switch in the mesh has GVRP enabled, then all switches in the mesh must have GVRP enabled. Otherwise, traffic on a dynamic VLAN may not pass through the mesh. Note that the 1600M/2400M/2424M/4000M/8000M switches do not offer GVRP. Thus, if you are using any of these switches in the same mesh domain with Series 5300xl, 3400cl, or 6400cl switches, then GVRP must be disabled on all switches in the mesh.

■ If a switch in the mesh has a particular static vlan configured, then all switches in the mesh must have that static vlan configured.

■ If a switch in the mesh has IGMP enabled, then all switches in the mesh must have IGMP enabled.

■ If a switch in the mesh has LLDP enabled, then all switches in the mesh must have LLDP enabled.

■ After adding or removing a port from the mesh, you must save the current configuration and reboot the switch in order for the change to take effect.

■ Multiple meshed domains require separation by either a non-meshed switch or a non-meshed link. For example:
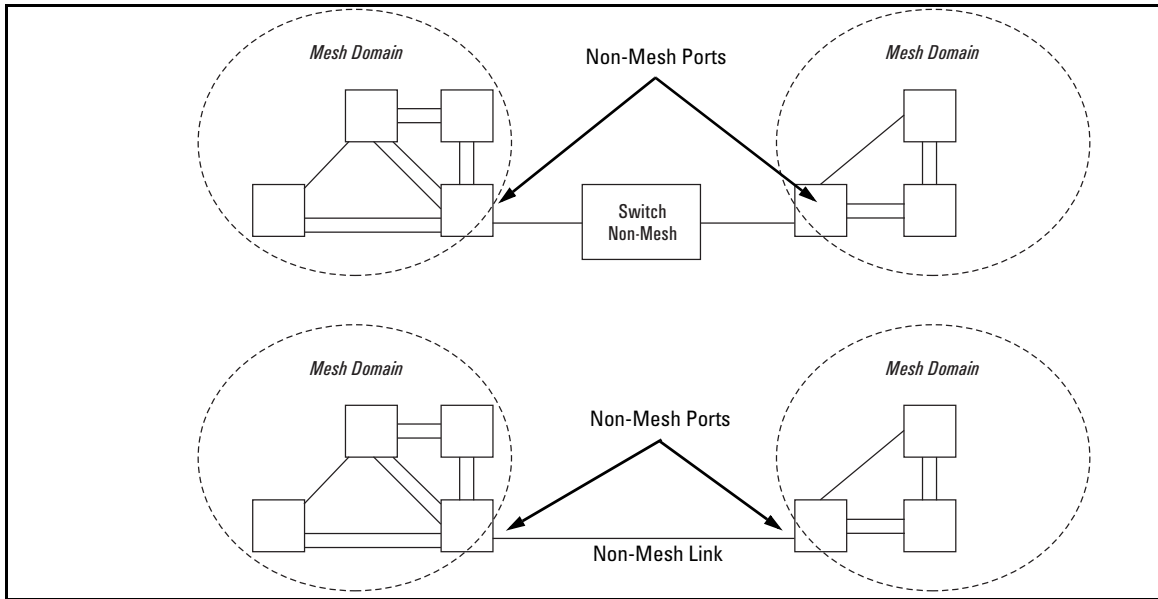
Figure 7-3. Example of Multiple Meshed Domains Separated by a Non-Mesh Switch or a Non-Mesh Link

■ If GVRP is enabled, meshed ports in a switch become members of any dynamic VLANs created in the switch in the same way that they would if meshing was not configured in the switch. (For more on GVRP, refer to chapter 3, "GVRP".)

GVRP Note   ProCurve 1600M/2400M/2424M/4000M/8000M switches do not offer the GVRP feature. If any of these switches are in your switch mesh, then GVRP must be disabled on any 3400cl, 6400cl, or 5300xl switches in the mesh.

Note
• A switch mesh domain (figure 7-1 on page 7-2) cannot include either a switch that is not configured for meshing, or a hub.
• Where a given pair of switches are linked with meshed ports, you must not also link the pair together through non-meshed ports unless you have also enabled STP, RSTP, or MSTP to prevent a loop from forming.

This topology forms a broadcast loop unless you configure STP or RSTP on the network.

Switch 1

Switch 3

Switch 2
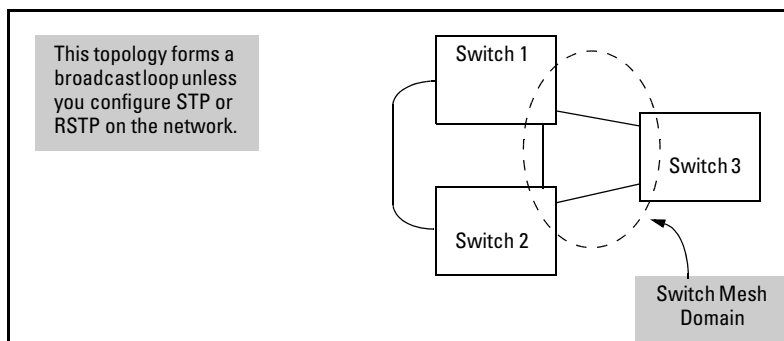
Switch Mesh Domain

Figure 7-4. Example of an Unsupported Topology

- The switch blocks traffic on a meshed port connected to a non-meshed port on another switch.
- Switch meshing does not allow trunked links (LACP or Trunk) between meshed ports.

Linking a non-mesh device or port into the mesh causes the meshed switch port(s) connected to that device to shut down.

Backward Compatibility Note

The ProCurve 3400cl, 6400cl, and 5300xl switches can interoperate with older devices in a switch mesh only after being placed in backwards compatibility mode. This is done with the **mesh backward-compat** command.

Using a Heterogeneous Switch Mesh

You can use 3400cl, 6400cl, and 5300xl switches together with any of the older ProCurve Switch 1600M/2400M/2424M/4000M/8000M models. These restrictions also apply:

■ All 3400cl, 6400cl, and 5300xl switches in the mesh must be placed in backward-compatible mode. This is done with the **mesh backward-compat** command.

■ The older models cannot be used in a mesh environment with 3400cl, 6400cl, and 5300xl switches where there is a duplicate MAC address on multiple switches and different VLANs. If you add an older model switch in this environment after the mesh is established, this switch will not be admitted to the mesh. If an older model switch is operating in a mesh with 3400cl, 6400cl, and/or 5300xl switches and you introduce a topology that creates a duplicate MAC address on multiple switches, the device accessed by these multiple switches will be blocked. For example:

Scenario 1: In a heterogenous mesh, creating the mesh with only one 5300xl, 3400cl, or 6400cl switch connected to the host (on VLAN 1, for example), and then connecting a second 5300xl, 3400cl, or 6400cl switch to the host (regardless of the VLAN used) results in connectivity issues with the host.

Scenario 2: Adding the Switch 4000M after bringing up the mesh with two later switches (5300xl, 3400xl, or 6400cl) already connected to the host as shown here (with or without separate VLANs) blocks the Switch 4000M from the mesh.

Untagged VLAN 1

Host (Both links use the same MAC address.)

Tagged VLAN 20

3400cl, 6400cl, or 5300xl Switch

3400cl, 6400cl, or 5300xl Switch

Switch 4000M

3400cl, 6400cl, or 5300xl Switch

The Switch 4000M is not supported in topologies allowing the same MAC address on multiple switches.

LAN

Mesh Domain

Figure 7-5. Example of an Unsupported Heterogeneous Topology Where Duplicate MAC Addresses Come Through Different Switches (Regardless of the VLANs Used)

Creating the mesh with only one Series 5300xl, 3400cl, or 6400cl switch connected to the host, and using tagged VLANs for multiple connections between the host and the meshed switch allows normal meshing operation.

Untagged VLAN 1

Host (Both links use the same MAC address.)

Tagged VLAN 20

5300xl, 6400cl, or 3400cl Switch

5300xl, 6400cl, or 3400cl Switch

Switch 4000M

5300xl, 6400cl, or 3400cl Switch

The Switch 4000M is not supported in topologies allowing the same MAC address on multiple switches.

LAN

Mesh Domain

Figure 7-6. Example of a Supported Heterogeneous Topology Where Duplicate MAC Addresses Come Through Different VLANs on the Same Switch

Note that in figures 7-5 and 7-6, if all switches are 3400cl, 6400cl, or 5300xl devices, then you can use either topology.

Also, if you have two separate switch meshes with the topology shown in figure 7-7, you cannot join them into a single mesh.

In this topology, the presence of a host using the same MAC address in multiple links to different meshed switches means that Mesh Domain "B" cannot be merged with Mesh Domain "A" to form a single mesh domain. This is because Mesh Domain "B" includes a Switch 1600M, 2400M 2424M, 4000M, or 8000M.
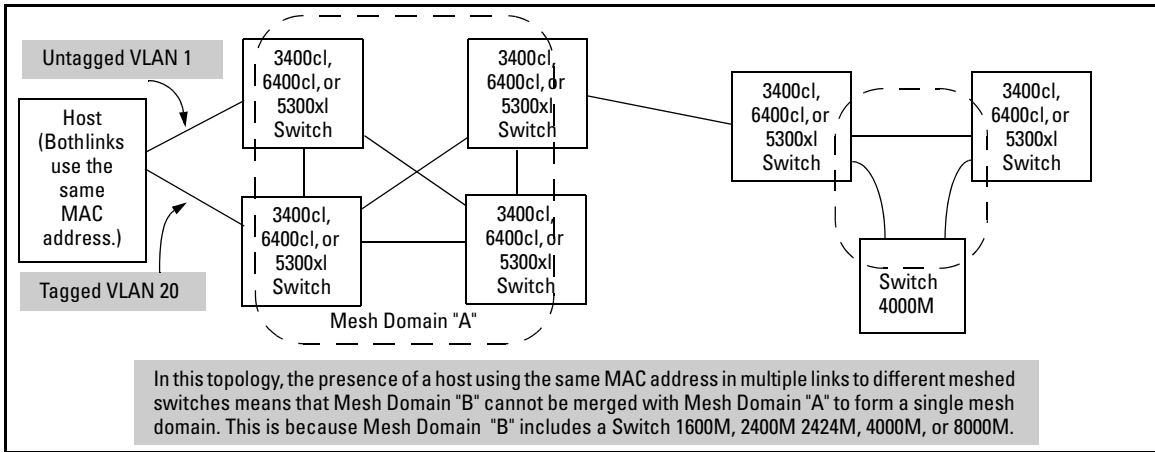
Figure 7-7. Example of Topology Where Adjacent Switch Meshes Cannot Be Merged Into a Single Mesh

■ Automatic Broadcast Control (ABC) on ProCurve 8000M/4000M/2424M/ 2400M/1600M switches is not supported when these switches are used in the same mesh domain with 3400cl, 6400cl,or 5300xl switches. Thus, in a mesh domain populated with all three types of switches, ABC must be disabled which is (the default setting) on all of the 8000M/4000M/2424M/ 2400M/1600M switches in the domain.

Bringing Up a Switch Mesh Domain:

When a meshed port detects a non-meshed port on the opposite end of a point-to-point connection, the link will be blocked. Thus, as you bring up switch meshing on various switches, you may temporarily experience blocked ports where meshed links should be running. These conditions should clear themselves after all switches in the mesh have been configured for meshing and their switches rebooted. To reduce the effect of blocked ports during bring-up, configure meshing and reboot the switches before installing the meshed switches in the network. Also, since adding (or removing) a meshed port requires a switch reboot to implement, you can avoid repeated system disruptions by waiting to implement the mesh until you have finished configuring meshing on all ports in your intended mesh domain.

Further Operating Information

Refer to "Operating Notes for Switch Meshing" on page 7-18.

Configuring Switch Meshing

Preparation

Before configuring switch meshing:

■   Review the Operating Rules (page 7-5), and particularly the restrictions
    and requirements for using switch meshing in environments that include
    static trunks, multiple static VLANs, GVRP, IGMP, and STP.

■   To avoid unnecessary system disruption, plan the mesh bring-up to min-
    imize temporary port-blocking. (Refer to "Bringing Up a Switch Mesh
    Domain:" on page 7-10.)

■   To view the current switch mesh status on the switch, use the CLI **show
    mesh** command (page 7-14).

Menu: To Configure Switch Meshing

1.   From the Main Menu, select:

     2. Switch Configuration
     **   2. Port/Trunk Settings**

2.   Press **[E]** (for **Edit**) to access the load balancing parameters.

```
===========================- CONSOLE - MANAGER MODE -============================
                   Switch Configuration - Port/Trunk Settings

    Port     Type        Enabled       Mode       Flow Ctrl   Group    Type
    ----   ---------- + -------    -------------   ---------   -----   -----
    A1     1000SX     | Yes         Auto           Disable
    A2     1000SX     | Yes         Auto           Disable
    A3     1000LX     | Yes         Auto           Disable
    A4     1000LX     | Yes         Auto           Disable
    B1     1000T      | Yes         Auto           Disable
    B2     1000T      | Yes         Auto           Disable
    B3     1000T      | Yes         Auto           Disable
    B4     1000T      | Yes         Auto           Disable
    C1     10/100TX   | Yes         Auto           Disable
    C2     10/100TX   | Yes         Auto           Disable
    C3     10/100TX   | Yes         Auto           Disable
    C4     10/100TX   | Yes         Auto           Disable

    Actions->   Cancel      Edit      Save       Help

Cancel changes and return to previous screen.
Use arrow keys to change action selection and <Enter> to execute action.
```

Figure 7-8. Example of the Screen for Configuring Ports for Meshing

3. In the Group column, move the cursor to the port you want to assign to the switch mesh.

4. Press **[M]** to choose **Mesh** for the selected port.

5. Use the **up-arrow or down-arrow** key to select the next port you want to include in your mesh domain, then press **[M]** again. For example, if you were adding ports A1 and A2 to your mesh domain, the screen would appear similar to figure 7-9:

```
==========================- CONSOLE - MANAGER MODE -==========================
                   Switch Configuration - Port/Trunk Settings

    Port    Type       Enabled     Mode       Flow Ctrl  Group   Type
    ----  ---------- + -------  -----------  ---------  -----  -----
    A1    1000SX     | Yes      Auto         Disable    Mesh
    A2    1000SX     | Yes      Auto         Disable    Mesh
    A3    1000LX     | Yes      Auto         Disable
    A4    1000LX     | Yes      Auto         Disable
    B1    1000T      | Yes      Auto         Disable
    B2    1000T      | Yes      Auto         Disable        Ports A1 and A2 configured
    B3    1000T      | Yes      Auto         Disable        for meshing.
    B4    1000T      | Yes      Auto         Disable
    C1    10/100TX   | Yes      Auto         Disable
    C2    10/100TX   | Yes      Auto         Disable
    C3    10/100TX   | Yes      Auto         Disable
    C4    10/100TX   | Yes      Auto         Disable

    Actions->   Cancel     Edit      Save      Help

 Select whether the port is part of a trunk or Mesh.
 Use arrow keys to change field selection, <Space> to toggle field choices,
 and <Enter> to go to Actions.
```

Figure 7-9. Example of Mesh Group Assignments for Several Ports

6. Repeat step 5 for all ports you want in the mesh domain.

Notes   For meshed ports, leave the **Type** setting blank. (Meshed ports do not accept a **Type** setting.)

All meshed ports in the switch automatically belong to the same mesh domain. (See figure 7-2 on page 7-4.)

7. When you finish assigning ports to the switch mesh, press **[Enter],** then **[S]** (for **Save**). You will then see the following screen.

The asterisk indicates that you must reboot the switch to cause the **Mesh** configuration change to take effect.

```
========================- CONSOLE - MANAGER MODE -=======================
                        Switch Configuration Menu

    1. System Information
   *2. Port/Trunk Settings
    3. Network Monitoring Port
    4. Spanning Tree Operation
    5. IP Configuration
    6. SNMP Community Names
    7. IP Authorized Managers
    8. VLAN Menu...
    0. Return to Main Menu...


Configures switch ports: Enabled, Mode, Flow Control, Trunking.
To select menu item, press item number, or highlight item and press <Enter
(*Needs reboot to activate changes.)
```

Figure 7-10. After Saving a Mesh Configuration Change, Reboot the Switch

8. Press **[0]** to return to the Main menu.

9. To activate the mesh assignment(s) from the Main menu, reboot the switch by pressing the following keys:

   a. **[6]** (for **Reboot Switch**)

   b. Space bar (to select **Yes**).

   c. **13** (to start the reboot process).

(The switch cannot dynamically reconfigure ports to enable or disable meshing, so it is always necessary to reboot the switch after adding or deleting a port in the switch mesh.)

CLI: To View and Configure Switch Meshing

Port Status and Configuration Features

Feature	Default	Menu	CLI	Web
viewing switch mesh status	n/a	n/a	below	n/a
configuring switch meshing	Disabled	n/a		n/a

Viewing Switch Mesh Status

Syntax: show mesh

> *Lists the switch ports configured for meshing, along with the*
> **State** *of each mesh-configured connection, the MAC address*
> *of the switch on the opposite end of the link (***Adjacent Switch***),*
> *and the MAC address of the port on the opposite end of the*
> *link (***Peer Port***).*

Reading the Show Mesh Output. For each port configured for meshing,
the State column indicates whether the port has an active link to the mesh or
is experiencing a problem. The status of the backwards compatibility option
is also displayed. For more details on the backwards compatibility option see
"CLI: Configuring Switch Meshing" on page 7-17.

```
ProCurve # show mesh

Status and Counters - Switch Mesh Information          Backwards
                                                       compatible
                                                       mode status
 Backward Compatibility mode enabled : No

 Port   State             | Adjacent Switch  Peer Port
 -----  ---------------   + ---------------  -------------
 C1     Established         0060b0-880a80    0060b0-880aff


   Port Configured      Operating State     MAC Address of the     MAC Address of the
   for Meshing          of the Link         Switch to which Port   Switch Port to which
                                            C1 Is Connected        Port C1 is Connected
```

Figure 7-11. Example of the Show Mesh Report

Table 7-1.   State Descriptions for Show Mesh Output

State	Meaning
Established	The port is linked to a meshed port on another switch and meshing traffic is flowing across the link. The **show mesh** listing includes the MAC addresses of the adjacent switch and direct connection port on the adjacent switch.
Not Established	The port may be linked to a switch on a port that is not configured for meshing or has gone down.
Initial	The port has just come up as a meshed port and is trying to negotiate meshing.
Disabled	The port is configured for meshing but is not connected to another device.
Error	Indicates a multiple MAC-address error. This occurs when you have two or more mesh ports from the same switch linked together through a hub.
Topology Error	Two meshed switches are connected via a hub, and traffic from other, non-meshed devices, is flowing into the hub. The **show mesh** listing includes the MAC addresses of the adjacent switch and direct connection port on the adjacent switch.

Topology Example with Show Mesh. Suppose that you have the following topology:
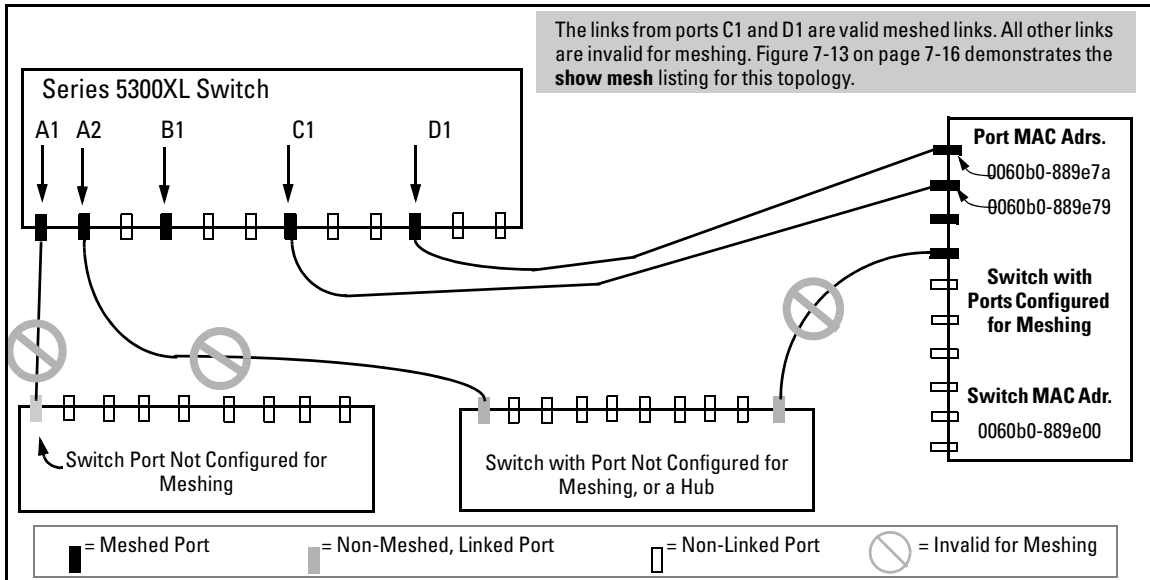
Figure 7-12. Example of a Meshed Topology with Some Mesh Ports Incorrectly Linked

Table 7-2 on page 7-16 describes the meshing operation in the above topology.

Table 7-2.   Operating Details for Figure 7-12

Port	Meshing?	Connection
A1	Yes	Connected to a port that may not be configured for meshing
A2	Yes	Connected to a switch port on a device that is not configured for meshing (another switch, or a hub). In this case, the **Topology Error** message indicates that the switch detects a meshed port on another, non-adjacent device that is also connected to the non-meshed switch or hub. ***However, meshing will not operate properly through this connection.***
B1	Yes	Not connected to another device.
C1	Yes	Connected to a meshed port on the same adjacent switch as D1 with meshing operating properly.
D1	Yes	Connected to a meshed port on the same adjacent switch as C1 with meshing operating properly.

Figure 7-13 lists the show mesh display for the topology and meshing config-uration in figure 7-12:

```
ProCurve# show mesh

Status and Counters - Switch Mesh Information

 Backward Compatibility mode enabled : No

 Port  State              | Adjacent Switch  Peer Port
 ----- ---------------- + --------------- -------------
 A1    Not Established
 A2    Topology Error     0060b0-889e00    0060b0-889e7b
 B1    Disabled
 C1    Established        0060b0-889e00    0060b0-889e7a
 D1    Established        0060b0-889e00    0060b0-889e79
```

Figure 7-13. Example of the Show Mesh Listing for the Topology in Figure 7-12

CLI: Configuring Switch Meshing

Syntax: [no] mesh [e] < *port-list* >

> *Enables or disables meshing operation on the specified ports.*

[no] mesh backward-compat

> *Enables or disables the switch for backward compatible mode. This allows the 3400cl, 6400cl, and 5300xl switches to interoperate with the* 8000M/4000M/2424M/2400M/1600M *switches in the same switch mesh.*
> ***Note:*** *Enabling this mode turns off some configuration checking done in a mesh with only 3400cl, 6400cl, or 5300xl switches. This command does not require a reboot to take effect.*

All meshed ports on a switch belong to the same mesh domain. Thus, to configure multiple meshed ports on a switch, you need to:

1. Specify the ports you want to operate in the mesh domain.

2. Use **write memory** to save the configuration to the startup-config file.

3. Reboot the switch

For example, to configure meshing on ports A1-A4, B3, C1, and D1-D3:

```
ProCurve (config)# mesh e a1-a4,b3,c1,d1-d3
Command will take effect after saving configuration and reboot.
ProCurve (config)# write memory
ProCurve (config)# boot
Device will be rebooted, do you want to continue [y/n]? y
```

Figure 7-14. Example of How To Configure Ports for Meshing

To remove a port from meshing, use the "**no**" version of **mesh**, followed by **write memory** and rebooting the switch. For example, to remove port C1 from the mesh:

```
ProCurve # config
ProCurve (config)# no mesh c1
Command will take effect after saving configuration and reboot.
ProCurve (config)# write memory
ProCurve (config)# boot
Device will be rebooted, do you want to continue [y/n]? y
```

Figure 7-15. Example of Removing a Port from the Mesh

Operating Notes for Switch Meshing

In a switch mesh domain traffic is distributed across the available paths with an effort to keep latency the same from path to path. The path selected at any time for a connection between a source node and a destination node is based on these latency and throughput cost factors:

- Outbound queue depth, or the current outbound load factor for any given outbound port in a possible path
- Port speed, such as 10Mbps versus 100Mbps; full-duplex or half-duplex
- Inbound queue depth, or how busy is a destination switch in a possible path
- Increased packet drops, indicating an overloaded port or switch

Paths having a lower cost will have more traffic added than those having a higher cost. Alternate paths and cost information is discovered periodically and communicated to the switches in the mesh domain. This information is used to assign traffic paths between devices that are newly active on the mesh. This means that after an assigned path between two devices has timed out, new traffic between the same two devices may take a different path than previously used.

To display information on the operating states of meshed ports and the identities of adjacent meshed ports and switches, see "Viewing Switch Mesh Status" on page 7-14.

Flooded Traffic

Broadcast and multicast packets will always use the same path between the source and destination edge switches unless link failures create the need to select new paths. (Broadcast and multicast traffic entering the mesh from different edge switches are likely to take different paths.) When an edge switch receives a broadcast from a non-mesh port, it floods the broadcast out all its other non-mesh ports, but sends the broadcast out only those ports in the mesh that represent the path from that edge switch through the mesh domain. (Only one copy of the broadcast packet gets to each edge switch for broadcast out of its nonmeshed ports. This helps to keep the latency for these packets to each switch as low as possible.)
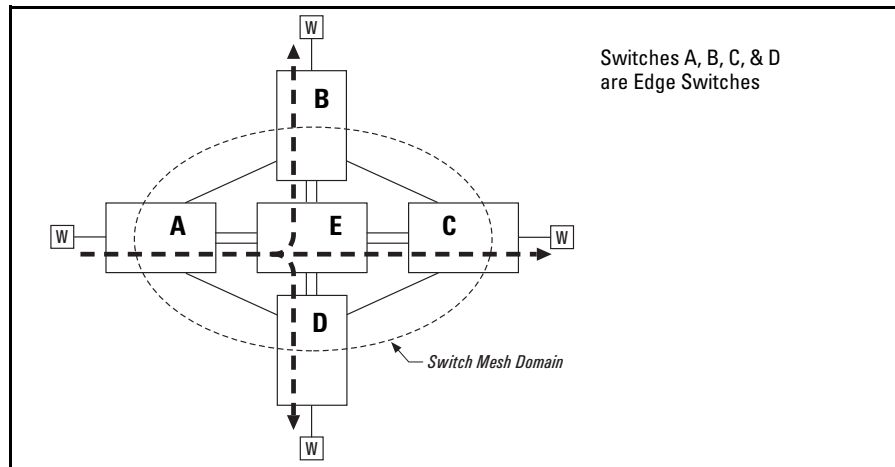
**Figure 7-16. Example of a Broadcast Path Through a Switch Mesh Domain**

Any mesh switches that are not edge switches will flood the broadcast packets only through ports (paths) that link to separate edge switches in the controlled broadcast tree. The edge switches that receive the broadcast will flood the broadcast out all non-meshed ports. Some variations on broadcast/multicast traffic patterns, including the situation where multiple VLANs are configured and a broadcast path through the mesh domain leads only to ports that are in the same VLAN as the device originating the broadcast.

Unicast Packets with Unknown Destinations

A meshed switch receiving a unicast packet with an unknown destination does not flood the packet onto the mesh. Instead, the switch sends a query on the mesh to learn the location of the unicast destination. The meshed switches then send 802.2 test packets through their non-meshed ports. After the unicast destination is found and learned by the mesh, subsequent packets having the same destination address will be forwarded. By increasing the **MAC Age Time** you can cause the switch address table to retain device addresses longer. (For more on **MAC Age Time**, refer to "System Information" in the chapter titled "Interface Access, System Information, and Friendly Port Names" in the *Management and Configuration Guide* for your switch.) Because the switches in a mesh exchange address information, this will help to decrease the number of unicast packets with unknown destinations, which improves latency within the switch mesh. Also, in an IP environment, HP recommends that you configure IP addresses on meshed switches. This makes the discovery mechanism more robust, which contributes to decreased latency.

Spanning Tree Operation with Switch Meshing

Using STP or RSTP with several switches and no switch meshing configured can result in unnecessarily blocking links and reducing available bandwidth. For example:
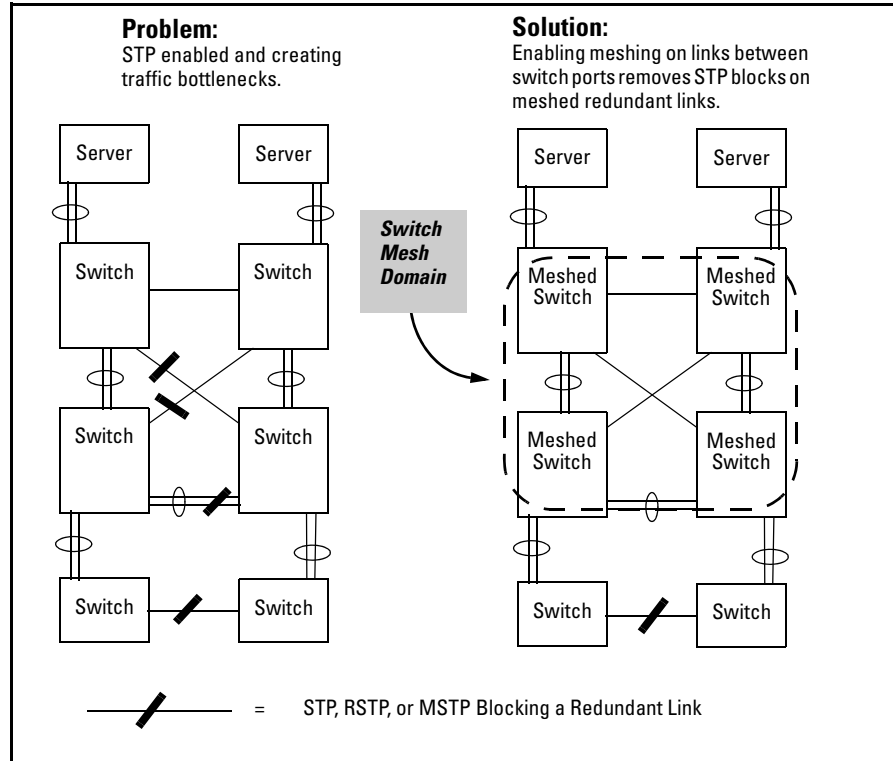
Figure 7-17. Example Using STP Without and With Switch Meshing

If you enable STP, RSTP, or MSTP on any meshed switch, you should enable the same spanning-tree protocol on all switches in the mesh. (That is, if you are going to use spanning-tree in a switch mesh, all switches in the mesh should be configured with the same type of spanning-tree: 802.1d/STP, 802.1w/RSTP, or 802.1s/MSTP.) Spanning-Tree interprets a meshed domain as a single link. However, on edge switches in the domain, STP and RSTP will manage non-meshed redundant links from other devices. For example:
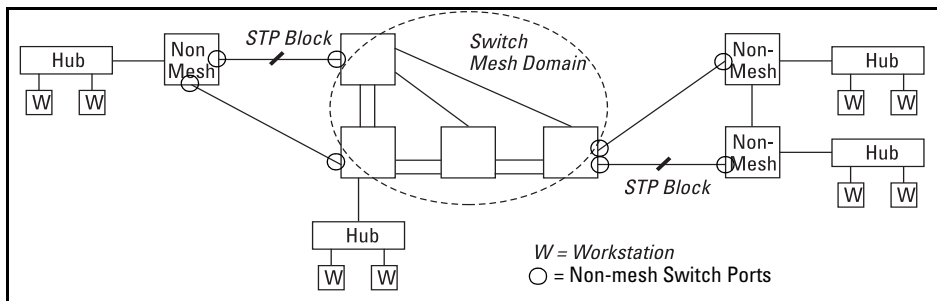
Figure 7-18. Connecting a Switch Mesh Domain to Non-Meshed Devices

Note on the Edge-Port Mode in RSTP and MSTP

When using RSTP or MSTP and interconnecting 3400cl, 6400cl, or 5300xl in a mesh with switches that are not in the mesh, all the non-mesh switch ports (as indicated in the figure above) should have the **edge-port** parameter disabled. For more information on RSTP edge-port parameter see "Optimizing the RSTP Configuration" on page 6-13

STP or RSTP should be configured on non-mesh devices that use redundant links to interconnect with other devices or with multiple switch mesh domains. For example:
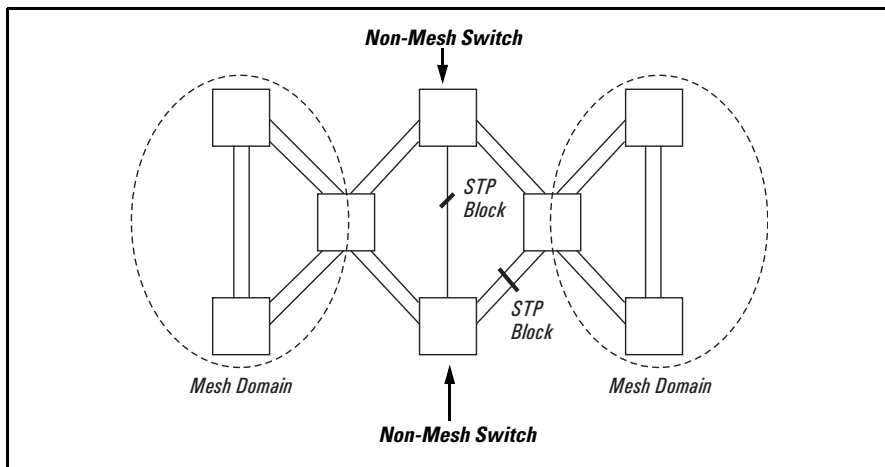
Figure 7-19. Interconnecting Switch Mesh Domains with Redundant Links

In the above case of multiple switch meshes linked with redundant trunks there is the possibility that spanning-tree will temporarily block a mesh link. This is because it is possible for spanning-tree to interpret the cost on an external trunked link to be less than the cost on a meshed link. However, if

this condition occurs, the meshed switch that has a blocked link will automatically increase the cost on the external (non-meshed) link to the point where STP or RSTP will block the external link and unblock the meshed link. This process typically resolves itself in approximately 30 seconds.

C a u t i o n

Spanning tree interprets a switch mesh as a single link. Because the switch automatically gives faster links a higher priority, the default spanning-tree parameter settings are usually adequate for spanning tree operation. Also, because incorrect STP or RSTP settings can adversely affect network performance, you should not make changes unless you have a strong understanding of how spanning tree operates.

In a mesh environment, the default RSTP and MSTP timer settings (**Hello Time** and **Forward Delay**) are usually adequate for RSTP or MSTP operation. Because a packet crossing a mesh may traverse several links within the mesh, using smaller-than-default settings for the RSTP **Hello Time** and **Forward Delay** timers can cause unnecessary topology changes and end-node connectivity problems.

For more on spanning-tree, refer to the chapter titled "Spanning-Tree Operation" in this manual. Also, you may want to examine the IEEE 802.1d, 802.1w, or 802.1s standards, depending on which version of spanning-tree you are using.

Filtering/Security in Meshed Switches

Because paths through the mesh can vary with network conditions, configuring filters on meshed ports can create traffic problems that are difficult to predict, and is not recommended. However, configuring filters on nonmeshed ports in an edge switch provides you with control and predictability.

IP Multicast (IGMP) in Meshed Switches

Like trunked ports, the switch mesh domain appears as a single port to IGMP. However, unlike trunked ports, IGMP protocol and multicast traffic may be sent out over several links in the mesh in the same manner as broadcast packets.

Static VLANs

In a network having a switch mesh domain and multiple static VLANs configured, all static VLANs must be configured on each meshed switch, even if no ports on the switch are assigned to any VLAN. (The switch mesh is a member of all static VLANs configured on the switches in the mesh.)

When static VLANs are configured, the mesh is seen as a single entity by each VLAN. All ports in the mesh domain are members of all VLANs and can be used to forward traffic for any VLAN. However, the non-mesh ports on edge switches that allow traffic to move between the mesh and non-meshed devices belong to specific VLANs and do not allow packets originating in a specific VLAN to enter non-meshed devices that do not belong to that same VLAN. (It is necessary to use a router to communicate between VLANs.) For example, in the following illustration, traffic from host A entering the switch mesh can only exit the mesh at the port for hosts B and E. Traffic from host A for any other host (such as C or D) will be dropped because only hosts B and E are in the same VLAN as host A.
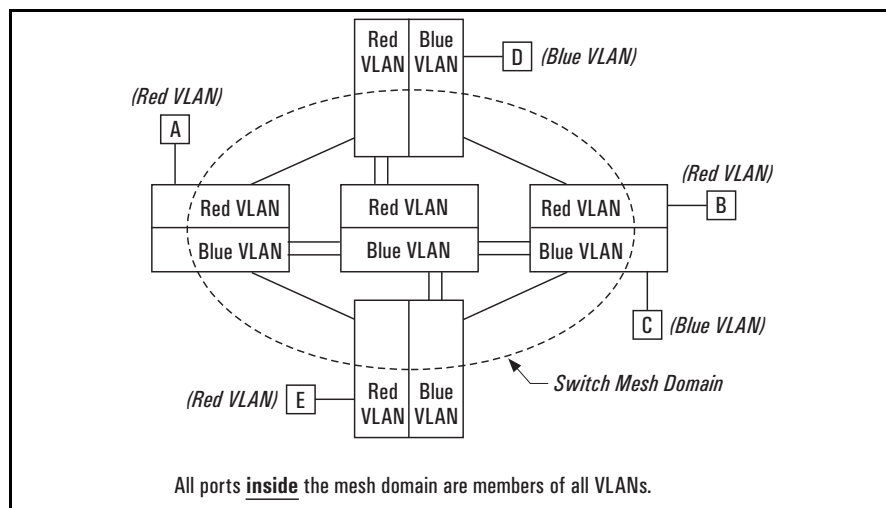

Figure 7-20. VLAN Operation with a Switch Mesh Domain

Dynamic VLANs

If GVRP is enabled, meshed ports in a switch become members of any dynamic VLANs created in the switch in the same way that they would if meshing was not configured in the switch. (For more on GVRP, refer to chapter 3, "GVRP".)

Jumbo Packets (3400cl and 6400cl Switches Only)

If you enable jumbo traffic on any VLAN is a 3400cl or 6400cl switch, then all meshed ports on the switch will be enabled to support jumbo traffic. (On a given meshed switch, every meshed port becomes a member of every VLAN configured on the switch.) If a port in a meshed domain does not belong to any VLANs configured to support jumbo traffic, then the port drops any jumbo packets it receives from other devices. In this regard, if a mesh domain includes any ProCurve Series 5300xl switches and/or ProCurve 1600M/2400M/ 2424M/4000M/8000M switches along with Series 3400cl and 6400cl switches configured to support jumbo traffic, only the 3400cl and 6400cl switches can transmit and receive jumbo packets. The other switch models in the mesh will drop such packets. For more information on jumbo packets, refer to the chapter titled "Port Traffic Controls" in the *Management and Configuration Guide* for your switch.

Mesh Design Optimization

Mesh performance can be enhanced by using mesh designs that are as small and compact as possible while still meeting the network design requirements. The following are limits on the design of meshes and have not changed:

1. Any switch in the mesh can have up to 24 meshed ports.

2. A mesh domain can contain up to 12 switches.

3. Up to 5 inter-switch meshed hops are allowed in the path connecting two nodes.

4. A fully interconnected mesh domain can contain up to 5 switches.

Mesh performance can be optimized by keeping the number of switches and the number of possible paths between any two nodes as small as possible. As mesh complexity grows, the overhead associated with dynamically calculating and updating the cost of all of the possible paths between nodes grows exponentially. Cost discovery packets are sent out by each switch in the mesh

every 30 seconds and are flooded to all mesh ports. Return packets include a cost metric based on inbound and outbound queue depth, port speed, number of dropped packets, etc. Also, as mesh complexity grows, the number of hops over which a downed link has to be reported may increase, thereby increasing the reconvergence time.

The simplest design is the two-tier design because the number of possible paths between any two nodes is kept low and any bad link would have to be communicated only to it's neighbor switch.

Other factors affecting the performance of mesh networks include the number of destination addresses that have to be maintained, and the overall traffic levels and patterns. However a conservative approach when designing new mesh implementations is to use the two-tier design and limit the mesh domain to eight switches where possible.

As shown here, meshing allows multiple redundant links between switches in the domain, and more than one link is allowed between any two switches in the domain.

Note also that a switch can have up to 24 ports configured for meshing.

Switch 1 — Switch 2

Switch 3 — Switch 4 — Switch 5 — Switch 6 — Switch 7 — Switch 8
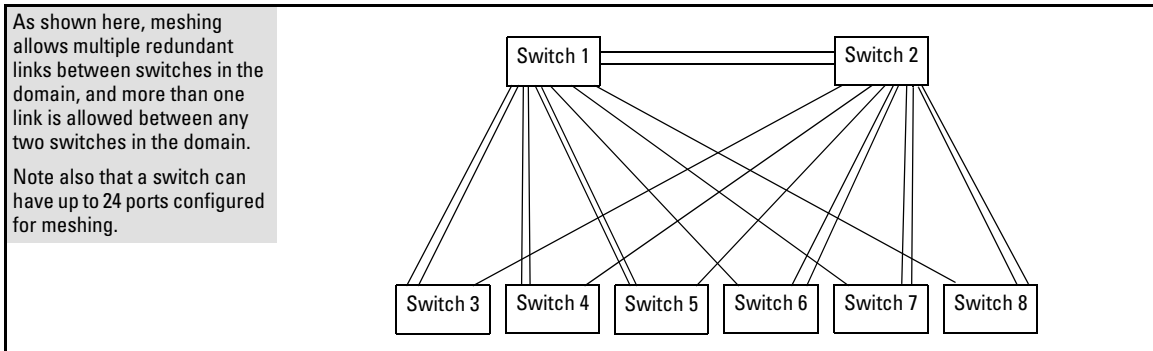
Figure 7-21. Example of a Two-Tier Mesh Design

Switch 1

Switch 5 — Switch 2

Switch 4 — Switch 3

As noted above, a switch can have up to 24 ports configured for meshing.
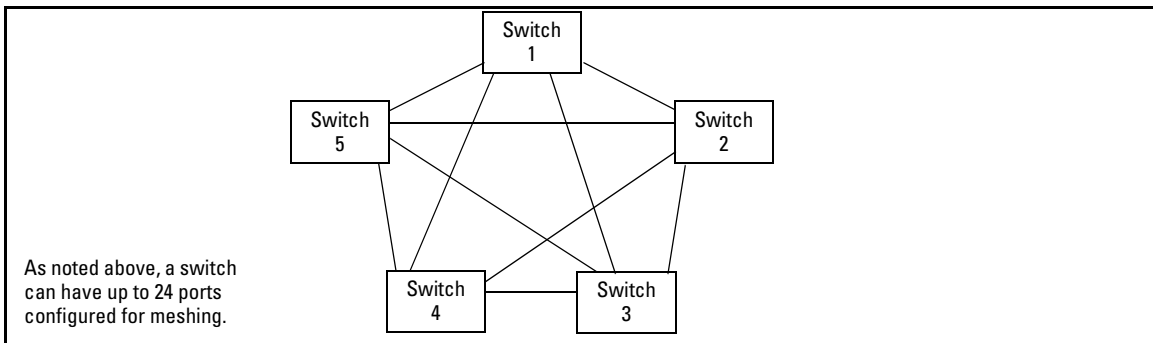
Figure 7-22. Example of a Fully Interconnected Mesh with the Maximum Switch Count

Other factors affecting the performance of mesh networks include the number of destination addresses that have to be maintained, and the overall traffic levels and patterns. However a conservative approach when designing new mesh implementations is to use the two-tier design and limit the mesh domain to eight switches where possible.

Other Requirements and Restrictions

■ **Mesh Support Within the Domain:** All switches in the mesh domain, including edge switches, must support the ProCurve switch meshing protocol.

■ **Switch Hop Count in the Mesh Domain:** A maximum of five (meshed) switch hops is allowed in the path connecting two nodes in a switch mesh domain. A path of six meshed hops is unusable. However, this does not interfere with other, shorter paths in the same domain.

■ **Connecting Mesh Domains:** To connect two separate switch meshing domains, you must use non-meshed ports. (The non-meshed link can be a port trunk or a single link.) Refer to figure 7-3 on page 7-7.

■ **Multiple Links Between Meshed Switches:** Multiple mesh ports can be connected between the same two switches, to provide higher band-width. Each port that you want in the mesh domain should be configured as **Mesh** (and not as a trunk—**Trk**). Note that if you configure a port as **Mesh**, there is no "Type" selection for that port.

■ **Automatic Broadcast Control:** Series 3400cl, 6400cl, and 5300xl switches do not offer this feature. Thus, in a switch mesh comprised of 3400cl, 6400cl, and/or 5300xl switches and any of the 1600M/2400M/ 2424M/4000M/8000M switches, ABC must be disabled (which is the default setting) on the 1600M/2400M/2424M/4000M/8000M switches.

■ **Network Monitor Port:** If a network monitor port is configured, broad-cast packets may be duplicated on this port if more than one port is being monitored and switch meshing is enabled.

■ **Compatibility with Older Switches:** Only after the Series 3400cl, 6400cl, and 5300xl switches are placed in backward compatibility mode will they operate with older switches. For more information see "CLI: Configuring Switch Meshing" on page 7-17. Each entry in a Series 3400cl, 6400cl, or 5300xl switch's MAC-address table consists of a MAC address and a VLAN ID (VID). In older switches there is no VID; just a MAC address. The older switches will therefore detect indistinguishable, dupli-cate addresses where the Series 3400cl, 6400cl, and 5300xl switches will detect multiple different addresses consisting of the same MAC address and different VIDs. In a switch mesh that includes any 1600M/2400M/ 2424M/4000M/8000M switches, duplicate MAC addresses entering the mesh on different switches are not allowed. (These older switches do not

recognize multiple instances of a particular MAC address on different VLANs.) If you try to add one of these switches to a mesh comprised entirely of Series 3400cl, 6400cl, and/or 5300xl switches, and any of these switches detects a duplicate MAC address entering the mesh through separate switches, the 1600M/2400M/2424M/4000M/8000M switch will not be allowed into the switch mesh.

■ **Rate-Limiting Not Recommended on Meshed Ports:** Rate-Limiting can reduce the efficiency of paths through a mesh domain.

(See also "Operating Rules" on page 7-5.)

For additional information on troubleshooting meshing problems, refer to "Using a Heterogeneous Switch Mesh" on page 7-8 and "Mesh-Related Problems" in appendix C, "Troubleshooting" of the Management and Configuration Guide for your switch.

— This page is intentionally unused. —

8

Quality of Service (QoS): Managing Bandwidth More Effectively

---

Contents

Introduction

QoS Feature	Default	Menu	CLI	Web
UDP/TCP Priority	Disabled	—	page 8-25	Refer to the Online Help.
IP-Device Priority	Disabled	—	page 8-31	"
IP Type-of-Service Priority	Disabled	—	page 8-37	"
LAN Protocol Priority	Disabled	—	page 8-50	"
VLAN-ID Priority	Disabled	—	page 8-52	"
Source-Port Priority	Disabled	—	page 8-58	"
DSCP Policy Table	Various	—	page 8-64	"

As the term suggests, *network policy* refers to the network-wide controls you can implement to:

■ Ensure uniform and efficient traffic handling throughout your network, while keeping the most important traffic moving at an acceptable speed, regardless of current bandwidth usage.

■ Exercise control over the priority settings of inbound traffic arriving in and travelling through your network.

Adding bandwidth is often a good idea, but it is not always feasible and does not completely eliminate the potential for network congestion. There will always be points in the network where multiple traffic streams merge or where network links will change speed and capacity. The impact and number of these congestion points will increase over time as more applications and devices are added to the network.

When (not *if*) network congestion occurs, it is important to move traffic on the basis of relative importance. However, without *Quality of Service* (QoS) prioritization, less important traffic can consume network bandwidth and slow down or halt the delivery of more important traffic. That is, without QoS, most traffic received by the switch is forwarded with the same priority it had upon entering the switch. In many cases, such traffic is "normal" priority and competes for bandwidth with all other normal-priority traffic, regardless of its relative importance to your organization's mission.

This section gives an overview of QoS operation and benefits, and describes how to configure QoS in the console interface.

Quality of Service is a general term for classifying and prioritizing traffic throughout a network. That is, QoS enables you to establish an end-to-end traffic priority policy to improve control and throughput of important data. You can manage available bandwidth so that the most important traffic goes first. For example, you can use Quality of Service to:

■ Upgrade or downgrade traffic from various servers.

■ Control the priority of traffic from dedicated VLANs or applications.

■ Change the priorities of traffic from various segments of your network as your business needs change.

■ Set priority policies in edge switches in your network to enable traffic-handling rules across the network.

Edge Switch	Honor Priority	Downstream Switch	Honor New Priority
Classify inbound traffic on these Class-of-Service (CoS) types: • IP-device (address) • Protocol (LAN) • VLAN-ID (VID). • Source-Port Apply 802.1p priority to selected outbound traffic on tagged VLANs. **Set Priority**	**Downstream Switch** Tagged VLANs on inbound and outbound ports. Traffic arrives with priority set by edge switch Forward with 802.1p priority.	Tagged VLANs on some or all inbound and outbound ports. Classify inbound traffic on CoS types. Change priority on selected CoS type(s). Forward with 802.1p priority. **Change Priority**	**Downstream Switch** Tagged VLANs on at least some inbound ports. Traffic arrives with the priority set in the VLAN tag. Carry priority downstream on tagged VLANs.

Figure 8-1. Example of 802.1p Priority Based on CoS (Class-of-Service) Types and Use of VLAN Tags

Edge Switch	Honor Policy	Downstream Switch	Honor New Policy
Classify inbound traffic on IP-device (address) and VLAN-ID (VID). Apply DSCP markers to selected traffic. **Set Policy**	**Downstream Switch** Traffic arrives with DSCP markers set by edge switch Classify on ToS DiffServ.	Classify on ToS DiffServ and Other CoS Apply new DSCP markers to selected traffic. **Change Policy**	**Downstream Switch** Classify on ToS Diffserv

Figure 8-2. Example Application of Differentiated Services Codepoint (DSCP) Policies

At the edge switch, QoS classifies certain traffic types and in some cases applies a DSCP policy. At the next hop (downstream switch) QoS honors the policies established at the edge switch. Further downstream, another switch may reclassify some traffic by applying new policies, and yet other downstream switches can be configured to honor the new policies.

QoS is implemented in the form of rules or policies that are configured on the switch. While you can use QoS to prioritize only the outbound traffic while it is moving through the switch, you derive the maximum benefit by using QoS in an 802.1Q VLAN environment (with 802.1p priority tags) or in an untagged VLAN environment (with DSCP policies) where QoS can set priorities that downstream devices can support without re-classifying the traffic.

By prioritizing traffic, QoS supports traffic growth on the network while optimizing the use of existing resources—and delaying the need for further investments in equipment and services. That is, QoS enables you to:

■ Specify which traffic has higher or lower priority, regardless of current network bandwidth or the relative priority setting of the traffic when it is received on the switch.

■ Change (upgrade or downgrade) the priority of outbound traffic.

■ Override "illegal" packet priorities set by upstream devices or applications that use 802.1Q VLAN tagging with 802.1p priority tags.

■ Avoid or delay the need to add higher-cost NICs (network interface cards) to implement prioritizing. (Instead, control priority through network policy.)

QoS on the switches covered by this guide supports these types of traffic marking:

■ **802.1p prioritization:** Controls the outbound port queue priority for traffic leaving the switch, and (if traffic exits through a VLAN-tagged port) sends the priority setting with the individual packets to the downstream devices.

■ **IP Type-of-Service (ToS):** Enables the switch to set, change, and honor prioritization policies by using the Differentiated Services (diffserv) bits in the ToS byte of IPv4 packet headers.

Terminology

Term	Use in This Document
802.1p priority	A traffic priority setting carried by a VLAN-tagged packet moving from one device to another through ports that are tagged members of the VLAN to which the packet belongs. This setting can be from 0 - 7. The switch handles an outbound packet on the basis of its 802.1p priority. However, if the packet leaves the switch through a VLAN on which the port is an untagged member, this priority is dropped, and the packet arrives at the next, downstream device without an 802.1p priority assignment.
802.1Q field	A four-byte field that is present in the header of Ethernet packets entering or leaving the switch through a port that is a tagged member of a VLAN. This field includes an 802.1p priority setting, a VLAN tag, or ID number (VID), and other data. A packet entering or leaving the switch through a port that is an untagged member of the outbound VLAN does not have this field in its header and thus does not carry a VID or an 802.1p priority. See also "802.1p priority".
codepoint	Refer to DSCP, below.
downstream device	A device linked directly or indirectly to an outbound switch port. That is, the switch sends traffic to downstream devices.
DSCP	***Differentiated Services Codepoint.*** (Also termed ***codepoint***.) A DSCP is comprised of the upper six bits of the ToS (Type-of-Service) byte in IP packets. There are 64 possible codepoints. In the default QoS configuration for the 5300xl switches, some codepoints are configured with default 802.1p priority settings for Assured-Forwarding and Expedited Forwarding. In the default QoS configuration for the 3400cl/6400cl switches, one codepoint (101110) is set for Expedited Forwarding. All other codepoints are unused (and listed with **No-override** for a priority).
DSCP policy	A DSCP configured with a specific 802.1p priority (0- 7). (Default: **No-override**). Using a DSCP policy, you can configure the switch to assign priority to IP packets. That is, for an IP packet identified by the specified classifier, you can assign a new DSCP and an 802.1p priority (0-7). For more on DSCP, refer to "Details of QoS IP Type-of-Service" on page 8-47. For the DSCP map, see figure 8-23 on page 8-48.
edge switch	In the QoS context, this is a switch that receives traffic from the edge of the LAN or from outside the LAN and forwards it to devices within the LAN. Typically, an edge switch is used with QoS to recognize packets based on classifiers such as TCP/UDP application type, IP-device (address), Protocol (LAN), VLAN-ID (VID), and Source-Port (although it can also be used to recognize packets on the basis of ToS bits). Using this packet recognition, the edge switch can be used to set 802.1p priorities or DSCP policies that downstream devices will honor.
inbound port	Any port on the switch through which traffic enters the switch.
IP Options	In an IPv4 packet, optional, these are extra fields in the packet header.
IP-precedence bits	The upper three bits in the Type of Service (ToS) field of an IP packet.
IPv4	Version 4 of the IP protocol.
outbound packet	A packet leaving the switch through any LAN port.
outbound port	Any port on the switch through which traffic leaves the switch.

Term	Use in This Document
outbound port queue	For any port, a buffer that holds outbound traffic until it can leave the switch through that port. There are four outbound queues for each port in the switch: high, medium, normal, and low. Traffic in a port's high priority queue leaves the switch before any traffic in the port's medium priority queue, and so-on.
re-marking (DSCP re-marking)	Assigns a new QoS policy to an outbound packet by changing the DSCP bit settings in the ToS byte.
tagged port membership	Identifies a port as belonging to a specific VLAN and enables VLAN-tagged packets belonging to that VLAN to carry an 802.1p priority setting when outbound from that port. Where a port is an untagged member of a VLAN, outbound packets belonging to that VLAN do not carry an 802.1p priority setting.
Type-of-Service (ToS) byte	Comprised of a three-bit (high-order) precedence field and a five-bit (low-order) Type-of-Service field. Later implementations may use this byte as a six-bit (high-order) Differentiated Services field and a two-bit (low-order) reserved field. See also "IP-precedence bits" and DSCP elsewhere in this table.
upstream device	A device linked directly or indirectly to an inbound switch port. That is, the switch receives traffic from upstream devices.

Overview

QoS settings operate on two levels:

■ **Controlling the priority of outbound packets moving through the switch:** Each switch port has four outbound traffic queues; "low", "normal", "medium", and "high" priority. Packets leave the switch port on the basis of their queue assignment and whether any higher queues are empty:

Table 8-1.   Port Queue Exit Priorities

Port Queue and 802.1p Priority Values	Priority for Exiting From the Port
Low (1 - 2)	Fourth
Normal (0, 3)	Third
Medium (4 - 5)	Second
High (6 - 7)	First

A QoS configuration enables you to set the outbound priority queue to which a packet is sent. (In an 802.1Q VLAN environment with VLAN-tagged ports, if QoS is *not* configured on the switch, but *is* configured on an upstream device, the priorities carried in the packets determine the forwarding queues in the switch.)

■ **Configuring a priority for outbound packets and a service (priority) policy for use by downstream devices:**

• **DSCP Policy:** This feature enables you to set a priority policy in outbound IP packets. (You can configure downstream devices to read and use this policy.) This method is not dependent on VLAN-tagged ports to carry priority policy to downstream devices, and can:

– Change the codepoint (the upper six bits) in the ToS byte.

– Set a new 802.1p priority for the packet.

(Setting DSCP policies requires IPv4 inbound packets. Refer to the "IPv4" entry under "Terminology" on page 8-6.)

• **802.1p Priority Rules:** An outbound, VLAN-tagged packet carries an 802.1p priority setting that was configured (or preserved) in the switch. This priority setting ranges from 0 to 7, and can be used by downstream devices having up to eight outbound port queues. Thus, while packets within the switch move at the four priority levels shown in table 8-1, above, they still can carry an 802.1p priority that can be used by downstream devices having more or less than the four priority levels in the switches covered by this guide. Also, if the packet enters the switch with an 802.1p priority setting, QoS can override this setting if configured with an 802.1p priority rule to do so.

N o t e s :

If your network uses only one VLAN (and therefore does not require VLAN-tagged ports) you can still preserve 802.1p priority settings in your traffic by configuring the ports as tagged VLAN members on the links between devices you want to honor traffic priorities.

Rule and Policy Limits: The switches covered by this guide have differing limits on the number of rules and policies they support:

– **5300xl Switches and 4200vl Switches:** Beginning with software release **E.08.01**, the switch allows up to **250** 802.1p priority rules and/or DSCP policies in any combination. For more information, refer to "Maximum QoS Configuration Entries" under "QoS Operating Notes and Restrictions" on page 8-73.

– **3400cl/6400cl Switches:** The combined number of 802.1p priority rules and DSCP policies the switch supports depends on the usage of rules by the ACL and other factions. Refer to "QoS Operating Notes and Restrictions" on page 8-73.

You can configure a QoS priority of 0 through 7 for an outbound packet. When the packet is then sent to a port, the QoS priority determines which outbound queue the packet uses:

Table 8-2.   QoS Priority Settings and Operation

QoS Priority Setting	Outbound Port Queue
1 - 2	low priority
0 - 3	normal priority
4 - 5	medium priority
6 - 7	high priority

If a packet is not in a VLAN-tagged port environment, then the QoS settings in table 8-2 control only to which outbound queue the packet goes. Without VLAN tagging, no 802.1p priority is added to the packet for downstream device use. But if the packet is in a VLAN-tagged environment, then the above setting is also added to the packet as an 802.1p priority for use by downstream devices and applications (shown in table 8-3). In either case, an IP packet can also carry a priority policy to downstream devices by using DSCP-marking in the ToS byte.

Table 8-3.   Mapping Series 5300xl and 3400cl/6400cl QoS Priority Settings to Device Queues

Priority Setting in thitherto	Outbound Port Queues in the Switch	802.1p Priority Setting Added to Tagged VLAN Packets Leaving the Switch	Queue Assignment in Downstream Devices With:		
			8 Queues	3 Queues	2 Queues
1	Queue 1	1 (low priority)	Queue 1	Queue 1	
2		2	Queue 2		Queue 1
0	Queue 2	0 (normal priority)	Queue 3	Queue 2	
3		3	Queue 4		
4	Queue 3	4 (medium priority)	Queue 5	Queue 3	
5		5	Queue 6		Queue 2
6	Queue 4	6 (high priority)	Queue 7		
7		7	Queue 8		

Classifiers for Prioritizing Outbound Packets

The classifiers used in the 3400cl/6400cl switches are a subset of the classifiers used in the 5300xl and 4200vl switches. Also, the 3400cl/6400cl switches search for classifier matches in the opposite order of that used in the 5300xl and 4200vl switches.

Note On Using Multiple Criteria

HP recommends that you configure a minimum number of the available QoS classifiers for prioritizing any given packet type. Increasing the number of active classifier options for a packet type increases the complexity of the possible outcomes and consumes switch resources.

5300xl and 4200vl Packet Classifiers and Evaluation Order

The 5300xl and 4200vl switches provide seven QoS classifiers (packet criteria) you can use to configure QoS priority.

Table 8-4.   5300xl and 4200vl Classifier Search Order and Precedence

Search Order	Precedence	QoS Classifier Type
1	1 (highest)	UDP/TCP Application Type (port)
2	2	Device Priority (destination or source IP address)
3	3	IP Type of Service (ToS) field (IP packets only)
4	4	Protocol Priority (IP, IPX, ARP, DEC LAT, AppleTalk, SNA, and NetBeui)
5	5	VLAN Priority
6	6	Incoming source-port on the switch
7	7 (lowest)	Incoming 802.1p Priority (present in tagged VLAN environments)

Where multiple classifier types are configured, a 5300xl or 4200vl switch uses the highest-to-lowest search order shown in table 8-4 to identify the highest-precedence classifier to apply to any given packet. When a match between a packet and a classifier is found, the switch applies the QoS policy configured for that classifier and the packet is handled accordingly.

Note that on a 5300xl or 4200vl switch, if the switch is configured with multiple classifiers that address the same packet, the switch uses only the QoS config-uration for the QoS classifier that has the highest precedence. (In this case, the QoS configuration for another, lower-precedence classifier that may apply is ignored.) For example, if QoS assigns high priority to packets belonging to VLAN 100, but normal priority to all IP protocol packets, since protocol priority (4) has precedence over VLAN priority (5), IP protocol packets on VLAN 100 will be set to normal priority.

3400cl/6400cl Packet Classifiers and Evaluation Order

The 3400cl/6400cl switches provide six QoS classifiers (packet criteria) you can use to configure QoS priority.

Table 8-5.   3400cl/6400cl Classifier Search Order and Precedence

Search Order	Precedence	QoS Classifier
1	6 (lowest)	Incoming 802.1p Priority (present in tagged VLAN environments)
2	5	Incoming source-port on the switch
3	4	VLAN Priority
4	3	IP Type of Service (ToS) field (IP packets only)
5	2	Device Priority (destination or source IP address)
6	1 (highest)	UDP/TCP Application Type (port)

A 3400cl or 6400cl switch uses the lowest-to-highest search order shown in table 8-5 to identify the highest-precedence classifier to apply to any given packet. (Note that this is the opposite of the order used in the 5300xl or 4200vl switches.) If there is only one configured classifier that matches a given packet, then the switch applies the QoS policy specified in that classifier. If multiple configured classifiers match a given packet, the switch applies each one in turn to the packet and concludes with the QoS policy for the highest-precedence classifier. Note that if the highest precedence classifier is configured to apply a DSCP policy, then both the DSCP in the packet and the 802.1p priority applied to the packet can be changed. However, if the highest precedence classifier is configured to apply an 802.1p priority rule, only the 802.1p priority in the final QoS match for the packet is changed.

N o t e

On the 3400cl/6400cl switches, intermixing lower-precedence classifiers configured with DSCP policies and higher-precedence classifiers configured with 802.1p priority rules is not recommended, as this can result in a packet with an 802.1p priority assigned by one classifier and a DSCP policy by another classifier. This is because the search order would allow a lower precedence classifier configured with a DSCP policy to change both the DSCP and the 802.1p setting in a packet, and then would allow a subsequent, higher precedence classifier configured with an 802.1p priority rule to change only the 802.1p setting. *To avoid this problem, a DSCP policy option should be applied only on the highest-precedence classifier in use on the switch or applied to all QoS classifiers in use on the switch.*

In general, the precedence of QoS classifiers should be considered when configuring QoS policies. For example, suppose that a system administrator has used an 802.1p priority rule to assign a high priority for packets received on VLAN 100, but has also used another 802.1p priority rule to assign a normal priority for TCP port 80 packets received on the switch. Since TCP/UDP port precedence supersedes VLAN precedence, all TCP port 80 packets on VLAN 100 will be set to normal priority. For a classifier precedence listing, see table 8-5, "3400cl/6400cl Classifier Search Order and Precedence", on page 8-11.

Table 8-6. Precedence Criteria for QoS Classifiers

Precedence	Criteria	Overview
1	UDP/TCP	Takes precedence based on a layer 4 UDP or TCP application, with a user-specified application port number (for example, Telnet). **Default state:** Disabled
		If a packet does not meet the criteria for UDP/TCP priority, then precedence defaults to the Device Priority classifier, below.
2	Device Priority (IP Address)	Takes precedence based on an inbound packet having a particular destination or source IP address. QoS applies the following IP address limits: – 5300xl and 4200vl Switches: Up to 256 IP addresses – 3400cl/6400cl Switches: Up to 60 IP address If a given packet has a destination IP address matching a QoS configuration, this packet takes precedence over another packet that has the matching IP address as a source address. (This can occur, for example, on an outbound port in a switch mesh environment.) Also, if the source and destination IP addresses (SA and DA) in the same packet match for different QoS policies, the DA takes precedence. **Default state:** No IP address prioritization.
		If a packet does not meet the criteria for device priority, then precedence defaults to the IP Type of Service (ToS) classifier, below.
3	IP Type-of-Service (IP ToS)	Takes precedence based on the TOS field in IP packets. (Applies only to IP packets.) The ToS field is configured by an upstream device or application before the packet enters the switch. • **IP Precedence Mode:** QoS reads an inbound packet's IP precedence (upper three) bits in the Type-of-Service (ToS) byte and automatically assigns an 802.1p priority to the packet (if specified in the QoS configuration) for outbound transmission. • **Differentiated Services (Diffserve) Mode:** QoS reads an inbound IP packet's differentiated services, or codepoint (upper six), bits of the Type-of-Service (TOS) byte. Packet prioritization depends on the configured priority for the codepoint. (Some codepoints default to the DSCP standard, but can be overridden.) For more on IP ToS, see "QoS IP Type-of-Service (ToS) Policy and Priority" on page 8-37. **Default state:** Disabled.
		If a packet does not meet the criteria for ToS priority, then precedence defaults as follows: – 5300xl and 4200vl switches: To the Protocol classifier – 3400cl/6400cl switches: To the VLAN classifier
		— Continued —

Precedence	Criteria	Overview
4	Layer 3 Protocol Priority	**Note:** This classifier is available in the 5300xl and 4200vl switches, but not in the 3400cl/6400cl switches. To prioritize traffic in a 3400cl or 6400cl switch according to protocol type, configure the switch to place traffic of the desired protocol type in a specific VLAN, and then apply the VLAN classifier. Takes precedence based on network protocols: IP, IPX, ARP, DEC LAT, AppleTalk, SNA, and NetBeui. **Default state:** No-override for any protocol. If a packet does not meet the criteria for Protocol priority, then precedence defaults to the VLAN classifier, below.
5	VLAN Priority	Takes precedence based on the ID number of the VLAN in which the inbound packet exists. For example, if the default VLAN (VID = 1) and the "Blue" VLAN (with a VID of 20) are both assigned to a port, and Blue VLAN traffic is more important, you can configure QoS to give Blue VLAN traffic a higher priority than default VLAN traffic. (Priority is applied on the outbound port.) **Default state:** No-override. If a packet does not meet the criteria for VLAN priority, then precedence defaults to the Source-Port classifier, below.
6	Source-Port	Takes precedence based on the source-port (that is, the port on which the packet entered the switch). If a packet does not meet the criteria for source-port priority, then precedence defaults to Incoming 802.1p criteria, below
7	Incoming 802.1p Priority	Where a VLAN-tagged packet enters the switch through a port that is a tagged member of that VLAN, if QoS is not configured to override the packet's priority setting, the switch uses the packet's existing 802.1p priority (assigned by an upstream device or application) to determine which inbound and outbound port queue to use. If there is no QoS policy match on the packet, and it then leaves the switch through a port that is a tagged member of the VLAN, then there is no change to its 802.1p priority setting. If the packet leaves the switch through a port that is an untagged member of the VLAN, the 802.1p priority is dropped. Entering (Inbound) 802.1p Priority / Outbound Port Queue / Exiting (Outbound) 802.1p Priority 1 - 2 / Low / 1 - 2 0 - 3 / Normal / 0 - 3 4 - 5 / Medium / 4 - 5 6 - 7 / High / 6 - 7 If a packet does not meet the criteria for Incoming 802.1p priority, then the packet goes to the "normal" outbound queue of the appropriate port. If the packet entered the switch through a port that is an untagged member of a VLAN, but exits through a VLAN-tagged port, then an 802.1Q field, including an 802.1p priority, is added to the packet header. If no QoS policy is configured or applied to the packet, then the 802.1p priority of 0 (normal) is assigned to the packet for outbound transmission.

Preparation for Configuring QoS

QoS operates in VLAN-tagged and VLAN-untagged environments. If your network does not use multiple VLANs, you can still implement the 802.1Q VLAN capability for packets to carry their 802.1p priority to the next downstream device. To do so, configure ports as VLAN-tagged members on the links between switches and routers in your network infrastructure.

Table 8-7.   Summary of QoS Capabilities

Outbound Packet Options	Port Membership in VLANs	
	Tagged	Untagged
Control Port Queue Priority for Packet Types	Yes	Yes
Carry 802.1p Priority Assignment to Next Downstream Device	Yes	No
Carry DSCP Policy to Downstream Devices. The policy includes:	Yes [1]	Yes [1]
Assigning a ToS Codepoint		
Assigning an 802.1p Priority [2] to the Codepoint		

[1] Except for non-IPv4 packets or packets processed using either the Layer 3 Protocol (5300xl and 4200vl only) or QoS IP-Precedence methods, which do not include the DSCP policy option. Also, to use a service policy in this manner, the downstream devices must be configured to interpret and use the DSCP carried in the IP packets.

[2] This priority corresponds to the 802.1p priority scheme and is used to determine the packet's port queue priority. When used in a VLAN-tagged environment, this priority is also assigned as the 802.1p priority carried outbound in packets having an 802.1Q field in the header.

Steps for Configuring QoS on the Switch

1. Determine the QoS policy you want to implement. This includes analyzing the types of traffic flowing through your network and identifying one or more traffic types to prioritize. In order of QoS precedence, these are:

    a. UDP/TCP applications

    b. Device Priority—destination or source IP address (Note that destination has precedence over source. See Table 8-6.)

    c. IP Type-of-Service Precedence Bits (Leftmost three bits in the ToS field of IP packets)

    d. IP Type-of-Service Differentiated Service bits (Leftmost six bits in the ToS field of IP packets)

    e. Protocol Priority (Series 5300xl switches only)

    f. VLAN Priority (requires at least one tagged VLAN on the network)

    g. Source-Port

      h.   Incoming 802.1p Priority (requires at least one tagged VLAN on the network)

For more on how QoS operates with the preceding traffic types, see "Precedence Criteria for QoS Classifiers", on page 8-12.)

Select the QoS option you want to use. Table 8-8 lists the traffic types (QoS classifiers) and the QoS options you can use for prioritizing or setting a policy on these traffic types:

Table 8-8.   Applying QoS Options to Traffic Types Defined by QoS Classifiers

QoS Options for Prioritizing Outbound Traffic		QoS Classifiers						
		UDP/ TCP	IP Device	IP-ToS Precedence	IP- DiffServ	L3 Protocol	VLAN -ID	Source -Port
Option 1: Configure 802.1p Priority Rules Only	Prioritize traffic by sending specific packet types (determined by QoS classifier) to different outbound port queues on the switch. Rely on VLAN-tagged ports to carry packet priority as an 802.1p value to downstream devices.	Yes	Yes	Yes [1]	Yes	Yes[2]	Yes	Yes
Option 2: Configure ToS DSCP Policies with 802.1p Priorities	Prioritize traffic by sending specific packet types (determined by QoS classifier) to different outbound port queues on the switch. Propagate a service policy by reconfiguring the DSCP in outbound IP packets according to packet type. The packet is placed in an outbound port queue according to the 802.1p priority configured for that DSCP policy. (The policy assumes that downstream devices can be configured to recognize the DSCP in IP packets and implement the service policy it indicates.) Use VLAN-tagged ports to include packet priority as an 802.1p value to downstream devices.	Yes	Yes	No	Yes	No	Yes	Yes

[1] In this mode the configuration is fixed. You cannot change the automatic priority assignment when using IP-ToS Precedence as a QoS classifier.

[2] Not available on the 3400cl/6400cl switches.

    2.   If you want 802.1p priority settings to be included in outbound packets, ensure that tagged VLANs are configured on the appropriate downstream links.

3. Determine the actual QoS configuration changes you will need to make on each QoS-capable device in your network in order to implement the desired policy. Also, if you want downstream devices to read and use DSCPs in IP packets from the switch, configure them to do so by enabling ToS Differentiated Service mode and making sure the same DSCP policies are configured.

4. If you are planning a QoS configuration on a 3400cl or 6400cl switch, refer to the next section, "Planning QoS for the Series 3400cl/6400cl Switches".

N o t e   On the 3400cl/6400cl switches, ACLs and Rate-Limiting use the same internal resources (per-port rules) as QoS. For this reason, plus the limit on per-port rule resources, it is important to consider rule usage when preparing to configure QoS on the switch. For more information, refer to the next section, "Planning QoS for the Series 3400cl/6400cl Switches".

Planning QoS for the Series 3400cl/6400cl Switches

QoS, ACLs, and Rate Limiting share certain internal, per-port resources on 3400cl/6400cl switches. Thus, QoS (and ACL) configurations can load internal resources in ways that require more careful attention to resource usage when planning a configuration using these features. Otherwise, there is an increased possibility of oversubscribing some switch resources, which means that at some point the switch would not support further QoS, ACL, and/or Rate-Limiting configuration. This section describes resource planning for QoS features on a 3400cl or 6400cl switch. For ACL planning, refer to chapter 10, "Access Control Lists (ACLs) for the Series 3400cl and Series 6400cl Switches". For information on Rate-Limiting, refer to the "Rate-Limiting" section in the chapter titled "Port Traffic Controls" of the *Management and Configuration Guide* for your switch.

Prioritizing and Monitoring QoS, ACL, and Rate Limiting Feature Usage on the 3400cl/6400cl Switches

If you want to configure QoS and either ACLs or Rate-Limiting (or both) on a 3400cl or 6400xl switch, plan and implement your configuration in descending order of feature importance. This will help to ensure that the most important features are configured first. Also, if insufficient resources become a problem, this approach can help you recognize how to distribute the desired feature implementations across multiple switches to achieve your objectives. For example, adding ACLs consumes resources faster than QoS rules and policies. If QoS is more important on a particular 3400cl or 6400cl switch than ACLs,

then you should plan and configure your QoS resource usage first for that switch. If insufficient resources remain for all of the ACL implementation you want, try spreading this implementation across multiple switches.

QoS Resource Usage and Monitoring on 3400cl/6400cl Switches

QoS, ACLs, multicast protocols, and Rate-Limiting configurations on the 3400cl/6400cl switches use rule resources on a per-port basis. Per-Port rule usage is reserved as shown below:

Table 8-9.   Rule Maximums on the 3400cl/6400xl Switches

Feature	Maximum Rules Available Per-Port
QoS and Rate-Limiting[1]	120 maximum in any feature combination
ACLs and IGMP[2]	

[1]Configuring Rate-Limiting on a port uses one per-port rule on that port.
[2]Configuring IGMP on any static VLAN uses one per-port ACL mask on all ports.

Table 8-10 describes rule resource use for each QoS classifier type.

Table 8-10.   QoS Rule Resource Usage

QoS Classifier	Port Application	Rules Used
TCP and UDP	All Ports in the Switch	2 per TCP or UDP Application
Device Priority	" " " " "	2 per IP Address
ToS IP-Precedence	" " " " "	8
ToS Diff-Services	" " " " "	1 per Codepoint[1]
VLAN	All Ports in the VLAN	1 per VLAN
Source Port	Specified Port(s)	1 per Port[2]

[1]When the ToS Diff-Services mode is enabled, each codepoint (DSCP) policy configured in the DSCP map and each (inbound) codepoint assigned to a DSCP policy use one rule per-port. When this mode is disabled, all rules used by the ToS Diff-Services option, including any DSCP policies configured in the DSCP map, become available for other uses.
[2]Enabling source-port QoS and rate-limiting on the same port uses one rule.

The following two CLI commands are unique to the 3400cl/6400cl switches and are useful for planning and monitoring rule usage in a QoS configuration.

Syntax: qos resources help

Provides a quick reference on how QoS and ACLs use rule resources for each configuration option. Includes most of the information in table 8-10, plus an ACL usage summary.

Syntax: show qos resources

Shows the number of rules currently available on each port. This command is useful for verifying rule availability as you proceed with configuring QoS, ACL, and/or Rate-Limiting features available on the switch.

Managing QoS Resource Consumption on the 3400cl/6400cl Switches

As shown in table 8-10, QoS classifiers use 1, 2, or 8 rules depending on the classifier selected. Extensive QoS and ACL configurations, with or without applying Rate-Limiting or a multicast protocol, can either fully subscribe the 120 rules available on a given port or leave an insufficient number of rules available for configuring another QoS policy on the switch. If there are not enough rules on the port to support another QoS policy, you cannot configure an additional policy on that port. Because most QoS features are applied to all ports, having one or more ports with insufficient rules remaining to support another QoS policy limits further QoS configurations on the switch to:

- Source-port QoS on ports that have sufficient unused rules
- VLAN QoS on VLANs where all of the member ports have sufficient unused rules

(This can also block any further ACL and—if not already configured—a Rate-Limiting configuration on the affected port.)

Problems with insufficient rules available on a port can occur in either of the following QoS scenarios:

■ Attempting to configure a policy when one or more of the affected ports have insufficient rules available

■ Attempting to add a port to a QoS-configured VLAN where the policy already on the VLAN requires more rule resources than the port has available.

Configuring a Policy When There Are Not Enough Rules Available On a Target Port. Attempting to configure a QoS policy on the switch, on a VLAN, or on selected ports when there are not enough rules available on one or more ports that are subject to the command results in the following:

■ The policy is not configured on any ports subject to the command.

■ The CLI displays the following message:

```
Unable to add this QoS rule. Maximum number (120)
already reached.
```

Adding a Port to a QoS-Configured VLAN Without Enough Rules Available on the Port. When you add a port to an existing, QoS-configured VLAN, the switch attempts to apply the VLAN's QoS configuration to the port. If the port has insufficient rule resources to add the VLAN's QoS configuration:

■ The port *is* added to the VLAN.

■ The QoS classifiers configured on the VLAN are *not* added to the port, which means that the port does not honor the QoS policies configured for the VLAN.

■ The switch generates this message in the Event Log:

```
cos: Vlan 1 QoS not configured on all new ports.
Some QoS resources exceeded
```

Troubleshooting a Shortage of Per-Port Rule Resources on the 3400cl/6400cl Switches

The lack of available rules is caused by existing QoS, ACL, and (if configured) Rate-Limiting configurations consuming the available rules on one or more ports. Do the following to enable configuration of the desired policy:

1. Use the **show qos resources** command to identify the port(s) on which there are insufficient rule resources. For example, figure 8-3 includes ports that can be the source of problems due to rule consumption by policies configured earlier:

```
ProCurve(config)# show qos resources
QoS/ACL Resource Usage

Port | Rules Available | ACL Masks Available
-----+-----------------+----------------
  1  |  104            |    8
  2  |   40            |    6
  3  |    2            |    6
  4  |    1            |    6
  5  |    0            |    6
  6  |   86            |    7
  .  |    .            |    .
  .  |    .            |    .
  .  |    .            |    .
```

At a minimum, the policies configured on port 5 must be reduced to free up enough rule resources to add a new QoS policy. Depending on the QoS policy you want to add, existing policies on ports 3 and 4 may have to be reduced.

Port 3 has enough rules available to accept any policy that uses 1 or 2 rules.

Port 4 can accept only a policy that uses one rule.

Port 5 is fully subscribed and cannot accept any new policies.

Figure 8-3. Example of Inspecting Available Rule Resources

2. Use **show** commands to identify the currently configured QoS, ACL, and Rate-Limiting policies.

3. Determine which of the existing policies you can remove to free up rule resources for the QoS policy you want to implement. Depending on your network topology and configuration, you can free up rule resources by moving some policies to other devices. Another alternative is to inspect the switch's existing QoS, ACL, and Rate-Limiting configurations for unnecessary entries or inefficient applications that could be removed or revised to achieve the desired policies with less resource usage. Tables 8-9 and 8-10 on page 8-17, or the information displayed by the **qos resources help** command, can help you to determine the resource usage of QoS and ACL policies.

Examples of QoS Resource Usage on 3400cl/6400cl Switches

Demonstrating Differing Resource Usage on Different Ports. Suppose that VLANs 111 and 222 on a 3400cl or 6400cl switch are configured for VLAN QoS. Also, device-priority QoS is configured for five IP addresses. The VLAN QoS affects only the ports that belong to VLANs 111 and 222. The device-priority QoS affects all ports on the switch. If ports 1 and 2 belong to both VLANs and ports 3 and 4 belong only to VLAN 222, then these two pairs of ports will differ in how many rules they use. Ports 5 through 24 do not belong to the VLANs, and so will use fewer rules than ports 1 through 4.
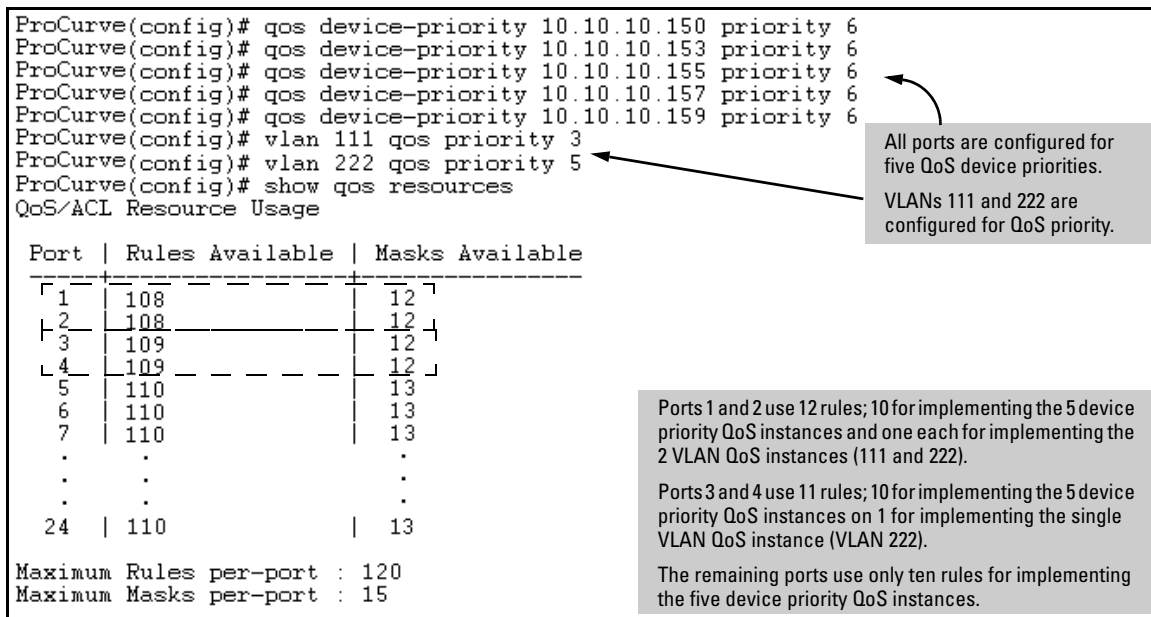
```
ProCurve(config)# qos device-priority 10.10.10.150 priority 6
ProCurve(config)# qos device-priority 10.10.10.153 priority 6
ProCurve(config)# qos device-priority 10.10.10.155 priority 6
ProCurve(config)# qos device-priority 10.10.10.157 priority 6
ProCurve(config)# qos device-priority 10.10.10.159 priority 6
ProCurve(config)# vlan 111 qos priority 3
ProCurve(config)# vlan 222 qos priority 5
ProCurve(config)# show qos resources
QoS/ACL Resource Usage

 Port | Rules Available | Masks Available
 -----+-----------------+----------------
   1  | 108             |   12
   2  | 108             |   12
   3  | 109             |   12
   4  | 109             |   12
   5  | 110             |   13
   6  | 110             |   13
   7  | 110             |   13
   .  |   .             |    .
   .  |   .             |    .
   .  |   .             |    .
  24  | 110             |   13

Maximum Rules per-port : 120
Maximum Masks per-port : 15
```

All ports are configured for five QoS device priorities.

VLANs 111 and 222 are configured for QoS priority.

Ports 1 and 2 use 12 rules; 10 for implementing the 5 device priority QoS instances and one each for implementing the 2 VLAN QoS instances (111 and 222).

Ports 3 and 4 use 11 rules; 10 for implementing the 5 device priority QoS instances on 1 for implementing the single VLAN QoS instance (VLAN 222).

The remaining ports use only ten rules for implementing the five device priority QoS instances.

Figure 8-4. Example of QoS Resource Usage with Device-Priority and VLAN QoS Implemented

Table 8-11. Per-Port Resource Usage in Figure 8-4

Port	Five QoS Device Priorities	VLAN 111	VLAN 222	Rules Usage
1	Yes (10 rules)	Yes (1 rule)	Yes (1 rule)	2 rules per device priority QoS instance 1 rule per VLAN QoS instance
2	Yes (10 rules)	Yes (1 rule)	Yes (1 rule)	2 rules per device priority QoS instance 1 rule per VLAN QoS instance
3	Yes (10 rules)	No	Yes	2 rules per device priority QoS instance
4	Yes (10 rules)	No	Yes	2 rules per device priority QoS instance
5 - 24	Yes (10 rules)	No	No	2 rules per device priority QoS instance

Demonstrating How the Switch Uses Resources in DSCP Configurations. In the default configuration, the DSCP map is configured with one DSCP policy (Expedited Forwarding; 101110 with a "7" priority) but, because no ToS Diff-Services options are configured, no rules are used. If ToS Diff-Services mode is enabled, then one rule is immediately used for this codepoint. Adding a new DSCP policy (for example, 001111 with a "5" priority) and then configuring ToS Diff-Services to assign inbound packets with a codepoint of 001010 to the 001111 policy implements all policies configured in the DSCP map and, in this case, uses three rules; one for each codepoint invoked in the switch's current DSCP configuration (101110-the default, 001111, and 001010). Adding another Diff-Services assignment, such as assigning inbound packets with a codepoint of 000111 to the Expedited Forwarding policy (101110), would use one more rule on all ports.

```
ProCurve(config)# show qos resources
QoS/ACL Resource Usage

 Port | Rules Available | ACL Masks Available
 -----+-----------------+--------------------
   1  | 120             |  8
   2  | 120             |  8
   3  | 120             |  8
   .  |  .              |  .
   .  |  .              |  .
   .  |  .              |  .
  24  | 120             |  8

Maximum Rules per-port : 120
Maximum Masks per-port : 8
```

Figure 8-5. Example of Rule Resources in the Default Configuration

```
ProCurve(config)# qos dscp-map 001111 priority 5
ProCurve(config)# qos type-of-service diff-services 001010 dscp 001111
ProCurve(config)# show qos resources
QoS/ACL Resource Usage

 Port | Rules Available | Masks Available
 -----+-----------------+----------------
   1  | 117             |  7
   2  | 117             |  7
   3  | 117             |  7
   .  |  .              |  .
   .  |  .              |  .
   .  |  .              |  .
  24  | 117             |  7

Maximum Rules per-port : 120
Maximum Masks per-port : 8
```

Assigning inbound packets with 001010 in the ToS byte to the newly created 001111 policy enables ToS Diff-Services mode. Because the default DSCP map already includes the Expedited Delivery (101110) policy, enabling ToS Diff- Services uses three rules on each port; one for each configured codepoint (101110, 001010, and 001111). As a result, the available rule count drops by 3 to 117.

Figure 8-6. Example of Rule Usage When a Configuration Includes DSCP-Map and Type-of-Service Options

Using QoS Classifiers To Configure Quality of Service for Outbound Traffic

QoS Feature	Default	Menu	CLI	Web
UDP/TCP Priority	Disabled	—	page 8-25	Refer to Online Help.
IP-Device Priority	Disabled	—	page 8-31	"
IP Type-of-Service Priority	Disabled	—	page 8-37	"
LAN Protocol Priority	Disabled	—	page 8-50	"
VLAN-ID Priority	Disabled	—	page 8-52	"
Source-Port Priority	Disabled	—	page 8-58	"

N o t e In addition to the information in this section on the various QoS classifiers, refer to "QoS Operating Notes and Restrictions" on page 8-73.

Viewing the QoS Configuration

All of these commands are available on the 5300xl and 4200vl switches. All except the **protocol-priority** command are available on the 3400cl/6400cl switches. Examples of the **show qos** output are included with the example for each priority type.

Syntax: show qos < *priority-classifier* >

tcp-udp-port-priority

Displays the current TCP/UDP port priority configuration. Refer to figure 8-11 on page 8-31.

device-priority

Displays the current device (IP address) priority configuration. Refer to figure 8-12 on page 8-33.

type-of-service

Displays the current type-of-service priority configu-ration. The display output differs according to the ToS option used:

■   *IP Precedence: Refer to figure 8-16 on page 8-38.*

■   *Diffserve: Refer to figure 8-18 on page 8-42.*

protocol-priority

Available on the 5300xl and 4200vl switches. Displays the current protocol priority configuration.

vlan-priority

Displays the current VLAN priority configuration. Refer to figure 8-26 on page 8-54.

port-priority

Displays the current source-port priority configura-tion. Refer to figure 8-31 on page 8-59.

No Override

By default, the IP ToS, Protocol, VLAN-ID, and (source) port **show** outputs automatically list **No-override** for priority options that have not been config-ured. This means that if you do not configure a priority for a specific option, QoS does not prioritize packets to which that option applies, resulting in the **No override** state. In this case, IP packets received through a VLAN-tagged port receive whatever 802.1p priority they carry in the 802.1Q tag in the packet's header. VLAN-Tagged packets received through an untagged port are handled in the switch with "normal" priority. For example, figure 8-7 below shows a qos VLAN priority output in a switch where non-default priorities exist for VLANs 22 and 33, while VLAN 1 remains in the default configuration.

```
ProCurve(config)# show qos vlan-priority

  VLAN priorities

  VLAN ID Apply rule  | DSCP    Priority
  ------- ----------- + ------  -----------
  1       No-override |         No-override
  22      Priority    |         0
  33      DSCP        | 000010  6
```

This output shows that VLAN 1 is in the default state, while VLANs 22 and 33 have been configured for 802.1p and DSCP Policy priorities respectively.

Figure 8-7. Example of the Show QoS Output for VLAN Priority

Note

As mentioned in table 8-6, the 3400cl/6400cl switches do not include the layer 3 protocol classifier. However, you can still apply a QoS priority to non-IP Layer 3 protocol traffic by grouping such traffic into separate VLANs, as desired, and then assigning a priority based on VLAN membership.

QoS UDP/TCP Priority

QoS Classifier Precedence: 1

When you use UDP or TCP and a layer 4 Application port number as a QoS classifier, traffic carrying the specified UDP/TCP port number(s) is marked with the UDP/TCP classifier's configured priority level, without regard for any other QoS classifiers in the switch.

N o t e

UDP/TCP QoS applications are supported only for IPv4 packets only. For more information on packet-type restrictions, refer to "Details of Packet Criteria and Restrictions for QoS Support", on page 8-73.

Options for Assigning Priority. Priority control options for TCP or UDP packets carrying a specified TCP or UDP port number include:

■ 802.1p priority
■ DSCP policy (Assigning a new DSCP and an associated 802.1p priority; inbound packets must be IPv4.)

For a given TCP or UDP port number, you can use only one of the above options at a time. However, for different port numbers, you can use different options.

TCP/UDP Port Number Ranges. There are three ranges:

■ Well-Known Ports: 0 - 1023
■ Registered Ports: 1024 - 49151
■ Dynamic and/or Private Ports: 49152 - 65535

For more information, including a listing of UDP/TCP port numbers, go to the *Internet Assigned Numbers Authority* (IANA) web site at:

www.iana.org

Then click on:

Protocol Number Assignment Services

P (Under "Directory of General Assigned Numbers" heading)

Port Numbers

Assigning an 802.1p Priority Based on TCP or UDP Port Number

This option assigns an 802.1p priority to (IPv4) TCP or UDP packets as described below.

Syntax: qos < udp-port | tcp-port > < *tcp or udp port number* > priority < 0 - 7 >

> *Configures an 802.1p priority for outbound packets having the specified TCP or UDP application port number. This priority determines the packet's queue in the outbound port to which it is sent. If the packet leaves the switch on a tagged port, it carries the 802.1p priority with it to the next downstream device. (Default: Disabled)*

> ***Note:*** *On 3400cl/6400cl switches, this feature is not supported for IPv4 packets with IP options. For more information on packet-type restrictions, refer to table 8-15 on page 8-73.*
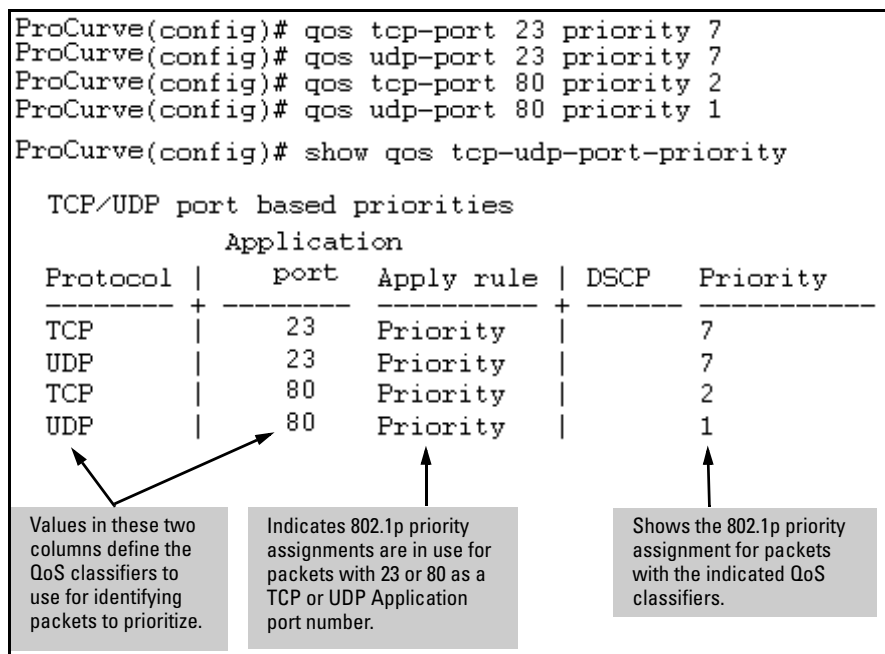
no qos < udp-port | tcp-port > < *tcp-udp port number* >

> *Deletes the specified UDP or TCP port number as a QoS classifier.*

show qos tcp-udp-port-priority

> *Displays a listing of all TCP and UDP QoS classifiers currently in the running-config file.*

For example, configure and list 802.1p priority for the following UDP and TCP port prioritization:

TCP/UDP Port	802.1p Priority for TCP	802.1p Priority for UDP
TCP Port 23 (Telnet)	7	7
UDP Port 23 (Telnet)	7	7
TCP Port 80 (World Wide Web HTTP)	2	2
UDP Port 80 (World Wide Web HTTP)	1	1

```
ProCurve(config)# qos tcp-port 23 priority 7
ProCurve(config)# qos udp-port 23 priority 7
ProCurve(config)# qos tcp-port 80 priority 2
ProCurve(config)# qos udp-port 80 priority 1

ProCurve(config)# show qos tcp-udp-port-priority

   TCP/UDP port based priorities
               Application
   Protocol |    port   Apply rule | DSCP    Priority
   -------- + -------   ---------- + ------  ----------
   TCP      |    23     Priority   |            7
   UDP      |    23     Priority   |            7
   TCP      |    80     Priority   |            2
   UDP      |    80     Priority   |            1
```

Values in these two columns define the QoS classifiers to use for identifying packets to prioritize.

Indicates 802.1p priority assignments are in use for packets with 23 or 80 as a TCP or UDP Application port number.

Shows the 802.1p priority assignment for packets with the indicated QoS classifiers.

Figure 8-8. Example of Configuring and Listing 802.1p Priority Assignments on TCP/UDP Ports

Assigning a DSCP Policy Based on TCP or UDP Port Number

N o t e   The Series 5300xl and Series 4200vl switches do not support DSCP policies on IPv4 packets with IP options. The 3400cl/6400cl switches do not support TCP/UDP QoS policies on packets with IP options. For more information on packet-type restrictions, refer to "Details of Packet Criteria and Restrictions for QoS Support", on page 8-73.

This option assigns a previously configured DSCP policy (codepoint and 802.1p priority) to (IPv4) TCP or UDP packets having the specified port number. That is, the switch:

1. Selects an incoming IP packet if the TCP or UDP port number it carries matches the port number specified in the TCP or UDP classifier (as shown in figure 8-8, above).

2. Overwrites (re-marks) the packet's DSCP with the DSCP configured in the switch for such packets.

3. Assigns the 802.1p priority configured in the switch for the new DSCP. (Refer to "Differentiated Services Codepoint (DSCP) Mapping" on page 8-64.)

4. Forwards the packet through the appropriate outbound port queue.

3400cl/6400cl Switch Restriction. On the 3400cl/6400cl switches, "mixing" ToS DSCP policies and 802.1p priorities is not recommended. Refer to the Note on page 8-11.

For more on DSCP, refer to "Terminology" on page 8-6.

Steps for Creating a DSCP Policy Based on TCP/UDP Port Number Classifiers. This procedure creates a DSCP policy for IPv4 packets carrying the selected UDP or TCP port-number classifier.

1. Identify the TCP or UDP port-number classifier you want to use for assigning a DSCP policy.

2. Determine the DSCP policy for packets carrying the selected TCP or UDP port number.
   a. Determine the DSCP you want to assign to the selected packets. (This codepoint will be used to overwrite (re-mark) the DSCP carried in packets received from upstream devices.)
   b. Determine the 802.1p priority you want to assign to the DSCP.

3. Configure the DSCP policy by using **qos dscp-map** to configure the priority to the codepoint you selected in step 2a. (For details, refer to the example later in this section, and to "Differentiated Services Codepoint (DSCP) Mapping" on page 8-64.)

Note   A codepoint must have an 802.1p priority assignment (0 - 7) before you can configure a policy for prioritizing packets by TCP or UDP port numbers. If a codepoint you want to use shows **No-override** in the **Priority** column of the DSCP map (**show qos dscp-map**), then you must assign a 0 - 7 priority before proceeding.

4. Configure the switch to assign the DSCP policy to packets with the specified TCP or UDP port number.

Syntax: qos dscp-map < *codepoint* > priority < 0 - 7 >

> *This command is optional if a priority has already been assigned to the <* **codepoint**>. *The command creates a DSCP policy by assigning an 802.1p priority to a specific DSCP. When the switch applies this policy to a packet, the priority determines the packet's queue in the outbound port to which it is sent. If the packet leaves the switch on a tagged port, it carries the 802.1p priority with it to the next downstream device. For IPv4 packets, the DSCP will be replaced by the codepoint specified in this command. (Default:* **No-override** *for most codepoints. See table 8-14 on page 8-65.)*

Syntax: qos < udp-port | tcp-port > < *tcp or udp port number* > dscp < *codepoint* >

> *Assigns a DSCP policy to outbound packets having the specified TCP or UDP application port number and overwrites the DSCP in these packets with the assigned* **<codepoint>** *value. This policy includes an 802.1p priority and determines the packet's queue in the outbound port to which it is sent. (The* **<codepoint>** *must be configured with an 802.1p setting. See step 3 on page 8-28.) If the packet leaves the switch on a tagged port, it carries the 802.1p priority with it to the next downstream device. (Default:* **No-override***)*

no qos < udp-port | tcp-port > < **tcp-udp port number** >

> *Deletes the specified UDP or TCP port number as a QoS classifier.*

show qos tcp-udp-port-priority

> *Displays a listing of all TCP and UDP QoS classifiers currently in the running-config file.*

For example, suppose you wanted to assign these DSCP policies to the packets identified by the indicated UDP and TDP port applications:

Port Applications	DSCP Policies	
	DSCP	Priority
23-UDP	000111	7
80-TCP	000101	5
914-TCP	000010	1
1001-UDP	000010	1

1. Determine whether the DSCPs already have priority assignments, which could indicate use by existing applications. (Also, a DSCP must have a priority configured before you can assign any QoS classifiers to use it.)

```
ProCurve(config)# show qos dscp-map
  DSCP -> 802.p priority mappings
  DSCP policy 802.1p tag  Policy name
  ----------- ----------- -------------------------------
  000000      No-override
  000001      No-override
  000010      No-override
  000011      No-override
  000100      No-override
  000101      No-override
  000110      No-override
  000111      No-override
     .            .
     .            .
     .            .
```

The DSCPs for this example have not yet been assigned an 802.1p priority level.

Figure 8-9. Display the Current DSCP-Map Configuration

2. Configure the DSCP policies for the codepoints you want to use.

```
ProCurve(config)# qos dscp-map 000111 priority 7
ProCurve(config)# qos dscp-map 000101 priority 5
ProCurve(config)# qos dscp-map 000010 priority 1
ProCurve(config)# show qos dscp-map
  DSCP -> 802.p priority mappings
  DSCP policy 802.1p tag  Policy name
  ----------- ----------- -------------
  000000      No-override
  000001      No-override
  000010      1
  000011      No-override
  000100      No-override
  000101      5
  000110      No-override
  000111      7
  001000      No-override
     .            .
     .            .
     .            .
```

DSCP Policies Configured in this Step

Figure 8-10. Assign Priorities to the Selected DSCPs

3. Assign the DSCP policies to the selected UDP/TCP port applications and display the result.

```
ProCurve(config)# qos udp-port 23 dscp 000111
ProCurve(config)# qos tcp-port 80 dscp 000101
ProCurve(config)# qos tcp-port 914 dscp 000010
ProCurve(config)# qos udp-port 1001 dscp 000010

ProCurve(config)# show qos tcp-udp-port-priority

  TCP/UDP port based priorities

            Application
  Protocol |   port  Apply rule|  DSCP   Priority
  -------- + -------- ----------+ ------ -----------
  UDP      | 23        DSCP     | 000111 7
  TCP      | 80        DSCP     | 000101 5
  TCP      | 914       DSCP     | 000010 1
  UDP      | 1001      DSCP     | 000010 1
```

Classifier   DSCP Policy

Figure 8-11. The Completed DSCP Policy Configuration for the Specified UDP/TCP Port Applications

The switch will now apply the DSCP policies in figure 8-11 to IPV4 packets received in the switch with the specified UDP/TCP port applications. This means the switch will:

■ Overwrite the original DSCPs in the selected packets with the new DSCPs specified in the above policies.

■ Assign the 802.1p priorities in the above policies to the selected packets.

QoS IP-Device Priority

QoS Classifier Precedence: 2

The IP device option, which applies only to IPv4 packets, enables you to use up to the following IP address limits (source or destination) as QoS classifiers:

■ 5300xl Switches: 256 IP addresses

■ 4200vl Switches: 256 IP addresses

■ 3400cl/6400cl Switches: 60 IP addresses

Where a particular device-IP address classifier has the highest precedence in the switch for traffic addressed to or from that device, then traffic received on the switch with that address is marked with the IP address classifier's configured priority level. Different IP device classifiers can have differing priority levels.

N o t e   The switch does not allow a QoS IP-device priority for the Management VLAN
IP address, if configured. If there is no Management VLAN configured, then
the switch does not allow configuring a QoS IP-device priority for the Default
VLAN IP address.

Ip address QoS does not support layer-2 SAP encapsulation. For more infor-
mation on packet-type restrictions, refer to table 8-15, "Details of Packet
Criteria and Restrictions for QoS Support", on page 8-73.

Options for Assigning Priority. Priority control options for packets carry-
ing a specified IP address include:

■   802.1p priority

■   DSCP policy (Assigning a new DSCP and an 802.1p priority; inbound
     packets must be IPv4.)

(For operation when other QoS classifiers apply to the same traffic, refer to
"Classifiers for Prioritizing Outbound Packets" on page 8-10.)

For a given IP address, you can use only one of the above options at a time.
However, for different IP addresses, you can use different options.

Assigning a Priority Based on IP Address

This option assigns an 802.1p priority to all IPv4 packets having the specified
IP address as either a source or destination. (If both match, the priority for
the IP destination address has precedence.)

Syntax: qos device-priority < *ip-address* > priority < 0 - 7 >

> *Configures an 802.1p priority for outbound packets
> having the specified IP address. This priority deter-
> mines the packet's queue in the outbound port to which
> it is sent. If the packet leaves the switch on a tagged
> port, it carries the 802.1p priority with it to the next
> downstream device. (Default: Disabled)*

no qos device-priority < *ip-address* >

> *Removes the specified IP device-priority QoS classifier
> and resets the priority for that VLAN to* **No-override**.

show qos device-priority

> *Displays a listing of all IP device-priority QoS classi-
> fiers currently in the running-config file.*

For example, configure and list the 802.1p priority for packets carrying the following IP addresses:

IP Address	802.1p Priority
10.28.31.1	7
10.28.31.130	5
10.28.31.100	1
10.28.31.101	1

```
ProCurve(config)# qos device-priority 10.28.31.1 priority 7
ProCurve(config)# qos device-priority 10.28.31.130 priority 5
ProCurve(config)# qos device-priority 10.28.31.100 priority 1
ProCurve(config)# qos device-priority 10.28.31.101 priority 1

ProCurve(config)# show qos device-priority
  Device priorities
  Device Address Apply rule | DSCP    Priority
  -------------- ---------- + ------ -----------
  10.28.31.1     Priority   |            7
  10.28.31.130   Priority   |            5
  10.28.31.100   Priority   |            1
  10.28.31.101   Priority   |            1
```

Figure 8-12. Example of Configuring and Listing 802.1p Priority Assignments for Packets Carrying Specific IP Addresses

Assigning a DSCP Policy Based on IP Address

N o t e   On 5300xl and 4200vl switches, DSCP policies cannot be applied to IPv4 packets having IP options. For more information on packet criteria and restrictions, refer to table 8-15 on page 8-73.

This option assigns a previously configured DSCP policy (codepoint and 802.1p priority) to outbound IP packets having the specified IP address (either source or destination). That is, the switch:

1. Selects an incoming IPv4 packet on the basis of the source or destination IP address it carries.

2. Overwrites the packet's DSCP with the DSCP configured in the switch for such packets, and assigns the 802.1p priority configured in the switch for the new DSCP. (Refer to "Differentiated Services Codepoint (DSCP) Mapping" on page 8-64.)

3. Forwards the packet through the appropriate outbound port queue.

3400cl/6400cl Switch Restriction. On the 3400cl/6400cl switches, "mixing" ToS DSCP policies and 802.1p priorities is not recommended. Refer to the Note on page 8-11.

For more on DSCP, refer to "Terminology" on page 8-6.

Steps for Creating a Policy Based on IP Address. This procedure creates a DSCP policy for IPv4 packets carrying the selected IP address (source or destination).

1. Identify the IP address to use as a classifier for assigning a DSCP policy.

2. Determine the DSCP policy for packets carrying the selected IP address:

   a. Determine the DSCP you want to assign to the selected packets. (This codepoint will be used to overwrite the DSCP carried in packets received from upstream devices.)

   b. Determine the 802.1p priority you want to assign to the DSCP.

3. Configure the DSCP policy by using **dscp-map** to configure the priority to the codepoint you selected in step 2a. (For details, refer to "Differentiated Services Codepoint (DSCP) Mapping" on page 8-64.)

N o t e s   A codepoint must have an 802.1p priority assignment (0 - 7) before you can configure a policy for prioritizing packets by IP address. If a codepoint you want to use shows **No-override** in the **Priority** column of the DSCP map (**show qos dscp-map**), then you must assign a 0 - 7 priority before proceeding.

On 5300xl and 4200vl switches, DSCP policies cannot be applied to IPv4 packets having IP options. For more information on packet criteria and restrictions, refer to 8-15 on page 8-73.

---

4. Configure the switch to assign the DSCP policy to packets with the specified IP address.

Syntax: qos dscp-map < *codepoint* > priority < 0 - 7 >

> *This command is optional if a priority is already assigned to the < **codepoint** >. The command creates a DSCP policy by assigning an 802.1p priority to a specific DSCP. When the switch applies this policy to a packet, the priority determines the packet's queue in the outbound port to which it is sent. If the packet leaves the switch on a tagged port, it carries the 802.1p priority with it to the next downstream device. If the packet is IPv4, the packet's DSCP will be replaced by the codepoint specified in this command. (Default: For most codepoints, **No-override**. See figure 8-14 on page 8-65.)*

Syntax: qos device-priority < *ip-address* > dscp < *codepoint* >

> *Assigns a DSCP policy to packets carrying the specified IP address, and overwrites the DSCP in these packets with the assigned < codepoint > value. This policy includes an 802.1p priority and determines the packet's queue in the outbound port to which it is sent. If the packet leaves the switch on a tagged port, it carries the 802.1p priority with it to the next downstream device. (Default: No-override)*

no qos device-priority < *ip-address* >

> *Deletes the specified IP address as a QoS classifier.*

show qos device-priority

> *Displays a listing of all QoS Device Priority classifiers currently in the running-config file.*

For example, suppose you wanted to assign these DSCP policies to the packets identified by the indicated IP addresses:

IP Address	DSCP Policies	
	DSCP	Priority
10.28.31.1	000111	7
10.28.31.130	000101	5
10.28.31.100	000010	1
10.28.31.101	000010	1

1. Determine whether the DSCPs already have priority assignments, which could indicate use by existing applications. This is not a problem if the configured priorities are acceptable for all applications using the same DSCP. (Refer to the "Note On Changing a Priority Setting" on page 8-67. Also, a DSCP must have a priority configured before you can assign any QoS classifiers to use it.

)

```
ProCurve(config)# show qos dscp-map
  DSCP -> 802.p priority mappings
  DSCP policy 802.1p tag  Policy name
  ----------- ----------- -----------------------------
  000000      No-override
  000001      No-override
  000010      No-override)  ←  The DSCPs for this
  000011      No-override         example have not yet
  000100      No-override         been assigned an
  000101      No-override)        802.1p priority level.
  000110      No-override
  000111      No-override)
    :           :
    :           :
    :           :
```

Figure 8-13. Display the Current DSCP-Map Configuration

2. Configure the priorities for the DSCPs you want to use.

```
ProCurve(config)# qos dscp-map 000111 priority 7
ProCurve(config)# qos dscp-map 000101 priority 5
ProCurve(config)# qos dscp-map 000010 priority 1
ProCurve(config)# show qos dscp-map

  DSCP -> 802.p priority mappings
  DSCP policy 802.1p tag  Policy name
  ----------- ----------- -------------
  000000      No-override
  000001      No-override
  000010      1
  000011      No-override
  000100      No-override                  DSCP Policies
  000101      5                            Configured in this step.
  000110      No-override
  000111      7
  001000      No-override
     :           :
     :           :
     :           :
```

Figure 8-14. Assigning 802.1p Priorities to the Selected DSCPs

3. Assign the DSCP policies to the selected device IP addresses and display
    the result.

```
ProCurve(config)# qos device-priority 10.28.31.1 dscp 000111
ProCurve(config)# qos device-priority 10.28.31.130 dscp 000101
ProCurve(config)# qos device-priority 10.28.31.100 dscp 000010
ProCurve(config)# qos device-priority 10.28.31.101 dscp 000010

ProCurve(config)# show qos device-priority
  Device priorities

  Device Address Apply rule | DSCP    Priority
  -------------- ---------- + ------  -----------
  10.28.31.1     DSCP       | 000111 7
  10.28.31.130   DSCP       | 000101 5
  10.28.31.100   DSCP       | 000010 1
  10.28.31.101   DSCP       | 000010 1
```

Figure 8-15. The Completed Device-Priority/Codepoint Configuration

The switch will now apply the DSCP policies in figure 8-14 to IPv4 packets
received on the switch with the specified IP addresses (source or destination).
This means the switch will:

■ Overwrite the original DSCPs in the selected packets with the new DSCPs
   specified in the above policies.

■ Assign the 802.1p priorities in the above policies to the appropriate
   packets.

QoS IP Type-of-Service (ToS) Policy and Priority

QoS Classifier Precedence: 3

This feature applies only to IPv4 traffic and performs either of the following:

- **ToS IP-Precedence Mode:** All IP packets generated by upstream devices and applications include precedence bits in the ToS byte. Using this mode, the switch uses these bits to compute and assign the corresponding 802.1p priority.

- **ToS Differentiated Services (Diffserv) Mode:** This mode requires knowledge of the codepoints set in IP packets by the upstream devices and applications. It uses the ToS codepoint in IP packets coming from upstream devices and applications to assign 802.1p priorities to the packets. You can use this option to do both of the following:

  - **Assign a New Prioritization Policy:** A "policy" includes both a codepoint and a corresponding 802.1p priority. This option selects an incoming IPv4 packet on the basis of its codepoint and assigns a new codepoint and corresponding 802.1p priority. (Use the **qos dscp-map** command to specify a priority for any codepoint—page 8-64.)

  - **Assign an 802.1p Priority:** This option reads the DSCP of an incoming IPv4 packet and, without changing this codepoint, assigns the 802.1p priority to the packet, as configured in the DSCP Policy Table (page 8-64). This means that a priority value of 0 - 7 must be configured for a DSCP before the switch will attempt to perform a QoS match on the packet's DSCP bits.

  Before configuring the ToS Diffserv mode, you must use the **dscp-map** command to configure the desired 802.1p priorities for the codepoints you want to use for either option. This command is illustrated in the following examples and is described under "Differentiated Services Codepoint (DSCP) Mapping" on page 8-64.

Unless IP-Precedence mode and Diffserv mode are both disabled (the default setting), enabling one automatically disables the other. *For more on ToS operation, refer to "Details of QoS IP Type-of-Service" on page 8-47.*

3400cl/6400cl Switch Restriction. On the 3400cl/6400cl switches, "mixing" ToS DSCP policies and 802.1p priorities is not recommended. Refer to the Note on page 8-11.

Assigning an 802.1p Priority to IPv4 Packets on the Basis of the ToS Precedence Bits

If a device or application upstream of the switch sets the precedence bits in the ToS byte of IPv4 packets, you can use this feature to apply that setting for prioritizing packets for outbound port queues. If the outbound packets are in a tagged VLAN, this priority is carried as an 802.1p value to the adjacent downstream devices.

Syntax: qos type-of-service ip-precedence

> *Causes the switch to automatically assign an 802.1p priority to all IPv4 packets by computing each packet's 802.1p priority from the precedence bits the packet carries. This priority determines the packet's queue in the outbound port to which it is sent. If the packet leaves the switch on a tagged port, it carries the 802.1p priority with it to the next downstream device. (ToS IP Precedence Default: Disabled)*

no qos type-of-service
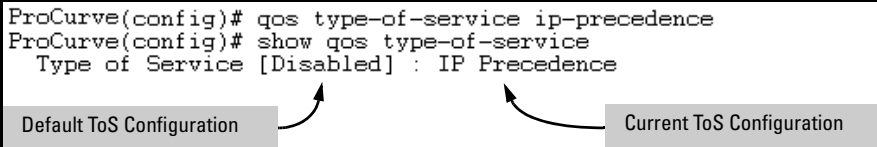
> *Disables all ToS classifier operation, including prioritization using the precedence bits.*

show qos type-of-service

> *When ip-precedence is enabled (or if neither ToS option is configured), shows the ToS configuration status. If diff-services is enabled, lists codepoint data as described under "Assigning a DSCP Policy on the Basis of the DSCP in IPv4 Packets Received from Upstream Devices" on page 8-43.*

With this option, prioritization of outbound packets relies on the IP-Precedence bit setting that IP packets carry with them from upstream devices and applications. To configure and verify this option:

```
ProCurve(config)# qos type-of-service ip-precedence
ProCurve(config)# show qos type-of-service
  Type of Service [Disabled] : IP Precedence
```
Default ToS Configuration   Current ToS Configuration

Figure 8-16. Example of Enabling ToS IP-Precedence Prioritization

To replace this option with the ToS diff-services option, just configure **diff-services** as described below, which automatically disables IP-Precedence. To disable IP-Precedence without enabling the diff-services option, use this command:

```
ProCurve(config)# no qos type-of-service
```

Assigning an 802.1p Priority to IPv4 Packets on the Basis of Incoming DSCP

One of the best uses for this option is on an interior switch where you want to honor (continue) a policy set on an edge switch. That is, it enables you to select incoming packets having a specific DSCP and forward these packets with the desired 802.1p priority. For example, if an edge switch "A" marks all packets received on port A5 with a particular DSCP, you can configure a downstream (interior) switch "B" to handle such packets with the desired priority (regardless of whether 802.1Q tagged VLANs are in use).
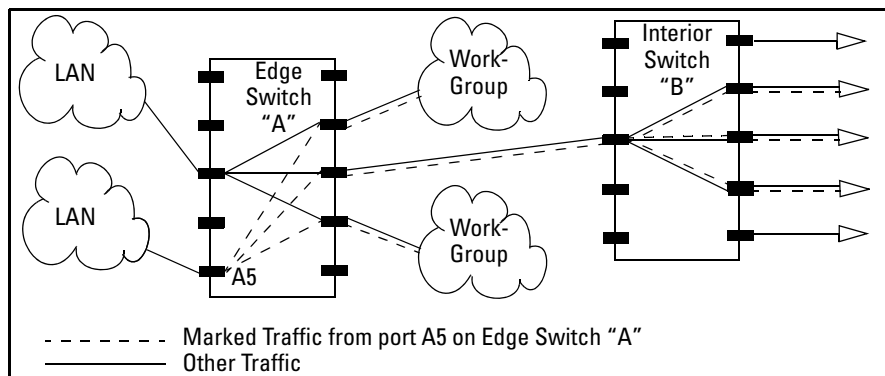
Figure 8-17. Interior Switch "B" Honors the Policy Established in Edge Switch "A"

To do so, assign the desired 802.1p priority to the same codepoint that the upstream or edge switch assigns to the selected packets. When the down-stream switch receives an IPv4 packet carrying one of these codepoints, it assigns the configured priority to the packet and sends it out the appropriate priority queue. (The packet retains the codepoint it received from the upstream or edge switch). You can use this option concurrently with the diffserv DSCP Policy option (described later in this section), as long as the DSCPs specified in the two options do not match.

Operating Notes   Different applications may use the same DSCP in their IP packets. Also, the same application may use multiple DSCPs if the application originates on different clients, servers, or other devices. Using an edge switch enables you to select the packets you want and mark them with predictable DSCPs that can be used by downstream switches to honor policies set in the edge switch.

When enabled, the switch applies direct 802.1p prioritization to all packets having codepoints that meet these criteria:

■ The codepoint is configured with an 802.1p priority in the DSCP table. (Codepoints configured with **No-override** are not used.)

■ The codepoint is not configured for a new DSCP policy assignment.

Thus, the switch does not allow the same incoming codepoint (DSCP) to be used simultaneously for directly assigning an 802.1p priority and also assigning a DSCP policy. For a given incoming codepoint, if you configure one option and then the other, the second overwrites the first.

To use this option:

1. Identify a DSCP used to set a policy in packets received from an upstream or edge switch.

2. Determine the 802.1p priority (0 - 7) you want to apply to packets carrying the identified DSCP. (You can either maintain the priority assigned in the upstream or edge switch, or assign a new priority.)

3. Use **qos dscp-map < _codepoint_ > priority < 0 - 7 >** to assign the 802.1p priority you want to the specified DSCP. (For more on this topic, refer to "Differentiated Services Codepoint (DSCP) Mapping" on page 8-64.)

4. Enable **diff-services**

.

Syntax: qos type-of-service diff-services < _codepoint_ >

> _Causes the switch to read the_ **< _codepoint_ >** (_DSCP) of an incoming IPv4 packet and, when a match occurs, assign a corresponding 802.1p priority, as configured in the switch's DSCP table (page 8-65)._

no qos type-of-service

> _Disables all ToS classifier operation._

no qos dscp-map < *codepoint* >

> *Disables direct 802.1p priority assignment to packets carrying the <* **codepoint** *> by reconfiguring the codepoint priority assignment in the DSCP table to* **No-override**. *Note that if this codepoint is in use as a DSCP policy for another diffserv codepoint, you must disable or redirect the other diffserv codepoint's DSCP policy before you can disable or change the codepoint. For example, in figure 8-18 you cannot change the priority for the 000000 codepoint until you redirect the DSCP policy for 000001 away from using 000000 as a policy. (Refer to "Note On Changing a Priority Setting" on page 8-67. Refer also to "Differentiated Services Codepoint (DSCP) Mapping" on page 8-64.)*

show qos type-of-service

> *Displays current Type-of-Service configuration. In diffserv mode it also shows the current direct 802.1p assignments and the current DSCP assignments covered later in this section.*

For example, an edge switch "A" in an untagged VLAN assigns a DSCP of 000110 on IP packets it receives on port A6, and handles the packets with high priority (7). When these packets reach interior switch "B" you want the switch to handle them with the same high priority. To enable this operation you would

configure an 802.1p priority of 7 for packets received with a DSCP of **000110**, and then enable **diff-services**:

```
ProCurve(config)# show qos type-of-service
 Type of Service [Disabled] : Disabled

 Codepoint DSCP Policy | Priority
 --------- ----------- + ----------
 000000                | 1
 000001    000000      | 1
 000010                | No-override
 000011                | No-override
 000100    001001      | 5
 000101                | No-override
 000110                | No-override
 000111                | No-override
 001000                | No-override
 001001                | 5
 001010                | 1
 001011                | No-override
    •          •            •
    •          •            •
    •          •            •
```

Executing this command displays the current ToS configuration and shows that the selected DSCP is not currently in use.

The **000110** codepoint is unused, and thus available for directly assigning an 802.1p priority without changing the packet's DSCP.

Note: All codepoints without a "DSCP Policy" entry are available for direct 802.1p priority assignment.

Figure 8-18. Example Showing Codepoints Available for Direct 802.1p Priority Assignments

```
ProCurve(config)# qos dscp-map 000110 priority 7
ProCurve(config)# qos type-of-service diff-services

ProCurve(config)# show qos type-of-service
  Type of Service [Disabled] : Differentiated Services

  Codepoint DSCP Policy | Priority
  --------- ----------- + ----------
  000000                | 1
  000001    000000      | 1
  000010                | No-override
  000011                | No-override
  000100    001001      | 5
  000101                | No-override
  000110                | 7
  000111                | No-override
  001000                | No-override
  001001                | 5
     •          •            •
     •          •            •
     •          •            •
```

Outbound IP packets with a DSCP of **000110** will have a priority of **7**.

Notice that codepoints **000000** and **001001** are named as DSCP policies by other codepoints (**000001** and **000110** respectively). This means they are not available for changing to a different 802.1p priority.

Figure 8-19. Example of a Type-of-Service Configuration Enabling Both Direct 802.1p Priority Assignment and DSCP Policy Assignment

Assigning a DSCP Policy on the Basis of the DSCP in IPv4 Packets Received from Upstream Devices

The preceding section describes how to forward a policy set by an edge (or upstream) switch. This option changes a DSCP policy in an IPv4 packet by changing its IP ToS codepoint and applying the priority associated with the new codepoint. (A DSCP policy consists of a differentiated services codepoint and an associated 802.1p priority.) You can use this option concurrently with the diffserv 802.1p priority option (above), as long as the DSCPs specified in the two options do not match.

To use this option to configure a change in policy:

1. Identify a DSCP used to set a policy in packets received from an upstream or edge switch.

2. Create a new policy by using **qos dscp-map < *codepoint* > priority < 0 - 7 >** to configure an 802.1p priority for the codepoint you will use to overwrite the DSCP the packet carries from upstream. (For more on this topic, refer to "Differentiated Services Codepoint (DSCP) Mapping" on page 8-64.)

3. Use **qos type-of-service diff-services < *incoming-DSCP* > dscp < *outgoing-DSCP* >** to change the policy on packets coming from the edge or upstream switch with the specified incoming DSCP.

    (Figure 8-17 on page 8-39 illustrates this scenario.)

N o t e   On 5300xl and 4200vl switches, DSCP policies (codepoint re-marking) cannot be applied to outbound IPv4 packets having IP options. (The 802.1p priority in the VLAN tag is applied.) For more information on packet criteria and restrictions, refer to 8-15 on page 8-73.

3400cl/6400cl Switch Restriction. On the 3400cl/6400cl switches, "mixing" ToS DSCP policies and 802.1p priorities is not recommended. Refer to the Note on page 8-11.

Syntax: qos type-of-service diff-services

Enables ToS diff-services.

Syntax: qos type-of-service diff-services < *current-codepoint* > dscp
< *new-codepoint* >

> *Configures the switch to select an incoming IP packet carrying the <**current-codepoint**> and then use the <**new-codepoint**> to assign a new, previously configured DSCP policy to the packet. The policy overwrites the <**current-codepoint**> with the < **new-codepoint** > and assigns the 802.1p priority specified by the policy. (Use the* **qos dscp-map** *command to define the priority for the DSCPs—page 8-64.)*

Syntax: no qos type-of-service

> *Disables all ToS classifier operation. Current ToS DSCP policies and priorities remain in the configuration and will become available if you re-enable ToS diff-services.*

Syntax: no qos type-of-service [diff-services < *codepoint* >]

> *Deletes the DSCP policy assigned to the < **codepoint** > and returns the < **codepoint** > to the 802.1p priority setting it had before the DSCP policy was assigned. (This will be either a value from 0 - 7 or* **No-override**.)

Syntax: show qos type-of-service
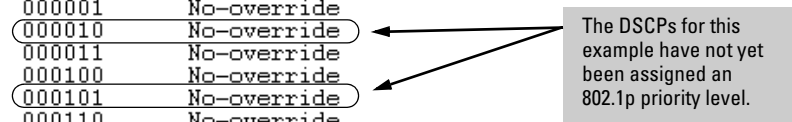
> *Displays a listing of codepoints, with any corresponding DSCP policy re-assignments for outbound packets. Also lists the (802.1p) priority for each codepoint that does not have a DSCP policy assigned to it.*

For example, suppose you want to configure the following two DSCP policies for packets received with the indicated DSCPs.

Received DSCP	Policy DSCP	802.1p Priority	Policy Name (Optional)
001100	000010	6	Level 6
001101	000101	4	Level 4

1. Determine whether the DSCPs already have priority assignments, which could indicate use by existing applications. This is not a problem as long as the configured priorities are acceptable for all applications using the same DSCP. (Refer to the "Note On Changing a Priority Setting" on page 8-67. Also, a DSCP must have a priority configured before you can assign any QoS classifiers to use it.)

```
ProCurve(config)# show qos dscp-map
  DSCP -> 802.p priority mappings
  DSCP policy 802.1p tag  Policy name
  ----------- ----------- -------------------------------
  000000      No-override
  000001      No-override
  000010      No-override
  000011      No-override
  000100      No-override
  000101      No-override
  000110      No-override
  000111      No-override
     .            .
     .            .
     .            .
```

The DSCPs for this example have not yet been assigned an 802.1p priority level.

Figure 8-20. Display the Current DSCP-Map Configuration

2. Configure the policies in the DSCP table:

```
ProCurve(config)# qos dscp-map 000010 priority 6 name 'Level 6'
ProCurve(config)# qos dscp-map 000101 priority 4 name 'Level 4'

ProCurve(config)# show qos dscp-map
  DSCP -> 802.p priority mappings
  DSCP policy 802.1p tag  Policy name
  ----------- ----------- -------------------------------
  000000      No-override
  000001      No-override
  000010      6           Level 6
  000011      No-override
  000100      No-override
  000101      4           Level 4
  000110      No-override
  000111      No-override
     .            .          .
     .            .          .
     .            .          .
```

Figure 8-21. Example of Policies Configured (with Optional Names) in the DSCP Table

3. Assign the policies to the codepoints in the selected packet types.

```
ProCurve(config)# qos type-of-service diff-services 001100 dscp 000010
ProCurve(config)# qos type-of-service diff-services 001101 dscp 000101

ProCurve(config)# show qos type-of-service
  Type of Service [Disabled] : Differentiated Services

  Codepoint DSCP Policy | Priority
  --------- ----------- + -----------
  000000                | No-override
  000001                | No-override
  000010                | 6
  000011                | No-override
  000100                | No-override
  000101                | 4
  000110                | No-override
  000111                | No-override
  001000                | No-override
  001001                | No-override
  001010                | 1
  001011                | No-override
  001100    000010      | 6
  001101    000101      | 4
  001110                | 2
  001111                | No-override
  010000                | No-override
  010001                | No-override
-- MORE --, next page: Space, next line: Enter, quit: Control-C
```

The specified DSCP policies overwrite the original DSCPs on the selected packets, and use the 802.1p priorities previously configured in the DSCP policies in step 2.

Figure 8-22. Example of Policy Assignment to Outbound Packets on the Basis of the DSCP in the Packets Received from Upstream Devices

Details of QoS IP Type-of-Service

IP packets include a Type of Service (ToS) byte. The ToS byte includes:

- **A Differentiated Services Codepoint (DSCP):** This element is comprised of the upper six bits of the ToS byte). There are 64 possible codepoints.

  - In the 5300xl and 4200vl switches, the default **qos** configuration includes some codepoints with 802.1p priority settings for Assured-Forwarding and Expedited Forwarding (codepoint 101110), while others are unused (and listed with **No-override** for a Priority).

  - In the 3400cl/6400cl switches, the default **qos** configuration includes the codepoint (7) having the 802.1p priority setting for Expedited Forwarding, while all others, including the Assured-Forwarding codepoints, are unused (and listed with **No-override** for a Priority).

  Refer to figure 8-14 on page 8-65 for an illustration of the default DSCP policy table.

  Using the **qos dscp map** command, you can configure the switch to assign different prioritization policies to IPv4 packets having different codepoints. As an alternative, you can configure the switch to assign a new codepoint to an IPv4 packet, along with a corresponding 802.1p priority (0-7). To use this option in the simplest case, you would:

  a. Configure a specific DSCP with a specific priority in an edge switch.
  b. Configure the switch to mark a specific type of inbound traffic with that DSCP (and thus create a policy for that traffic type).
  c. Configure the internal switches in your LAN to honor the policy.

  (For example, you could configure an edge switch to assign a codepoint of 000001 to all packets received from a specific VLAN, and then handle all traffic with that codepoint at high priority.)

  For a codepoint listing and the commands for displaying and changing the DSCP Policy table, refer to "Differentiated Services Codepoint (DSCP) Mapping" on page 8-64.

  Restriction: On the 3400cl/6400cl switches, "mixing" ToS DSCP policies and 802.1p priorities is not recommended. Refer to the Note on page 8-11.

- **Precedence Bits:** This element is a subset of the DSCP and is comprised of the upper three bits of the ToS byte. When configured to do so, the switch uses the precedence bits to determine a priority for handling the associated packet. (The switch does not change the setting of the precedence bits.) Using the ToS Precedence bits to prioritize IPv4 packets relies on priorities set in upstream devices and applications.

Figure 8-23 shows an example of the ToS byte in the header for an IPv4 packet, and illustrates the diffserv bits and precedence bits in the ToS byte. (Note that the Precedence bits are a subset of the Differentiated Services bits.)

Field:	Destination MAC Address	Source MAC Address	802.1Q Field	Type & Version	ToS Byte	. . .	
Packet:	FF FF FF FF FF FF	08 00 09 00 00 16	08 00	45	**E 0**	...	

	Differentiated Services Codepoint							
	Precedence Bits						Rsvd.	
1	1	1	0	0	0		0	0
	E				0			
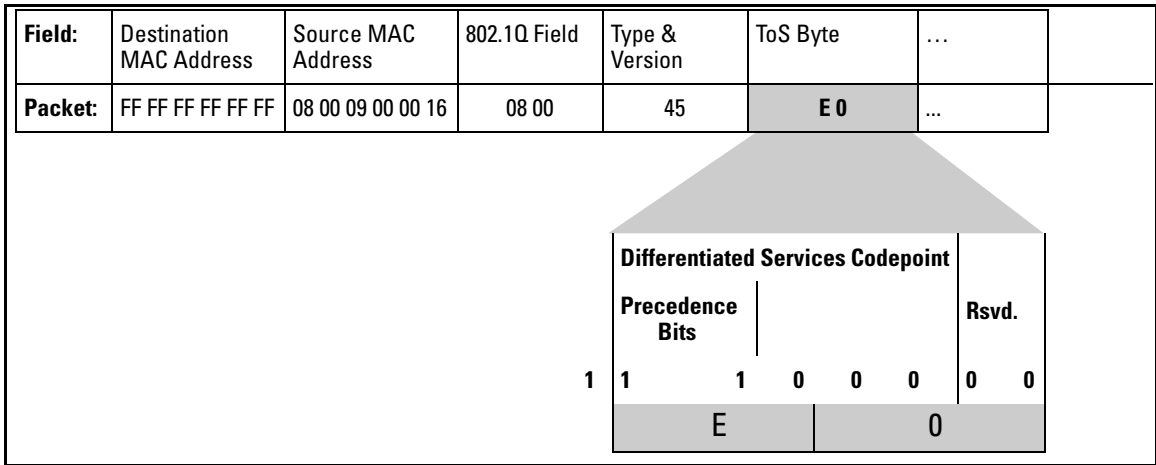
Figure 8-23. The ToS Codepoint and Precedence Bits

Table 8-12.   How the Switch Uses the ToS Configuration

Outbound Port	ToS Option:	
	IP Precedence (Value = 0 - 7)	**Differentiated Services**
IP Packet Sent Out an Untagged Port in a VLAN	Depending on the value of the IP Precedence bits in the packet's ToS field, the packet will go to one of four outbound port queues in the switch: 1 - 2 = low priority 0 - 3 = normal priority 4 - 5 = high priority 6 - 7 = high priority	For a given packet carrying a ToS codepoint that the switch has been configured to detect: • Change the codepoint according to the configured policy and assign the 802.1p priority specified for the new codepoint in the DSCP Policy Table (page 8-64). • Do not change the codepoint, but assign the 802.1p priority specified for the existing codepoint in the DSCP Policy Table (page 8-64). Depending on the 802.1p priority used, the packet will leave the switch through one of the following queues: 1 - 2 = low priority 0 - 3 = normal priority 4 - 5 = high priority 6 - 7 = high priority If **No-override** (the default) has been configured for a specified codepoint, then the packet is not prioritized by ToS and, by default, is sent to the "normal priority" queue.

Outbound Port	ToS Option:	
	IP Precedence (Value = 0 - 7)	**Differentiated Services**
IP Packet Sent Out an Untagged Port in a VLAN	Same as above, plus the IP Precedence value (0 - 7) will be used to set a corresponding 802.1p priority in the VLAN tag carried by the packet to the next downstream device. Refer to table 8-13, below.	Same as above, plus the Priority value (0 - 7) will be used to set a corresponding 802.1p priority in the VLAN tag carried by the packet to the next downstream device. Where **No-override** is the assigned priority, the VLAN tag carries a "0" (normal priority) 802.1p setting if not prioritized by other QoS classifiers.

Table 8-13.   ToS IP-Precedence Bit Mappings to 802.1p Priorities

ToS Byte IP Precedence Bits	Corresponding 802.1p Priority	Service Priority Level
000	1	Lowest
001	2	Low
002	0	Normal
003	3	
004	4	
005	5	
006	6	
007	7	Highest

QoS Layer-3 Protocol Priority (5300xl and 4200vl Switches Only)

(This feature is available only on the Series 5300xl and 4200vl switches.)

QoS Classifier Precedence: 4

The QoS protocol option enables you to use these protocols as QoS classifiers:

- IP
- ARP
- Appletalk
- Netbeui
- IPX
- DEC_LAT
- SNA

Options for Assigning Priority. Priority control for the Layer-3 protocol classifier includes assigning only the 802.1p priority. The switch does not use this classifier for assigning DSCP-based priority.

Assigning a Priority Based on Layer-3 Protocol

When QoS on the switch is configured with a Layer-3 protocol as the highest-precedence classifier and the switch receives traffic carrying that protocol, then this traffic is assigned the priority configured for this classifier. (For operation when other QoS classifiers apply to the same traffic, refer to "Classifiers for Prioritizing Outbound Packets" on page 8-10.)

Syntax: qos protocol

< ip | ipx | arp | dec_lat | appletalk | sna | netbeui > priority < 0 - 7 >

Configures an 802.1p priority for outbound packets having the specified protocol. This priority determines the packet's queue in the outbound port to which it is sent. If the packet leaves the switch on a tagged port, it carries the 802.1p priority with it to the next downstream device. You can configure one QoS classifier for each protocol type. (Default: **No-override***)*

no qos protocol

< ip | ipx | arp | dec_lat | appletalk | sna | netbeui >

Disables use of the specified protocol as a QoS classifier and resets the protocol priority to **No-override***.*

show qos protocol

Lists the QoS protocol classifiers with their priority settings.

For example:

1. Configure QoS protocol classifiers with IP at 0 (normal), ARP at 5 (medium), and AppleTalk at 7 (high) and display the QoS protocol configuration.

2. Disable the QoS IP protocol classifier, downgrade the ARP priority to 4, and again display the QoS protocol configuration.

Figure 8-24 shows the command sequence and displays for the above steps.

```
ProCurve(config)# qos protocol ip priority 0
ProCurve(config)# qos protocol appletalk priority 7          Configures IP, Appletalk,
ProCurve(config)# qos protocol arp priority 5                and ARP as QoS classifiers.

ProCurve(config)# show qos protocol                          Displays the result of the
                                                             above commands.
   Protocol priorities

   Protocol  Priority
   --------  ----------
   IP        0
   IPX       No-override
   ARP       5
   DEC_LAT   No-override
   AppleTalk 7
   SNA       No-override
   NetBEUI   No-override

ProCurve(config)# no qos protocol ip                         Removes IP as a QoS
ProCurve(config)# qos protocol arp priority 4                classifier.

ProCurve(config)# show qos protocol                          Changes the priority of the
                                                             ARP QoS classifier.
   Protocol priorities
                                                             Displays the result of these
   Protocol  Priority                                        changes.
   --------  ----------
   IP        No-override
   IPX       No-override
   ARP       4
   DEC_LAT   No-override
   AppleTalk 7
   SNA       No-override
   NetBEUI   No-override
```

Figure 8-24. Adding, Displaying, Removing, and Changing QoS Protocol Classifiers

QoS VLAN-ID (VID) Priority

QoS Classifier Precedence: 5

The QoS protocol option enables you to use the VLAN-ID quantities listed below as QoS classifiers.

- 5300xl Switches: Up to 256 VIDs
- 4200vl Switches: Up to 256 VIDs
- 3400cl/6400cl Switches: Up to 120 VIDs

Where a particular VLAN-ID classifier has the highest precedence in the switch for traffic in that VLAN, then traffic received in that VLAN is marked with the VID classifier's configured priority level. Different VLAN-ID classifiers can have differing priority levels.

Options for Assigning Priority. Priority control options for packets carrying a specified VLAN-ID include:

- 802.1p priority
- DSCP policy (Assigning a new DSCP and an associated 802.1p priority; inbound packets must be IPv4.)

(For operation when other QoS classifiers apply to the same traffic, refer to "Classifiers for Prioritizing Outbound Packets" on page 8-10.)

N o t e   QoS with VID priority applies to static VLANs only, and applying QoS to dynamic VLANs created by GVRP operation is not supported. A VLAN must exist while a subject of a QoS configuration, and eliminating a VLAN from the switch causes the switch to clear any QoS features configured for that VID.

Assigning a Priority Based on VLAN-ID

This option assigns a priority to all outbound packets having the specified VLAN-ID (VID). You can configure this option by either specifying the VLAN-ID ahead of the **qos** command or moving to the VLAN context for the VLAN you want to configure for priority.

Syntax: vlan < *vid* > qos priority < 0 - 7 >

> *Configures an 802.1p priority for outbound packets belong-
> ing to the specified VLAN. This priority determines the
> packet's queue in the outbound port to which it is sent. If the
> packet leaves the switch on a tagged port, it carries the 802.1p
> priority with it to the next downstream device. You can
> configure one QoS classifier for each VLAN-ID. (Default:* **No-
> override***)*

Syntax: no vlan < *vid* > qos

> *Removes the specified VLAN-ID as a QoS classifier and resets
> the priority for that VLAN to* **No- override***.*

Syntax: show qos vlan-priority

> *Displays a listing of the QoS VLAN-ID classifiers currently
> in the running-config file, with their priority data.*

1. For example, suppose that you have the following VLANs configured on
    the switch and want to prioritize them as shown:

Figure 8-25. Example of a List of VLANs Available for QoS Prioritization

2. You would then execute the following commands to prioritize the VLANs by VID:

```
ProCurve(config)# vlan 1 qos priority 2
ProCurve(config)# vlan 20 qos priority 5
ProCurve(config)# vlan 30 qos priority 5
ProCurve(config)# vlan 40 qos priority 7

ProCurve(config)# show qos vlan

  VLAN priorities

  VLAN ID Apply rule  | DSCP   Priority
  ------- ----------- + ------ -----------
  1        Priority   |           2
  20       Priority   |           5
  30       Priority   |           5
  40       Priority   |           7
```

Figure 8-26. Configuring and Displaying QoS Priorities on VLANs

If you then decided to remove VLAN_20 from QoS prioritization:

```
ProCurve(config)# no vlan 20 qos
ProCurve(config)# show qos vlan

  VLAN priorities

  VLAN ID Apply rule   | DSCP   Priority
  ------- -----------  + ------ -----------
  1        Priority    |           2
  20       No-override |       No-override
  30       Priority    |           5
  40       Priority    |           7
```

In this instance, **No- override** indicates that VLAN 20 is not prioritized by QoS.

Figure 8-27. Returning a QoS-Prioritized VLAN to "No-override" Status

Assigning a DSCP Policy Based on VLAN-ID (VID)

This option assigns a previously configured DSCP policy (codepoint and 802.1p priority) to outbound IP packets having the specified VLAN-ID (VID). That is, the switch:

1. Selects an incoming IP packet on the basis of the VLAN-ID it carries.

2. Overwrites the packet's DSCP with the DSCP configured in the switch for such packets.

3. Assigns the 802.1p priority configured in the switch for the new DSCP. (Refer to "Differentiated Services Codepoint (DSCP) Mapping" on page 8-64.)

4. Forwards the packet through the appropriate outbound port queue.

3400cl/6400cl Switch Restriction. On the 3400cl and 6400cl switches, "mixing" ToS DSCP policies and 802.1p priorities is not recommended. Refer to the Note on page 8-11.

For more on DSCP, refer to "Terminology" on page 8-6.

Steps for Creating a Policy Based on VLAN-ID Classifier.

1. Determine the VLAN-ID classifier to which you want to assign a DSCP policy.

2. Determine the DSCP policy for packets carrying the selected VLAN-ID:

   a. Determine the DSCP you want to assign to the selected packets. (This codepoint will be used to overwrite the DSCP carried in packets received from upstream devices.)

   b. Determine the 802.1p priority you want to assign to the DSCP.

3. Configure the DSCP policy by using **qos dscp-map** to configure the priority for each codepoint. (For details, see the example later in this section, and to "Differentiated Services Codepoint (DSCP) Mapping" on page 8-64.)

N o t e   A codepoint must have an 802.1p priority (0 - 7) before you can configure the codepoint for use in prioritizing packets by VLAN-ID. If a codepoint you want to use shows **No-override** in the **Priority** column of the DSCP Policy table (**show qos dscp-map**), then assign a priority before proceeding.

4. Configure the switch to assign the DSCP policy to packets with the specified VLAN-ID.

Syntax: qos dscp-map < *codepoint* > priority < 0 - 7 >

> *This command is optional if a priority has already been assigned to the <* ***codepoint****>. The command creates a DSCP policy by assigning an 802.1p priority to a specific DSCP. When the switch applies this priority to a packet, the priority determines the packet's queue in the outbound port to which it is sent. If the packet leaves the switch on a tagged port, it carries the 802.1p priority with it to the next downstream device. If the packet is IPv4, the packet's DSCP will be replaced by the codepoint specified in this command. (Default: For most codepoints,* **No-override***. See figure 8-14 on page 8-65 on page 8-65.)*

Syntax: vlan < *vid* > qos dscp < *codepoint* >

*Assigns a DSCP policy to packets carrying the specified VLAN-ID, and overwrites the DSCP in these packets with the assigned < **codepoint** > value. This policy includes an 802.1p priority and determines the packet's queue in the outbound port to which it is sent. If the packet leaves the switch on a tagged port, it carries the 802.1p priority with it to the next downstream device. (Default:* **No-override***)*

Syntax: no vlan < *vid* > qos

Removes QoS classifier for the specified VLAN.

Syntax: show qos device-priority

Displays a listing of all QoS VLAN-ID classifiers currently in the running-config file.

For example, suppose you wanted to assign this set of priorities:

VLAN-ID	DSCP	Priority
40	000111	7
30	000101	5
20	000010	1
1	000010	1

1. Determine whether the DSCPs already have priority assignments, which could indicate use by existing applications. This is not a problem as long as the configured priorities are acceptable for all applications using the same DSCP. (Refer to the "Note On Changing a Priority Setting" on page 8-67. Also, a DSCP must have a priority configured before you can assign any QoS classifiers to use it.)

```
ProCurve (config)# show qos dscp-map
   DSCP -> 802.p priority mappings
   DSCP policy 802.1p tag  Policy name
   ----------  -----------  ----------------------------
   000000      No-override
   000001      No-override
   000010      No-override
   000011      No-override
   000100      No-override
   000101      No-override
   000110      No-override
   000111      No-override
       .            .
       .            .
       .            .
```

The DSCPs for this example have not yet been assigned an 802.1p priority level.

Figure 8-28. Display the Current Configuration in the DSCP Policy Table

2. Configure the priorities for the DSCPs you want to use.

```
ProCurve(config)# qos dscp-map 000111 priority 7
ProCurve(config)# qos dscp-map 000101 priority 5
ProCurve(config)# qos dscp-map 000010 priority 1
ProCurve(config)# show qos dscp-map

  DSCP -> 802.p priority mappings
  DSCP policy 802.1p tag  Policy name
  ----------- ----------- ------------
   000000      No-override
   000001      No-override
   000010      1
   000011      No-override
   000100      No-override
   000101      5
   000110      No-override
   000111      7
   001000      No-override
      :           :
      :           :
      :           :
```

Priorities
Configured
in this step.

Figure 8-29. Assign Priorities to the Selected DSCPs

3. Assign the DSCP policies to the selected VIDs and display the result.

```
ProCurve(config)# vlan 1 qos dscp 000010
ProCurve(config)# vlan 20 qos dscp 000010
ProCurve(config)# vlan 30 qos dscp 000101
ProCurve(config)# vlan 40 qos dscp 000111

ProCurve(config)# show qos vlan-priority

  VLAN priorities

  VLAN ID Apply rule | DSCP    Priority
  ------- ---------- + ------ -----------
   1       DSCP        | 000010 1
   20      DSCP        | 000010 1
   30      DSCP        | 000101 5
   40      DSCP        | 000111 7
```

Figure 8-30. The Completed VID-DSCP Priority Configuration

The switch will now apply the DSCP policies in figure 8-30 to packets received
on the switch with the specified VLAN-IDs. This means the switch will:

■ Overwrite the original DSCPs in the selected packets with the new DSCPs
   specified in the above policies.

■ Assign the 802.1p priorities in the above policies to the appropriate
   packets.

QoS Source-Port Priority

QoS Classifier Precedence: 6

The QoS source-port option enables you to use a packet's source-port on the switch as a QoS classifier. Where a particular source-port classifier has the highest precedence in the switch for traffic entering through that port, then traffic received from the port is marked with the source-port classifier's configured priority level. Different source-port classifiers can have different priority levels.

Options for Assigning Priority on the Switch. Priority control options for packets from a specified source-port include:

- 802.1p priority
- DSCP policy (Assigning a new DSCP and an associated 802.1p priority; inbound packets must be IPv4.)

(For operation when other QoS classifiers apply to the same traffic, refer to "Classifiers for Prioritizing Outbound Packets" on page 8-10.)

Options for Assigning Priority From a RADIUS Server. You can use a RADIUS server to impose a QoS source-port priority during an 802.1X port-access authentication session. Refer to the RADIUS chapter in the Access Security Guide for your switch (January 2005 or later).

Assigning a Priority Based on Source-Port

This option assigns a priority to all outbound packets having the specified source-port. You can configure this option by either specifying the source-port ahead of the **qos** command or moving to the port context for the port you want to configure for priority. (If you are configuring multiple source-ports with the same priority, you may find it easier to use the **interface < *port-list* >** command to go to the port context instead of individually configuring the priority for each port.)

Syntax: interface < *port-list* > qos priority < 0 - 7 >

> *Configures an 802.1p priority for packets entering the switch through the specified (source) ports. This priority determines the packet queue in the outbound port(s) to which traffic is sent. If a packet leaves the switch on a tagged port, it carries the 802.1p priority with it to the next downstream device. You can configure one QoS classifier for each source-port or group of source-ports. (Default:* **No-override***)*

Syntax: no interface < *port-list* > qos

> *Disables use of the specified source-port(s) for QoS classi-fier(s) and resets the priority for the specified source-port(s) to* **No-override**.

Syntax: show qos port-priority

> *Lists the QoS port-priority classifiers with their priority data.*

For example, suppose that you want to prioritize inbound traffic on the following source-ports:

Source-Port	Priority
A1 - A3	2
A4	3
B1, B4	5
C1-C3	6

You would then execute the following commands to prioritize traffic received on the above ports:

```
ProCurve(config)# interface e c1-c3 qos priority 6
ProCurve(config)# interface e b1,b4 qos priority 5
ProCurve(config)# interface e a4 qos priority 3
ProCurve(config)# interface e a1-a3 qos priority 2

ProCurve(config)# show qos port-priority
  Port priorities

  Port Apply rule  | DSCP   Priority     Radius Override
  ---- ----------- + ------ -----------  ---------------
  A1   Priority    |          2          No-override
  A2   Priority    |          2          No-override
  A3   Priority    |          2          No-override
  A4   Priority    |          3          No-override
  B1   Priority    |          5          No-override
  B2   No-override |        No-override  No-override
  B3   No-override |        No-override  No-override
  B4   Priority    |          5          No-override
  C1   Priority    |          6          No-override
  C2   Priority    |          6          No-override
  C3   Priority    |          6          No-override
  C4   No-override |        No-override  No-override
  C5   No-override |        No-override  No-override
  .        .       |            .
  .        .       |            .
  .        .       |            .
```

Figure 8-31. Configuring and Displaying Source-Port QoS Priorities

If you then decided to remove port A1 from QoS prioritization:

```
ProCurve(config)# no interface e a1 qos
ProCurve(config)# show qos port-priority
  Port priorities

  Port Apply rule  | DSCP   Priority     Radius Override
  ---- ----------- + ------ ----------- --------------
  A1   No-override |        No-override No-override
  A2   Priority    |    2               No-override
  A3   Priority    |    2               No-override
  A4   Priority    |    3               No-override
```

In this instance, **No-override** indicates that port A1 is not prioritized by QoS.

Figure 8-32. Returning a QoS-Prioritized VLAN to "No-override" Status

Assigning a DSCP Policy Based on the Source-Port

This option assigns a previously configured DSCP policy (codepoint and 802.1p priority) to outbound IP packets (received from the specified source-ports). That is, the switch:

1. Selects an incoming IP packet on the basis of its source-port on the switch.

2. Overwrites the packet's DSCP with the DSCP configured in the switch for such packets.

3. Assigns the 802.1p priority configured in the switch for the new DSCP. (Refer to "Differentiated Services Codepoint (DSCP) Mapping" on page 8-64.)

4. Forwards the packet through the appropriate outbound port queue.

3400cl/6400cl Switch Restriction. On the 3400cl/6400cl switches, "mixing" ToS DSCP policies and 802.1p priorities is not recommended. Refer to the Note on page 8-11.

For more on DSCP, refer to "Terminology" on page 8-6.

Steps for Creating a Policy Based on Source-Port Classifiers.

N o t e   You can select one DSCP per source-port. Also, configuring a new DSCP for a source-port automatically overwrites (replaces) any previous DSCP or 802.1p priority configuration for that port.)

1. Identify the source-port classifier to which you want to assign a DSCP policy.

2. Determine the DSCP policy for packets having the selected source-port:

   a. Determine the DSCP you want to assign to the selected packets. (This codepoint will be used to overwrite the DSCP carried in packets received through the source-port from upstream devices.)

   b. Determine the 802.1p priority you want to assign to the DSCP.

3. Configure the DSCP policy by using **qos dscp-map** to configure the priority for each codepoint. (For details, refer to the example later in this section and to "Differentiated Services Codepoint (DSCP) Mapping" on page 8-64.)

N o t e   A codepoint must have an 802.1p priority assignment (0 - 7) before you can configure that codepoint as a criteria for prioritizing packets by source-port. If a codepoint shows **No-override** in the **Priority** column of the DSCP Policy Table (**show qos dscp-map**), then you must assign a 0 - 7 priority before proceeding.

4. Configure the switch to assign the DSCP policy to packets from the specified source-port.

Syntax: qos dscp-map < *codepoint* > priority < 0 - 7 >

*This command is optional if a priority has already been assigned to the < **codepoint** >. The command creates a DSCP policy by assigning an 802.1p priority to a specific DSCP. When the switch applies this priority to a packet, the priority determines the packet's queue in the outbound port to which it is sent. If the packet leaves the switch on a tagged port, it carries the 802.1p priority with it to the next downstream device. (Default: For most codepoints, **No-override**. See figure 8-14 on page 8-65 on page 8-65.)*

Syntax: interface < *port-list* > qos dscp < *codepoint* >

*Assigns a DSCP policy to packets from the specified source-port(s), and overwrites the DSCP in these packets with the assigned < **codepoint** > value. This policy includes an 802.1p priority and determines the packet's queue in the outbound port to which it is sent. If the packet leaves the switch on a tagged port, it carries the 802.1p priority with it to the next downstream device. (Default: **No-override**)*

Syntax: no interface [e] < *port-list* > qos

> *Removes QoS classifier for the specified source-port(s).*

Syntax: show qos source-port
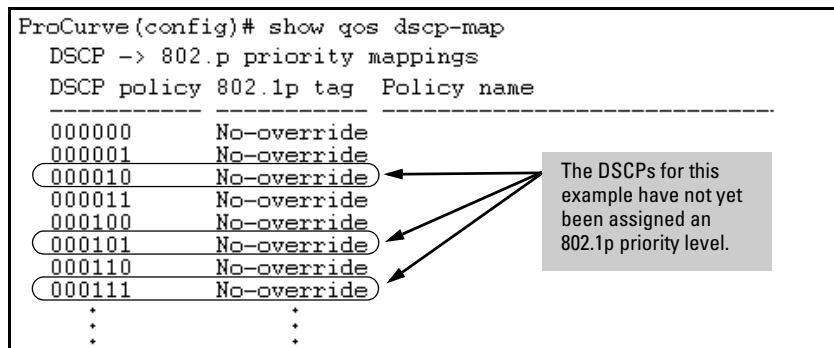
> *Displays a listing of all source-port QoS classifiers currently in the running-config file.*

For example, suppose you wanted to assign this set of priorities:

Source-Port	DSCP	Priority
A2	000111	7
B1-B3	000101	5
B4, C2	000010	1

1. Determine whether the DSCPs already have priority assignments, which could indicate use by existing applications. This is not a problem as long as the configured priorities are acceptable for all applications using the same DSCP. (Refer to the "Note On Changing a Priority Setting" on page 8-67. Also, a DSCP must have a priority configured before you can assign any QoS classifiers to use it.)

```
ProCurve(config)# show qos dscp-map
   DSCP -> 802.p priority mappings
   DSCP policy 802.1p tag  Policy name
   -----------  -----------  ----------------------------
   000000      No-override
   000001      No-override
   000010      No-override
   000011      No-override
   000100      No-override
   000101      No-override
   000110      No-override
   000111      No-override
       .            .
       .            .
       .            .
```

The DSCPs for this example have not yet been assigned an 802.1p priority level.

Figure 8-33. Display the Current Configuration in the DSCP Policy Table

2. Configure the priorities for the DSCPs you want to use.

```
ProCurve(config)# qos dscp-map 000111 priority 7
ProCurve(config)# qos dscp-map 000101 priority 5
ProCurve(config)# qos dscp-map 000010 priority 1
ProCurve(config)# show qos dscp-map

  DSCP -> 802.p priority mappings
  DSCP policy  802.1p tag  Policy name
  -----------  ----------  ------------
  000000       No-override
  000001       No-override
  000010       1
  000011       No-override
  000100       No-override                    Priorities
  000101       5                              Configured in
  000110       No-override                    this step.
  000111       7
  001000       No-override
     .            .
     .            .
     .            .
```

Figure 8-34. Assign Priorities to the Selected DSCPs

3. Assign the DSCP policies to the selected source-ports and display the result.

```
ProCurve(eth-A2)# int e b4,c2
ProCurve(eth-B4,C2)# qos dscp 000010
ProCurve(eth-B4,C2)# int e b1-b3
ProCurve(eth-B1-B3)# qos dscp 000101
ProCurve(eth-B1-B3)# int e a2
ProCurve(eth-A2)# qos dscp 000111

 ProCurve(eth-A2)# show qos port-priority

  Port priorities

  Port Apply rule  | DSCP     Priority      Radius Override
  ---- ----------- + ------   -----------   ---------------
   A1  No-override |          No-override   No-override
   A2  DSCP        | 000111 7               No-override
   A3  No-override |          No-override   No-override
   A4  No-override |          No-override   No-override
   B1  DSCP        | 000101 5               No-override
   B2  DSCP        | 000101 5               No-override
   B3  DSCP        | 000101 5               No-override
   B4  DSCP        | 000010 1               No-override
   C1  No-override |          No-override   No-override
   C2  DSCP        | 000010 1               No-override
   C3  No-override |          No-override   No-override
   C4  No-override |          No-override   No-override
```

Figure 8-35. The Completed Source-Port DSCP-Priority Configuration

Radius Override Field. During a client session authenticated by a RADIUS server, the server can impose a port priority that applies only to that client session. Refer to the RADIUS chapter in the *Access Security Guide* for your switch (January 2005 or later).

Differentiated Services Codepoint (DSCP) Mapping

The DSCP Policy Table associates an 802.1p priority with a specific ToS byte codepoint in an IPv4 packet. This enables you to set a LAN policy that operates independently of 802.1Q VLAN-tagging.

In the default state, most of the 64 codepoints do not assign an 802.1p priority, as indicated by **No-override** in table 8-14 on page 8-65.

You can use the following command to list the current DSCP Policy table, change the codepoint priority assignments, and assign optional names to the codepoints.

Syntax: show qos dscp-map

Displays the DSCP Policy Table.

qos dscp-map **< codepoint >** priority < 0 - 7 > [name **< ascii-string >**]

Configures an 802.1p priority for the specified codepoint and, optionally, an identifying (policy) name.

no qos dscp-map **< codepoint >**

*Reconfigures the 802.1p priority for <**codepoint**> to **No-override**. Also deletes the codepoint policy name, if configured.*

no qos dscp-map < codepoint > name

*Deletes only the **policy name, if** configured, for < **codepoint** >.*

Table 8-14. The Default DSCP Policy Table

DSCP Policy	802.1p Priority	DSCP Policy	802.1p Priority	DSCP Policy	802.1p Priority
000000	No-override	010110	3*	101011	No-override
000001	No-override	010111	No-override	101100	No-override
000010	No-override	011000	No-override	101101	No-override
000011	No-override	011001	No-override	101110	7**
000100	No-override	011010	4*	101111	No-override
000101	No-override	011011	No-override	110000	No-override
000110	No-override	011100	4*	110001	No-override
000111	No-override	011101	No-override	110010	No-override
001000	No-override	011110	5*	110011	No-override
001001	No-override	011111	No-override	110100	No-override
001010	1*	100000	No-override	110101	No-override
001011	No-override	100001	No-override	110110	No-override
001100	1*	100010	6*	110111	No-override
001101	No-override	100011	No-override	111000	No-override
001110	2*	100100	6*	111001	No-override
001111	No-override	100101	No-override	111010	No-override
010000	No-override	100110	7*	111011	No-override
010001	No-override	100111	No-override	111100	No-override
010010	0 *	101000	No-override	111101	No-override
010011	No-override	101001	No-override	111110	No-override
010100	0 *	101010	No-override	111111	No-override
010101	No-override				

*Assured Forwarding codepoints; configured by default on the Series 5304xl switches. These codepoints are configured as "No-override" in the Series 3400cl, Series 6400cl and Series 2600/2800 switches.

**Expited Forwarding codepoint configured by default.

Default Priority Settings for Selected Codepoints

In a few cases, such as 001010 and 001100, a default policy (implied by the DSCP standards for Assured-Forwarding and Expedited-Forwarding) is used. You can change the priorities for the default policies by using **qos dscp-map < *codepoint* > priority < 0 - 7 >**.(These policies are not in effect unless you have either applied the policies to a QoS classifier or configured QoS Type-of-Service to be in **diff-services** mode.)

Quickly Listing Non-Default Codepoint Settings

Table 8-14 lists the switch's default codepoint/priority settings. If you change
the priority of any codepoint setting to a non-default value and then execute
write memory, the switch will list the non-default setting in the show config
display. For example, in the default configuration, the following codepoint
settings are true:

Codepoint	Default Priority
001100	1
001101	No-override
001110	2

If you change all three settings to a priority of 3, and then execute **write memory**,
the switch will reflect these changes in the show config listing:

```
ProCurve(config)# qos dscp-map 001100 priority 3
ProCurve(config)# qos dscp-map 001101 priority 3
ProCurve(config)# qos dscp-map 001110 priority 3
ProCurve(config)# write memory

ProCurve(config)# show config
Startup configuration:

; J4850A Configuration Editor; Created on release #E.05.01

hostname "HPswitch"
time daylight-time-rule None
cdp run
qos dscp-map 001100 priority 3
qos dscp-map 001101 priority 3
qos dscp-map 001110 priority 3
module 2 type J4821A
module 3 type J4820A
    .    .    .        .
    .    .    .        .
    .    .    .        .
```

Configure these three codepoints
with non-default priorities.

Show config lists the non default
codepoint settings.

**Figure 8-36. Example of Show Config Listing with Non-Default Priority Settings in
the DSCP Table**

Effect of "No-override". In the QoS Type-of-Service differentiated services
mode, a **No-override** assignment for the codepoint of an outbound packet
means that QoS is effectively disabled for such packets. That is, QoS does not

affect the packet queuing priority or VLAN tagging. In this case, the packets are handled as follows (as long as no other QoS feature creates priority assignments for them):

802.1Q Status	Outbound 802.1p Priority
Received and Forwarded on a tagged port member of a VLAN.	Unchanged
Received on an Untagged port member of a VLAN; Forwarded on a tagged port member of a VLAN.	0 (zero)—"normal"
Forwarded on an Untagged port member of a VLAN.	None

Note On Changing a Priority Setting

If a QoS classifier is using a policy (codepoint and associated priority) in the DSCP Policy table, you must delete or change this usage before you can change the priority setting on the codepoint. Otherwise the switch blocks the change and displays this message:

Cannot modify DSCP Policy < *codepoint* > - in use by other qos rules.

In this case, use **show qos < *classifier* >** to identify the specific classifiers using the policy you want to change; that is:

    show qos device-priority
    show qos port-priority
    show qos tcp-udp-port-priority
    show qos vlan-priority
    show qos type-of-service

Note that protocol-priority is not included because a DSCP policy is not meaningful for this classifier and therefore not configurable in this case.

For example, suppose that the 000001 codepoint has a priority of 6, and several classifiers use the 000001 codepoint to assign a priority to their respective types of traffic. If you wanted to change the priority of codepoint 000001 you would do the following:

1. Identify which QoS classifiers use the codepoint.

2. Change the classifier configurations by assigning them to a different DSCP policy, or to an 802.1p priority, or to **No-override**.

3. Reconfigure the desired priority for the 000001 codepoint.

4. Either reassign the classifiers to the 00001 codepoint policy or leave them as they were after step 2, above.

Example of Changing the Priority Setting on a Policy When One or More Classifiers Are Currently Using the Policy

Suppose that codepoint 000001 is in use by one or more classifiers. If you try to change its priority, you see a result similar to the following:

```
ProCurve(config)# qos dscp-map 000001 priority 2
Cannot modify DSCP Policy 000001 - in use by other qos rules.
```

Figure 8-37. Example of Trying To Change the Priority on a Policy In Use by a Classifier

In this case, you would use steps similar to the following to change the priority.

1. Identify which classifiers use the codepoint you want to change.

```
                          ProCurve(config)# show qos device-priority
Three classifiers use
the codepoint that is      Device priorities
to be changed.
                           Device Address Apply rule | DSCP  Priority
                           -------------- ---------- + ------ -----------
                           10.26.50.104   DSCP       | 000001 6

                          ProCurve(config)# show qos port-priority

                           Port priorities

                           Port Apply rule  | DSCP  Priority     Radius Override
                           ---- ----------- + ----- -----------  ---------------
                           A1   No-override |        No-override No-override
                           A2   No-override |        No-override No-override
                           A3   DSCP        | 000001 6           No-override
                           A4   No-override |        No-override No-override
                           A5   No-override |        No-override No-override
                            .        .           .        .
                            .        .           .        .
                            .        .           .        .
                          ProCurve(config)# show qos tcp-udp-port-priority

                           TCP/UDP port based priorities

                                    | Application              |
                           Protocol | Port        Apply rule | DSCP  Priority
                           -------- + ----------- ---------- + ------ -----------
Two classifiers do not     UDP      | 1260        DSCP       | 000001 6
use the codepoint that
is to be changed.
                          ProCurve(config)# show qos vlan-priority

                           VLAN priorities

                           VLAN ID Apply rule  | DSCP  Priority
                           ------- ----------- + ----- -----------
                           1       No-override |        No-override

                          ProCurve(config)# show qos type-of-service
                           Type of Service [Disabled] : Disabled
```

Figure 8-38. Example of a Search to Identify Classifiers Using a Codepoint You Want To Change

2. Change the classifier configurations by assigning them to a different DSCP policy, or to an 802.1p priority, or to **No-override**. For example:

   a. Delete the policy assignment for the **device-priority** classifier. (That is, assign it to **No-override**.)

   b. Create a new DSCP policy to use for re-assigning the remaining classifiers.

   c. Assign the **port-priority** classifier to the new DSCP policy.

   d. Assign the **udp-port 1260** classifier to an 802.1p priority.

```
(a) ProCurve(config)# no qos device-priority 10.26.50.104

(b) ProCurve(config)# qos dscp-map 000100 priority 6

(c) ProCurve(config)# int e a3 qos dscp 000100

(d) ProCurve(config)# qos udp-port 1260 priority 2
```

3. Reconfigure the desired priority for the 000001 codepoint.

   ```
   ProCurve(config)# qos dscp-map 000001 priority 4
   ```

4. You could now re-assign the classifiers to the original policy codepoint or leave them as currently configured.

IP Multicast (IGMP) Interaction with QoS

IGMP high-priority-forward causes the switch to service the subscribed IP multicast group traffic at high priority, even if QoS on the switch has relegated the traffic to a lower priority. This does not affect any QoS priority settings, so the QoS priority is honored by downstream devices. However, QoS does take precedence over IGMP normal-priority traffic.

The switch's ability to prioritize IGMP traffic for either a normal or high priority outbound queue overrides any QoS criteria, and does not affect any 802.1p priority settings the switch may assign. For a given packet, if both IGMP high priority and QoS are configured, the QoS classification occurs and the switch marks the packet for downstream devices, but the packet is serviced by the high-priority queue when leaving the switch.

IGMP High Priority	QoS Configuration Affects Packet	Switch Port Output Queue	Outbound 802.1p Setting (Requires Tagged VLAN)
Not Enabled	Yes	Determined by QoS	Determined by QoS
Enabled	See above paragraph.	High	As determined by QoS if QoS is active.

QoS Messages in the CLI

Message	Meaning
DSCP Policy < *decimal-codepoint* > not configured	You have attempted to map a QoS classifier to a codepoint for which there is no configured priority (**No-override**). Use the **qos dscp-map** command to configure a priority for the codepoint, then map the classifier to the codepoint.
Cannot modify DSCP Policy < *codepoint* > - in use by other qos rules.	You have attempted to map a QoS classifier to a codepoint that is already in use by other QoS classifiers. Before remapping the codepoint to a new priority, you must reconfigure the other QoS classifiers so that they do not use this codepoint. You can have multiple QoS classifiers use this same codepoint as long as it is acceptable for all such classifiers to use the same priority.

QoS Operating Notes and Restrictions

Table 8-15. **Details of Packet Criteria and Restrictions for QoS Support**

Packet Criteria or Restriction	QoS Classifiers							DSCP Overwrite (Re-Marking)
	UDP/TCP	Device Priority (IP Address)	IP Type-of-Service	Layer 3 Protocol	VLAN	Source Port	Incoming 802.1p	
Restricted to IPv4 Packets Only	Yes	Yes	Yes	No	No	No	No	Yes
Allow Packets with IP Options[1]	3400cl and 6400cl: No 5300xl: Yes 4200vl: Yes[3]	3400cl and 6400cl: Yes 5300xl: Yes[3] 4200vl: Yes[3]	3400cl and 6400cl: Yes 5300xl: Yes[3] 4200vl: Yes[3]	3400cl and 6400cl: Yes 5300xl: Yes[3] 4200vl: Yes[3]	3400cl and 6400cl: Yes 5300xl: Yes[3] 4200vl: Yes[3]	3400cl and 6400cl: Yes 5300xl: Yes[3] 4200vl: Yes[3]	3400cl and 6400cl: Yes 5300xl: Yes[3] 4200vl: Yes[3]	3400cl and 6400cl: Yes 5300xl: No 4200vl: No[3]
Support IPv6 Packets[2]	No	No	No	3400cl and 6400cl: n/a 5300xl: Yes	Yes	Yes	Yes	No
Support Layer-2 SAP Encapsulation	3400cl and 6400cl: No 5300xl: Yes 4200vl: Yes	3400cl and 6400cl: No 5300xl: Yes 4200vl: Yes	3400cl and 6400cl: No 5300xl: Yes 4200vl: Yes	3400cl and 6400cl: No 5300xl: Yes 4200vl: Yes	3400cl and 6400cl: Yes 5300xl: Yes 4200vl: Yes	3400cl and 6400cl: Yes 5300xl: Yes 4200vl: Yes	3400cl and 6400cl: Yes 5300xl: Yes 4200vl: Yes	3400cl and 6400cl: No 5300xl: Yes 4200vl: Yes

[1]An "IP Option" is an optional, extra field in the header of an IP packet. If a 3400cl or 6400cl switch is configured with a UDP/TCP classifier and a packet with an IP option is received, the switch uses the next-highest classifier that is configured and applicable to actually match and classify the packet.

[2]All Switches: For explicit QoS support of IPv6 packets, force IPv6 traffic into its own set of VLANs and then configure VLAN-based classifiers for those VLANs.

[3]On IPv4 packets with IP options, the 5300xl and 4200vl switches support QoS for 802.1p priority policies, but does **not** do any DSCP re-marking for DSCP policies.

■ **All Switches:** For explicit QoS support of IP subnets, HP recommends forcing IP subnets onto separate VLANs and then configuring VLAN-based classifiers for those VLANs.

■ **For Devices that Do Not Support 802.1Q VLAN-Tagged Ports:** For communication between these devices and the switch, connect the device to a switch port configured as **Untagged** for the VLAN in which you want the device's traffic to move.

■ **Port Tagging Rules:** For a port on the switch to be a member of a VLAN, the port must be configured as either **Tagged** or **Untagged** for that VLAN. A port can be an untagged member of only one VLAN of a given protocol type. Otherwise, the switch cannot determine which

VLAN should receive untagged traffic. For more on VLANs, refer to
chapter 2, "Static Virtual LANs (VLANs)".

■ **3400cl and 6400cl Switches Only—SAP-Encapsulated Packet
Restriction:** Except for source-port QoS and VLAN QoS, the 3400cl/
6400cl switches do not support QoS (or ACL) operation for SAP-
Encapsulated packets.

■ **3400cl/6400cl Switches Only—Packets with IP Option Fields in
the Header:** UDP/TCP QoS is not supported for IP packets carrying
optional fields in their headers.

■ **Maximum QoS Configuration Entries:** The switches covered by this
guide accept the maximum outbound priority and/or DSCP policy config-
uration entries shown in table 8-16.

Table 8-16.   Maximum QoS Entries.

Switch	Software Version	Maximum QoS Entries	Notes for All Switch Models
Series 5300xl	E.08.01 and greater	250*	• Each device (IP address) QoS configuration uses two entries.
Series 3400cl and Series 6400cl	All	120*	• Each TCP/UDP port QoS configuration uses four entries. • All other classifier configurations use one entry each.
*Configuring device (IP address) or TCP/UDP QoS entries reduces this maximum. See the "Notes" column.			

Attempting to exceed the above limits generates the following message
in the CLI:

```
Unable to add this QoS rule. Maximum number (entry-#)
already reached.
```

5300xl and 4200vl: Where a 5300xl switch is running a software release
earlier than E.08.01 and is configured with more than 250 QoS rules,
downloading software release E.08.01 (or greater) causes the switch to:

• Implement the first 250 QoS rules in its configuration, but ignore the
configured rules exceeding that limit.

• Generate these Event Log messages:
  – ```Too many QoS configuration items - limit of 250```
  – ```Some QoS configuration items will not be active```

■ **5300xl and 4200vl Switches—Non-Supported IP Packets:** The DSCP policy codepoint-remarking operation is not supported in any QoS classifier for packets carrying IP options in the packet header.

■ **All Switches—Not Supported:** Use of an inbound 802.1p packet priority as a classifier for remapping a packet's outbound priority to different 802.1p priority. For example, where inbound packets carry an 802.1p priority of 1, QoS cannot be configured use this priority as a classifier for changing the outbound priority to 0.

—This page is intentionally unused —

9

Access Control Lists (ACLs) for the Series 5300xl Switches

Contents

Introduction

This chapter applies only to the Series 5300xl Switches. For ACL operation on Series 3400cl and Series 6400clswitches, refer to the chapter 10, "Access Control Lists (ACLs) for the Series 3400cl and Series 6400cl Switches".

Feature	Default	Menu	CLI	Web
Numbered ACLs				
Standard ACLs	None	—	9-33	—
Extended ACLs	None	—	9-38	—
Named ACLs		—	9-44	—
Enable or Disable an ACL		—	9-46	—
Display ACL Data	n/a	—	9-48	—
Delete an ACL	n/a	—	9-47	—
Configure an ACL from a TFTP Server	n/a	—	9-55	—
Enable ACL Logging	n/a	—	9-60	—

Layer 3 IP filtering with ACLs on the Series 5300XL switches can help improve network performance and restrict network use by creating policies for:

■ **Switch Management Access:** Permits or denies in-band management access. This includes preventing the use of certain TCP or UDP applications (such as Telnet, SSH, web browser, and SNMP) for transactions between specific source and destination IP addresses.)

■ **Application Access Security:** Eliminates unwanted IP, TCP, or UDP traffic in a path by filtering packets where they enter or leave the switch on specific VLAN interfaces.

ACLs on the 5300xl switches can filter traffic to or from a host, a group of hosts, or entire subnets.

This chapter describes how to configure, apply, and edit ACLs in a network populated with ProCurve Series 5300XL switches (with IP routing support enabled) and how to monitor the results of ACL actions.

Notes   ACLs can enhance network security by blocking selected IP traffic, and can serve as part of your network security program. *However, because ACLs do not provide user or device authentication, or protection from malicious manipulation of data carried in IP packet transmissions, they should not be relied upon for a complete security solution.*

Series 5300XL ACLs do not screen non-IP traffic such as AppleTalk and IPX.

For ACL filtering to take effect, configure an ACL and then assign it to either the inbound or outbound traffic on a statically configured VLAN on the switch. (Except for ACEs that screen traffic to an IP address on the switch itself, ACLs assigned to VLANs can operate only while IP routing is enabled. Refer to "Notes on IP Routing" on page 9-11.)

Table 9-1.   Comprehensive Command Summary

Action	Command	Page
Configuring Standard (Numbered) ACLs	ProCurve(config)# [no] access-list < 1-99 > < deny | permit > < any | host *<src-ip-addr>* | *src-ip-address/mask* > [1] [log][2]	9-3 3

Configuring Extended (Numbered) ACLs	ProCurve(config)# [no] access-list <100-199> < deny | permit >
	        ip < any | host *<src-ip-addr>* | *src-ip-address*/mask > [1]
	        [log][2]
		9-3
		8

	ProCurve(config)# [no] access-list < 100-199 > < deny | permit >
	        < tcp | udp >
	        < any | host *<src-ip-addr>* | *src-ip-address*/mask > [1]
	          [operator < *src-port tcp/udp-id* >]
	        < any | host *<dest-ip-addr>* | *dest-ip-address*/mask > [1]
	          [operator < *dest-port tcp/udp-id* >]
	        [log][2]	

Configuring Standard (Named) ACLs	ProCurve(config)# [no] ip access-list standard < *name-str* | 1-99 >	9-4 4
	ProCurve(config-std-nacl)# < deny | permit > < any | host *<src-ip-addr>* | *src-ip-address/mask* > [1] [log] [2]	
Configuring Extended (Named) ACLs	ProCurve(config)# [no] ip access-list extended < *name-str* | 100-199 >	
	ProCurve(config-std-nacl)# < deny | permit > ip < any | host *<src-ip-addr>* | *src-ip-address*/mask >[1] < any | host *<dest-ip-addr>* | *dest-ip-address*/mask >[1] [log][2]	

	ProCurve(config-std-nacl)# < deny | permit > < tcp | udp >
	        < any | host *<src-ip-addr>* | *src-ip-address*/mask > [1]
	          [operator < *src-port tcp/udp-id* >]
	        < any | host *<dest-ip-addr>* | *dest-ip-address*/mask > [1]
	          [operator < *dest-port tcp/udp-id* >]
	        [log][2]	

Enabling or Disabling an ACL	ProCurve(config)# [no] vlan < *vid* > ip access-group < *name-str* | 1-99 | 100-199 > < in/out >	9-4 6

[1] The mask can be in either dotted-decimal notation (such as 0.0.15.255) or CIDR notation (such as /20).
[2] The [log] function applies only to "deny" ACLs, and generates a message only when there is a "deny" match.

Action	Command	Page
Deleting an ACL from the Switch	ProCurve(config)# no ip access-list < standard | extended > < *name-str* | 1-99 | 100 -199 > < in | out >	9-4 7
Displaying ACL Data	ProCurve(config)# show access-list	9-4 8
	ProCurve(config)# show access-list config	
	ProCurve(config)# show access-list vlan < *vid* >	
	ProCurve(config)# show config	
	ProCurve(config)# show running	

Terminology

Access Control Entry (ACE): An ACE is a policy consisting of criteria and an action to take (permit or deny) on a packet if it meets the criteria. The elements composing the criteria include:

- Source IP address and mask (standard and extended ACLs)
- Destination IP address and mask (extended ACLs only)
- TCP or UDP application port numbers (optional, extended ACLs only)

Access Control List (ACL): A list (or set) consisting of one or more explicitly configured Access Control Entries (ACEs) and terminating with an implicit "deny" default which drops any packets that do not have a match with any explicit ACE in the named ACL. The two classes of ACLs are "standard" and "extended". See "Standard ACL" and "Extended ACL".

ACE: See "Access Control Entry".

ACL: See "Access Control List".

ACL ID: A number or alphanumeric string used to identify an ACL. A *standard* ACL ID can have either a number from 1 to 99 or an alphanumeric string. An *extended* ACL ID can have either a number from 100 to 199 or an alphanumeric string.

ACL Mask: Follows any IP address (source or destination) listed in an ACE. Defines which bits in a packet's corresponding IP addressing must exactly match the IP addressing in the ACE, and which bits need not match (wildcards). See also "How an ACE Uses a Mask To Screen Packets for Matches" on page 9-20.)

Connection-Rate ACL: An optional feature used with Connection-Rate filtering based on virus-throttling technology, and available in 5300xl switches running software release E.09.*xx* or greater. For more information, refer to the chapter titled "Virus Throttling" in the Access Security Guide for your 5300xl switch.

DA: The acronym for *Destination IP Address*. In an IP packet, this is the destination IP address carried in the header, and identifies the destination intended by the packet's originator. In an extended ACE, this is the second of two IP addresses required by the ACE to determine whether there is a match between a packet and the ACE. See also "SA".

Deny: An ACE configured with this action causes the switch to drop a packet for which there is a match within an applicable ACL.

Extended ACL: This type of Access Control List uses layer-3 IP criteria composed of source and destination IP addresses and (optionally) TCP or UDP port criteria to determine whether there is a match with an IP packet. You can apply extended ACLs to either inbound or outbound routed traffic and to any inbound switched or routed traffic with a DA belonging to the switch itself. Extended ACLs require an identification number (ID) in the range of 100 - 199 or an alphanumeric name.

Implicit Deny: If the switch finds no matches between a routed packet and the configured criteria in an applicable ACL, then the switch denies (drops) the packet with an implicit "deny IP any" operation. You can preempt the implicit "deny IP any" in a given ACL by configuring **permit IP any** (standard) or **permit IP any any** (extended) as the last explicit ACE in the ACL. Doing so permits any routed packet that is not explicitly permitted or denied by other ACEs configured sequentially earlier in the ACL. Unless otherwise noted, "implicit deny IP any" refers to the "deny" action enforced by both standard and extended ACLs.

Inbound Traffic: For the purpose of defining where the switch applies ACLs to filter traffic, inbound traffic is any IP packet that:

- *Enters the switch* on a given subnet.
- Has a destination IP address (DA) that meets either of these criteria:
    - The packet's DA is for an external device on a different subnet.

– The packet's DA is for an IP address configured on the switch itself. (This increases your options for protecting the switch from unauthorized management access.)

Because ACLs are assigned to VLANs, an ACL that filters inbound traffic on a particular VLAN examines packets meeting the above criteria that have entered the switch through any port on that VLAN.

Outbound Traffic: For defining the points where the switch applies ACLs to filter traffic, outbound traffic is routed traffic *leaving the switch* through a physical port; that is, traffic received on a port in one VLAN (subnet) and sent through a port on another VLAN to another device. This requires that you enable IP routing on the switch. The switch does not apply ACLs internally where routed traffic moves between VLANs. Note that for ACL purposes, "outbound traffic" does not include traffic received on one port and switched to the outbound queue of another port on the same VLAN (subnet); that is, traffic arriving on and leaving the switch on the same VLAN. (Refer also to "ACL Inbound and Outbound Application Points" on page 9-8.)

Permit: An ACE configured with this action allows the switch to forward a routed packet for which there is a match within an applicable ACL.

SA: The acronym for *Source IP Address*. In an IP packet, this is the source IP address carried in the IP header, and identifies the packet's sender. In an extended ACE, this is the first of two IP addresses used by the ACE to determine whether there is a match between a packet and the ACE. See also "DA".

Standard ACL: This type of Access Control List uses layer-3 IP criteria of source IP address to determine whether there is a match with an IP packet. You can apply standard ACLs to either inbound or outbound routed traffic and to any inbound switched or routed traffic with a DA belonging to the switch itself. Standard ACLs require an identification number (ID) in the range of 100 - 199 or an alphanumeric name.

Wildcard: The part of a mask that indicates the bits in a packet's IP addressing that do not need to match the corresponding bits specified in an ACL. See also **ACL Mask** on page 9-6.

Overview

Types of IP ACLs

Standard ACL: Use a standard ACL when you need to permit or deny traffic based on source IP address only. Standard ACLs are also useful when you need to quickly control a performance problem by limiting traffic from a subnet, group of devices, or a single device. (This can block all IP traffic from the configured source, but does not hamper traffic from other sources within the network.) This ACL type uses a numeric ID of 1 through 99 or an alphanumeric ID string. You can specify a single host, a finite group of hosts, or any host.

Extended ACL: Extended ACLs are useful whenever simple IP source address restrictions do not provide the breadth of traffic selection criteria you want to exercise on a VLAN interface. Extended ACLs allow use of the following criteria:

■ Source and destination IP addresses

■ TCP application criteria

■ UDP application criteria

Connection-Rate ACL. An optional feature used with Connection-Rate filtering based on virus-throttling technology, and available in 5300xl switches running software release E.09.*xx* or greater. For more information, refer to the chapter titled "Virus Throttling" in the Access Security Guide for your 5300xl switch.

ACL Inbound and Outbound Application Points

You can apply ACL filtering to the following types of traffic:

■ IP traffic routed between different subnets. (IP routing *must* be enabled.)

■ IP traffic carrying a destination address (DA) on the switch itself. In figure 9-1, below, this is any of the IP addresses shown in VLANs "A", "B", and "C" on the switch. (IP routing need not be enabled.)

The switch can apply ACL filtering to traffic *entering or leaving the switch* on VLANs configured to apply ACL filters. (When you assign an ACL to a VLAN, you must specify whether the ACL will filter inbound or outbound traffic.) For example, in figure 9-1:

■   You would assign either an inbound ACL on VLAN "A" or an outbound ACL on VLAN "B" to filter a packet routed between subnets; that is, from the workstation 18.28.10.5 on VLAN "A" to the server at 18.28.20.99 on VLAN "B". (An outbound ACL on VLAN "A" or an inbound ACL on VLAN "B" would not filter the packet.)

■   Where multiple subnets are configured on the same VLAN, *if*:

   •   Traffic you want to filter moves between subnets on the same VLAN.

   •   The traffic source and destination IP addresses are on devices external to the switch.

   Then you can use either inbound or outbound ACLs to filter the traffic on the VLAN (because the traffic moves between subnets but enters and leaves the switch in the same VLAN.)
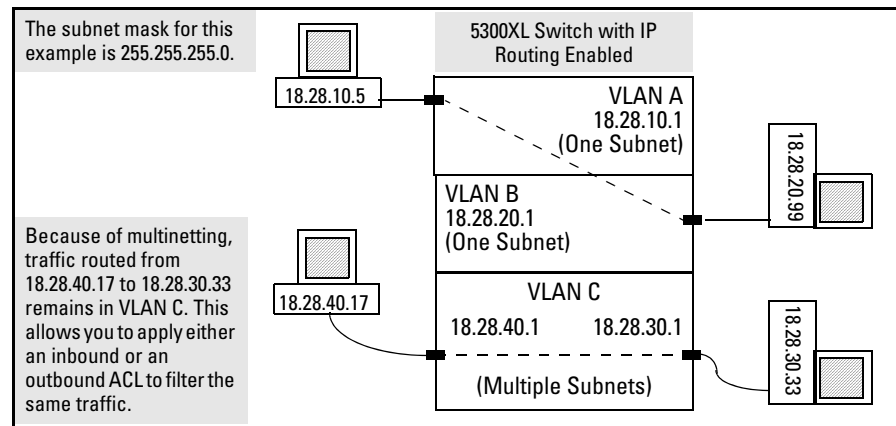
Figure 9-1. Example of Filter Applications

Note   ACLs do not filter traffic that remains in the same subnet from source to destination (switched traffic) unless the destination IP address (DA) is on the switch itself.

Features Common to All per-VLAN ACLs

■ On any VLAN you can apply one ACL to inbound traffic and one ACL to outbound traffic. You can use the same ACL or different ACLs for the inbound and outbound traffic.

■ Any ACL can have multiple entries (ACEs).

■ You can apply any one ACL to multiple VLANs.

■ A source or destination IP address and a mask, together, can define a single host, a range of hosts, or all hosts.

■ The IP address(es) assigned to a VLAN must not be configured from a DHCP server.

■ Every standard ACL includes an implied "**deny IP any**" as the last entry, and every extended ACL includes an implied "**deny IP any any**" as the last entry. The switch applies this action to any packets that do not match other criteria in the ACL.

■ In any ACL, you can apply an ACL log function to ACEs that have a "deny" action. The logging occurs when there is a match on a "deny" ACE. (The switch sends ACL logging output to Syslog and, optionally, to a console session.)

You can configure ACLs using either the CLI or a text editor. The text-editor method is recommended when you plan to create or modify an ACL that has more entries than you can easily enter or edit using the CLI alone. Refer to "Editing ACLs and Creating an ACL Offline" on page 9-53.

General Steps for Planning and Configuring ACLs

1. Identify the traffic type to filter. Options include:
   • Any routed IP traffic
   • Routed TCP traffic only
   • Routed UDP traffic only

2. The SA and/or the DA of routed traffic you want to permit or deny.

3. Determine the best points at which to apply specific ACL controls. For example, you can improve network performance by filtering unwanted traffic at the edge of the network instead of in the core. Also, on the switch itself, you can improve performance by filtering unwanted traffic where it is inbound to the switch instead of outbound.

4. Design the ACLs for the control points you have selected. Where you are using explicit "deny" ACEs, you can optionally use the ACL logging feature to help verify that the switch is denying unwanted packets where intended. Remember that excessive ACL logging activity can degrade the switch's performance. (Refer to "Enable ACL "Deny" Logging" on page 9-59.)

5. Create the ACLs in the selected switches.

6. Assign the ACLs to filter the inbound and/or outbound traffic on static VLAN interfaces configured on the switch.

7. Enable IP routing on the switch. (Except for an ACL configured to filter traffic having the switch itself as the destination IP address, IP routing must be enabled before ACLs will operate.)

8. Test for desired results.

For more details on ACL planning considerations, refer to "Planning an ACL Application" on page 9-16.

Notes on IP Routing
To activate an ACL to screen inbound traffic for routing between subnets, assign the ACL to the statically configured VLAN on which the traffic enters the switch. Also, ensure that IP routing is enabled. Similarly, to activate an ACL to screen routed, outbound traffic, assign the ACL to the statically configured VLAN on which the traffic exits from the switch. The only exception to these rules is for an ACL configured to screen inbound traffic with a destination IP address on the switch. In this case, an ACL assigned to a VLAN screens traffic addressed to an IP address on the switch, regardless of whether IP routing is also enabled. (ACLs do not screen outbound traffic generated by the switch, itself. Refer to "ACL Screening of Traffic Generated by the Switch" on page 9-63.)

Caution Regarding the Use of Source Routing
Source routing is enabled by default on the switch and can be used to override ACLs. For this reason, if you are using ACLs to enhance network security, the recommended action is to use the **no ip source-route** command to disable source routing on the switch. (If source routing is disabled in the running-config file, the **show running** command includes "**no ip source-route**" in the running-config file listing.)

ACL Operation

Introduction

An ACL is a list of one or more Access Control Entries (ACEs), where each ACE consists of a matching criteria and an action (permit or deny). An ACL applies only to the switch in which it is configured. ACLs operate on assigned static VLANs, and filter these traffic types:

■ Routed traffic entering or leaving the switch on a VLAN. (Note that ACLs do not screen traffic at the internal point where traffic moves between VLANs or subnets within the switch. Refer to "ACL Inbound and Outbound Application Points" on page 9-8.)

■ Switched or routed traffic entering the switch on a VLAN and having an IP address on the switch as the destination

You can apply one inbound ACL and one outbound ACL to each static VLAN configured on the switch. The complete range of options per VLAN includes:

■ **No ACL** assigned to a static VLAN. (In this case, all traffic entering or leaving the switch on the VLAN does so without any ACL filtering, which is the default.)

■ **One ACL** assigned to filter *either* the inbound or the outbound traffic entering or leaving the switch on a static VLAN.

■ **One ACL** assigned to filter *both* the inbound and the outbound traffic entering or leaving the switch on a static VLAN.

■ **Two different ACLs** assigned to a static VLAN; one for filtering traffic entering the switch and one for filtering traffic leaving the switch.

Note   On a given switch, after you assign an ACL to a static VLAN, the default action for all physical ports belonging to the VLAN is to deny any IP traffic that is not specifically permitted by the ACL. (This applies only in the direction of traffic flow filtered by the ACL.)

The Packet-Filtering Process

Sequential Comparison and Action. When the switch uses an ACL to fil-
ter a packet, it sequentially compares each ACE's filtering criteria to the
corresponding data in the packet until it finds a match.

For a packet with a source IP address of
18.28.156.3, the switch:

1. Compares the packet to this ACE first.

2. Since there is not a match with the first
    ACE, the switch then compares the
    packet to the second ACE, where there
    is also not a match.

3. The switch compares the packet to the
    third ACE. There is a match because
    the 0.0.0.15 mask includes the source
    IP address. The then switch denies
    (drops) the packet.

4. The packet is not compared to the
    fourth ACE.

```
ip access-list standard "1"
    permit 18.28.136.24 0.0.0.0
    permit 18.28.144.116 0.0.0.0
    deny 18.28.156.7 0.0.0.15 log
    permit 18.28.156.1 0.0.0.255
    exit
```
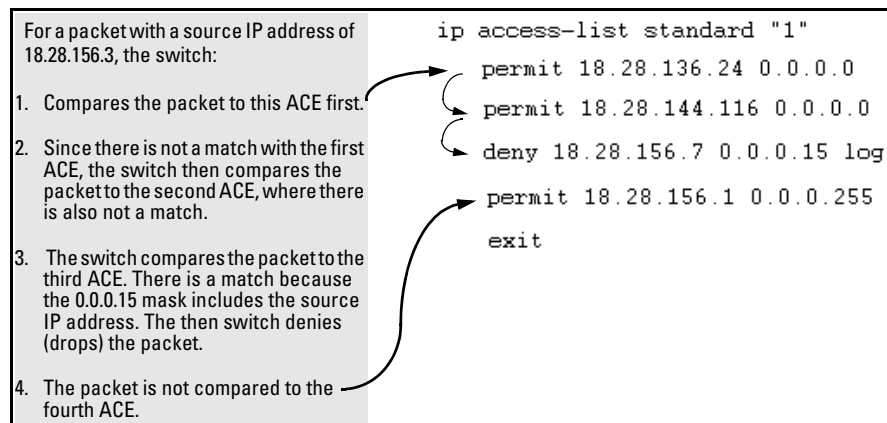
Figure 9-2. Example of Sequential Comparison

That is, the switch tries the first ACE in the list. If there is not a match, it tries
the second ACE, and so on. When a match is found, the switch invokes the
configured action for that entry (permit or drop the packet) and no further
comparisons of the packet are made with the remaining ACEs in the ACL. This
means that when the switch finds an ACE whose criteria matches a packet, it
invokes the action configured for that ACE, and any remaining ACEs in the
ACL are ignored. *Because of this sequential processing, successfully imple-*
menting an ACL depends in part on configuring ACEs in the correct order
for the overall policy you want the ACL to enforce.

Implicit Deny. If a packet does not have a match with the criteria in any of
the ACEs in the ACL, the switch denies (drops) the packet. (This is termed
implicit deny.) If you need to override the implicit deny so that any packet
that does not have a match will be permitted, then you can enter **permit any** as
the last ACE in the ACL. This directs the switch to permit (forward) any
packets that do not have a match with any earlier ACE listed in the ACL, and
prevents these packets from being filtered by the implicit deny.

Note on Implicit Deny   For ACLs configured to filter inbound packets on a VLAN, remember that Implicit Deny filters routed packets *and any bridged packets with a DA specifying the switch itself*. This operation helps to prevent management access from unauthorized IP sources.
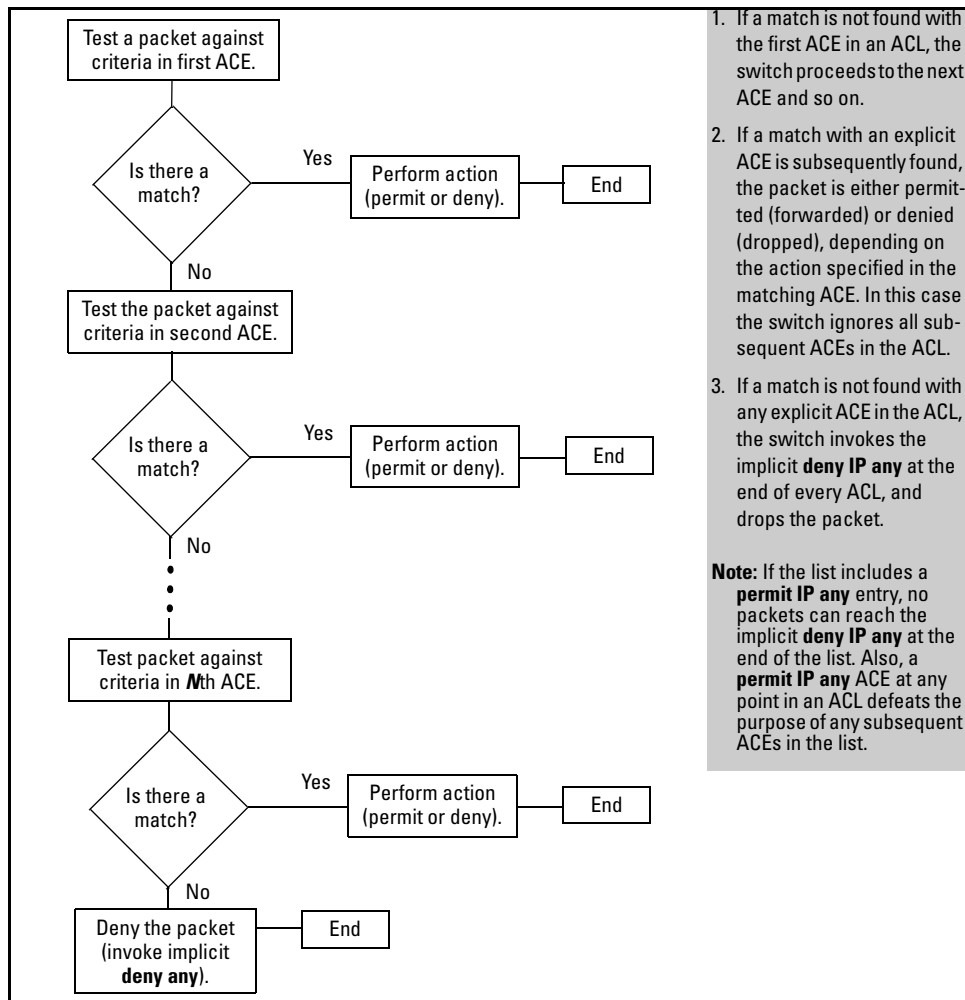
Figure 9-3. The Packet-Filtering Process in an ACL with *N* Entries (ACEs)

1. If a match is not found with the first ACE in an ACL, the switch proceeds to the next ACE and so on.

2. If a match with an explicit ACE is subsequently found, the packet is either permitted (forwarded) or denied (dropped), depending on the action specified in the matching ACE. In this case the switch ignores all subsequent ACEs in the ACL.

3. If a match is not found with any explicit ACE in the ACL, the switch invokes the implicit **deny IP any** at the end of every ACL, and drops the packet.

Note: If the list includes a **permit IP any** entry, no packets can reach the implicit **deny IP any** at the end of the list. Also, a **permit IP any** ACE at any point in an ACL defeats the purpose of any subsequent ACEs in the list.

Note

The order in which an ACE occurs in an ACL is significant. For example, if an ACL contains six ACEs, but the first ACE is a "permit IP any", then the ACL permits all IP traffic, and the remaining ACEs in the list do not apply, even if they specify criteria that would make a match with any of the traffic permitted by the first ACE.

For example, suppose you want to configure an ACL on the switch (with an ID of "100") to invoke these policies:

1. Permit all inbound traffic on VLAN 12 routed from IP address 11.11.11.42.

2. Deny *only* the inbound Telnet traffic routed from address 11.11.11.101.

3. Permit *only* inbound Telnet traffic routed from IP address 11.11.11.33.

4. Deny *all other* inbound routed traffic on VLAN 12.

The following ACL model, when assigned to inbound filtering on VLAN 12, supports the above case:

```
ProCurve(config)# show access-list config

ip access-list extended "100"
 ❶ permit ip 11.11.11.42 0.0.0.0 0.0.0.0 255.255.255.255
 ❷ deny tcp 11.11.11.101 0.0.0.0 0.0.0.0 255.255.255.255 eq 23
 ❸ permit ip 11.11.11.101 0.0.0.0 0.0.0.0 255.255.255.255
 ❹ permit tcp 11.11.11.33 0.0.0.0 0.0.0.0 255.255.255.255 eq 23
 ❺ < implicit deny IP any >

ProCurve(config)# vlan 12 ip access-group 100 in
```

1. **Permits** IP traffic routed from source address 11.11.11.42. Packets matching this criterion are permitted and will not be compared to any later ACE in the list. Packets not matching this criterion will be compared to the next entry in the list.

2. **Denies** Telnet traffic routed from source address 11.11.11.101. Packets matching this criterion are dropped and are not compared to later criteria in the list. Packets not matching this criterion are compared to the next entry in the list.

3. **Permits** any IP traffic routed from source address 11.11.11.101. Any packets matching this criterion will be permitted and will not be compared to any later criteria in the list. Because this entry comes after the entry blocking Telnet traffic from this same address, there will not be any Telnet packets to compare with this entry; they have already been dropped as a result of matching the preceding entry.

4. **Permits** Telnet traffic routed from source address 11.11.11.33. Packets matching this criterion are permitted and are not compared to any later criteria in the list. Packets not matching this criterion are compared to the next entry in the list.

5. This entry does not appear in an actual ACL, but is implicit as the last entry in every ACL. Any routed packets that do not match any of the criteria in the ACL's preceding entries will be denied (dropped), and will not cross VLAN 12.

Figure 9-4. Example of How an ACL Filters Packets

It is important to remember that this ACL (and all ACLs) include an implicit "deny IP any". That is, routed IP packets (and switched packets having the switch as the destination IP address) that the ACL does not *explicitly* permit or deny will be *implicitly* denied, and therefore dropped instead of forwarded on the VLAN. You can preempt the implicit deny by inserting a "permit IP any" at the end of an ACL, but this solution does not apply in the preceding example, where the intention is for the switch to forward only explicitly permitted packets routed on VLAN 12.

Overriding the Implicit "deny IP any". If you want an ACL to permit any routed packets that are not explicitly denied by other entries in the ACL, you can do so by configuring a **permit any** entry as the last entry in the ACL. Doing so permits any packet not explicitly denied by earlier entries.

Planning an ACL Application

Before creating and implementing ACLs, you need to define the policies you want your ACLs to enforce, and understand how your ACLs will impact your network users.

Traffic Management and Improved Network Performance

You can use ACLs to block unnecessary traffic caused by individual hosts, workgroups, or subnets, and to block user access to subnets, devices, and services. Answering the following questions can help you to design and properly position ACLs for optimum network usage.

■ What are the logical points for minimizing unwanted traffic? In many cases it makes sense to prevent unwanted traffic from reaching the core of your network by configuring ACLs to drop unwanted traffic at or close to the edge of the network. (The earlier in the network path you can block unwanted traffic, the greater the benefit for network performance.)

■ What traffic should you explicitly block? Depending on your network size and the access requirements of individual hosts, this can involve creating a large number of ACEs in a given ACL (or a large number of ACLs), which increases the complexity of your solution.

<tag_name>**Access Control Lists (ACLs) for the Series 5300xl Switches**
Planning an ACL Application

■   What traffic can you implicitly block by taking advantage of the
implicit **deny IP any** to deny traffic that you have not explicitly
permitted? This can reduce the number of entries needed in an ACL.

■   What traffic should you permit? In some cases you will need to
explicitly identify permitted traffic. In other cases, depending on your
policies, you can insert a **permit any** entry at the end of an ACL. This
means that all IP traffic not specifically matched by earlier entries in
the list will be permitted.

Security

ACLs can enhance security by blocking routed IP traffic carrying an unautho-
rized source IP address (SA). This can include:

■   Blocking access to or from subnets in your network

■   Blocking access to or from the internet

■   Blocking access to sensitive data storage or restricted equipment

■   Preventing the use of specific TCP or UDP functions (such as Telnet,
SSH, web browser) for unauthorized access

You can also enhance switch management security by using ACLs to block
bridged IP traffic that has the switch itself as the destination address (DA).

Caution   ACLs can enhance network security by blocking selected IP traffic, and can
serve as one aspect of maintaining network security. *However, because ACLs
do not provide user or device authentication, or protection from malicious
manipulation of data carried in IP packet transmissions, they should not
be relied upon for a complete security solution.*

Note   ACLs in the Series 5300XL switches do not screen non-IP traffic such as
AppleTalk and IPX.

9-17

Guidelines for Planning the Structure of an ACL

The first step in planning a specific ACL is to determine where you will apply it. (Refer to "ACL Inbound and Outbound Application Points" on page 9-8.) You must then determine the order in which you want the individual ACEs in the ACL to filter traffic.

■ The first match dictates the action on a packet. Subsequent matches are ignored.

■ On any ACL, the switch implicitly denies packets that are not explicitly permitted or denied by the ACEs configured in the ACL. If you want the switch to forward packets for which there is not a match in an ACL, add the "permit IP any" function as the last ACE in an ACL. This ensures that no packets reach the implicit "deny IP any" case.

■ Generally, you should list ACEs from the most specific (individual hosts) to the most general (subnets or groups of subnets) unless doing so permits traffic that you want dropped. For example, an ACE allowing a small group of workstations to use a specialized printer should occur earlier in an ACL than an entry used to block widespread access to the same printer.

ACL Configuration and Operating Rules

■ **Routing.** Except for any IP traffic with a DA on the switch itself, ACLs filter only routed traffic. Thus, if routing is not enabled on the switch, there is no routed traffic for ACLs to filter. (To enable routing, execute **ip routing** at the global configuration level.) For more on routing, refer to the chapter titled "IP Routing Features" in this manual.

■ **Per-Switch ACL Limits.** At a minimum an ACL must have one, explicit "permit" or "deny" Access Control Entry. You can configure up to 255 ACL assignments to VLANs, as follows:

   • Standard ACLs: Up to 99; numeric range: 1 - 99
   • Extended ACLs: Up to 100; numeric range: 100 - 199
   • Named (Extended or Standard) ACLs: Up to 255 (minus any numeric ACL assignments)
   • Total ACEs in all ACLs: 1024

■ **Implicit "deny any":** In any ACL, the switch automatically applies an implicit "deny IP any" that does not appear in **show** listings. This means that the ACL denies any packet it encounters that does not have a match with an entry in the ACL. Thus, if you want an ACL to

permit any packets that you have not expressly denied, you must enter a **permit any** or **permit ip any any** as the last ACE in an ACL. Because, for a given packet the switch sequentially applies the ACEs in an ACL until it finds a match, any packet that reaches the **permit any** or **permit ip any any** entry will be permitted, and will not encounter the "deny ip any" ACE the switch automatically includes at the end of the ACL. For an example, refer to figure 9-4 on page 9-15.

■ **Explicitly Permitting Any IP Traffic:** Entering a **permit any** or a **permit ip any any** ACE in an ACL permits all IP traffic not previously permitted or denied by that ACL. Any ACEs listed after that point do not have any effect.

■ **Explicitly Denying Any IP Traffic:** Entering a **deny any** or a **deny ip any any** ACE in an ACL denies all IP traffic not previously permitted or denied by that ACL. Any ACEs listed after that point have no effect.

■ **Replacing One ACL with Another:** The last ACL assigned for inbound ("in") or outbound ("out") packet filtering on an interface replaces any other ACL previously configured for the same purpose. For example, if you configured ACL 100 to filter inbound traffic on VLAN 20, but later, you configured ACL 112 to filter inbound traffic on this same VLAN, ACL 112 replaces ACL 100 as the ACL to use for filtering inbound traffic on VLAN 20.

■ **ACLs Operate On Static VLANs:** You can assign an ACL to any VLAN that is statically configured on the switch. ACLs do not operate with dynamic VLANs.

■ **An ACL Affects All Physical Ports in a Static VLAN:** An ACL assigned to a VLAN applies to all physical ports on the switch that belong to that VLAN, including ports that have dynamically joined the VLAN.

■ **ACLs Screen Traffic Entering or Leaving the Switch on a VLAN:** On a given VLAN, ACLs can screen inbound or outbound traffic at the point where it enters or leaves the switch. ACLs do not screen traffic moving between VLANs within the switch or between subnets in a multinetted VLAN. (See figure 9-1.)

■ **ACLs Do Not Filter Switched Traffic Unless the Switch Itself is the DA:** ACLs do not filter:

• Traffic moving between ports belonging to the same subnet

• Traffic leaving the switch with an SA on the switch itself

ACLs *do* filter switched or routed traffic having a DA on the switch.

How an ACE Uses a Mask To Screen Packets for Matches

When the switch applies an ACL to inbound or outbound traffic in a VLAN, each ACE in the ACL uses an IP address and *ACL mask* to enforce a selection policy on the packets being screened. That is, the mask determines the range of IP addresses (SA only or SA/DA) that constitute a match between the policy and a packet being screened.

What Is the Difference Between Network (or Subnet) Masks and the Masks Used with ACLs?

In common IP addressing, a network (or subnet) mask defines which part of the IP address to use for the network number and which part to use for the hosts on the network. For example:

IP Address	Mask	Network Address	Host Address
18.38.252.195	255.255.255.0	first three octets	The fourth octet.
18.38.252.195	255.255.248.0	first two octets and the left-most five bits of the third octet	The right most three bits of the third octet and all bits in the fourth octet.

Thus, the bits set to 1 in a network mask define the part of an IP address to use for the network number, and the bits set to 0 in the mask define the part of the address to use for the host number.

In an ACL, IP addresses and masks provide the criteria for determining whether to deny or permit a packet, or to pass it to the next ACE in the list. If there is a match, the deny or permit action occurs. If there is not a match, the packet is compared with the next ACE in the ACL. Thus, where a standard network mask defines how to identify the network and host numbers in an IP address, the mask used with ACEs defines which bits in a packet's IP address must match the corresponding bits in the IP address listed in an ACE, and which bits can be *wildcards*.

Rules for Defining a Match Between a Packet and an Access Control Entry (ACE)

■ For a given ACE, when the switch compares an IP address and corresponding mask in the ACE to an IP address carried in a packet:

- **A mask-bit setting of 0 ("off")** requires that the corresponding bit in the packet's IP address and in the ACE's IP address must be the same. That is, if a bit in the ACE's IP address is set to 1 ("on"), the same bit in the packet's IP address must also be 1.

- **A mask-bit setting of 1 ("on")** means the corresponding bit in the packet's IP address and in the ACE's IP address do not have to be the same. That is, if a bit in the ACE's IP address is set to 1, the same bit in the packet's IP address can be either 1 or 0 ("on" or "off").

For an example, refer to "Example of How the Mask Bit Settings Define a Match" on page 9-23.

■ In any ACE, a mask of all ones means *any* IP address is a match. Conversely, a mask of all zeros means the *only* match is an IP address identical to the host IP address specified in the ACL.

■ Depending on your network, a single ACE that allows a match with more than one source or destination IP address may allow a match with multiple subnets For example, in a network with a prefix of 31.30.240 and a subnet mask of 255.255.240.0 (the left most 20 bits), applying an ACL mask of 0.0.31.255 causes the subnet mask and the ACL mask to overlap one bit, which allows matches with hosts in two subnets: 31.30.224.0 and 31.30.240.0.

Bit Position in the Third Octet of Subnet Mask 255.255.240.0								
Bit Values	128	64	32	16	8	4	2	1
Subnet Mask Bits	1	1	1	1	n/a	n/a	n/a	n/a
Mask Bit Settings Affecting Subnet Addresses	0	0	0	**1 or 0**	n/a	n/a	n/a	n/a

This ACL supernetting technique can help to reduce the number of ACLs you need. You can apply it to a multinetted VLAN and to multiple VLANs. However, ensure that you exclude subnets that do not belong in the policy. If this creates a problem for your network, you can eliminate the unwanted match by making the ACEs in your ACL as specific as possible, and using multiple ACEs carefully ordered to eliminate unwanted matches.

■ Every IP address and mask pair (source or destination) used in an ACE creates one of the following policies:

- **Any IP address fits the matching criteria.** In this case, the switch automatically enters the IP address and mask in the ACE. For example:

  access-list 1 deny any

  produces this policy in an ACL listing:

IP Address	Mask
0.0.0.0	255.255.255.255

  This policy states that every bit in every octet of a packet's SA is a wildcard, which covers any IP address.

- **One IP address fits the matching criteria.** In this case, you provide the IP address and the switch provides the mask. For example:

  access-list 1 permit host 18.28.100.15

  produces this policy in an ACL listing:

IP Address	Mask
18.28.100.15	0.0.0.0

  This policy states that every bit in every octet of a packet's SA must be the same as the corresponding bit in the SA defined in the ACE.

- **A group of IP addresses fits the matching criteria.** In this case you provide both the IP address and the mask. For example:

  access-list 1 permit 18.28.32.1 0.0.0.31

IP Address	Mask
18.28.32.1	0.0.0.31

  This policy states that:
  – In the first three octets of a packet's SA, every bit must be set the same as the corresponding bit in the SA defined in the ACE.
  – In the last octet of a packet's SA, the first three bits must be the same as in the ACE, but the last five bits are wildcards and can be any value.

■ Unlike subnet masks, the wildcard bits in an ACL mask need not be contiguous. For example, 0.0.7.31 is a valid ACL mask. However, a subnet mask of 255.255.248.224 is not a valid subnet mask.

Example of How the Mask Bit Settings Define a Match . Assume an ACE where the second octet of the mask for an SA is 7 (the rightmost three bits are "on", or "1") and the second octet of the corresponding SA in the ACE is 31 (the rightmost five bits). In this case, a match occurs when the second octet of the SA in a packet being filtered has a value in the range of 24 to 31. Refer to table 9-1, below.

Table 9-1.   Example of How the Mask Defines a Match

Location of Octet	Bit Position in the Octet							
	128	64	32	16	8	4	2	1
SA in ACE	0	0	0	1	1	1	1	1
Mask for SA	0	0	0	0	0	1	1	1
Corresponding Octet of a Packet's SA	0	0	0	1	1	0/1	0/1	0/1

The shaded area indicates bits in the packet that must exactly match the bits in the source IP in the ACE. Wherever the mask bits are ones (wildcards), the IP bits in the packet can be any value, and where the mask bits are zeros, the IP bits in the packet must be the same as the IP bits in the ACE. **Note:** This example covers only one octet of an IP address. An actual ACE applies this method to all four octets of an IP address.

Example of Allowing Only One IP Address ("Host" Option). Suppose, for example, that you have configured the ACL in figure 9-5 to filter inbound packets on VLAN 20. Because the mask is all zeros, the ACE policy dictates that a match occurs only when the source IP address on such packets is identical to the IP address configured in the ACE.

This ACL (a standard ACL named "Fileserver") includes an ACE (Access Control Entry) that permits matches only with the packets received from IP address 18.28.252.117 (the SA). Packets from any other source do not match and are denied.

```
ip access-list standard Fileserver
  permit 18.28.252.117   0.0.0.0
  exit
```

ACE

Source IP Address (SA)

Mask (All zeros mean that a match occurs only when an IP packet's source address is identical to the source address configured in the ACE.

Inbound Packet "A" On VLAN 20
  – Destination Address: 18.35.248.184
  – Source Address: 18.28.252.117

The VLAN permits packet "A" because it s source IP address matches the source address in the ACE.

Inbound Packet "B" On VLAN 20
  – Destination Address: 18.35.248.184
  – Source Address: 18.28.252.120

The VLAN denies packet "B" because its source IP address does not match the source address in the ACE.

Figure 9-5. Example of an ACL with an Access Control Entry (ACE) that Allows Only One Source IP Address

Examples Allowing Multiple IP Addresses. Table 9-2 provides examples of how to apply masks to meet various filtering requirements.

Table 9-2.   Example of Using an IP Address and Mask in an Access Control Entry

IP Address in the ACE	Mask	Policy for a Match Between a Packet and the ACE	Allowed IP Addresses
A: 18.38.252.195	0.0.0.255	Exact match in first three octets only.	18.38.252.< 0-255 > (See row A in table 9-3, below.)
B: 18.38.252.195	0.0.7.255	Exact match in the first two octets and the leftmost five bits (248) of the third octet.	18.38.< 248-255 >.< 0-255 > (In the third octet, only the rightmost three bits are wildcard bits. The leftmost five bits must be a match, and in the ACE, these bits are all set to 1. See row B in table 9-3, below.)
C: 18.38.252.195	0.0.0.0	Exact match in all octets.	18.38.252.195 (There are no wildcard bits in any of the octets. See row C in table 9-3, below.)
D: 18.38.252.195	0.15.255.255	Exact match in the first octet and the leftmost four bits of the second octet.	18.< 32-47 >.< 0-255 >.<0-255> (In the second octet, the rightmost four bits are wildcard bits. See row D in table 9-3, below.)

Table 9-3.   Mask Effect on Selected Octets of the IP Addresses in Table 9-2

IP Addr	Octet	Mask	Octet Range	128	64	32	16	8	4	2	1
A	3	0 all bits	252	1	1	1	1	1	1	0	0
B	3	7 last 3 bits	248-255	1	1	1	1	1	0 or 1	0 or 1	0 or 1
C	4	0 all bits	195	1	1	0	0	0	0	1	1
D	2	15 last 4 bits	32-47	0	0	1	0	0 or 1	0 or 1	0 or 1	0 or 1

Shaded areas indicate bit settings that must be an exact match.

If there is a match between the policy in the ACE and the IP address in a packet, then the packet is either permitted or denied, according to how the ACE is configured. If there is not a match, the next ACE in the ACL is then applied to the packet. The same operation applies to a destination IP address (DA) used in an extended ACE. (Where an ACE includes both source and destination IP addresses, there is one IP-address/ACL-mask pair for the source address, and another IP-address/ACL-mask pair for the destination address. See "Configuring and Assigning an ACL" on page 9-25.)

CIDR Notation. For information on using CIDR notation to specify ACL masks, refer to "Using CIDR Notation To Enter the ACL Mask" on page 9-32.

Configuring and Assigning an ACL

ACL Feature	Page
Configuring and Assigning a Numbered, Standard ACL	9-33
Configuring and Assigning a Numbered, Extended ACL	9-38
Configuring a Named ACL	9-44
Enabling or Disabling ACL Filtering	9-46

Overview

General Steps for Implementing ACLs

1. Configure at least one ACL. This creates and stores the ACL(s) in the switch configuration.

2. Assign an ACL. This applies the ACL to either the inbound or outbound traffic on a designated VLAN.

3. Enable IP routing. Except for instances where the switch is the destination, assigned ACLs screen IP traffic only when routing is enabled on the switch.

Caution Regarding the Use of Source Routing

Source routing is enabled by default on the switch and can be used to override ACLs. For this reason, if you are using ACLs to enhance network security, the recommended action is to disable source routing on the switch. To do so, execute **no ip source-route**.

Types of ACLs

■ **Standard ACL:** Uses only a packet's source IP address as a criterion for permitting or denying the packet. For a standard ACL ID, use either a unique numeric string in the range of 1-99 or a unique name string of up to 64 alphanumeric characters.

■ **Extended ACL:** Offers the following criteria as options for permitting or denying a packet:

• Source IP address

• Destination IP address

• TCP or UDP criteria

For an extended ACL ID, use either a unique number in the range of 100-199 or a unique name string of up to 64 alphanumeric characters.

You should carefully plan your ACL application before configuring specific ACLs. For more on this topic, refer to "Planning an ACL Application" on page 9-16.

ACL Configuration Structure

After you enter an ACL command, you may want to inspect the resulting configuration. This is especially true where you are entering multiple ACEs into an ACL. Also, it will be helpful to understand the configuration structure when using later sections in this chapter.

The basic ACL structure includes three elements:

1. List type and name: This identifies the ACL as **standard** or **extended** and shows the ACL name.

2. One or more deny/permit list entries (ACEs): One entry per line.

Element	Stnd	Ext	Notes
ID Range	1 - 99	100 - 199	You can also use an alphanumeric name of up to 64 characters, including spaces.
Minimum ACEs per ACL	1		
Maximum ACEs Per ACL and per Switch	1024		The switch allows a total of 1024 ACEs across all ACLs.

3. Implicit **deny any**: Where an ACL is in use, the switch denies any packets that do not have a match with the ACEs explicitly configured in the ACL. The implicit **deny any** does not appear in ACL configuration listings, but

always functions when the switch uses an ACL to filter packets. (You cannot delete the implicit "deny any", but you can supersede it with a "permit any" statement.)

Standard ACL Structure

Individual ACEs in a standard ACL include only a permit/deny "type" statement, the source IP addressing, and an optional **log** command (available with "deny" statements).

```
ip access-list < type > "< id-string >"
    permit host < source-ip-address >
    deny < source-ip-address > < acl-mask > [ log ]
    .
    .
    .
    permit any
    exit
```

Figure 9-6. Example of the General Structure for a Standard ACL

For example, figure 9-7 shows how to interpret the entries in a standard ACL.
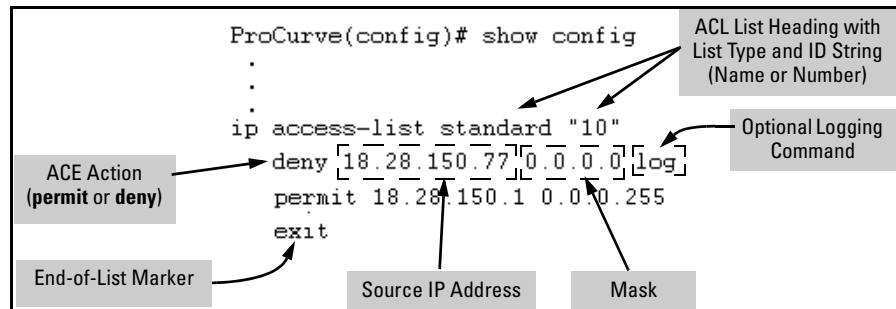
Figure 9-7. Example of a Displayed Standard ACL Configuration with Two ACEs

Extended ACL Configuration Structure

Individual ACEs in an extended ACL include:

- A permit/deny "type" statement

- Source IP addressing

- Optional TCP or UDP port type with optional source port ID and operator and/or optional destination port ID and operator

- Destination IP addressing

- Optional ACL **log** command

```
ip access-list < type > "< id-string >"< permit | deny > ip
      < source-ip-address > < source-acl-mask >
      < destination-ip-address > < destination-acl-mask >[log]

   < permit | deny > tcp
      < source-ip-address > < source-acl-mask > [< operator > < port-id >]
      < destination-ip-address > < destination-acl-mask > [< operator > < port-id >] [log]

   < permit | deny > udp
      < source-ip-address > < source-acl-mask > [< operator > < port-id >]
      < destination-ip-address > < destination-acl-mask > [< operator > < port-id >] [log]
   .
   .
   .
```

Note: The optional log function appears only with "deny" aces.
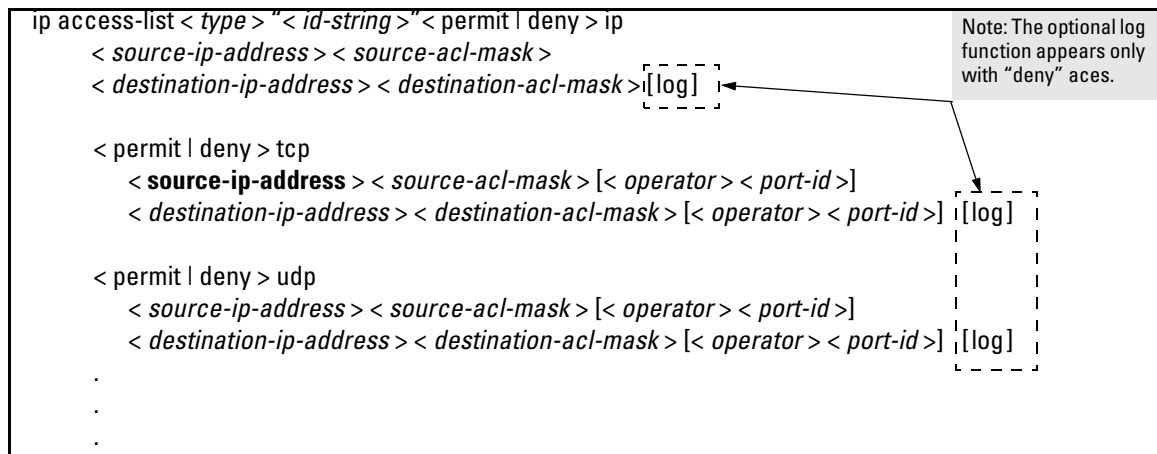
Figure 9-8. General Structure for an Extended ACL

For example, figure 9-9 shows how to interpret the entries in an extended ACL.


```
ProCurve(config)# show config
   .
   .           Protocol Types
   .
ip access-list extended "101"
    permit ip 18.38.130.55 0.0.0.0 18.38.130.240 0.0.0.0
    permit tcp 0.0.0.0 255.255.255.255 0.0.0.0 255.255.255.255 eq 23
    permit tcp 18.38.131.14 0.0.0.0 eq 80 0.0.0.0 255.255.255.255 eq 3871
    deny tcp 0.0.0.0 255.255.255.255 0.0.0.0 255.255.255.255 eq 80 log
    deny udp 18.42.120.19 0.0.0.0 eq 69 18.38.140.44 0.0.0.0 range 3680 3690 log
    deny udp 0.0.0.0 255.255.255.255 18.38.99.121 0.0.0.0 log
    exit
```

Callouts:

ACL List Heading with List Type and ID String (Name or Number)

Specifies all destination IP addresses.

Denies TCP Port 80 traffic to any destination from any source.

End-of-List Marker

ACE Action (**permit** or **deny**)

Source IP Addresses and Masks.

Upper entry denies certain UDP packets from a single host. Lower entry denies all UDP packets from all hosts.

Optional Source UDP or TCP Operator and Port Number

In this case, the ACL specifies UDP port 69 packets coming from the source IP address.

Destination IP Address and Mask

Optional Destination UDP or TCP Operator and Range of Port Numbers

In this case, the ACL specifies UDP port numbers 3680-3690.

Figure 9-9. Example of a Displayed Extended ACL Configuration

ACL Configuration Factors

The Sequence of Entries in an ACL Is Significant

When the switch uses an ACL to determine whether to permit or deny a packet on a particular VLAN, it compares the packet to the criteria specified in the individual Access Control Entries (ACEs) in the ACL, beginning with the first ACE in the list and proceeding sequentially until a match is found. When a match is found, the switch applies the indicated action (permit or deny) to the packet. This is significant because, once a match is found for a packet, subsequent ACEs in the same ACL will not be used for that packet, regardless of whether they match the packet.

For example, suppose that you have applied the ACL shown in figure 9-10 to inbound traffic on VLAN 1 (the default VLAN):

```
1   ip access-list extended "101"   Source              Destination

2     deny ip 18.28.235.10 0.0.0.0 0.0.0.0 255.255.255.255

3     deny ip 18.28.245.89 0.0.0.0 0.0.0.0 255.255.255.255

4     permit tcp 18.28.18.100 0.0.0.0 18.28.237.1 0.0.0.0

5     deny tcp 18.28.18.100 0.0.0.0 0.0.0.0 255.255.255.255

6     permit ip 0.0.0.0 255.255.255.255 0.0.0.0 255.255.255.255

7     exit
```

Source and Destination IP Addresses for the ACE in line 4 of the ACL.

Following the last explicit ACE in the ACL there is always an implicit "deny any". However, in this case it will not be used because the last, explicit permit statement allows all IP packets that earlier ACEs have not already permitted or denied.

Figure 9-10. Example of a Standard ACL that Permits All Traffic Not Implicitly Denied

Table 9-4.   Effect of the Above ACL on Inbound Traffic in the Assigned VLAN

Line #	Action
1	Shows list type (extended) and ID (101).
2	A packet from IP source address 18.28.235.10 will be denied (dropped). This line filters out all packets received from 18.28.235.10. As a result, IP traffic from that device will not be routed and packets from that device will not be compared against any later entries in the list.
3	A packet from IP source 18.28.245.89 will be denied (dropped). This line filters out all packets received from 18.28.245.89. As the result, IP traffic from that device will not be routed and packets from that device will not be compared against any later entries in the list.
4	A packet from TCP source address 18.28.18.100 with a destination address of 18.28.237.1 will be permitted (forwarded). Since no earlier lines in the list have filtered TCP packets from 18.28.18.100 and destined for 18.28.237.1, the switch will use this line to evaluate such packets. Any packets that meet this criteria will be forwarded. (Any packets that do not meet this TCP source-destination criteria are not affected by this line.)
5	A packet from TCP source address 18.28.18.100 to **any** destination address will be denied (dropped). Since, in this example, the intent is to block TCP traffic from 18.28.18.100 to any destination **except** the destination stated in line 4, this line must follow line 4. (If their relative positions were exchanged, all TCP traffic from 18.28.18.100 would be dropped, including the traffic for the 18.28.18.1 destination.)
6	Any packet from any IP source address to any destination address will be permitted (forwarded). The only traffic to reach this line will be IP packets not specifically permitted or denied in the earlier lines.
n/a	The "implicit deny any any" is a function automatically added as the last action in all ACLs. It denies (drops) any IP traffic from any source to any destination that has not found a match with earlier entries in the list. In this example, line 6 permits (forwards) any IP traffic not already permitted or denied by the earlier entries in the list, so there is no traffic remaining for action by the "implicit deny any any" function.
7	Indicates the end of the ACL.

In Any ACL, There Will Always Be a Match

As indicated in figure 9-10, the switch automatically uses an implicit "deny IP any" (Standard ACL) or "deny IP any any" (Extended ACL) as the last ACE in any ACL. This means that if you configure the switch to use an ACL for filtering either inbound or outbound traffic on a VLAN, any packets not specifically permitted or denied by the explicit entries you create will be denied by the implicit "deny" action. Note that if you want to preempt the implicit "deny" action, insert an explicit **permit any** or **permit ip any any** as the last line of the ACL.

A Configured ACL Has No Effect Until You Apply It to an Interface

The switch stores ACLs in the configuration file. Thus, until you actually assign an ACL to a VLAN interface, it is present in the configuration, but not used.

You Can Assign an ACL Name or Number to a VLAN Even if the ACL Does Not Yet Exist in the Switch's Configuration

In this case, if you subsequently create an ACL with that name or number, the switch automatically applies each ACE as soon as you enter it in the running-config file. Similarly, if you modify an existing ACE in an ACL you already applied to a VLAN, the switch automatically implements the new ACE as soon as you enter it. (See "General ACL Operating Notes" on page 9-63.) The switch allows a maximum of 255 ACLs in any combination of numeric and alphanumeric names, and determines the total from the number of unique ACL names in the configuration. For example, if you configure two ACLs, but assign only one of them to a VLAN, the ACL total is two, for the two unique ACL names. If you then assign the name of a nonexistent ACL to a VLAN, the new ACL total is three, because the switch now has three unique ACL names in its configuration.

Using the CLI To Create an ACL

Command	Page
access-list (standard ACLs)	9-33
access-list (extended ACLs)	9-38
ip access-list (named ACLs)	9-44

You can use either the switch CLI or an offline text editor to create an ACL. This section describes the CLI method, which is recommended for creating short ACLs. (To use the offline method, refer to "Editing ACLs and Creating an ACL Offline" on page 9-53.)

General ACE Rules

These rules apply to all ACEs you create or edit using the CLI:

■   ACEs are placed in an ACL according to the sequence in which you enter them (last entered, last listed).

■   You can use the CLI to delete an ACE from anywhere in a given ACL by using the "no" form of the command to enter that ACE. However, when you use the CLI to add an ACE, the new entry is always placed *at the end of the ACL.*

■   Duplicate ACEs are allowed in an ACL. However, multiple instances of an ACE have no effect on filtering because the first instance preempts any subsequent duplicates.

For more information, refer to "Editing ACLs and Creating an ACL Offline" on page 9-53.

Using CIDR Notation To Enter the ACL Mask

You can use CIDR (Classless Inter-Domain Routing) notation to enter ACL masks. The switch interprets the bits specified with CIDR notation as the IP address bits in an ACL and the corresponding IP address bits in a packet. The switch then converts the mask to inverse notation for ACL use.

Table 9-5.   Examples of CIDR Notation for Masks

IP Address Used In an ACL with CIDR Notation	Resulting ACL Mask	Meaning
18.38.240.125/15	0.1.255.255	The leftmost 15 bits must match; the remaining bits are wildcards.
18.38.240.125/20	0.0.15.255	The leftmost 20 bits must match; the remaining bits are wildcards.
18.38.240.125/21	0.0.7.255	The leftmost 21 bits must match; the remaining bits are wildcards.
18.38.240.125/24	0.0.0.255	The leftmost 24 bits must match; the remaining bits are wildcards.
18.38.240.125/32	0.0.0.0	All bits must match.

Configuring and Assigning a Numbered, Standard ACL

This section describes how to configure numbered, standard ACLs.

■   To configure named ACLs, refer to "Configuring a Named ACL" on page 9-44.

■   To configure extended, numbered ACLs, refer to "Configuring and Assigning a Numbered, Extended ACL" on page 9-38.

A standard ACL uses only source IP addresses in its ACEs. This type of ACE is useful when you need to:

■   Permit or deny traffic based on source IP address only.

■   Quickly control the IP traffic from a specific address. This allows you to isolate traffic problems generated by a specific device, group of contiguous devices, or a subnet threatening to degrade network performance. This gives you an opportunity to troubleshoot without sacrificing performance for users outside of the problem area.

You can configure up to 255 standard ACL assignments, depending on how many extended ACL assignments are already configured. (The switch allows a maximum of 255 unique ACL identities; standard and extended combined.) You can identify each standard ACL with a number in the range of 1 - 99, or an alphanumeric string of up to 64 characters. The CLI command process for using an alphanumeric string to name an ACL differs from the command process for a numeric name. For a description of naming an ACL with an alphanumeric character string, refer to "Configuring a Named ACL" on page 9-44. To view the command differences, refer to table 9-1, "Comprehensive Command Summary" on page 9-4.

Note   For a summary of ACL commands, refer to table 9-1, "Comprehensive Command Summary", on page 9-4.

Syntax: [no] access-list

> *Creates an ACE in the specified (1-99) access list and indicates the action (deny or permit) to take on a packet if there is a match between the packet and the criterion in the entry. If the ACL does not already exist, this command creates the specified ACL and its first ACE. To create a named ACL, refer to "Configuring a Named ACL" on page 9-44.*

< 1-99 >

> *Specifies the ACL ID number. The switch interprets an ACL with a value in this range as a standard ACL.*
>
> ***Note:*** *To create an access list with an alphanumeric name (**name-str**) instead of a number, refer to "Configuring a Named ACL" on page 9-44.*

< deny | permit >

> *Specifies whether to deny (drop) or permit (forward) a packet that matches the ACE criteria.*

< any | host < *src-ip-addr* > | *ip-addr* / *mask-length* >

- **any**—*Performs the specified action on any IP packet. Use this criterion to designate packets from any IP address.*

- **host** < *host ip-address* >—*Performs the specified action on any IP packet having the < host ip-address > as the source. Use this criterion to designate packets from a single IP address.*

- *IP-addr* / *mask-length* — *Performs the specified action on any IP packet having a source address within the range defined by either*

  > **<** *src-ip-addr* / *cidr-mask-bits* **>**
  > *or*
  > **<** *src-ip-addr* < *mask* **>>**

  Use this criterion to filter packets received from either a subnet or a group of IP addresses. The mask can be in either dotted-decimal format or CIDR format with the number of significant bits. Refer to "Using CIDR Notation To Enter the ACL Mask" on page 9-32.

The mask is applied to the IP address in the ACL to define which bits in a packet's source IP address must exactly match the IP address configured in the ACL and which bits need not match. Note that specifying a group of contiguous IP addresses may require more than one ACE. For more on how masks operate in ACLs, refer to "How an ACE Uses a Mask To Screen Packets for Matches" on page 9-20.

[log]

Optionally generates an ACL log message if:

- *The action is* **deny***.*
- *There is a match.*
- *ACL logging is enabled on the switch. (Refer to "Enable ACL "Deny" Logging" on page 9-59.)*

(Use the debug command to direct ACL logging output to the current console session and/or to a Syslog server. Note that you must also use the **logging < i***p-addr* **>** *command to specify the IP addresses of Syslog servers to which you want log messages sent. See also "Enable ACL "Deny" Logging" on page 9-59.)*

Syntax: vlan < *vid* > ip access-group < *ASCII-STR* > < in | out >

Assigns an ACL, designated by an ACL ID (< ***ASCII-STR*** *>), to a VLAN.*

Example of a Standard ACL. Suppose you wanted to configure a standard ACL and assign it to filter inbound traffic on VLAN 10 in a particular switch:

- The ID you selected for this ACL is "50".

- You want the ACL to deny IP traffic from all hosts except these three:
  - 18.128.100.10
  - 18.128.100.27
  - 18.128.100.14

```
ProCurve(config)# access-list 50 permit host 18.128.100.10
ProCurve(config)# access-list 50 permit host 18.128.100.27
ProCurve(config)# access-list 50 permit host 18.128.80.14
ProCurve(config)# vlan 10 ip access-group 50 in
ProCurve(config)# write mem
ProCurve(config)# show config

Startup configuration:

; J4850A Configuration Editor; Created on release #E.07.2X

hostname "ProCurve"
cdp run
module 1 type J4820A
ip routing
snmp-server community "public" Unrestricted
ip access-list standard "50"
    permit 18.128.100.10 0.0.0.0
    permit 18.128.100.27 0.0.0.0
    permit 18.128.80.14 0.0.0.0
    exit
vlan 1
    name "DEFAULT_VLAN"
    untagged A1-A6,A19-A24
    ip address 15.30.248.180 255.255.248.0
    no untagged A7-A18
    exit
vlan 10
    name "VLAN_10"
    untagged A7-A12
    ip address 13.28.227.10 255.255.248.0
    ip access-group "50" in
    exit
```

• Permits IP traffic from the indicated IP address. Since, for this example, ACL 50 is a new list, this command also creates the ACL.

• Permits IP traffic from the indicated IP address.

• The **deny any** that the switch implicitly includes in all standard ACLs denies IP packets from IP sources not included in the above three commands.

Show config lists any ACLs and ACL assignments configured in the startup-config.

ACL "50" is listed in the switch configuration.

ACL "50" is assigned to filter inbound traffic on VLAN 10.

Note: To enable traffic filtering with an ACL assigned to a VLAN such as the one shown in this example, IP routing must be enabled on the switch. Otherwise, no ACL filtering occurs.

Figure 9-11. Example of Configuring a Standard ACL To Permit Only Traffic from Specific IP Addresses

In a situation opposite to the above, suppose that you wanted to deny inbound IP traffic received on VLAN 20 from 18. 128.93.17 and 18.130.93.25, but permit all other IP traffic on this VLAN. The next ACL achieves this:

```
ProCurve(config)# access-list 60 deny host 18.128.93.17
ProCurve(config)# access-list 60 deny host 18.28.93.25
ProCurve(config)# access-list 60 permit any
ProCurve(config)# vlan 20 ip access-group 60 in
ProCurve(config)# write mem
ProCurve(config)# show config

Startup configuration:

; J4850A Configuration Editor; Created on release #E.07.2X

hostname "ProCurve"
cdp run
module 1 type J4820A
ip routing
snmp-server community "public" Unrestricted
ip access-list standard "60"
   deny 18.128.93.17 0.0.0.0
   deny 18.28.93.25 0.0.0.0
   permit 0.0.0.0 255.255.255.255
   exit
vlan 1
   name "DEFAULT_VLAN"
   untagged A1-A6,A19-A24
   ip address 15.30.248.180 255.255.248.0
   no untagged A7-A18
   exit
vlan 10
   name "VLAN_10"
   untagged A7-A12
   ip address 13.28.227.10 255.255.248.0
   exit
vlan 20
   name "VLAN_20"
   untagged A13-A18
   ip address 13.30.227.10 255.255.248.0
   ip access-group "60" in
   exit
```

- Denies IP traffic from the indicated IP address. Since, for this example, ACL 60 is a new list, this command also creates the ACL.
- Denies IP traffic from the indicated IP address.
- Permits IP traffic from all sources. (Traffic from the IP sources in the first two lines is already filtered and dropped.) The **deny any** with which the switch implicitly concludes all ACLs is preempted by this line.

Show config lists any ACLs and ACL assignments configured in the startup-config.

ACL "60" is listed in the switch configuration.

ACL "60" is assigned to filter inbound traffic on VLAN 20.

Note: To enable traffic filtering with an ACL assigned to a VLAN such as the one shown in this example, IP routing must be enabled on the switch. Otherwise, no ACL filtering will occur.

Figure 9-12. Example of Configuring a Standard ACL To Deny Inbound Traffic from Specific IP Addresses

Configuring and Assigning a Numbered, Extended ACL

This section describes how to configure numbered, extended ACLs.

■ To configure named ACLs, refer to "Configuring a Named ACL" on page 9-44.

■ To configure standard, numbered ACL, refer to "Configuring and Assigning a Numbered, Standard ACL" on page 9-33.

While standard ACLs use only source IP addresses for filtering criteria, extended ACLs allow multiple ACE criteria. This enables you to more closely define your IP packet-filtering criteria. These criteria include:

■ Source and destination IP addresses (required), in one of the following options:
  • Specific host IP
  • Subnet or group of IP addresses
  • Any IP address

■ IP protocol (IP, TCP, or UDP)

■ Source TCP or UDP port (if the IP protocol is TCP or UDP)

■ Destination TCP or UDP port (if the IP protocol is TCP or UDP)

■ TCP or UDP comparison operator (if the IP protocol is TCP or UDP)

You can configure up to 100 extended ACLs with a numeric name in the range of 100 -199. You can also configure extended ACLs with alphanumeric names. (Refer to "Configuring a Named ACL" on page 9-44.) The switch allows a maximum of 255 ACLs in any combination of numeric and alphanumeric names, and determines the total from the number of unique ACL names in the configuration. For example, if you configure two ACLs, but assign only one of them to a VLAN, the ACL total is two, for the two unique ACL names. If you then assign the name of a nonexistent ACL to a VLAN, the new ACL total is three, because the switch now has three unique ACL names in its configuration. (The switch allows up to 1024 ACEs total in all ACLs.)

Note   For a summary of ACL commands, refer to table 9-1, "Comprehensive Command Summary", on page 9-4.

Syntax: [no] access-list

> *Creates an ACE in the specified (100-199) access list and:*
> - *Indicates the action (deny or permit) to take on a packet if there is a match between the packet and the criteria in the complete ACE.*
> - *Specifies the packet protocol type (IP, TCP, or UDP).*
> - *Specifies the source and destination addressing options described in the remainder of this section.*
> - *Allows optional ACL logging where a packet has a match with a* **deny** *ACE.*
>
> *If the ACL does not already exist, this command creates the specified ACL and its first ACE. If the ACL already exists, this command adds a new, explicit ACE to the end of the ACL. For a match to occur, the packet must have the source and destination IP addressing criteria specified by this command, as well as any protocol-specific (TCP or UDP port number) criteria specified by the command. To create a named ACL, refer to "Configuring a Named ACL" on page 9-44.*

< 100-199 >

> *Specifies the ACL ID number. The switch interprets an ACL with a value in this range as an extended ACL.*
>
> ***Note:*** *To create an access list with an alphanumeric name instead of a number, refer to "Configuring a Named ACL" on page 9-44.*

< deny | permit >

> Specifies whether to deny (drop) or permit (forward) a packet that matches the ACE criteria.

< ip | tcp | udp >

> *Specifies the packet protocol type required for a match:*
> - **ip** — *any IP packet*
> - **tcp** — *only tcp packets*
> - **udp** — *only udp packets*

< any | host < *src-ip-addr* > | *ip-addr/mask -length* >

In an extended ACL, this parameter defines the source IP address (SA) that a packet must carry in order to have a match with the ACE.

- **any** — *Specifies all inbound IP packets.*
- **host** < ***src-ip-addr*** > — *Specifies only inbound packets from a single IP address. Use this option when you want to match only the IP packets from one source IP address.*
- ***src-ip-addr/mask-length*** — *Performs the specified action on any IP packet having a source address within the range defined by either*

  < *src-ip-addr / cidr-mask-bits* >
  or
  < *src-ip-addr < mask >* >

  Use this criterion to filter packets received from either a subnet or a group of IP addresses. The mask can be in either dotted-decimal format or CIDR format with the number of significant bits. Refer to "Using CIDR Notation To Enter the ACL Mask" on page 9-32.
  The mask is applied to the IP address in the ACL to define which bits in a packet's source IP address must exactly match the IP address configured in the ACL and which bits need not match. Note that specifying a group of contiguous IP addresses may require more than one ACE. For more on how masks operate in ACLs, refer to "How an ACE Uses a Mask To Screen Packets for Matches" on page 9-20.

[*operator* < *src-port tcp/udp-id* >]

In an extended ACL where you have selected either **tcp** *or* **udp** *as the packet protocol type (see above), you can optionally use a TCP or UDP source port number or range of numbers to further define the criteria for a match. To specify a TCP or UDP port number, (1) select a comparison operator from the following list and (2) enter the port number or a well-known port name.*

Comparison Operators:

- **eq** < *tcp/udp-port-nbr* > — *"Equal To"; to have a match with the ACE entry, the TCP or UDP source port number in a packet must be equal to* < *tcp/udp-port-nbr* >.

- **gt** < *tcp/udp-port-nbr* > — *"Greater Than"; to have a match with the ACE entry, the TCP or UDP source port number in a packet must be greater than* < *tcp/udp-port-nbr* >.

- **lt** < *tcp/udp-port-nbr* > — *"Less Than"; to have a match with the ACE entry, the TCP or UDP source port number in a packet must be less than* < *tcp/udp-port-nbr* >.

- **neq** < *tcp/udp-port-nbr* > — *"Not Equal"; to have a match with the ACE entry, the TCP or UDP source port number in a packet must not be equal to* < *tcp/udp-port-nbr* >.

- **range** < *start-port-nbr* > < *end-port-nbr* > — *To have a match with the ACE entry, the TCP or UDP source port number in a packet must be in the range* < *start-port-nbr* > < *end-port-nbr* >.

Port Number or Well-Known Port Name:

Use the TCP or UDP port number required by your application. The switch also accepts these well-known TCP or UDP port names as an alternative to their corresponding port numbers:

- *TCP*: bgp, dns, ftp, http, imap4, ldap, nntp, pop2, pop3, smtp, ssl, telnet

- *UDP*: bootpc, dns, ntp, radius, radius-old, rip, snmp, snmp-trap, tftp

To list the above names, press the **[Shift] [?]** *key combination after entering an operator. For a comprehensive listing of port numbers, visit www.iana.org/assignments/port-numbers.*

< any | host < *dest-ip-addr* > | *ip-addr/mask*-length >

In an extended ACL, this parameter defines the destination IP address (DA) that a packet must carry in order to have a match with the ACE. The options are the same as shown for < *src-ip-addr* >.

[< *dest-port tcp/udp-id* >]

In an extended ACL, this parameter defines the TCP or UDP destination port number a packet must carry in order to have a match with the extended ACE. The options are the same as shown above on the preceding page for the source IP address.

[log]

Optional; generates an ACL log message if:

- *The action is* **deny.** *(This option is not configurable for Permit.)*
- *There is a match.*
- *ACL logging is enabled on the switch. (Refer to "Enabling ACL Logging on the Switch" on page 9-60)*

Syntax: vlan < *vid* > ip access-group < *list-#* | *ascii-str* > < in | out >

Assigns an ACL, designated by an ACL list number or ASCII string (alphanumeric list name), to a VLAN to filter either inbound or outbound IP traffic on that VLAN. To configure named ACLs, refer to "Configuring a Named ACL" on page 9-44.

Example of an Extended ACL. Suppose that you want to implement these policies on a Series 5300XL switch configured for IP routing and membership in VLANs 10, 20, and 30:

A. Permit Telnet traffic from 10.10.10.44 to 10.10.20.78, deny all other IP traffic from network 10.10.10.0 (VLAN 10) to 10.10.20.0 (VLAN 20), and permit all other IP traffic from any source to any destination. (See "A" in figure 9-13, below.)

B. Permit FTP traffic from IP address 10.10.20.100 (on VLAN 20) to 10.10.30.55 (on VLAN 30). Deny FTP traffic from other hosts on network10.10.20.0 to any destination, but permit all other traffic.
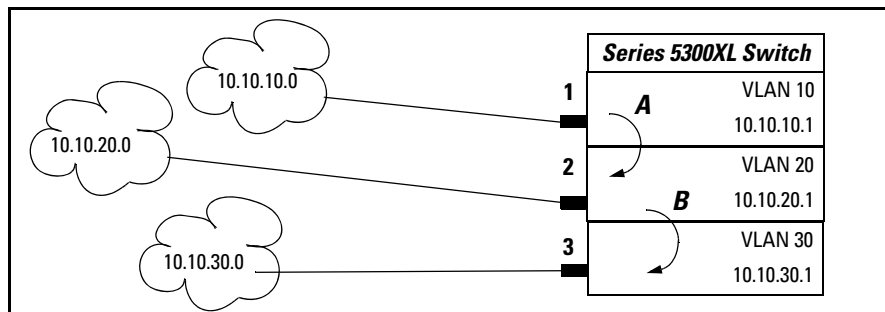
Figure 9-13. Example of an Extended ACL

A *(Refer to figure 9-13, above.)*

```
ProCurve(config)# access-list 110 permit tcp host 10.10.10.44
                  host 10.10.20.78 eq telnet
ProCurve(config)# access-list 110 deny ip 10.10.10.1 0.0.0.255 10.10.20.1 0.0.0.255
ProCurve(config)# access-list 110 permit ip any any
ProCurve(config)# vlan 10 ip access-group 110 in
```

B *(Refer to figure 9-13, above.)*

```
ProCurve(config)# access-list 120 permit tcp host 10.10.20.100
                  host 10.10.30.55 eq ftp
ProCurve(config)# access-list 120 deny tcp any any eq ftp
ProCurve(config)# access-list 120 permit ip any any
ProCurve(config)# vlan 20 ip access-group 120 in
```

```
ProCurve(config)# ip routing
ProCurve(config)# write mem
```

Enabling ip routing activates ACL operation on routed traffic.

Executing **write memory** saves the configuration changes to the startup-config file.

Figure 9-14. Example of Configuration Commands for an Extended ACL

Configuring a Named ACL

You can use the "Named ACL" context to configure a standard or extended ACL with an alphanumeric name instead of a number. Note that the command structure for configuring a named ACL differs from that for a numbered ACL.

Syntax: ip access-list standard < *name-str* | 1-99 >
          < deny | permit >
          < any | host < *src-ip-addr* > | *ip-addr / mask-length* >
          [log]

          ip access-list extended < **name-str** | 100-199 >
            < deny | permit > ip
            < any | host < *src-ip-addr* > | *ip-addr / mask-length* >
            < any | host < *dest-ip-addr* > | *ip-addr / mask-length* >
            [log]

          ip access-list extended < *name-string* >
          < deny | permit > < tcp | udp >
              < any | host < *src-ip-addr* > | *ip-addr / mask-length* >
                  [*oper* < *src-port tcp/udp-id* >]
              < any | host < *dest-ip-addr* > | *ip-addr / mask-length* >
                  [*oper* < *dest-port tcp/udp-id* >]
            [log]

              These commands create an ACE in the named ACL list and:
              - *Indicate the action (deny or permit) to take on a packet if there is a match between a packet and the criteria in the complete ACE.*
              - *Specify the packet protocol type (IP, TCP, or UDP) and (if TCP or UDP) the comparison operator.*
              - *Specify the source and destination addressing options required for a match.*
              - *Allow optional ACL logging where a packet has a match with a* **deny** *ACE. The* **log** *option does not appear when* **permit** *is the action.*

              If the ACL does not already exist, these commands create the specified ACL and its first ACE. If the ACL already exists, these commands add a new, explicit ACE to the end of the ACL. For a match to occur, the packet must have the source and destination IP addressing criteria specified by this command, as well as any protocol-specific (TCP or UDP port number) criteria specified by the command.

< *name-str* | 1-99 | 100-199 >

> *Consists of an alphanumeric string of up to 64 case-sensitive characters. If you include a space in the string, you must also enclose the string with quotes. For example,* **"ACL # 1"**. *You can also enter numbers in the ranges associated with standard (1-99) and extended (100-199) ACLs.*

> *For explanations of the individual parameters in the preceding syntax statements, refer to the syntax descriptions under "Configuring and Assigning a Numbered, Standard ACL" on page 9-33 or "Configuring and Assigning a Numbered, Extended ACL" on page 9-38.*

For example, figure 9-15 shows the commands for creating an ACL in the "Named ACL" context with these parameters:

ACL Name:	VLAN 10 Inbound
Action:	Deny
Protocol:	TCP
Source IP Address and Mask	10.10.20.100 0.0.0.0
Destination IP Address and Mask	10.10.10.1 0.0.0.255
Protocol Operator and Port Number at Destination	eq telnet

```
ProCurve(config)# ip access-list extended "VLAN 10 Inbound"
ProCurve(config-ext-nacl)# permit tcp host 10.10.20.100 10.10.10.1 0.0.0.255
                           eq telnet
ProCurve(config-ext-nacl)# exit
ProCurve(config)# write mem
ProCurve(config)# show config

  Startup configuration:

  ; J4850A Configuration Editor; Created on release #E.07.2X

  hostname "ProCurve"
  cdp run
  module 1 type J4820A
  ip default-gateway 13.30.248.1
  ip routing
  logging 13.28.227.2
  snmp-server community "public" Unrestricted

  ip access-list extended "VLAN 10 Inbound"
     permit tcp 10.10.20.100 0.0.0.0 10.10.10.1 0.0.0.255 eq 23
     exit
  .
  .
  .
```

Command Entry for Source IP Address and Mask

Command Entry for Destination IP Address and Mask

Configured Source IP Address and Mask

Configured Destination IP Address and Mask

Figure 9-15. Using the "Named ACL" Context To Configure an ACL

Enabling or Disabling ACL Filtering on a VLAN

For a given interface, you can configure one ACL to filter inbound traffic and one ACL to filter outbound traffic. You can also use the same ACL for both inbound and outbound traffic, and for assignment to multiple VLANs. For limits and operating rules, refer to "ACL Configuration and Operating Rules" on page 9-18.

Syntax: [no] vlan < *vid* > ip access-group < *ascii-string* > < in | out >
     where: < *ascii-string* > = either a ACL name or an ACL ID number.

> *Assigns an ACL to a VLAN. You can use either the global configuration level or the VLAN context level to assign an ACL to a VLAN or remove an ACL from a VLAN.*
>
> ***Note:*** *The switch allows you to assign a nonexistent ACL name or number to a VLAN. In this case, if you subsequently configure an ACL with that name or number, it will automatically become active on the assigned VLAN. Also, if you delete an assigned ACL from the switch without subsequently using the "**no**" form of this command to remove the assignment to a VLAN, the ACL assignment remains and will automatically activate any new ACE if you create with the same ACL name.*

Enabling an ACL from the Global
Configuration Level

Enabling an ACL from a VLAN
Context.

Disabling an ACL from the
Global Configuration Level

Disabling an ACL from a VLAN
Context.

```
HP5300(config)# vlan 10 ip access-group 155 in

HP5300(config)# vlan 10
HP5300(vlan-10)# ip access-group 155 in
HP5300(vlan-10)# exit

HP5300(config)# no vlan 10 ip access-group 155 in

HP5300(config)#
HP5300(config)# vlan 10
HP5300(vlan-10)# no ip access-group 155 in
```

Figure 9-16. Methods for Enabling and Disabling ACLs

Deleting an ACL from the Switch

Syntax: no ip access-list standard < *name-str* | 1-99 >

no ip access-list extended < *name-str* | 100-199 >

Removes the specified ACL from the switch's running-config file.

Note: Deleting an ACL does not delete any assignment of that ACL to a specific VLAN. If you need to delete an ACL assignment, refer to "Enabling or Disabling ACL Filtering on a VLAN" on page 9-46.

Displaying ACL Data

ACL Commands	Function	Page
show access-list	View a brief listing of all ACLs on the switch.	9-48
show access-list config	Display the CLI commands for generating the ACL commands configured in the switch.	9-49
show access-list vlan < ***vid*** >	List the name and type of ACLs assigned to a particular VLAN on the switch.	9-50
show access-list < ***acl-id*** >	Display detailed content information for a specific ACL.	9-50
show config	**show config** includes configured ACLs and assignments existing in the startup-config file.	
show running	**show running** includes configured ACLs and assignments existing in the running-config file.	

Display an ACL Summary

This command lists the configured ACLs, regardless of whether they are assigned to any VLANs.

Syntax: show access-list

List a summary table of the name, type, and application status of all ACLs configured on the switch.

For example:

```
ProCurve(config)# show access-list

Access Control Lists

 Type  Appl  Name
 ----  ----  --------------
  std   yes  1
  ext   yes  103
  ext   [no] 105
  std   yes  2
  std   [no] Red VLAN Inbound
```
In this switch, ACLs 105 and "Red VLAN Inbound" exist in the configuration but are not applied to any VLANs and thus do not affect packet

Figure 9-17. Example of a Summary Table of Access lists

Term	Meaning
Type	Shows whether the listed ACL is **std** (Standard; source-address only) or **ext** (Extended; protocol, source, and destination data).
Appl	Shows whether the listed ACL has been applied to a VLAN (**yes**/**no**).
Name	Shows the name or ID number assigned to each ACL configured in the switch.

Display the Content of All ACLs on the Switch

This command lists the configuration details for every ACL configured in the running-config file, regardless of whether you have assigned any to filter traffic on VLANs configured on the switch.

Syntax: show access-list config

> *List the configured syntax for all ACLs currently configured on the switch.*

Note

Notice that you can use the output from this command for input to an offline text file in which you can edit, add, or delete ACL commands. Refer to "Editing ACLs and Creating an ACL Offline" on page 9-53.

This information also appears in the **show running** display. If you executed **write memory** after configuring an ACL, it appears in the **show config** display.

For example, with two ACLs configured in the switch, you will see results similar to the following:

```
ProCurve(config)# show access-list config
ip access-list standard "1"
    deny 18.28.236.77 0.0.0.0
    deny 18.29.140.107 0.0.0.0
    permit 0.0.0.0 255.255.255.255
    exit
ip access-list extended "105"
    permit tcp 18.30.133.27 0.0.0.0 0.0.0.0 255.255.255.255
    permit tcp 18.30.155.101 0.0.0.0 0.0.0.0 255.255.255.255
    deny ip 18.30.133.1 0.0.0.255 0.0.0.0 255.255.255.255
    deny ip 18.30.155.1 0.0.0.255 0.0.0.0 255.255.255.255
    permit ip 0.0.0.0 255.255.255.255 0.0.0.0 255.255.255.255
    exit
```

Figure 9-18. Example of an ACL Configured Syntax Listing

Display the ACL Assignments for a VLAN

This command briefly lists the identification and type(s) of ACLs currently assigned to a particular VLAN in the running-config file. (The switch allows up to two ACL assignments per VLAN; one inbound and one outbound.)

Syntax: show access-list vlan < *vid* >

List the ACLs assigned to a VLAN in the running config file.

N o t e   This information also appears in the **show running** display. If you executed **write memory** after configuring an ACL, it appears in the **show config** display.

For example, if you assigned a standard ACL with an ACL-ID of "1" to filter inbound traffic on VLAN 10, you could quickly verify this assignment as follows:
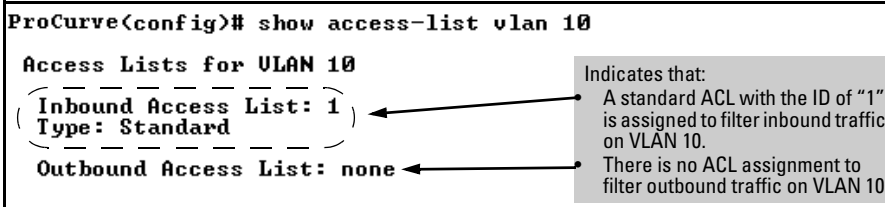
```
ProCurve(config)# show access-list vlan 10

 Access Lists for VLAN 10

  Inbound Access List: 1
  Type: Standard

  Outbound Access List: none
```

Indicates that:
- A standard ACL with the ID of "1" is assigned to filter inbound traffic on VLAN 10.
- There is no ACL assignment to filter outbound traffic on VLAN 10.

Figure 9-19. Example of Listing the ACL Assignments for a VLAN

Displaying the Content of a Specific ACL

This command displays a specific ACL configured in the running config file in an easy-to-read tabular format.

N o t e   This information also appears in the **show running** display. If you executed **write memory** after configuring an ACL, it appears in the **show config** display.

Syntax: show access-list < *acl-id* >

Display detailed information on the content of a specific ACL configured in the running-config file.

For example, suppose you configured the following two ACLs in the switch:

ACL ID	ACL Type	Desired Action
1	Standard	• Deny IP traffic from 18.28.236.77 and 18.29.140.107. • Permit IP traffic from all other sources.
105	Extended	• Permit any TCP traffic from 18.30.133.27 to any destination. • Deny any other IP traffic from 18.30.133.(1-255). • Permit all other IP traffic from any source to any destination.

Inspect the ACLs as follows:


```
ProCurve(config)# show access-list 1
Access Control Lists
  Name: 1              Indicates whether the ACL
  Type: Standard       is assigned to a VLAN.
  Applied: Yes                                              Listing for a Standard ACL

 ID   action      IP                Mask              Log
 ----------------------------------------------------------
 1     deny  std  18.28.236.77      0.0.0.0
 2     deny  std  18.29.140.107     0.0.0.0
 3   permit  std  0.0.0.0           255.255.255.255          Listing for an Extended ACL

ProCurve(config)# show access-list 105
Access Control Lists
                       Indicates whether the ACL is assigned to a VLAN.   Indicates that the source
  Name: 105                                                               TCP port can be any value.
  Type: Extended       Indicates source and destination entries in the ACL.
  Applied: No

 ID   action    IP              Mask              proto   oper   port(s)   Log
 -----------------------------------------------------------------------------
 1   permit  src: 18.30.133.27  0.0.0.0           TCP     none     0
             dst: 0.0.0.0       255.255.255.255   TCP     eq      23
 2    deny   src: 18.30.133.1   0.0.0.255         IP
             dst: 0.0.0.0       255.255.255.255   IP                        log
 3   permit  src: 0.0.0.0       255.255.255.255   IP
             dst: 0.0.0.0       255.255.255.255   IP
```

Figure 9-20. Examples of Listings Showing the Content of Standard and Extended ACLs

Table 9-6.   Descriptions of Data Types Included in Show Access-List < *acl-id* > Output

Field	Description
Name	The ACL identifier. Can be a number from 1 to 199, or a name.
Type	Standard or Extended. The former uses only source IP addressing. The latter uses both source and destination IP addressing and also allows TCP or UDP port specifiers.
Applied	"Yes" means the ACL has been applied to a VLAN. "No" means the ACL exists in the switch configuration, but has not been applied to any VLANs, and is therefore not in use.
ID	The sequential number of the Access Control Entry (ACE) in the specified ACL.
action	Permit (forward) or deny (drop) a packet when it is compared to the criteria in the applicable ACE and found to match.
IP	**In Standard ACLs:** The source IP address to which the configured mask is applied to determine whether there is a match with a packet. **In Extended ACLs:** The source and destination IP addresses to which the corresponding configured masks are applied to determine whether there is a match with a packet.
Mask	The mask configured in an ACE and applied to the corresponding IP address in the ACE to determine whether a packet matches the filtering criteria.
proto	Used only in extended ACLs to specify the packet protocol type to filter. Must be either IP, TCP, or UDP.
oper	Used only in extended ACLs where a TCP or UDP port type and number have been entered. Specifies how to compare the corresponding TCP or UDP port number in a packet to the port number in the ACE.
port(s)	Used only in extended ACLs to show any TCP or UDP port number that has been entered in the ACE.
Log	Shows the status of logging for the entry (ACE). A blank space indicates ACL logging is not enabled for that ACE.

Display All ACLs and Their Assignments in the Switch Startup-Config File and Running-Config File

The **show config** and **show running** commands include in their listings any configured ACLs and any ACL assignments to VLANs. Refer to figure 9-11 (page 9-36) and figure 9-12 (page 9-37) for examples. Remember that **show config** lists the startup-config file and **show running** lists the running-config file.

Editing ACLs and Creating an ACL Offline

Earlier sections of this chapter describe how to use the CLI to create an ACL. Beginning with "Using the CLI To Edit ACLs", below, describes how to use the CLI to edit existing ACLs. However, you can also create or edit an ACL offline, then use a TFTP server to upload the ACL as a command file. The offline method (page 9-55) provides a useful alternative to using the CLI for creating or editing large ACLs.

Using the CLI To Edit ACLs

The switch applies individual ACEs in the order in which they occur in an ACL. You can use the CLI to delete individual ACEs from anywhere in an ACL and to append new ACEs to the end of an ACL. However, the CLI method does not allow you to insert a new ACE between two existing ACEs.

Using the CLI To Edit a Short ACL. To insert a new ACE between existing ACEs in a short ACL, you may want to delete the ACL and then re-configure it by entering your updated list of ACEs in the correct order.

Using the CLI to Edit a Longer ACL. To insert a new ACE between existing ACEs in a longer ACL:

a. Delete the first ACE that is out of sequence and all following ACEs through the end of the ACL.
b. Re-Enter the desired ACEs in the correct sequence.

General Editing Rules

■   You can delete any ACE from an ACL by repeating the ACE's entry command, preceded by the "no" statement. When you enter a new ACE, the switch inserts it as the last entry of the specified ACL.

■ Deleting the last ACE from a *numeric* ACL, removes the ACL from the configuration. Deleting the last ACE from a *named* ACL leaves the ACL in memory. In this case, the ACL is "empty" and cannot perform any filtering tasks. (In any ACL the implicit "deny any" does not apply unless the ACL includes at least one explicit ACE.)

■ When you create a new ACL, the switch inserts it as the last ACL in the startup-config file. (Executing **write memory** saves the running-config file to the startup-config file.)

Deleting Any ACE from an ACL

You can delete an ACE from an ACL by repeating the ACE's entry command, preceded by the "**no**" statement.

Syntax: no access-list < *acl-id* > < permit | deny > < any | host | *ip-addr/mask-length* >

Deletes an ACE from a standard ACL. All variable parameters in the command must be an exact match with their counterparts in the ACE you want to delete.

no access-list < *acl-id* > < permit | deny > < ip | tcp | udp >
< src-addr: any | host | *ip-addr/mask-length* > [operator < src-port-num >]
< dest-addr: any | host | ip-addr-mask-length > [operator < dest-port-num >
[log]

Deletes an ACE from a standard ACL. All variable parameters in the command must be an exact match with their counterparts in the ACE you want to delete.

For example, the first of the following two commands creates an ACE in ACL 22 and the second deletes the same ACE:

Creates an ACE in ACL 22.

```
ProCurve(config)# access-list 22 permit host 18.28.152.64
ProCurve(config)# no access-list 22 permit host 18.28.152.64
```

Removes the same ACE from ACL 22, regardless of the ACE's position in the ACL.
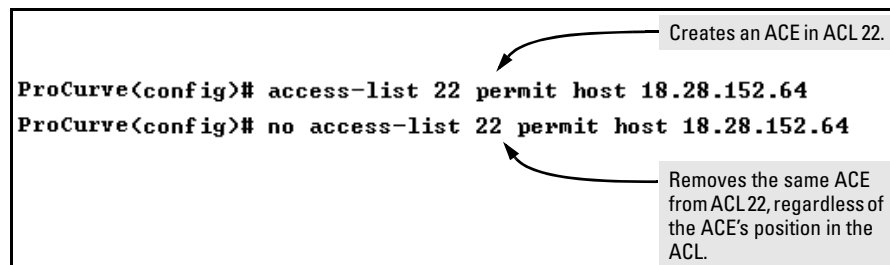
Figure 9-21. Example of Deleting an ACE from a Standard ACL

Figure 9-22 shows an example of deleting an ACE from an extended ACL.

```
ProCurve(config)# show config              ACL 103 Before Removing
Startup configuration:                     the Second "deny" ACE.
   .
   .
   .
ip access-list extended "103"
    deny tcp 0.0.0.0 255.255.255.255 10.10.10.2 0.0.0.0 eq 23 log
    deny tcp 0.0.0.0 255.255.255.255 10.10.20.2 0.0.0.0 eq 23 log          Use no access-list
    permit ip 0.0.0.0 255.255.255.255 0.0.0.0 255.255.255.255              to remove this line
    exit                                                                   from ACL 103.
vlan 1
    name "DEFAULT_VLAN"
    untagged A1
   .
   .
   .
ProCurve(config)# no access-list 103 deny tcp any host 10.10.20.2 eq 23 log
ProCurve(config)# write mem
ProCurve(config)# show config
                                           ACL 103 After Removing
Startup configuration:                     the Second "deny" ACE.
   .
   .
   .
ip access-list extended "103"
    deny tcp 0.0.0.0 255.255.255.255 10.10.10.2 0.0.0.0 eq 23 log
    permit ip 0.0.0.0 255.255.255.255 0.0.0.0 255.255.255.255
    exit
vlan 1
    name "DEFAULT_VLAN"
    untagged A1
```

Figure 9-22. Example of Deleting an ACE from an ACL

Working Offline To Create or Edit an ACL

For longer ACLs that would be difficult or time-consuming to accurately create or edit in the CLI, you can use the offline method:

1. Begin by doing one of the following:

   - To edit one or more existing ACLs, use **copy command-output tftp** to copy the current version of the ACL configuration to a file in your TFTP server. For example, to copy the ACL configuration to a file named **acl02.txt** in the TFTP directory on a server at 18.28.227.2:

     ```
     ProCurve# copy command-output 'show access-list
     config' tftp 18.28.227.2 acl02.txt pc
     ```

   - To create a new ACL, just open a text file in the appropriate directory on a TFTP server accessible to the switch.

2. Use the text editor to create or edit the ACL(s).

3. Use **copy tftp command-file** to download the file as a list of commands to the switch.

Creating an ACL Offline

Use a text editor that allows you to create an ASCII text file (.txt).

If you are replacing an ACL on the switch with a new ACL that uses the same number or name syntax, begin the command file with a "no" command to remove the earlier version of the ACL from the switch's running-config file. Otherwise, the switch will append the new ACEs in the ACL you download to the existing ACL. For example, if you plan to use the Copy command to *replace* ACL "103", you would place this command at the beginning of the edited file:

```
no ip access-list extended 103
```

```
no ip access-list extended 103
ip access-list extended "103"
   deny tcp 0.0.0.0 255.255.255.255 10.10.10.2 0.0.0.0 eq 23 log
   permit ip 0.0.0.0 255.255.255.255 0.0.0.0 255.255.255.255
   exit
```

Removes an existing ACL and replaces it with a new version with the same identity. To append new ACEs to the ACL instead of replacing it, you would omit the first line.

Figure 9-23. Example of an Offline ACL File Designed To Replace An Existing ACL

For example, suppose that you wanted to create an extended ACL to fulfill the following requirements (Assume a subnet mask of 255.255.255.0.):

■   ID: "Controls for VLAN 20"

■   Deny Telnet access to a server at 10.10.10.100 on VLAN 10 from these three IP addresses on VLAN 20 (with ACL logging):

•   10.10.20.17

•   10.10.20.23

•   10.10.20.40

■   Allow any access to the server from all other addresses on VLAN 20:

■   Permit internet access to these two IP address on VLAN 20, but deny access to all other addresses on VLAN 20 (without ACL logging).

•   10.10.20.98

•   10.10.20.21

■   Deny all other traffic from VLAN 20 to VLAN 10.

■ Deny all traffic from VLAN 30 (10.10.30.0) to the server at 10.10.10.100 on VLAN 10 (without ACL logging), but allow any other traffic from VLAN 30 to VLAN 10.

■ Deny all other inbound traffic to VLAN 20. (Hint: The implicit "deny any" can achieve this objective.)

1. You would create a **.txt** file with the content shown in figure 9-24.

Figure 9-24 content:

```
ip access-list extended "153"

; APPLIES INBOUND ON VLAN 20.

; ANY SOURCE TO VLAN 20 DESTINATIONS.

permit tcp 0.0.0.0 255.255.255.255 10.10.20.98 0.0.0.0 eq http
permit tcp 0.0.0.0 255.255.255.255 10.10.20.21 0.0.0.0 eq http
deny tcp 0.0.0.0 255.255.255.255 10.10.20.1 0.0.0.255 eq http

; VLAN 20 SOURCES TO VLAN 10 DESTINATIONS.

deny tcp 10.10.20.17 0.0.0.0 10.10.10.100 0.0.0.0 eq telnet log
deny tcp 10.10.20.23 0.0.0.0 10.10.10.100 0.0.0.0 eq telnet log
deny tcp 10.10.20.40 0.0.0.0 10.10.10.100 0.0.0.0 eq telnet log
permit ip 10.10.20.1 0.0.0.255 10.10.10.100 0.0.0.0

; VLAN 30 POLICY.

deny ip 10.10.30.1 0.0.0.255 10.10.10.100 0.0.0.0
permit ip 10.10.30.1 0.0.0.255 10.10.10.1 0.0.0.255

exit

vlan 20 ip access-group "Controls for VLAN 20" in
```

Callout (left): Enables a comment in the file.

Callout (top right): You can use the ";" character to denote a comment. The file stored on your TFTP server retains comments, and they appear when you use **copy** to download the ACL command file. (Comments are not saved in the switch configuration.)

Callout (right): Blank lines in the file cause breaks in the displayed line-numbering sequence when you copy the command file to the switch. This is normal operation. (See figure 9-25.)

Figure 9-24. Example of a.txt File Designed for Creating an ACL

2. After you copy the above .txt file to a TFTP server the switch can access, you would then execute the following command:

```
ProCurve(config)# copy tftp command-file 13.28.227.2 acl-vlan20.txt pc
Running configuration may change, do you want to continue [y/n]?   y
   1. ip access-list extended "153"
   3. ; APPLIES INBOUND ON VLAN 20.
   5. ; ANY SOURCE TO VLAN 20 DESTINATIONS.
   7. permit tcp 0.0.0.0 255.255.255.255 10.10.20.98 0.0.0.0 eq http
   8. permit tcp 0.0.0.0 255.255.255.255 10.10.20.21 0.0.0.0 eq http
   9. deny tcp 0.0.0.0 255.255.255.255 10.10.20.1 0.0.0.255 eq http
  11. ; VLAN 20 SOURCES TO VLAN 10 DESTINATIONS.
  13. deny tcp 10.10.20.17 0.0.0.0 10.10.10.100 0.0.0.0 eq telnet log
  14. deny tcp 10.10.20.23 0.0.0.0 10.10.10.100 0.0.0.0 eq telnet log
  15. deny tcp 10.10.20.40 0.0.0.0 10.10.10.100 0.0.0.0 eq telnet log
  16. permit ip 10.10.20.1 0.0.0.255 10.10.10.100 0.0.0.0
  18. ; VLAN 30 POLICY.
  20. deny ip 10.10.30.1 0.0.0.255 10.10.10.100 0.0.0.0
  21. permit ip 10.10.30.1 0.0.0.255 10.10.10.1 0.0.0.255
  23. exit
  25. vlan 20 ip access-group "153" in
ProCurve(config)#
```

Figure 9-25. Example of Using "copy tftp command-file" To Configure an ACL in the Switch

---

Note

If a transport error occurs, the switch does not execute the command and the ACL is not configured.

---

3. Next, assign the new ACL to the intended VLAN which, in this example, is for inbound traffic on VLAN 20.

    ```
    ProCurve(config)# vlan 20 ip access-group "Controls
    for VLAN 20" in
    ```

4. Inspect the new running configuration:

    ```
    ProCurve(config)# show running
    ```

5. If the configuration appears satisfactory, save it to the startup-config file:

    ```
    ProCurve(config)# write memory
    ```

Enable ACL "Deny" Logging

ACL logging enables the switch to generate a message when IP traffic meets the criteria for a match with an ACE that results in an explicit "deny" action. You can use ACL logging to help:

■   Test your network to ensure that your ACL configuration is detecting and denying the traffic you do not want forwarded

■   Receive notification when the switch detects attempts to transmit traffic you have designed your ACLs to reject

The switch sends ACL messages to Syslog and optionally to the current console, Telnet, or SSH session. You can configure up to six Syslog server destinations.

Requirements for Using ACL Logging

■   The switch configuration must include an ACL (1) assigned to a static VLAN and (2) containing an ACE configured with the **deny** action and the **log** option.

■   To screen routed packets with destination IP addresses outside of the switch, IP routing must be enabled.

■   For ACL logging to a Syslog server, the server must be accessible to the switch and identified (with the **logging < *ip-addr* >** command) in the switch configuration.

■   Debug must be enabled for ACLs and one or both of the following:
   •   logging (for sending messages to Syslog)
   •   Session (for sending messages to the current console interface)

ACL Logging Operation

When the switch detects a packet match with an ACE and the ACE includes both the **deny** action and the optional **log** parameter, an ACL log message is sent to the designated debug destination. The first time a packet matches an ACE with **deny** and **log** configured, the message is sent immediately to the destination and the switch starts a wait-period of approximately five minutes. (The exact duration of the period depends on how the packets are internally routed.) At the end of the collection period, the switch sends a single-line

summary of any additional "deny" matches for that ACE (and any other "deny" ACEs for which the switch detected a match). If no further log messages are generated in the wait-period, the switch suspends the timer and resets itself to send a message as soon as a new "deny" match occurs. The data in the message includes the information illustrated in figure 9-26.

Note: To fit this illustration on the page, the portion of the message generated by the Syslog server itself is shown in the line above the portion of the message generated by the switch. Depending on the terminal emulator you use, you may see information similar to this, which includes the date and time the log message was received, the IP address of the default VLAN on the sending switch, and the message type (ACL).

Mar 1 10:04:45 18.28.227.101 ACL:

ACL 03/01/03 10:04:45: denied tcp 13.28.227.3 (1025) (VLAN 227) -> 10.10.20.2 (23), 1 packets

Begins the actual message generated by the ACL itself, and indicates message type (ACL), date and time of generation.

ACL action (denied), protocol type (IP, TCP, or UDP), source IP address of the denied packet(s) and the TCP or UDP port number at the source device. The port number is "0" if the protocol is "IP" or you do not specify a port number.

Indicates the VLAN on which the ACL is assigned, the packet's destination IP address, the destination TCP or UDP port number ("0" if the protocol is "IP" or you do not specify a port

Indicates the number of packets (deny instances) detected.

Figure 9-26. Content of an ACL-Generated Message

Enabling ACL Logging on the Switch

1. Use the debug command to:
    a. Configure one or more log destinations.
    b. If you are using a Syslog server, use the **logging** command to configure the server's IP address. (You can configure up to six Syslog servers.)
    c. Ensure that the switch can access any Syslog servers you specify.
2. Configure one or more ACLs with the deny action and the log option.

For example, suppose that you want to:

■   On VLAN 100 configure an extended ACL with an ACL-ID of 143 to deny Telnet traffic from IP address 18.38.100.127 on VLAN 100.

■   Configure the switch to send an ACL log message to the console and to a Syslog server at IP address 18.38.110.54 on VLAN 110 if the switch detects a match denying Telnet access from 18.38.100.127.

(This example assumes that IP routing is already configured on the switch.)
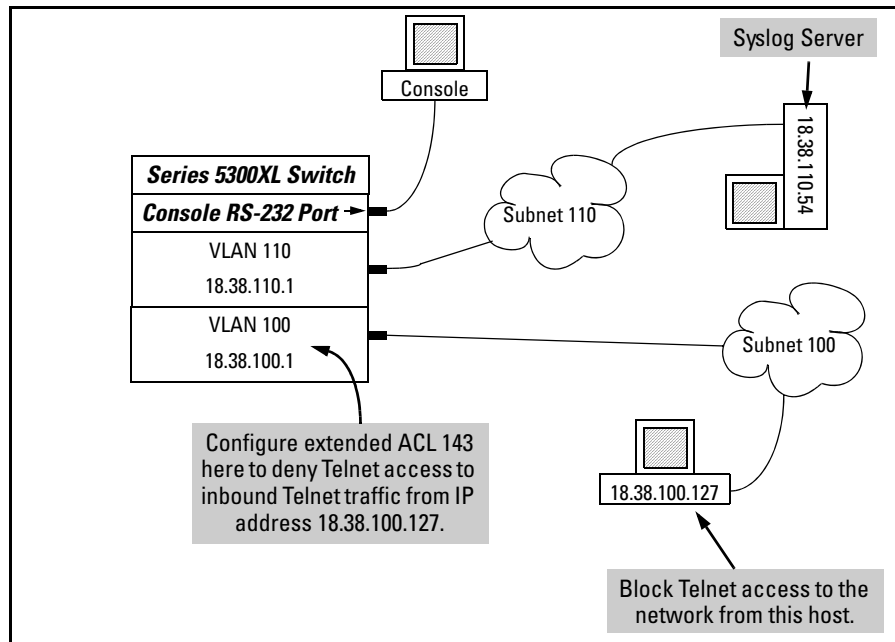
Figure 9-27. Example of an ACL Log Application

```
ProCurve(config)# access-list 143 deny tcp host 18.38.100.127 any eq telnet log
ProCurve(config)# access-list 143 permit ip any any
ProCurve(config)# vlan 100 ip access-group 143 in
ProCurve(config)# logging 18.38.110.54
ProCurve(config)# debug acl
ProCurve(config)# debug destination logging
ProCurve(config)# debug destination session
ProCurve(config)# write memory
ProCurve(config)# show debug
Debug Logging
 Destination:
  Logging
    18.38.110.54
  Session
 Enabled debug types:
  event
  acl log
```

Figure 9-28. Commands for Applying an ACL with Logging to Figure 9-27

Operating Notes for ACL Logging

■   The ACL logging feature generates a message only when packets are explicitly denied as the result of a match, and not when explicitly permitted or implicitly denied. To help test ACL logging, configure an ACL with an explicit **deny any** and **log** statements at the end of the list, and apply the ACL to an appropriate VLAN.

■   Logging enables you to selectively test specific devices or groups. However, excessive logging can affect switch performance. For this reason, HP recommends that you remove the logging option from ACEs for which you do not have a present need. Also, avoid config-uring logging where it does not serve an immediate purpose. (Note that ACL logging is not designed to function as an accounting method.) See also "Apparent Failure To Log All "Deny" Matches" in the section titled "ACL Problems", found in appendix C, "Trouble-shooting" of the Management and Configuration Guide for your switch.

■   When configuring logging, you can reduce excessive use by config-uring the appropriate ACEs to match with specific hosts instead of entire subnets.

General ACL Operating Notes

ACLs do not provide DNS hostname support.

Protocol Support: ACL criteria includes IP, TCP, and UDP. ACLs do not use these protocols:

■   TOS (Type-of-Service)

■   Precedence

■   MAC information

■   QoS

ACLs do not affect switch serial port access.

When the ACL configuration includes TCP or UDP options, the switch operates in "strict" TCP and UDP mode for increased control. The switch compares all TCP and UDP packets against the ACLs. (In the ProCurve Series 9300 Routing Switches, the Strict TCP and Strict UDP modes are optional and must be specifically invoked.)

Replacing or Adding To an Active ACL Policy.   If you assign an ACL to a VLAN and subsequently add or replace ACEs in that ACL, each new ACE becomes active when you enter it.

Note   When an ACE becomes active, it screens the packets resulting from new traffic connections. It does not screen packets resulting from currently open traffic connections. If you invoke a new ACE to screen packets in a currently open traffic connection, you must force the connection to close before the ACE can begin screening packets from that source.

ACL Screening of Traffic Generated by the Switch.   Outbound ACLs on a switch do not screen traffic (such as broadcasts, Telnet, Ping, and ICMP replies) *generated by the switch itself*. Note that ACLs do screen this type of traffic when other devices generate it. Similarly, ACLs can screen responses from other devices to unscreened traffic the switch generates.

— This page is intentionally unused. —

10

Access Control Lists (ACLs) for the Series 3400cl and Series 6400cl Switches

Contents

Introduction

Feature	Default	Menu	CLI	Web
Numbered ACLs				
Standard ACLs	None	—	10-47	—
Extended ACLs	None	—	10-52	—
Named ACLs		—	10-58	—
Enable or Disable an ACL		—	10-61	—
Display ACL Data	n/a	—	10-62	—
Delete an ACL	n/a	—	10-62	—
Configure an ACL from a TFTP Server	n/a	—	10-72	—
Enable ACL Logging	n/a	—	10-77	—
Show ACL Resources				
Access-List Resources Help				

ACL Applications on Series 3400cl and 6400cl Switches

ACLs can filter traffic from a host, a group of hosts, or from entire subnets. Where it is necessary to apply ACLs to filter traffic from outside a network or subnet, applying ACLs at the edge of the network or subnet removes unwanted traffic as soon as possible, and thus helps to improve system performance. ACLs on the 3400cl/6400cl switches filter inbound traffic only and can rapidly consume switch resources. Also, ACLs, QoS, and Rate-Limiting share the same per-port mask resources on these switches. For these reasons, the best places to apply ACLs on the 3400cl/6400cl switches are on "edge" ports where ACLs are likely to be less complex and resource-intensive than in core network applications where the per-VLAN and inbound/outbound ACL filtering offered by the Series 5300xl switches may be the best ACL solution.

General Application Options

Layer 3 IP filtering with Access Control Lists (ACLs) on the 3400cl/6400cl switches enables you to improve network performance and restrict network use by creating policies for:

- **Switch Management Access:** Permits or denies in-band management access. This includes preventing the use of certain TCP or UDP applications (such as Telnet, SSH, web browser, and SNMP) for transactions between specific source and destination IP addresses.

- **Application Access Security:** Eliminates inbound, unwanted IP, TCP, or UDP traffic by filtering packets where they enter the switch on specific physical ports or trunks.

This chapter describes how to configure, apply, and edit ACLs in ProCurve Series 3400cl and Series 6400cl switches and how to monitor the results of ACL actions.

Notes   Unlike the ProCurve Series 5300xl switches, it is not necessary to enable routing on 3400cl/6400cl switches to support ACL operation.

ACLs can enhance network security by blocking selected IP traffic, and can serve as one aspect of maintaining network security. *However, because ACLs do not provide user or device authentication, or protection from malicious manipulation of data carried in IP packet transmissions, they should not be relied upon for a complete security solution.*

ACLs in the 3400cl/6400cl switches do not screen non-IP traffic such as AppleTalk and IPX.

For ACL filtering to take effect, configure an ACL and then assign it to the inbound traffic on a statically configured port or trunk.

Table 10-1. Comprehensive Command Summary

Action	Command	Page
Configuring Standard (Numbered) ACLs	ProCurve(config)# [no] access-list < 1-99 > < deny | permit > < any | host *<src-ip-addr >* | *src-ip-address/mask* > [1] [log][2]	10-47

Configuring Extended (Numbered) ACLs	ProCurve(config)# [no] access-list <100-199> < deny | permit >
	        ip < any | host *<src-ip-addr >* | *src-ip-address*/mask > [1]
	        [log][2]
		10-52

	ProCurve(config)# [no] access-list < 100-199 > < deny | permit >
	        < tcp | udp >
	        < any | host *<src-ip-addr>* | *src-ip-address*/mask > [1]
	          [eq < *src-port tcp/udp-id* >]
	        < any | host *<dest-ip-addr>* | *dest-ip-address*/mask > [1]
	          [eq < *dest-port tcp/udp-id* >]
	        [log][2]
		10-52

Configuring Standard (Named) ACLs	ProCurve(config)# [no] ip access-list standard < *name-str* | 1-99 >	10-58
	ProCurve(config-std-nacl)# < deny | permit > < any | host *<src-ip-addr>* | *src-ip-address/mask* > [1] [log] [2]	10-58
Configuring Extended (Named) ACLs	ProCurve(config)# [no] ip access-list extended < *name-str* | 100-199 >	10-58
	ProCurve(config-std-nacl)# < deny | permit > ip < any | host *<src-ip-addr>* | *src-ip-address/mask* >[1] < any | host *<dest-ip-addr>* | *dest-ip-address/mask* >[1] [log][2]	10-58

	ProCurve(config-std-nacl)# < deny | permit > < tcp | udp >
	        < any | host *<src-ip-addr>* | *src-ip-address*/mask > [1]
	          [eq < *tcp/udp-port-#* | *well-known-port-name* >]
	        < any | host *<dest-ip-addr>* | *dest-ip-address*/mask > [1]
	          [eq < *tcp/udp-port-#* | *well-known-port-name* >]
	        [log][2]
		10-58

Enabling or Disabling an ACL	ProCurve(config)# [no] interface < *port-list* > access-group < *name-str* | 1-99 | 100-199 > in	10-61
Deleting an ACL from the Switch	ProCurve(config)# no ip access-list < standard < *name-str* | 1-99 >> in ProCurve(config)# no ip access-list < extended < *name-str* | 100 -199 >> in	10-62

[1] The mask can be in either dotted-decimal notation (such as 0.0.15.255) or CIDR notation (such as /20).
[2] The [log] function applies only to "deny" ACLs, and generates a message only when there is a "deny" match.

Action	Command	Page
Displaying ACL Data	ProCurve(config)# show access-list	10-62
	ProCurve(config)# show access-list [*acl-name-string*]	
	ProCurve(config)# show access-list config	
	ProCurve(config)# show access-list ports < *port-list* >	
	ProCurve(config)# show access-list resources	
	ProCurve(config)# access-list resources help	
	ProCurve(config)# show config	
	ProCurve(config)# show running	

Terminology

3400cl/6400cl Switches: An all-inclusive reference to the ProCurve 3400cl and 6400cl switches.

Access Control Entry (ACE): An ACE is a policy consisting of criteria and an action to take (permit or deny) on a packet if it meets the criteria. The elements composing the criteria include:

- Source IP address and mask (standard and extended ACLs)
- Destination IP address and mask (extended ACLs only)
- TCP or UDP application port numbers (optional, extended ACLs only)

Access Control List (ACL): A list (or set) consisting of one or more explicitly configured Access Control Entries (ACEs) and terminating with an implicit "deny" default which drops any packets that do not have a match with any explicit ACE in the named ACL. The two classes of ACLs are "standard" and "extended". See "Standard ACL" and "Extended ACL".

ACE: See "Access Control Entry".

ACL: See "Access Control List".

ACL ID: A number or alphanumeric string used to identify an ACL. A *standard* ACL ID can have either a number from 1 to 99 or an alphanumeric string. An *extended* ACL ID can have either a number from 100 to 199 or an alphanumeric string.

ACL Mask: Follows an IP address (source or destination) listed in an ACE to specify either a subnet or a group of devices. Defines which bits in a packet's corresponding IP addressing must exactly match the IP addressing in the ACE, and which bits need not match (wildcards). For example:

```
10.10.10.1 0.0.0.255
10.10.10.1/24
```

Dotted-Decimal and CIDR versions of the same mask. In both cases, zeros in the mask indicate significant digits. Ones in the mask indicate wildcard digits.
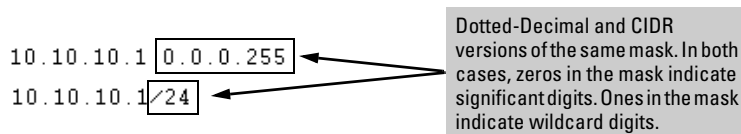
As shown above, zeros in an ACL mask specify an exact match requirement for IP addresses, and ones specify a wildcard. In this example, a matching IP address would be any address in the range 10.10.10.1-255. (See also "How an ACE Uses a Mask To Screen Packets for Matches" on page 10-32, and Per-Port Mask on page 10-9.)

DA: The acronym for *Destination IP Address*. In an IP packet, this is the destination IP address carried in the header, and identifies the destination intended by the packet's originator. In an extended ACE, this is the second of two IP addresses required by the ACE to determine whether there is a match between a packet and the ACE. See also "SA".

Deny: An ACE configured with this action causes the switch to drop an inbound packet for which there is a match within an applicable ACL. As an option, you can configure the switch to generate a logging output to a Syslog server and a console session.)

Extended ACL: This type of Access Control List uses layer-3 IP criteria composed of source and destination IP addresses and (optionally) TCP or UDP port criteria to determine whether there is a match with an IP packet. You can apply extended ACLs to either inbound or outbound routed traffic and to any inbound switched or routed traffic with a DA belonging to the switch itself. Extended ACLs require an identification number (ID) in the range of 100 - 199 or an alphanumeric name.

Implicit Deny: If the switch finds no matches between an inbound packet and the configured criteria in an applicable ACL, then the switch denies (drops) the packet with an implicit "deny IP any" operation. You can preempt the implicit "deny IP any" in a given ACL by configuring **permit any** (standard) or **permit IP any any** (extended) as the last explicit ACE in the ACL. Doing so permits an inbound packet that is not explicitly permitted or denied by other ACEs configured sequentially earlier in the ACL. Unless otherwise noted, "implicit deny IP any" refers to the "deny" action enforced by both standard and extended ACLs.

Inbound Traffic: For the purpose of defining where the switch applies ACLs to filter traffic, inbound traffic is any IP packet that:

- *Enters the switch through a physical port.*
- Has a destination IP address (DA) that meets either of these criteria:
  - The packet's DA is for an external device.
  - The packet's DA is for an IP address configured on the switch itself. (This increases your options for protecting the switch from unauthorized management access.)

Because ACLs are assigned to physical ports or port trunks, an ACL that filters inbound traffic on a particular port or trunk examines packets meeting the above criteria that enter the switch through that port or trunk.

Outbound Traffic: This is any traffic *leaving the switch* through a physical port or trunk. The switch does not apply ACLs to outbound traffic or internally where routed traffic moves between VLANs. That is, ACL operation is not affected by enabling or disabling routing on the switch. (Refer also to "ACL Inbound Application Points" on page 10-10.)

Permit: An ACE configured with this action allows a port or trunk to permit an inbound packet for which there is a match within an applicable ACL.

Per-Port Mask: An internally applied template for all ACL and IGMP configurations. The significance of per-port masks is that a maximum of 8 masks are available (per-port) for ACL (and IGMP) use.

```
ProCurve(config)# show access-list resources
QoS/ACL Resource Usage

 Port | Rules Available | ACL Masks Available
 -----+-----------------+---------------------
   1  |  120            |   8
   2  |  120            |   8
   3  |  120            |   8
   .  |   .             |   .
   .  |   .             |   .
   .  |   .             |   .
```

Figure 10-1. Example of Per-Port Mask Allocation in the Default Configuration

For more information, refer to "Planning an ACL Application on a Series 3400cl or Series 6400cl Switch" on page 10-17. See also "ACL Mask" on page 10-8.

SA: The acronym for *Source IP Address*. In an IP packet, this is the source IP address carried in the IP header, and identifies the packet's sender. In an extended ACE, this is the first of two IP addresses used by the ACE to determine whether there is a match between a packet and the ACE. See also "DA".

Standard ACL: This type of Access Control List uses layer-3 IP criteria of source IP address to determine whether there is a match with an inbound IP packet. You can apply a standard ACL to inbound traffic on a port or trunk, including any inbound traffic with a DA belonging to the switch itself. Standard ACLs require an identification number (ID) in the range of 1 - 99 or an alphanumeric name.

Wildcard: The part of a mask that indicates the bits in a packet's IP addressing that do not need to match the corresponding bits specified in an ACL. See also **ACL Mask** on page 10-8.

Overview

Types of IP ACLs

Standard ACL: Use a standard ACL when you need to permit or deny traffic based on source IP address. Standard ACLs are also useful when you need to quickly control a performance problem by limiting traffic from a subnet, group of devices, or a single device. (This can block all inbound IP traffic from the configured source, but does not block traffic from other sources within the network.) This ACL type uses a numeric ID of 1 through 99 or an alphanumeric ID string. You can specify a single host, a finite group of hosts, or any host.

Extended ACL: Use extended ACLs whenever simple IP source address restrictions do not provide the breadth of traffic selection criteria you want for a port or trunk. Extended ACLs allow use of the following criteria:

■ Source and destination IP addresses

■ TCP application criteria

■ UDP application criteria

ACL Inbound Application Points

You can apply ACL filtering to IP traffic inbound on a physical port or static trunk with a destination (DA):

■ On another device. (ACLs are not supported on dynamic LACP trunks.)

■ On the switch itself. In figure 10-2, below, this would be any of the IP addresses shown in VLANs "A", "B", and "C" on the switch. (IP routing need not be enabled.)

The switch can apply ACL filtering to traffic *entering the switch* on ports and/ or trunks configured to apply ACL filters. For example, in figure 10-2 you would assign an inbound ACL on port 1 to filter a packet from the workstation 10.28.10.5 to the server at 10.28.20.99. Note that all ACL filtering is performed on the inbound port or trunk. Routing may be enabled or disabled on the switch, and any permitted inbound traffic may have any valid destination.
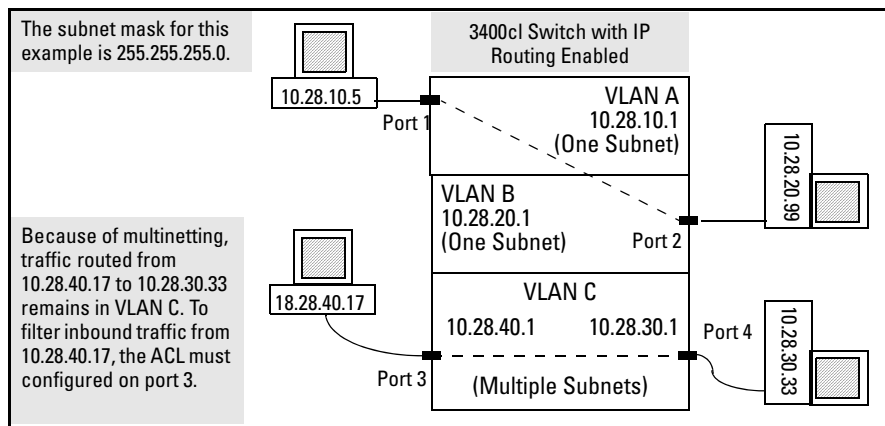
Figure 10-2. Example of Filter Applications

Features Common to All ACLs

■   On any port or static trunk you can apply one ACL to inbound traffic.

■   Any ACL can have multiple entries (ACEs).

■   You can apply any one ACL to multiple ports and trunks.

■   A source or destination IP address and a mask, together, can define a single host, a range of hosts, or all hosts.

■   Before changing the content of an ACL assigned to one or more ports or trunks, you must first remove the ACL from those ports or trunks.

■   Every standard ACL includes an implied "**deny any**" as the last entry, and every extended ACL includes an implied "**deny IP any any**" as the last entry. The switch applies this action to any packets that do not match other criteria in the ACL.

■   In any ACL, you can apply an ACL log function to ACEs that have a "deny" action. The logging occurs when there is a match on a "deny" ACE. (The switch sends ACL logging output to Syslog and, optionally, to a console session.)

■   Standard and Extended ACL features cannot be combined in one ACL.

You can configure ACLs using either the CLI or a text editor. The text-editor method is recommended when you plan to create or modify an ACL that has more entries than you can easily enter or edit using the CLI alone. Refer to "Editing ACLs and Creating an ACL Offline" on page 10-69.

General Steps for Planning and Configuring ACLs

1. Identify the traffic type to filter. Options include:
   - Any inbound IP traffic
   - Inbound TCP traffic only
   - Inbound UDP traffic only

2. The SA and/or the DA of inbound traffic you want to permit or deny.

3. Determine the best points at which to apply specific ACL controls. For example, you can improve network performance by filtering unwanted traffic at the edge of the network instead of in the core.

4. Design the ACLs for the selected control points. Where you are using explicit "deny" ACEs, you can optionally use the ACL logging feature to help verify that the switch is denying unwanted packets where intended. Remember that excessive ACL logging activity can degrade the switch's performance. (Refer to "Enable ACL "Deny" Logging" on page 10-75.)

5. Create the ACLs in the selected switches.

6. Assign the ACLs to filter the inbound traffic on ports and/or static trunk interfaces configured on the switch.

7. Test for desired results.

For more details on ACL planning considerations, refer to "Planning an ACL Application on a Series 3400cl or Series 6400cl Switch" on page 10-17.

Caution Regarding the Use of Source Routing

Source routing is enabled by default on the switch and can be used to override ACLs. For this reason, if you are using ACLs to enhance network security, the recommended action is to use the **no ip source-route** command to disable source routing on the switch. (If source routing is disabled in the running-config file, the **show running** command includes "**no ip source-route**" in the running-config file listing.)

ACL Operation

Introduction

An ACL is a list of one or more Access Control Entries (ACEs), where each ACE consists of a matching criteria and an action (permit or deny). An ACL applies only to the switch in which it is configured. ACLs operate on assigned ports and static trunks, and filter these traffic types:

■   Traffic entering the switch. (Note that ACLs do not screen traffic at any internal point.where traffic moves between VLANs or subnets within the switch; only on inbound ports and static trunks. Refer to "ACL Inbound Application Points" on page 10-10.)

■   Switched or routed traffic entering the switch and having an IP address on the switch as the destination

You can apply one inbound ACL to each port and static trunk configured on the switch. The complete range of options per interface includes:

■   **No ACL** assigned. (In this case, all traffic entering the switch on the interface does so without any ACL filtering, which is the default.)

■   **One ACL** assigned to filter the inbound traffic entering the switch on the interface.

■   **Multiple Assignments for the same ACL.** (The switch allows one ACL assignment to an interface, but you can assign the same ACL to multiple interfaces.)

Note   On a given port or trunk, after you assign an ACL, the default action is to deny any traffic that is not specifically permitted by the ACL. (This applies only to the inbound traffic flow filtered by the ACL.)

The Packet-Filtering Process

Sequential Comparison and Action. When the switch uses an ACL to filter a packet, it sequentially compares each ACE's filtering criteria to the corresponding data in the packet until it finds a match.

```
                                     ip access-list standard "1"
For a packet with a source IP address of     permit 10.28.136.24 0.0.0.0
10.28.156.3, the switch:
                                             permit 10.28.144.116 0.0.0.0
1.  Compares the packet to this ACE first.
                                             deny 10.28.156.7 0.0.0.15 log
2.  Since there is not a match with the first
    ACE, the switch then compares the        permit 10.28.156.1 0.0.0.255
    packet to the second ACE, where there
    is also not a match.                      exit

3.  The switch compares the packet to the
    third ACE. There is a match because            0.0.0.0 is an explicit
    the 0.0.0.15 mask includes the source          host mask. 0.0.0.15
    IP address. The then switch denies             and 0.0.0.255 allow
    (drops) the packet.                            multiple hosts.

4.  The packet is not compared to the
    fourth ACE.
```

Figure 10-3. Example of Sequential Comparison

That is, the switch tries the first ACE in the list. If there is not a match, it tries the second ACE, and so on. When a match is found, the switch invokes the configured action for that entry (permit or drop the packet) and no further comparisons of the packet are made with the remaining ACEs in the ACL. This means that when the switch finds an ACE whose criteria matches a packet, it invokes the action configured for that ACE, and any remaining ACEs in the ACL are ignored. *Because of this sequential processing, successfully implementing an ACL depends in part on configuring ACEs in the correct order for the overall policy you want the ACL to enforce.*

Implicit Deny. If a packet does not have a match with the criteria in any of the ACEs in the ACL, the switch denies (drops) the packet. (This is termed *implicit deny.*) If you need to override the implicit deny so that any packet that does not have a match will be permitted, then you can enter **permit any** as the last ACE in the ACL. This directs the switch to permit (forward) any packets that do not have a match with any earlier ACE listed in the ACL, and prevents these packets from being filtered by the implicit deny.

Note on Implicit Deny

For ACLs configured to filter inbound packets, note that Implicit Deny filters *any packets, including those with a DA specifying the switch itself.* This operation helps to prevent management access from unauthorized IP sources.
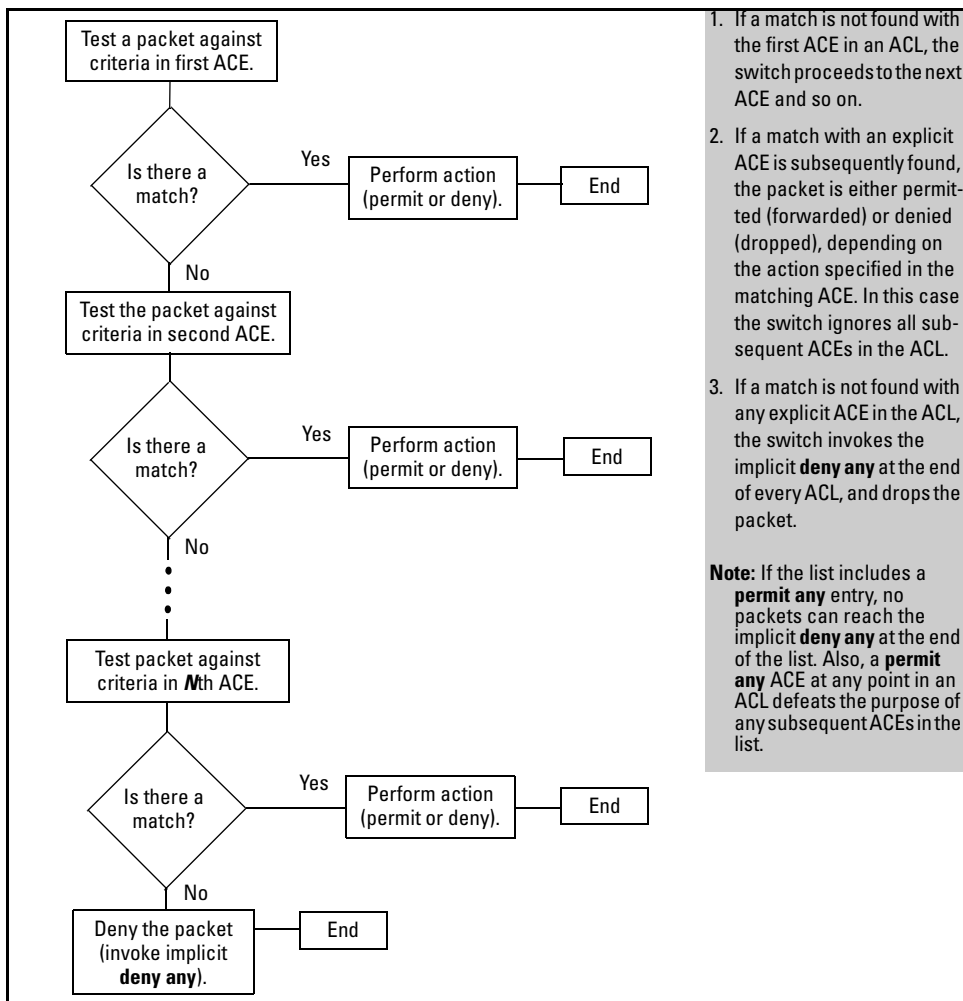
Figure 10-4. The Packet-Filtering Process in an ACL with *N* **Entries (ACEs)**

Note

The order in which an ACE occurs in an ACL is significant. For example, if an ACL contains six ACEs, but the first ACE is a "permit IP any", then the ACL permits all IP traffic, and the remaining ACEs in the list do not apply, even if they specify criteria that would make a match with any of the traffic permitted by the first ACE.

For example, suppose you want to configure an ACL on the switch (with an ID of "100") to invoke these policies:

1. Permit all inbound traffic on port 12 sent from IP address 11.11.11.42.

2. Deny *only* the inbound Telnet traffic sent from IP address 11.11.11.101.

3. Permit *only* inbound Telnet traffic sent from IP address 11.11.11.33.

4. Deny *all other* inbound traffic on port 12.

The following ACL model, when assigned to inbound filtering on port 12, supports the above case:

```
ProCurve(config)# show access-list config

ip access-list extended "100"
```
1️⃣ `permit ip 11.11.11.42 0.0.0.0 0.0.0.0 255.255.255.255`
2️⃣ `deny tcp 11.11.11.101 0.0.0.0 0.0.0.0 255.255.255.255 eq 23`
3️⃣ `permit ip 11.11.11.101 0.0.0.0 0.0.0.0 255.255.255.255`
4️⃣ `permit tcp 11.11.11.33 0.0.0.0 0.0.0.0 255.255.255.255 eq 23`
5️⃣ *< implicit deny IP any >*
```
ProCurve(config)# access-group 100 in
```

1. **Permits** IP traffic inbound from source address 11.11.11.42. Packets matching this criterion are permitted and will not be compared to any later ACE in the list. Packets not matching this criterion will be compared to the next entry in the list.

2. **Denies** Telnet traffic from source address 11.11.11.101. Packets matching this criterion are dropped and are not compared to later criteria in the list. Packets not matching this criterion are compared to the next entry in the list.

3. **Permits** any IP traffic from source address 11.11.11.101. Any packets matching this criterion will be permitted and will not be compared to any later criteria in the list. Because this entry comes after the entry blocking Telnet traffic from this same address, there will not be any Telnet packets to compare with this entry; they have already been dropped as a result of matching the preceding entry.

4. **Permits** Telnet traffic from source address 11.11.11.33. Packets matching this criterion are permitted and are not compared to any later criteria in the list. Packets not matching this criterion are compared to the next entry in the list.

5. This entry does not appear in an actual ACL, but is implicit as the last entry in every ACL. Any inbound packets on port 12 that do not match any of the criteria in the ACL's preceding entries will be denied (dropped).

Figure 10-5. Example of How an ACL Filters Packets

It is important to remember that this ACL (and all ACLs) include an implicit **deny any**. That is, inbound IP packets (including switched packets having the switch as the destination IP address) that the ACL does not *explicitly* permit or deny will be *implicitly* denied, and therefore dropped. You can preempt the implicit deny by inserting a "permit IP any" at the end of an ACL, but this solution does not apply in the preceding example, where the intention is for the switch to allow only explicitly permitted packets inbound on port 12.

Overriding the Implicit "Deny Any".   If you want an ACL to permit any inbound packets that are not explicitly denied by other entries in the ACL, you can do so by configuring a **permit any** entry as the last entry in the ACL. Doing so permits any packet not explicitly denied by earlier entries. (On extended ACLs, you must configure **permit ip any any**.)

Planning an ACL Application on a Series 3400cl or Series 6400cl Switch

Before creating and implementing ACLs, you should understand the Series 3400cl and Series 6400cl switch resources available per-port to support ACL operation, define the policies you want your ACLs to enforce, and understand how your ACLs will impact your network users.

Switch Resource Usage

ACLs, IGMP, QoS, and Rate Limiting share certain 3400cl/6400cl switch per-port resources and load these resources in ways that require more careful attention to per-port resource usage when planning a configuration using these features. Otherwise, there is an increased possibility of fully consuming some port resources, which means that at some point the switch would not support further ACL, QoS, and/or Rate-Limiting configurations on one or more ports (and/or IGMP on the switch). This section describes resource planning for ACLs on a 3400cl or 6400cl switch. For QoS resource planning, refer to chapter 8, "Quality of Service (QoS): Managing Bandwidth More Effectively". For Rate-Limiting resource planning, refer to the "Rate Limiting" section in the chapter titled "Port Traffic Controls" of the *Management and Configuration Guide* for your switch.

Prioritizing and Monitoring ACL, IGMP, QoS, and Rate Limiting Feature Usage

If you want to configure ACLs and either QoS or Rate-Limiting (or both) on the same 3400cl or 6400cl port(s), plan and implement your per-port configuration in descending order of feature importance. This will help to ensure that the most important features are configured first on any given port. Also, if insufficient resources become a problem, this approach can help you recognize how to distribute the desired feature implementations across multiple switches to achieve your objectives.

Note   ACLs on the Series 3400cl and Series 6400cl switches are applied per-port. Except for the source-port classifier, QoS on 3400cl/6400cl switches is applied across either all physical interfaces on the switch or across all physical interfaces on a specified VLAN. This means that in most cases a QoS configuration applies to multiple ports while an ACL configuration applies only to specifically designated ports.

Adding ACLs consumes per-port ACL mask resources rapidly. If ACLs are more important on particular 3400cl or 6400cl switch ports than IGMP, then you should plan and configure your ACL resource usage first for those ports, then give attention to configuration of IGMP. If insufficient resources remain for IGMP, try applying IGMP on other switches.

ACL Resource Usage and Monitoring

ACL configurations on the 3400cl/6400cl switches use internal rule and mask resources on a per-port basis. Per-Port rule and mask usage is reserved as shown below:

Feature	Maximum Internal Masks Available Per-Port	Maximum Internal Rules Available Per-Port
ACLs and IGMP*	8 ACL Masks*	120 maximum

*Enabling IGMP on one or more VLANs consumes one per-port ACL mask on all ports. If all per-port ACL masks are used up on any port in the switch, IGMP cannot be configured. If all rules are used, but at least one mask remains, IGMP can be configured.

The switch consumes per-port (internal) rule and mask resources required by the ACEs in an ACL when you apply the ACL to one or more port and/or static trunk interfaces.

Standard ACLs:

■   Each ACE, including the implicit **deny any** ACE in a standard ACL, uses one port rule.

■   Contiguous ACE entries with the same subnet mask use the same port mask. Contiguous ACE entries with different subnet masks use one port mask per entry. To conserve ACL mask resources, group ACEs with identical subnet masks together. For example:

Table 10-2.   Minimizing Per-Port Mask Usage

Contiguous ACEs with the Same Subnet Mask		Contiguous ACEs with Different Subnet Masks	
The ACEs in this sequence use **two** port masks because entries with identical subnet masks are contiguous. This method optimizes the capacity of an ACL to accept ACEs requiring different port masks because it minimizes port mask usage.		This sequence uses the same entries as the column to the left, but each consecutive entry has a subnet mask that differs from its predecessor, and requires **four** port masks. This method of ordering ACEs unnecessarily consumes port masks and reduces the capacity of an ACL to accept ACEs requiring different port masks.	
15.28.247.1/24	(15.28.247.1 255.255.255.0)	15.28.247.1/24	(15.28.247.1 255.255.255.0)
15.28.253.1/24	(15.28.253.1 255.255.255.0)	10.0.8.0/32	(10.0.8.0 0.0.0.0)
10.0.8.0/32	(10.0.8.0 0.0.0.0)	15.28.253.1/24	(15.28.253.1 255.255.255.0)
10.0.8.105/32	(10.0.8.0 0.0.0.0)	10.0.8.105/32	(10.0.8.0 0.0.0.0)

- An ACL with no ACEs except a **permit any** or a **deny any** uses only one rule and one mask because the IP address and subnet mask are duplicates of the IP address and subnet mask used for the implicit **deny any** ACE that the switch automatically includes at the end of each ACL.

Table 10-3 on page 10-21 summarizes switch use of resources to support ACES.

Extended ACLs:

- Each ACE, including the implicit **deny ip any any** ACE in an extended ACL uses one port rule.

- Contiguous ACE entries with the same subnet mask and the same IP or TCP/UDP protocol applications use the same port mask. Contiguous ACE entries with different subnet masks or different IP-TCP/UDP applications use one port mask per entry. To conserve ACL mask resources, group ACEs with identical subnet masks and IP or TCP/UDP applications together. (The effect of this grouping is the same as above for the standard ACLs, but with more elements to consider.)

- An extended ACL with no ACEs except a **permit ip any any** or **deny ip any any** uses one rule and one mask. This is because the IP address and subnet mask are duplicates of the IP address and subnet mask used for the implicit **deny ip any any** ACE that the switch automatically includes at the end of every ACL.

Table 10-3. ACL Rule and Mask Resource Usage

ACE Type	Per-Port Rule Usage	Per-Port Masks Usage
Standard ACLs		
Implicit **deny any** (automatically included in any standard ACL, but not displayed by **show access-list** < *acl-#* > command).	1	1
First ACE entered	1	1
Next ACE entered with same ACL mask[1]	1	0
Next ACE entered with a different ACL mask[1]	1	1
Closing ACL with a deny any or permit any ACE having the same ACL mask as the preceding ACE	0	0
Closing ACL with a deny any or permit any ACE having a different ACL mask than the preceding ACE	1	1
Extended ACLs		
Implicit **deny ip an any** (automatically included in any standard ACL, but not displayed by **show access-list** < *acl-#* > command).	1	1
First ACE entered	1	1
Next ACE entered with same SA/DA ACL mask and same IP or TCP/UDP protocols specified[2]	1	0
Next ACE entered with any of the following differences from preceding ACE in the list: – Different SA or DA ACL mask – Different protocol (IP as opposed to TCP/UDP) specified in either the SA or DA[3]	1	1
Closing an ACL with a **deny ip any any** or **permit ip any any** ACE preceded by an IP ACE with the same SA and DA ACL masks	0	0
Closing an ACL with a **deny ip any any** or **permit ip any any** ACE preceded by an IP ACE with different SA and/or DA ACL masks	1	1

[1]In a given standard ACL, consecutive ACEs must have identical ACL masks in their SA entries to avoid using a separate per-port mask for each ACE. In a given standard ACL, If two ACEs having identical SA ACL masks are separated by an ACE with a different SA ACL mask, then three per-port masks are used instead of two; one for each sequential change in SA ACL masks. Thus, you can conserve per-port resources by grouping SA entries with the same ACL mask together.

[2]In a given extended ACL, consecutive ACEs must have the same SA and DA ACL mask and the same protocol application (IP as opposed to TCP/UDP) to avoid using a separate per-port mask for each ACE. If consecutive ACEs have different SA or DA ACL masks, or different protocol applications, then each such ACE consumes a separate per-port mask.

[3]TCP and UDP are the same for the purpose of determining per-port mask use. Also, actual TCP or UDP port numbers can vary between ACEs without affecting per-port mask usage. However, if one ACE specifies a TCP/UDP source port and another does not, another per-port mask will be used.

The following two CLI commands are unique to the 3400cl/6400cl switches and are useful for planning and monitoring rule and mask usage in an ACL configuration.

Syntax: access-list resources help

> *Provides a quick reference on how ACL, QoS and Rate-Limiting use rule resources and how ACL uses mask resources for each configuration option. Includes most of the information in table 10-3, plus an ACL usage summary.*

Syntax: show access-list resources

> *Shows the number of rules and ACL masks currently available on each port. This command is useful for verifying rule and ACL mask availability as you proceed with configuring ACL, IGMP, QoS, and/or Rate-Limiting features available on the switch.*

Managing ACL Resource Consumption

As shown in table 10-3, changes in IP subnet masks or changes in IP or TCP/UDP applications among consecutive ACEs in an assigned ACL can rapidly consume per-port mask resources. Also, in almost all cases, adding a new ACE to an ACL consumes one per-port rule. An extensive ACL configuration can fully subscribe the 120 rule resources available on one or more ports, especially when QoS and Rate-Limiting are also configured on the switch. (Configuring IGMP uses one per-port ACL mask, but does not use any per-port rules.) However, a relatively short ACL can fully subscribe the eight mask resources available on one or more ports. (The switch allows one ACL per-port.)

Oversubscribing Available Resources

If a given ACL requires more mask or rule resources on a port than are available, then the switch cannot apply the ACL to *any* of the interfaces specified for that ACL. In this case, the **access-group** command fails and the CLI displays the following:

■   In the CLI:

```
Unable to apply access control list.
```

■   In the Event Log (and in a Syslog server, if configured on the switch):

```
ACL: unable to apply ACL < acl-# > to port < port-# >, failed
to add entry < # >
```

(Note that < *port-#* > is the first port in the assignment command that was unable to support the ACL.)

Troubleshooting a Shortage of Per-Port Resources

As noted above, a lack of available per-port rules can be caused by a combination of ACL, IGMP, QoS, and Rate-Limiting applications. A lack of available ACL masks is caused by configuring an ACL to oversubscribe the number of per-port masks available for ACLs. (Also, note that enabling IGMP on a VLAN consumes one ACL mask per-port for all ports on the switch, leaving seven available per-port masks for ACL applications.)

Do the following to determine how to change resource usage to allow the ACL you want to configure:

1. Use the **show access-list resources** command to identify the port(s) on which there are insufficient rule resources. For example, figure 10-6 includes ports that can be the source of problems due to rule consumption by policies configured earlier:

```
ProCurve(config)# show qos resources
QoS/ACL Resource Usage

Port | Rules Available | ACL Masks Available
-----+-----------------+-----------------
  1  |  104            |  8
  2  |   40            |  6
  3  |    2            |  6
  4  |    1            |  6
  5  |    0            |  6
  6  |   86            |  7
  .  |    .            |  .
  .  |    .            |  .
  .  |    .            |  .
```

In this example, suppose that earlier configuration of QoS policies have depleted the rule resources on ports 4 and 5 to the point where there are not enough rules remaining for applying an ACL, and only enough rules on port 3 for a minimal ACL.

At a minimum, the policies previously configured on ports 4 and 5 must be reduced to free up enough rule resources to allow you to apply an ACL to these ports. Depending on the ACL you want to apply to port 3, existing QoS policies on port 3 may have to be reduced.

Port 3 has enough rules available to accept an ACL that uses 1 or 2 rules.

Port 4 can accept only an ACL with one entry that has either the same (standard) ACL mask as **deny any** or the same (extended) ACL that has the same SA/DA ACL mask and same IP protocol.

Figure 10-6. Example of Inspecting Available Rule (and Mask) Resources

2. Use **show** commands to identify the currently configured ACL, QoS, and Rate-Limiting policies, and any per-VLAN IGMP configuration.

3. Determine which of the existing policies you can remove to free up rule resources for the ACL policy you want to implement. Depending on your network topology and configuration, you can free up rule resources by moving some policies to other devices. Another alternative is to inspect

the switch's existing configuration for unnecessary QoS and rate-limiting entries or inefficient applications that could be removed or revised to achieve the desired policies with less resource usage. Tables10-3 on page 10-21 and 8-10 on page 8-17, and the information displayed by the **access-list resources help** command, can help you to determine the resource usage of ACL and QoS policies.

Guidelines for Reconfiguring an ACL to Use Fewer Port Masks. If an ACL requires more mask resources than are reserved on a port for ACL use (maximum: eight per-port; seven if IGMP is configured), then the remedy is to reduce mask consumption by:

a. Ensuring that the ACEs in the list are in a sequence that takes optimum advantage of the switch's ability to re-use a mask on consecutive ACEs in a list. (Refer to table 10-2 on page 10-20.)

b. Removing enough ACEs from the ACL to reduce mask consumption to no more than the available maximum.

If an ACL requires more rule resources on a port than are available (a maximum of 120), then the remedy is to reduce rule consumption by:

a. Examining the ACEs in the list and, where feasible, combining multiple ACEs into a single ACE with a broader application.

b. If QoS or Rate-Limiting are applied to the same port(s) where you want to apply the ACL, prioritize your use of resources and eliminate enough of the lower-priority applications to allow you to apply the ACL. This may include shifting some applications to other switches.

Example of ACL Resource Usage

This example illustrates how to check for current per-port rule and mask availability, and then how to create and assign an ACL, and then to verify its effect on per-port rule and mask resources. (For more detailed information on configuring and applying ACLs, refer to the later sections of this chapter.)

Viewing the Current Per-Port Rule and Mask Usage

The **show access-list resources** command displays the currently available per-port rules and masks.

```
ProCurve(config)# show access-list resources
QoS/ACL Resource Usage

Port | Rules Available | ACL Masks Available
-----+-----------------+--------------------
  1  | 120             | 8
  2  | 120             | 8
  3  | 120             | 8
  4  | 120             | 8
  5  | 120             | 8
  6  | 120             | 8
  7  | 120             | 8
  8  | 120             | 8
  .  |  .              | .
  .  |  .              | .
  .  |  .              | .
```

In the default configuration, there are 120 rules and 8 per-port ACL masks available on each port. These masks are reserved for ACLs and IGMP use.

Figure 10-7. Example of Available Per-Port Rules and ACL Masks

Standard ACL Using a Subset of the Switch's Ports. Suppose that ports 1 - 4 on a 3400cl or 6400cl switch belong to the following VLANs:

■ VLAN 1: 10.10.10.1

■ VLAN 2: 10.10.11.1

■ VLAN 3: 10.10.12.1

(Assume that ports 1-4 are tagged members of VLAN 22, although tagged/untagged ports do not affect ACL operation because ACLs examine all inbound traffic, regardless of VLAN membership.)

The system administrator wants to:

■ Permit inbound VLAN 1 traffic on all ports

■ Permit inbound VLAN 2 traffic on ports 1 - 4 from hosts 10.10.10.1-30

■ Deny inbound VLAN 2 traffic on ports 1 - 4 from hosts 10.10.10.31-255

■   Permit inbound VLAN 3 traffic on all ports.

Because all ports in the example have the same inbound traffic requirements for ACL filtering, the system administrator needs to create only one ACL for application to all four ports.

■   All inbound 10.10.10.x (VLAN 1) traffic is allowed on all ports.

■   For the inbound 10.10.11.x (VLAN 2) traffic, the fourth octet of the ACL mask includes an overlap of permit and deny use on the "16" bit, which will require two different ACEs in the ACL. That is:

•   To deny hosts in the range of 31-255 in the fourth octet, it is necessary to use an ACE that specifies the leftmost four bits of the octet.

•   To permit hosts in the range of 1-30 in the fourth octet, it is necessary to use and ACE that specifies the rightmost five bits of the octet.

The overlap[1] can be illustrated as shown here:

Bit Values in the Fourth Octet	128	64	32	16	8	4	2	1
Bits Needed To Deny Hosts 31 - 255 (4th Octet Mask: 0.0.0.224)	■	■	■	■				
Bits Needed To Permit Hosts 1 - 30 (4th Octet Mask: 0.0.0.31)				░	░	░	░	░
[1]For more on this topic, refer to "Rules for Defining a Match Between a Packet and an Access Control Entry (ACE)" on page 10-33, and "Using CIDR Notation To Enter the ACL Mask" on page 10-45.								

The overlap on the "16" bit means that it is necessary for the ACL to deny the host at 10.10.11.31 before permitting the hosts in the range of 10.10.10.1 - 30. The complete sequence is:

1.   Permit all inbound traffic from 10.10.10.x.

2.   Permit all inbound traffic from 10.10.12.x.

3.   Deny the host at 10.10.11.31.

4.   Permit the hosts in the range of 10.10.11.1 - 30.

5.   Allow the implicit deny (automatically present in all ACLs) to deny all other traffic, which will automatically include the hosts in the range 10.10.10.32 - 255.

```
ProCurve(config)# access-list 1 permit 10.10.10.1/24
ProCurve(config)# access-list 1 permit 10.10.12.1/24
ProCurve(config)# access-list 1 deny host 10.10.11.31
ProCurve(config)# access-list 1 permit 10.10.11.1/27
ProCurve(config)# show access-list 1

Access Control Lists

  Name: 1
  Type: Standard
  Applied: No

 ID   action        IP               Mask             Log
------------------------------------------------------------
  1   permit  std  10.10.10.1       0.0.0.255
  2   permit  std  10.10.12.1       0.0.0.255
  3    deny   std  10.10.11.31      0.0.0.0
  4   permit  std  10.10.11.1       0.0.0.31

ProCurve(config)# interface 1-4 access-group 1 in
ProCurve(config)# show access-list resources

QoS/ACL Resource Usage

 Port | Rules Available | Masks Available
------+-----------------+------------------
   1  |  115            |  4
   2  |  115            |  4
   3  |  115            |  4
   4  |  115            |  4
   5  |  120            |  8
   6  |  120            |  8
   7  |  120            |  8
   8  |  120            |  8
   .  |   .             |  .
   .  |   .             |  .
   .  |   .             |  .
```

Every standard ACL has at least two ACEs; the first ACE that you configure, and the implicit **deny any** ACE that follows all other configured ACEs in the ACL. The first ACE and the implied **deny any** together consume two per-port rules and two per-port masks.

ACE # 2 consumes one per-port rule. It does not consume a per-port mask because both entries use the same ACL mask (0.0.0.255).

ACE # 3 consumes one per-port rule and one per-port mask. The additional per-port mask is used because the ACL mask for ACE # 3 is different from the ACL mask used in the immediately preceding ACE (0.0.0.0 as opposed to 0.0.0.255).

ACE # 4 consumes one per-port rule and one per-port mask. The additional per-port mask is used because, again, it is not a duplicate of the ACL mask for the preceding ACE.

The **show access-list resources** command shows that the applied access list consumes five per-port rules and four per-port (ACL) masks.

Figure 10-8. Example of Show Access List Command

Traffic Management and Improved Network Performance

You can use ACLs to block unnecessary traffic caused by individual hosts, workgroups, or subnets, and to block user access to subnets, devices, and services. Answering the following questions can help you to design and properly position ACLs for optimum network usage.

■   What are the logical points for minimizing unwanted traffic? In many cases it makes sense to block unwanted traffic from the core of your network by configuring ACLs to drop such traffic at or close to the edge of the network. (The earlier in the network path you block unwanted traffic, the greater the benefit for network performance.)

■   What traffic should you explicitly block? Depending on your network size and the access requirements of individual hosts, this can involve creating a large number of ACEs in a given ACL (or a large number of ACLs), which increases the complexity of your solution and rapidly consumes the per-port rule and mask resources.

■   What traffic can you implicitly block by taking advantage of the implicit **deny any** to deny traffic that you have not explicitly permitted? This can reduce the number of entries needed in an ACL and make more economical use of switch resources.

■   What traffic should you permit? In some cases you will need to explicitly identify permitted traffic. In other cases, depending on your policies, you can insert a **permit any** (standard ACL) or **permit ip any any** (extended ACL) entry at the end of an ACL. This means that all IP traffic not specifically matched by earlier entries in the list will be permitted.

Security

ACLs can enhance security by blocking inbound IP traffic carrying an unauthorized source IP address (SA). This can include:

■   Blocking access to or from subnets in your network

■   Blocking access to or from the internet

■   Blocking access to sensitive data storage or restricted equipment

■   Preventing the use of specific TCP or UDP functions (such as Telnet, SSH, web browser) for unauthorized access

You can also enhance switch management security by using ACLs to block inbound IP traffic that has the switch itself as the destination address (DA).

C a u t i o n

ACLs can enhance network security by blocking selected IP traffic, and can serve as one aspect of maintaining network security. *However, because ACLs do not provide user or device authentication, or protection from malicious manipulation of data carried in IP packet transmissions, they should not be relied upon for a complete security solution.*

N o t e

ACLs in the 3400cl/6400cl switches do not screen non-IP traffic such as AppleTalk and IPX.

Guidelines for Planning the Structure of an ACL

The first step in planning a specific ACL is to determine where you will apply it. (Refer to "ACL Inbound Application Points" on page 10-10.) You must then determine the order in which you want the individual ACEs in the ACL to filter traffic. Some applications require high usage of the per-port resources the switch uses to support ACLs (as well as the rules used by QoS and Rate-Limiting applications). In these cases it is important to order the individual ACEs in a list to avoid unnecessarily using resources. For more on this topic, refer to "Planning an ACL Application on a Series 3400cl or Series 6400cl Switch" on page 10-17.

■ The first match dictates the action on a packet. possible, subsequent matches are ignored.

■ On any ACL, the switch implicitly denies packets that are not explicitly permitted or denied by the ACEs configured in the ACL. If you want the switch to forward a packet for which there is not a match in an ACL, add **permit any** as the last ACE in an ACL. This ensures that no packets reach the implicit **deny any** case.

■ Generally, you should list ACEs from the most specific (individual hosts) to the most general (subnets or groups of subnets) unless doing so permits traffic that you want dropped. For example, an ACE allowing a small group of workstations to use a specialized printer should occur earlier in an ACL than an entry used to block widespread access to the same printer.

ACL Configuration and Operating Rules

■ **Per-Interface ACL Limits.** At a minimum an ACL must have one, explicit "permit" or "deny" Access Control Entry. You can assign one ACL per interface, as follows:

- Standard ACLs—Numeric range: 1 - 99
- Extended ACLs—Numeric range: 100 - 199
- Named (Extended or Standard) ACLs: Up to the maximum number of ports on the switch (minus any numeric ACL assignments)

■ **Implicit "deny any":** In any ACL, the switch automatically applies an implicit "deny IP any" that does not appear in **show** listings. This means that the ACL denies any packet it encounters that does not have a match with an entry in the ACL. Thus, if you want an ACL to permit any packets that you have not expressly denied, you must enter a **permit any** or **permit ip any any** as the last visible ACE in an ACL. Because, for a given packet the switch sequentially applies the ACEs in an ACL until it finds a match, any packet that reaches the **permit any** or **permit ip any any** entry will be permitted, and will not encounter the "deny ip any" ACE the switch automatically includes at the end of the ACL. For an example, refer to figure 10-5 on page 10-16.

■ **Explicitly Permitting Any IP Traffic:** Entering a **permit any** or a **permit ip any any** ACE in an ACL permits all IP traffic not previously permitted or denied by that ACL. Any ACEs listed after that point do not have any effect and unnecessarily use rule and mask resources.

■ **Explicitly Denying Any IP Traffic:** Entering a **deny any** or a **deny ip any any** ACE in an ACL denies all IP traffic not previously permitted or denied by that ACL. Any ACEs listed after that point have no effect.

■ **An ACL Assignment Is Exclusive:** The switch allows one ACL assignment on an interface. If a port or static trunk already has an ACL assigned, you cannot assign another ACL to the interface without first removing the currently assigned ACL.

■ **Replacing One ACL with Another:** Where an ACL is already assigned to an interface, you must remove the current ACL assignment before assigning another ACL to that interface. If an assignment command fails because one or more interfaces specified in the command already have an ACL assignment, the switch generates this message in the CLI and in the Event Log:

< *acl-list-#* >: Unable to apply access control list.

■ **ACLs Operate On Ports and Static Trunk Interfaces:** You can assign an ACL to any port and/or any statically configured trunk on the switch. ACLs do not operate with dynamic (LACP) trunks.

■ **ACLs Screen Only the Traffic Entering the Switch on a Port or Static Trunk Interface:** On a given interface, ACLs can screen inbound traffic at the point where it enters the switch. In the 3400cl/ 6400cl switches, ACLs do not screen traffic routed between VLANs within the switch, between subnets in a multinetted VLAN, or at the interface where the traffic exits from the switch. (See figure 10-2 on page 10-11.)

■ **Before Modifying an Applied ACL, You Must First Remove It from All Assigned Interfaces:** An ACL cannot be changed while it is assigned to an interface.

■ **Before Deleting an Applied ACL, You Must First Remove It from All Interfaces to Which It Is Assigned:** An assigned ACL cannot be deleted.

■ **Port and Static Trunk Interfaces:**
   • Removing a port from an ACL-assigned trunk returns the port to its default settings.
   • To add a port to a trunk when an ACL is already assigned to the port, you must first remove the ACL assignment from the port.
   • Adding a new port to an ACL-assigned trunk automatically applies the ACL to the new port.

How an ACE Uses a Mask To Screen Packets for Matches

When the switch applies an ACL to inbound traffic on an interface, each ACE in the ACL uses an IP address and *ACL mask* to enforce a selection policy on the packets being screened. That is, the mask determines the range of IP addresses (SA only or SA/DA) that constitute a match between the policy and a packet being screened.

What Is the Difference Between Network (or Subnet) Masks and the Masks Used with ACLs?

In common IP addressing, a network (or subnet) mask defines which part of the IP address to use for the network number and which part to use for the hosts on the network. For example:

IP Address	Mask	Network Address	Host Address
18.38.252.195	255.255.255.0	first three octets	The fourth octet.
18.38.252.195	255.255.248.0	first two octets and the left-most five bits of the third octet	The right most three bits of the third octet and all bits in the fourth octet.

Thus, the bits set to 1 in a network mask define the part of an IP address to use for the network number, and the bits set to 0 in the mask define the part of the address to use for the host number.

In an ACL, IP addresses and masks provide the criteria for determining whether to deny or permit a packet, or to pass it to the next ACE in the list. If there is a match, the deny or permit action occurs. If there is not a match, the packet is compared with the next ACE in the ACL. Thus, where a standard network mask defines how to identify the network and host numbers in an IP address, the mask used with ACEs defines which bits in a packet's IP address must match the corresponding bits in the IP address listed in an ACE, and which bits can be *wildcards*.

Rules for Defining a Match Between a Packet and an Access Control Entry (ACE)

■ For a given ACE, when the switch compares an IP address and corresponding mask in the ACE to an IP address carried in a packet:

- **A mask-bit setting of 0 ("off")** requires that the corresponding bit in the packet's IP address and in the ACE's IP address must be the same. That is, if a bit in the ACE's IP address is set to 1 ("on"), the same bit in the packet's IP address must also be 1.

- **A mask-bit setting of 1 ("on")** means the corresponding bit in the packet's IP address and in the ACE's IP address do not have to be the same. That is, if a bit in the ACE's IP address is set to 1, the same bit in the packet's IP address can be either 1 or 0 ("on" or "off").

For an example, refer to "Example of How the Mask Bit Settings Define a Match" on page 10-35.

■ In any ACE, a mask of all ones means *any* IP address is a match. Conversely, a mask of all zeros means the *only* match is an IP address identical to the host IP address specified in the ACL.

■ Depending on your network, a single ACE that allows a match with more than one source or destination IP address may allow a match with multiple subnets For example, in a network with a prefix of 31.30.240 and a subnet mask of 255.255.240.0 (the leftmost 20 bits), applying an ACL mask of 0.0.31.255 causes the subnet mask and the ACL mask to overlap one bit, which allows matches with hosts in two subnets: 31.30.224.0 and 31.30.240.0.

Bit Position in the Third Octet of Subnet Mask 255.255.240.0								
Bit Values	128	64	32	16	8	4	2	1
Subnet Mask Bits	1	1	1	1	n/a	n/a	n/a	n/a
Mask Bit Settings Affecting Subnet Addresses	0	0	0	**1 or 0**	n/a	n/a	n/a	n/a

This ACL supernetting technique can help to reduce the number of ACLs you need. You can apply it to a multinetted VLAN and to multiple VLANs. However, ensure that you exclude subnets that do not belong in the policy. If this creates a problem for your network, you can eliminate the unwanted match by making the ACEs in your ACL as specific as possible, and using multiple ACEs carefully ordered to eliminate unwanted matches.

■ Every IP address and mask pair (source or destination) used in an ACE creates one of the following policies:

- **Any IP address fits the matching criteria.** In this case, the switch automatically enters the IP address and mask in the ACE. For example:

    access-list 1 deny any

    produces this policy in an ACL listing:

IP Address	Mask
0.0.0.0	255.255.255.255

    This policy states that every bit in every octet of a packet's SA is a wildcard, which covers any IP address.

- **One IP address fits the matching criteria.** In this case, you provide the IP address and the switch provides the mask. For example:

    access-list 1 permit host 18.28.100.15

    produces this policy in an ACL listing:

IP Address	Mask
18.28.100.15	0.0.0.0

    This policy states that every bit in every octet of a packet's SA must be the same as the corresponding bit in the SA defined in the ACE.

- **A group of IP addresses fits the matching criteria.** In this case you provide both the IP address and the mask. For example:

    access-list 1 permit 18.28.32.1 0.0.0.31

IP Address	Mask
18.28.32.1	0.0.0.31

    This policy states that:
    – In the first three octets of a packet's SA, every bit must be set the same as the corresponding bit in the SA defined in the ACE.
    – In the last octet of a packet's SA, the first three bits must be the same as in the ACE, but the last five bits are wildcards and can be any value.

■ Unlike subnet masks, the wildcard bits in an ACL mask need not be contiguous. For example, 0.0.7.31 is a valid ACL mask. However, a subnet mask of 255.255.248.224 is not a valid subnet mask.

Example of How the Mask Bit Settings Define a Match . Assume an ACE where the second octet of the mask for an SA is 7 (the rightmost three bits are "on", or "1") and the second octet of the corresponding SA in the ACE is 31 (the rightmost five bits). In this case, a match occurs when the second octet of the SA in a packet being filtered has a value in the range of 24 to 31. Refer to table 10-4, below.

Table 10-4. Example of How the Mask Defines a Match

Location of Octet	Bit Position in the Octet							
	128	64	32	16	8	4	2	1
SA in ACE	0	0	0	1	1	1	1	1
Mask for SA	0	0	0	0	0	1	1	1
Corresponding Octet of a Packet's SA	0	0	0	1	1	0/1	0/1	0/1

The shaded area indicates bits in the packet that must exactly match the bits in the source IP in the ACE. Wherever the mask bits are ones (wildcards), the IP bits in the packet can be any value, and where the mask bits are zeros, the IP bits in the packet must be the same as the IP bits in the ACE. **Note:** This example covers only one octet of an IP address. An actual ACE applies this method to all four octets of an IP address.

Example of Allowing Only One IP Address ("Host" Option). Suppose, for example, that you have configured the ACL in figure 10-9 to filter inbound packets on port 20. Because the mask is all zeros, the ACE policy dictates that a match occurs only when the source IP address on such packets is identical to the IP address configured in the ACE.

.

```
ip access-list standard Fileserver
  permit 10.28.252.117  0.0.0.0
exit
```

This ACL (a standard ACL named "Fileserver") includes an ACE (Access Control Entry) that permits matches only with the packets received from IP address 10.28.252.117 (the SA). Packets from any other source do not match and are denied.

Source IP Address (SA)

Mask (All zeros mean that a match occurs only when an IP packet's source address is identical to the source address configured in the ACE.

ACE

Inbound Packet "A" On Port 20
  – Destination Address: 10.35.248.184
  – Source Address: 10.28.252.117

The port permits packet "A" because its source IP address matches the source address in the ACE.

Inbound Packet "B" On Port 20
  – Destination Address: 10.35.248.184
  – Source Address: 10.28.252.120

The port denies (drops) packet "B" because its source IP address does not match the source address in the ACE.

Figure 10-9. Example of an ACL with an Access Control Entry (ACE) that Allows Only One Source IP Address

Examples Allowing Multiple IP Addresses. Table 10-5 provides examples of how to apply masks to meet various filtering requirements.

Table 10-5.   Example of Using an IP Address and Mask in an Access Control Entry

IP Address in the ACE	Mask	Policy for a Match Between a Packet and the ACE	Allowed IP Addresses
A: 10.38.252.195	0.0.0.255	Exact match in first three octets only.	10.38.252.< 0-255 > (See row A in table 10-6, below.)
B: 10.38.252.195	0.0.7.255	Exact match in the first two octets and the leftmost five bits (248) of the third octet.	10.38.< 248-255 >.< 0-255 > (In the third octet, only the rightmost three bits are wildcard bits. The leftmost five bits must be a match, and in the ACE, these bits are all set to 1. See row B in table 10-6, below.)
C: 10.38.252.195	0.0.0.0	Exact match in all octets.	10.38.252.195 (There are no wildcard bits in any of the octets. See row C in table 10-6, below.)
D: 10.38.252.195	0.15.255.255	Exact match in the first octet and the leftmost four bits of the second octet.	10.< 32-47 >.< 0-255 >.<0-255> (In the second octet, the rightmost four bits are wildcard bits. See row D in table 10-6, below.)

Table 10-6. Mask Effect on Selected Octets of the IP Addresses in Table 10-5

IP Addr	Octet	Mask	Octet Range	128	64	32	16	8	4	2	1
A	3	0 all bits	252	1	1	1	1	1	1	0	0
B	3	7 last 3 bits	248-255	1	1	1	1	1	0 or 1	0 or 1	0 or 1
C	4	0 all bits	195	1	1	0	0	0	0	1	1
D	2	15 last 4 bits	32-47	0	0	1	0	0 or 1	0 or 1	0 or 1	0 or 1

Shaded areas indicate bit settings that must be an exact match.

If there is a match between the policy in the ACE and the IP address in a packet, then the packet is either permitted or denied, according to how the ACE is configured. If there is not a match, the next ACE in the ACL is then applied to the packet. The same operation applies to a destination IP address (DA) used in an extended ACE. (Where an ACE includes both source and destination IP addresses, there is one IP-address/ACL-mask pair for the source address, and another IP-address/ACL-mask pair for the destination address. See "Configuring and Assigning an ACL" on page 10-38.)

CIDR Notation. For information on using CIDR notation to specify ACL masks, refer to "Using CIDR Notation To Enter the ACL Mask" on page 10-45.

Configuring and Assigning an ACL

ACL Feature	Page
Configuring and Assigning a Numbered, Standard ACL	10-47
Configuring and Assigning a Numbered, Extended ACL	10-52
Configuring a Named ACL	10-58
Enabling or Disabling ACL Filtering	10-61

Overview

General Steps for Implementing ACLs

1. Configure at least one ACL. This creates and stores the ACL in the switch configuration.

2. Assign an ACL. This applies the ACL to the inbound traffic on one or more designated interfaces.

Caution Regarding the Use of Source Routing

Source routing is enabled by default on the switch and can be used to override ACLs. For this reason, if you are using ACLs to enhance network security, the recommended action is to disable source routing on the switch. To do so, execute **no ip source-route**.

Types of ACLs

■ **Standard ACL:** Uses only a packet's source IP address as a criterion for permitting or denying the packet. For a standard ACL ID, use either a unique numeric string in the range of 1-99 or a unique name string of up to 64 alphanumeric characters.

■ **Extended ACL:** Offers the following criteria as options for permitting or denying a packet:

   • Source IP address
   • Destination IP address
   • TCP or UDP criteria

   For an extended ACL ID, use either a unique number in the range of 100-199 or a unique name string of up to 64 alphanumeric characters.

You should carefully plan your ACL application before configuring specific ACLs. For more on this topic, refer to "Planning an ACL Application on a Series 3400cl or Series 6400cl Switch" on page 10-17.

ACL Configuration Structure

After you enter an ACL command, you may want to inspect the resulting configuration. This is especially true where you are entering multiple ACEs into an ACL. Also, it will be helpful to understand the configuration structure when using later sections in this chapter.

The basic ACL structure includes three elements:

1. ACL type and name: This identifies the ACL as **standard** or **extended** and shows the ACL name.

2. One or more deny/permit list entries (ACEs): One entry per line.

Element	Stnd	Ext	Notes
ID Range	1 - 99	100 - 199	You can also use an alphanumeric name of up to 64 characters, including spaces.
Minimum ACEs per ACL	1		
Maximum ACEs Per ACL	120		
Maximum ACEs per Switch	1024		In some cases, rule usage by ACLs, IGMP, QoS, and Rate-Limiting, and mask usage by ACLs may consume available resources to the point where this limit cannot be reached.

3. Implicit **deny any**: Where an ACL is in use, the switch denies any packets that do not have a match with the ACEs explicitly configured in the ACL. The implicit **deny any** does not appear in ACL configuration listings, but always functions when the switch uses an ACL to filter packets. (You cannot delete the implicit "deny any", but you can supersede it with a "permit any" statement.)

Standard ACL Structure

Individual ACEs in a standard ACL include only a permit/deny "type" statement, the source IP addressing, and an optional **log** command (available with "deny" statements).

```
ip access-list < type > "< id-string >"
  permit host < source-ip-address >
  deny < source-ip-address > < acl-mask > [log]
  .
  .
  .
  permit any
  exit
```

Figure 10-10. Example of the General Structure for a Standard ACL

For example, figure 10-11 shows how to interpret the entries in a standard ACL.
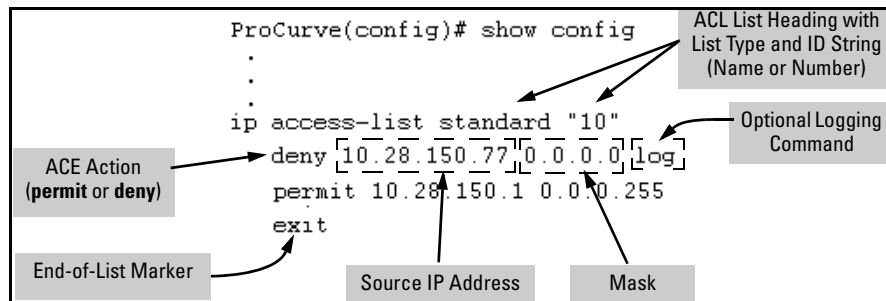
Figure 10-11. Example of a Displayed Standard ACL Configuration with Two ACEs

Extended ACL Configuration Structure

Individual ACEs in an extended ACL include:

■ A permit/deny "type" statement

■ Source IP addressing

■ Optional TCP or UDP port type with optional source port ID and operator and/or optional destination port ID and operator

■ Destination IP addressing

■   Optional ACL **log** command (available for "Deny" ACLs only)

```
ip access-list < type > "< id-string >"< permit | deny > ip                    Note: The optional log
    < source-ip-address > < source-acl-mask >                                   function appears only
    < destination-ip-address > < destination-acl-mask >[log]                    with "deny" aces.

    < permit | deny > tcp
        < source-ip-address > < source-acl-mask > [< operator > < port-id >]
        < destination-ip-address > < destination-acl-mask > [< operator > < port-id >] [log]

    < permit | deny > udp
        < source-ip-address > < source-acl-mask > [< operator > < port-id >]
        < destination-ip-address > < destination-acl-mask > [< operator > < port-id >] [log]
    . . .
    exit
```

Figure 10-12. General Structure for an Extended ACL

For example, figure 10-13 shows how to interpret the entries in an extended ACL.

Figure 10-13. Example of a Displayed Extended ACL Configuration

ACL Configuration Factors

ACL Resource Consumption

Consumption of per-port rules and masks can be a significant factor in switches using extensive ACL applications. In this case, resource usage takes precedence over other factors when planning and configuring ACLs. For more information on this topic, refer to "Planning an ACL Application on a Series 3400cl or Series 6400cl Switch" on page 10-17.

The Sequence of Entries in an ACL Is Significant

When the switch uses an ACL to determine whether to permit or deny a packet on a particular interface, it compares the packet to the criteria specified in the individual Access Control Entries (ACEs) in the ACL, beginning with the first ACE in the list and proceeding sequentially until a match is found. When a match is found, the switch applies the indicated action (permit or deny) to the packet. This is significant because, once a match is found for a packet, subsequent ACEs in the same ACL will not be used for that packet, regardless of whether they match the packet.

For example, suppose that you have applied the ACL shown in figure 10-11 to inbound traffic on port 10:

```
1   ip access-list extended "101"          Source              Destination
                                                                                        Source and
                                                                                        Destination
2     deny ip 10.28.235.10 0.0.0.0 0.0.0.0 255.255.255.255                               IP Addresses
                                                                                        for the ACE in
3     deny ip 10.28.245.89 0.0.0.0 0.0.0.0 255.255.255.255                               line 4 of the
                                                                                        ACL.
4     permit tcp 10.28.18.100 0.0.0.0 10.28.237.1 0.0.0.0

5     deny tcp 10.28.18.100 0.0.0.0 0.0.0.0 255.255.255.255

6     permit ip 0.0.0.0 255.255.255.255 0.0.0.0 255.255.255.255

                              Following the last explicit ACE in the ACL there is always an implicit "deny
7     exit                    any". However, in this case it will not be used because the last, explicit
                              permit statement allows all IP packets that earlier ACEs have not already
                              permitted or denied.
```

Figure 10-14. Example of a Standard ACL that Permits All Traffic Not Implicitly Denied

Table 10-7.   Effect of the ACL in Figure 10-14 on Inbound Traffic on the Assigned Port

Line #	Action
1	Shows list type (extended) and ID (101).
2	A packet from IP source address 10.28.235.10 will be denied (dropped). This line filters out all packets received from 10.28.235.10. As a result, IP traffic from that device will not be routed or switched, and packets from that device will not be compared against any later entries in the list.
3	A packet from IP source 10.28.245.89 will be denied (dropped). This line filters out all packets received from 10.28.245.89. As the result, IP traffic from that device will not be routed or switched and packets from that device will not be compared against any later entries in the list.
4	A packet from TCP source address 10.28.18.100 with a destination address of 10.28.237.1 will be permitted (forwarded). Since no earlier lines in the list have filtered TCP packets from 10.28.18.100 and destined for 10.28.237.1, the switch will use this line to evaluate such packets. Any packets that meet this criteria will be forwarded. (Any packets that do not meet this TCP source-destination criteria are not affected by this line.)
5	A packet from TCP source address 10.28.18.100 to *any* destination address will be denied (dropped). Since, in this example, the intent is to block TCP traffic from 10.28.18.100 to any destination *except* the destination stated in line 4, this line must follow line 4. (If their relative positions were exchanged, all TCP traffic from 10.28.18.100 would be dropped, including the traffic for the 10.28.18.1 destination.)
6	Any packet from any IP source address to any destination address will be permitted (forwarded). The only traffic to reach this line will be IP packets not specifically permitted or denied in the earlier lines.
n/a	The "implicit deny any any" is a function automatically added as the last action in all ACLs. It denies (drops) any IP traffic from any source to any destination that has not found a match with earlier entries in the list. In this example, line 6 permits (forwards) any IP traffic not already permitted or denied by the earlier entries in the list, so there is no traffic remaining for action by the "implicit deny any any" function.
7	Indicates the end of the ACL.

```
ProCurve(config)# show access-list resources
QoS/ACL Resource Usage

Port | Rules Available | ACL Masks Available
-----+-----------------+--------------------
  1  | 120             |  8
  2  | 120             |  8
  3  | 120             |  8
  4  | 120             |  8
  5  | 120             |  8
  6  | 120             |  8
  7  | 120             |  8
  8  | 120             |  8
  9  | 120             |  8
 10  | 115             |  4
 11  | 120             |  8
 12  | 120             |  8
  .  |  .              |  .
  .  |  .              |  .
  .  |  .              |  .
```

The ACL in figure 10-14, when applied to port 10, uses five per-port rules and four per-port masks. Note that the last ACE in the list is a duplicate of the implicit **deny ip any any** that is automatically included at the end of every extended ACL. As a result, the last configured (visible) ACE and the implicit **deny** ACE use the same rule and mask. For more on this topic, refer to table Table 10-3 on page 10-21.

Figure 10-15. Per-Port Rule and Mask Usage for the ACL in Figure 10-14

In Any ACL, There Will Always Be a Match

As indicated in figure 10-14, the switch automatically uses an implicit "deny IP any" (Standard ACL) or "deny IP any any" (Extended ACL) as the last ACE in any ACL. This means that if you configure the switch to use an ACL for filtering inbound traffic, any packets not specifically permitted or denied by the explicit entries you create will be denied by the implicit "deny" action. Note that if you want to preempt the implicit "deny" action, insert an explicit **permit any** or **permit ip any any** as the last line of the ACL.

A Configured ACL Has No Effect Until You
Apply It to an Interface

The switch stores ACLs in the configuration file. Thus, until you actually assign an ACL to an interface, it is present in the configuration, but not used.

Using the CLI To Create an ACL

Command	Page
access-list (standard ACLs)	10-47
access-list (extended ACLs)	10-52
ip access-list (named ACLs)	10-58

You can use either the switch CLI or an offline text editor to create an ACL. This section describes the CLI method, which is recommended for creating short ACLs. (To use the offline method, refer to "Editing ACLs and Creating an ACL Offline" on page 10-69.)

General ACE Rules

These rules apply to all ACEs you create or edit using the CLI:

■   ACEs are placed in an ACL according to the sequence in which you enter them (last entered, last listed).

■   You can use the CLI to delete an ACE from anywhere in a given ACL by using the "no" form of the command to enter that ACE. However, when you use the CLI to add an ACE, the new entry is always placed *at the end of the ACL.*

■   Duplicate ACEs are allowed in an ACL. However, multiple instances of an ACE have no effect on filtering because the first instance preempts any subsequent duplicates. Also, duplicate entries unnecessarily consume additional resources on assigned ACLs.

Using CIDR Notation To Enter the ACL Mask

You can use CIDR (Classless Inter-Domain Routing) notation to enter ACL masks. The switch interprets the bits specified with CIDR notation as the IP address bits in an ACL and the corresponding IP address bits in a packet. The switch then converts the mask to inverse notation for ACL use.

Table 10-8.   Examples of CIDR Notation for Masks

IP Address Used In an ACL with CIDR Notation	Resulting ACL Mask	Meaning
18.38.240.125/15	0.1.255.255	The leftmost 15 bits must match; the remaining bits are wildcards.
18.38.240.125/20	0.0.15.255	The leftmost 20 bits must match; the remaining bits are wildcards.
18.38.240.125/21	0.0.7.255	The leftmost 21 bits must match; the remaining bits are wildcards.
18.38.240.125/24	0.0.0.255	The leftmost 24 bits must match; the remaining bits are wildcards.
18.38.240.125/32	0.0.0.0	All bits must match.

Configuring and Assigning a Numbered, Standard ACL

Configuring Named ACLs	"Configuring a Named ACL" on page 10-58
Configuring Extended, Numbered ACLs	"Configuring and Assigning a Numbered, Extended ACL" on page 10-52

- To configure named ACLs, refer to "Configuring a Named ACL" on page 10-58.

- To configure extended, numbered ACLs, refer to "Configuring and Assigning a Numbered, Extended ACL" on page 10-52.

A standard ACL uses only source IP addresses in its ACEs. This type of ACE is useful when you need to:

- Permit or deny traffic based on source IP address only.

- Quickly control the IP traffic from a specific address, a group of addresses, or a subnet. This allows you to isolate traffic problems generated by a specific device, group of contiguous devices, or a subnet threatening to degrade network performance. This gives you an opportunity to troubleshoot without sacrificing performance for users outside of the problem area.

You can identify each standard ACL with a number in the range of 1 - 99, or an alphanumeric string of up to 64 characters. The CLI command process for using an alphanumeric string to name an ACL differs from the command process for a numeric name. For a description of how to name an ACL with an alphanumeric character string, refer to "Configuring a Named ACL" on page 10-58. To view the command differences, refer to table 10-1, "Comprehensive Command Summary" on page 10-6.

Note   For a summary of ACL commands, refer to table 10-1, "Comprehensive Command Summary", on page 10-6.

Syntax: [no] access-list

> *Creates an ACE in the specified (1-99) access list and indicates the action (deny or permit) to take on a packet if there is a match between the packet and the criterion in the entry. If the ACL does not already exist, this command creates the specified ACL and its first ACE. To create a named ACL, refer to "Configuring a Named ACL" on page 10-58*

< 1-99 >

> *Specifies the ACL ID number. The switch interprets an ACL with a value in this range as a standard ACL.*
>
> ***Note:*** *To create an access list with an alphanumeric name (**name-str**) instead of a number, refer to "Configuring a Named ACL" on page 10-58.*

< deny | permit >

> *Specifies whether to deny (drop) or permit (forward) a packet that matches the ACE criteria.*

< any | host < *src-ip-addr* > | *ip-addr / mask-length* >

- **any**—*Performs the specified action on any IP packet. Use this criterion to designate packets from any IP address.*

- **host** < *host ip-address* >—*Performs the specified action on any IP packet having the < host ip-address > as the source. Use this criterion to designate packets from a single IP address.*

- ***IP-addr / mask-length*** — *Performs the specified action on any IP packet having a source address within the range defined by either*

  < *src-ip-addr / cidr-mask-bits* **>**
  or
  < *src-ip-addr* < *mask* **>>**

  Use this criterion to filter packets received from either a subnet or a group of contiguous IP addresses. The mask can be in either dotted-decimal format or CIDR format with the number of significant bits. Refer to "Using CIDR Notation To Enter the ACL Mask" on page 10-45.

The mask is applied to the IP address in the ACL to define which bits in a packet's source IP address must exactly match the IP address configured in the ACL and which bits need not match. Note that specifying a group of contiguous IP addresses may require more than one ACE. For more on how masks operate in ACLs, refer to "How an ACE Uses a Mask To Screen Packets for Matches" on page 10-32.

[log]

Optionally generates an ACL log message if:

- *The action is **deny**.*
- *There is a match.*
- *ACL logging is enabled on the switch. (Refer to "Enable ACL "Deny" Logging" on page 10-75.)*

*(Use the debug command to direct ACL logging output to the current console session and/or to a Syslog server. Note that you must also use the **logging < i*p-addr **>** command to specify the IP addresses of Syslog servers to which you want log messages sent. See also "Enable ACL "Deny" Logging" on page 10-75.)*

Syntax: interface < *port-list* | *trunk* > access-group < *ASCII-STR* > in

Assigns an ACL, designated by an ACL ID (< ASCII-STR >), to an interface (list of one or more ports and/or one or more static trunks).

Example of a Standard ACL. Suppose you wanted to configure a standard ACL and assign it to filter inbound traffic on port 10 in a particular switch:

- The ID you selected for this ACL is "50".

- You want the ACL to deny IP traffic from all hosts except these three:
  - 10.128.100.10
  - 10.128.100.27
  - 10.128.100.14

```
ProCurve(config)# access-list 50 permit host 10.128.100.10
ProCurve(config)# access-list 50 permit host 10.128.100.27
ProCurve(config)# access-list 50 permit host 10.128.80.14
ProCurve(config)# interface 10 access-group 50 in
ProCurve(config)# write mem
ProCurve(config)# show config

Startup configuration:

; J4905A Configuration Editor; Created on release #M.08.01

hostname "ProCurve"

ip access-list standard "50"
    permit 10.128.100.10 0.0.0.0
    permit 10.128.100.27 0.0.0.0
    permit 10.128.80.14 0.0.0.0
    exit

interface 10
    access-group "50" in
    no lacp

exit
snmp-server community "public" Unrestricted
vlan 1
    name "DEFAULT_VLAN"
    untagged 1-24
    ip address dhcp-bootp
    exit

ProCurve(config)# show access-list resources
QoS/ACL Resource Usage

 Port | Rules Available | Masks Available
 -----+-----------------+-----------------
   1  | 120             |    15
   2  | 120             |    15
   3  | 120             |    15
   .  |   .             |     .
   .  |   .             |     .
   .  |   .             |     .
  10  | 116             |    13
   .  |   .             |     .
   .  |   .             |     .
   .  |   .             |     .
```

- Permits IP traffic from the indicated IP address. Since, for this example, ACL 50 is a new list, this command also creates the ACL.
- Permits IP traffic from the indicated IP address.
- The **deny any** that the switch implicitly includes in all standard ACLs denies IP packets from IP sources not included in the above three commands.

Show config lists any ACLs and ACL assignments configured in the startup-config.

ACL "50" is listed as assigned to filter inbound traffic on port 10.

show access-list resources shows the per-port rule and ACL mask usage on port 10 (and all other ports on the switch.

Figure 10-16. Example of Configuring a Standard ACL To Permit Only Traffic from Specific IP Addresses

In a situation opposite to the above, suppose that you wanted to deny inbound IP traffic received on port 20 from 10. 128.93.17 and 10.130.93.25, but permit all other IP traffic on this VLAN. The next ACL achieves this:

```
ProCurve(config)# access-list 60 deny host 10.128.93.17
ProCurve(config)# access-list 60 deny host 10.28.93.25
ProCurve(config)# access-list 60 permit any
ProCurve(config)# interface 20 access-group 60 in
ProCurve(config)# write mem
ProCurve(config)# show config
Startup configuration:

; J4905A Configuration Editor; Created on release #M.08.01
hostname "ProCurve"
ip access-list standard "50"
   permit 10.128.100.10 0.0.0.0
   permit 10.128.100.27 0.0.0.0
   permit 10.128.80.14 0.0.0.0
   exit
ip access-list standard "60"
   deny 10.128.93.17 0.0.0.0
   deny 10.28.93.25 0.0.0.0
   permit 0.0.0.0 255.255.255.255
   exit
interface 10
   access-group "50" in
   no lacp
exit
interface 20
   access-group "60" in
   no lacp
exit
snmp-server community "public" Unrestricted
vlan 1
   name "DEFAULT_VLAN"
   untagged 1-24
   ip address dhcp-bootp
   exit

ProCurve(config)# show access-list resources

QoS/ACL Resource Usage

 Port | Rules Available | Masks Available
 -----+-----------------+-----------------
    1 | 120             | 15
    . |    .            |    .
    . |    .            |    .
    . |    .            |    .
   10 | 116             | 13
    . |    .            |    .
    . |    .            |    .
    . |    .            |    .
   20 | 117             | 13
    . |    .            |    .
    . |    .            |    .
    . |    .            |    .
```

Denies IP traffic from the indicated IP address. Since, for this example, ACL 60 is a new list, this command also creates the ACL.

Denies IP traffic from the indicated IP address.

Permits IP traffic from all sources. (Traffic from the IP sources in the first two lines is already filtered and dropped.) The **deny any** with which the switch implicitly concludes all ACLs is preempted by this ACE (but is still present in the ACL).

Show config lists any ACLs and ACL assignments configured in the startup-config.

ACL "50" from the preceding example.

ACL "60" is listed in the switch configuration.

ACL "60" is assigned to filter inbound traffic on port 20.

Figure 10-17. Example of Configuring a Standard ACL To Deny Inbound Traffic from Specific IP Addresses

Configuring and Assigning a Numbered, Extended ACL

This section describes how to configure numbered, extended ACLs. To configure other ACL types, refer to the following table.

To Configure:	Refer To:
Standard, numbered ACLs	"Configuring and Assigning a Numbered, Standard ACL" on page 10-47
Named ACLs	"Configuring a Named ACL" on page 10-58

While standard ACLs use only source IP addresses for filtering criteria, extended ACLs allow multiple ACE criteria. This enables you to more closely define your IP packet-filtering criteria. These criteria include:

■ Source and destination IP addresses (required), in one of the following options:

  • Specific host IP

  • Subnet or group of IP addresses

  • Any IP address

■ IP protocol (IP, TCP, or UDP)

■ Source TCP or UDP port (if the IP protocol is TCP or UDP)

■ Destination TCP or UDP port (if the IP protocol is TCP or UDP)

■ TCP or UDP **eq** operator (if the IP protocol is TCP or UDP)

You can configure extended ACLs with a numeric name in the range of 100 - 199. You can also configure extended ACLs with alphanumeric names. (Refer to "Configuring a Named ACL" on page 10-58.)

Note

For a summary of ACL commands, refer to table 10-1, "Comprehensive Command Summary", on page 10-6.

Syntax: [no] access-list

Creates an ACE in the specified (100-199) access list and:
- *Indicates the action (deny or permit) to take on a packet if there is a match between the packet and the criteria in the complete ACE.*
- *Specifies the packet protocol type (IP, TCP, or UDP).*
- *Specifies the source and destination addressing options described in the remainder of this section.*
- *Allows optional ACL logging where a packet has a match with a* **deny** *ACE.*

If the ACL does not already exist, this command creates the specified ACL and its first ACE. If the ACL already exists, this command adds a new, explicit ACE to the end of the ACL. For a match to occur, the packet must have the source and destination IP addressing criteria specified by this command, as well as any protocol-specific (TCP or UDP port number) criteria specified by the command. To create a named ACL, refer to "Configuring a Named ACL" on page 10-58.

< 100-199 >

Specifies the ACL ID number. The switch interprets an ACL with a value in this range as an extended ACL.

Note: *To create an access list with an alphanumeric name instead of a number, refer to "Configuring a Named ACL" on page 10-58.*

< deny | permit >

Specifies whether to deny (drop) or permit (forward) a packet that matches the ACE criteria.

< ip | tcp | udp >

Specifies the packet protocol type required for a match:
- **ip** *— any IP packet*
- **tcp** *— only tcp packets*
- **udp** *— only udp packets*

< any | host < *src-ip-addr* > | *ip-addr/mask -length* >

In an extended ACL, this parameter defines the source IP address (SA) that a packet must carry in order to have a match with the ACE.

- **any** — *Specifies all inbound IP packets.*
- **host** < ***src-ip-addr*** > — *Specifies only inbound packets from a single IP address. Use this option when you want to match only the IP packets from one source IP address (device).*
- ***src-ip-addr/mask-length*** — *Performs the specified action on any IP packet having a source address within the range defined by either*

  < *src-ip-addr / cidr-mask-bits* >
  or
  < *src-ip-addr < mask >>*

Use this criterion to filter packets received from either a subnet or a group of IP addresses. The mask can be in either dotted-decimal format or CIDR format with the number of significant bits. Refer to "Using CIDR Notation To Enter the ACL Mask" on page 10-45.
The mask is applied to the IP address in the ACL to define which bits in a packet's source IP address must exactly match the IP address configured in the ACL and which bits need not match. Note that specifying a group of contiguous IP addresses may require more than one ACE. For more on how masks operate in ACLs, refer to "How an ACE Uses a Mask To Screen Packets for Matches" on page 10-32.

[*operator* < *src-port tcp/udp-id* >]

In an extended ACL where you have selected either **tcp** *or* **udp** *as the packet protocol type (see above), you can optionally use a TCP or UDP source port number or range of numbers to further define the criteria for a match. To specify a TCP or UDP port number, (1) select the* **eq** *comparison operator and (2) enter the port number or a well-known port name.*

Comparison Operator:

- **eq** < ***tcp/udp-port-nbr*** > — *"Equal To"; to have a match with the ACE entry, the TCP or UDP source port number in a packet must be equal to* < ***tcp/udp-port-nbr*** >.

Port Number or Well-Known Port Name:

Use the TCP or UDP port number required by your application. The switch also accepts these well-known TCP or UDP port names as an alternative to their corresponding port numbers:

- ***TCP***: bgp, dns, ftp, http, imap4, ldap, nntp, pop2, pop3, smtp, ssl, telnet
- ***UDP***: bootpc, bootps, dns, ntp, radius, radius-old, rip, snmp, snmp-trap, tftp

To list the above names, press the **[Shift] [?]** *key combination after entering an operator. For a comprehensive listing of port numbers, visit www.iana.org/assignments/port-numbers.*

< any | host < *dest-ip-addr* > | *ip-addr/mask*-length >

In an extended ACL, this parameter defines the destination IP address (DA) that a packet must carry in order to have a match with the ACE. The options are the same as shown for < ***src-ip-addr*** >.

[< *dest-port tcp/udp-id* >]

In an extended ACL, this parameter defines the TCP or UDP destination port number a packet must carry in order to have a match with the extended ACE. The options are the same as shown above on the preceding page for the source IP address.

[log]

Optional; generates an ACL log message if:

- *The action is* **deny.** *(This option is not configurable for* **Permit.***)*
- *There is a match.*
- *ACL logging is enabled on the switch. (Refer to "Enabling ACL Logging on the Switch" on page 10-77)*

Syntax: interface < *port-list* > access-group < *list-#* | *ascii-str* > in

Assigns an ACL, designated by an ACL list number or ASCII string (alphanumeric list name), to an interface to filter inbound IP traffic on that interface. To configure named ACLs, refer to "Configuring a Named ACL" on page 10-58.

Example of an Extended ACL. Suppose that you want to implement these policies on ports 1, 2, and 3:

A. Permit Telnet traffic from 10.10.10.44 inbound on port 1 to 10.10.20.78, deny all other inbound IP traffic from network 10.10.10.0 (VLAN 10) to 10.10.20.0 (VLAN 20), and permit all other IP traffic from any source to any destination. (See "A" in figure 10-18, below.)

B. Permit FTP traffic from IP address 10.10.20.100 on port 2 to 10.10.30.55. Deny FTP traffic from other hosts on network10.10.20.0 to any destination, but permit all other traffic.
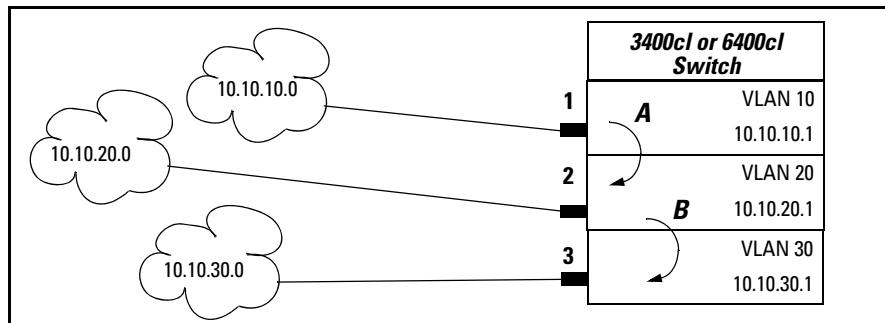
Figure 10-18. Example of an Extended ACL

```
ProCurve(config)# access-list 110 permit tcp host 10.10.10.44 host 10.10.20.78
eq telnet
ProCurve(config)# access-list 110 deny ip 10.10.10.1/24 10.10.20.1/24
ProCurve(config)# access-list 110 permit ip any any          A (Refer to figure 10-18, above.)
ProCurve(config)# interface 1 access-group 110 in
```

```
ProCurve(config)# access-list 120 permit tcp host 10.10.20.100 host 10.10.30.55
eq ftp
ProCurve(config)# access-list 120 deny tcp any any eq ftp
ProCurve(config)# access-list 120 permit ip any any          B (Refer to figure 10-18, above.)
ProCurve(config)# interface 2 access-group 120 in
```

```
ProCurve(config)# write mem ◄────────────    write memory writes the configuration
ProCurve(config)# show config                 changes to the startup-config file.

Startup configuration:
; J4905A Configuration Editor; Created on release #M.08.01
hostname "ProCurve"
ip access-list extended "110"
    permit tcp 10.10.10.44 0.0.0.0 10.10.20.78 0.0.0.0 eq 23
    deny ip 10.10.10.1 0.0.0.255 10.10.20.1 0.0.0.255
    permit ip 0.0.0.0 255.255.255.255 0.0.0.0 255.255.255.255
    exit
ip access-list extended "120"
    permit tcp 10.10.20.100 0.0.0.0 10.10.30.55 0.0.0.0 eq 21
    deny tcp 0.0.0.0 255.255.255.255 0.0.0.0 255.255.255.255 eq 21
    permit ip 0.0.0.0 255.255.255.255 0.0.0.0 255.255.255.255
    exit
interface 1
    access-group "110" in
    no lacp
exit
interface 2
    access-group "120" in            Access-List configuration in the
    no lacp                          switch's startup-config file.
exit
snmp-server community "public" Unrestricted
vlan 1
    name "DEFAULT_VLAN"
    .
    .
    .
ProCurve(config)# show access-list resources
QoS/ACL Resource Usage

 Port | Rules Available | ACL Masks Available
 -----+-----------------+--------------------
    1 | 118             |    5          ◄──    ACL 110, applied to port 1, consumes two
    2 | 118             |    5          ◄──    per-port rules and three ACL masks.
    3 | 120             |    8
    4 | 120             |    8                 ACL 120, applied to port 2, also consumes
    . |  .              |    .                 two per-port rules and three ACL masks.
    . |  .              |    .
    . |  .              |    .
```

Figure 10-19. Example of Configuration Commands for an Extended ACL

Configuring a Named ACL

You can use the "Named ACL" context to configure a standard or extended ACL with an alphanumeric name instead of a number. Note that the command structure for configuring a named ACL differs from that for a numbered ACL.

Syntax: ip access-list standard < *name-str* | 1-99 >
         < deny | permit >
         < any | host < *src-ip-addr* > | *ip-addr / mask-length* >
         [log]

         ip access-list extended < *name-str* | 100-199 >
          < deny | permit > ip
          < any | host < *src-ip-addr* > | *ip-addr / mask-length* >
          < any | host < *dest-ip-addr* > | *ip-addr / mask-length* >
          [log]

         ip access-list extended < *name-string* >
          < deny | permit > < tcp | udp >
            < any | host < *src-ip-addr* > | *ip-addr / mask-length* >
              [*oper* < *src-port tcp/udp-id* >]
            < any | host < *dest-ip-addr* > | *ip-addr / mask-length* >
                [*oper* < *dest-port tcp/udp-id* >]
          [log]

            These commands create an ACE in the named ACL list and:
            - *Indicate the action (deny or permit) to take on a packet if there is a match between a packet and the criteria in the complete ACE.*
            - *Specify the packet protocol type (IP, TCP, or UDP) and (if TCP or UDP) the comparison operator.*
            - *Specify the source and destination addressing options required for a match.*
            - *Allow optional ACL logging where a packet has a match with a* **deny** *ACE. The* **log** *option does not appear when* **permit** *is the action.*

            If the ACL does not already exist, these commands create the specified ACL and its first ACE. If the ACL already exists, these commands add a new, explicit ACE to the end of the ACL. For a match to occur, the packet must have the source and destination IP addressing criteria specified by this command, as well as any protocol-specific (TCP or UDP port number) criteria specified by the command.

< *name-str* | 1-99 | 100-199 >

Consists of an alphanumeric string of up to 64 case-sensitive characters. If you include a space in the string, you must also enclose the string with quotes. For example, **"ACL # 1"**. *You can also enter numbers in the ranges associated with standard (1-99) and extended (100-199) ACLs.*

For explanations of the individual parameters in the preceding syntax statements, refer to the syntax descriptions under "Configuring and Assigning a Numbered, Standard ACL" on page 10-47 or "Configuring and Assigning a Numbered, Extended ACL" on page 10-52.

For example, figure 10-20 shows the commands for creating an ACL in the "Named ACL" context with these parameters:

ACL Name:	150
Action:	Deny
Protocol:	TCP
Source IP Address and Mask	10.10.20.100 0.0.0.0
Destination IP Address and Mask	10.10.10.1 0.0.0.255
Protocol Operator and Port Number at Destination	eq telnet

```
ProCurve(config)# ip access-list extended 150
ProCurve(config-ext-nacl)# permit tcp host 10.10.20.100 10 10.10.1/24 eq telnet
ProCurve(config-ext-nacl)# exit
ProCurve(config)# write mem
ProCurve(config)# interface 10 access-group        Command Entry for      Command Entry for
ProCurve(config)# show config                       Source IP Address      Destination IP Address
                                                     and Mask               and Mask
Startup configuration:

; J4903A Configuration Editor; Created on release #M.08.5X

hostname "ProCurve"
ip access-list extended "150"
    permit tcp 10.10.20.100 0.0.0.0 10.10.10.1 0.0.0.255 eq 23
    exit
snmp-server community "public" Unrestricted
vlan 1                            Configured Source IP         Configured Destination IP
    name "DEFAULT_VLAN"           Address and Mask             Address and Mask
    untagged 1-24
    ip address dhcp-bootp
    exit

ProCurve(config)# show access-list resources
QoS/ACL Resource Usage

 Port | Rules Available | Masks Available
 -----+-----------------+-----------------
   1  | 120             | 15
   2  | 120             | 15
   3  | 120             | 15
   .  |   .             |   .
   .  |   .             |   .
   .  |   .             |   .
   9  | 120             | 15
  10  | 119             | 13
  11  | 120             | 15
   .  |   .             |   .
   .  |   .             |   .
   .  |   .             |   .
```

Figure 10-20. Using the "Named ACL" Context To Configure an ACL

Enabling or Disabling ACL Filtering on an Interface

You can configure one ACL to filter inbound traffic on multiple interfaces. For limits and operating rules, refer to "ACL Configuration and Operating Rules" on page 10-30.

Syntax: [no] interface < *port-list* > ip access-group < *ascii-string* > in
     where: < *ascii-string* > = either a ACL name or an ACL ID number.

> *Assigns an ACL to a physical interface, which can be any combination of ports and/or trunks that do not already have an ACL assignment. You can use either the global configuration level or the interface context level to assign an ACL to an interface or remove an ACL from an interface.*

Enabling an ACL from the Global Configuration Level →
```
ProCurve(config)# interface 10 access-group 150 in
```

Enabling an ACL from a interface Context. →
```
ProCurve(config)# interface 12
ProCurve(eth-12)# access-group 150 in
ProCurve(eth-12)# exit
```

Disabling an ACL from the Global Configuration Level →
```
ProCurve(config)# no interface 10 access-group 150 in
```

Disabling an ACL from an Interface Context. →
```
ProCurve(config)# interface 12
ProCurve(eth-12)# no access-group 150 in
```

Figure 10-21. Methods for Enabling and Disabling ACLs

Deleting an ACL from the Switch

Syntax: no ip access-list standard < *name-str* | 1-99 >

no ip access-list extended < *name-str* | 100-199 >

> *Removes the specified ACL from the switch's running-config file.*

> ***Note:*** *You cannot delete an ACL from the switch while the ACL is assigned to any interfaces. Thus, before deleting an ACL from the switch, remove all assignments of the ACL to specific interfaces.   If you need to delete an ACL assignment, refer to "Enabling or Disabling ACL Filtering on an Interface" on page 10-61.*

Displaying ACL Data

ACL Commands	Function	Page
show access-list	View a brief listing of all ACLs on the switch.	10-63
show access-list config	Display the ACL lists configured in the switch.	10-63
show access-list ports < all | < *interface* >>	List the name and type of ACLs assigned to all ports on the switch or to a particular port or static trunk configured on the switch.	10-64
show access-list < *acl-name-string* >	Display detailed content information for a specific ACL.	10-65
show access-list resources	Displays the currently available per-port rule and ACL mask resources.	10-67
show config show running	**show config** includes configured ACLs and assignments existing in the startup-config file. **show running** includes configured ACLs and assignments existing in the running-config file.	

Display an ACL Summary

This command lists the configured ACLs, regardless of whether they are assigned to any interfaces.

Syntax: show access-list

> *List a summary table of the name, type, and application status of all ACLs configured on the switch.*

For example:

```
ProCurve(config)# show access-list

Access Control Lists

  Type  Appl  Name
  ----  ----  ----------
  std   yes   1
  ext   yes   103
  ext   no    105
  std   yes   2
  std   no    144
```

In this switch, ACLs 105 and "Red VLAN Inbound" exist in the configuration but are not applied to any interfaces s and thus do not perform packet filtering.

Figure 10-22. Example of a Summary Table of Access lists

Term	Meaning
Type	Shows whether the listed ACL is **std** (Standard; source-address only) or **ext** (Extended; protocol, source, and destination data).
Appl	Shows whether the listed ACL has been applied to an interface (**yes**/**no**).
Name	Shows the name or ID number assigned to each ACL configured in the switch.

Display the Content of All ACLs on the Switch

This command lists the configuration details for every ACL configured in the running-config file, regardless of whether you have assigned any to filter traffic on switch interfaces.

Syntax: show access-list config

> *List the configured syntax for all ACLs currently configured on the switch.*

N o t e

Notice that you can use the output from this command for input to an offline text file in which you can edit, add, or delete ACL commands. Refer to "Editing ACLs and Creating an ACL Offline" on page 10-69.

This information also appears in the **show running** display. If you executed **write memory** after configuring an ACL, it appears in the **show config** display.

For example, with two ACLs configured in the switch, you will see results similar to the following:

```
ProCurve(config)# show access-list config
ip access-list standard "1"
   deny 18.28.236.77 0.0.0.0
   deny 18.29.140.107 0.0.0.0
   permit 0.0.0.0 255.255.255.255
   exit
ip access-list extended "105"
   permit tcp 18.30.133.27 0.0.0.0 0.0.0.0 255.255.255.255
   permit tcp 18.30.155.101 0.0.0.0 0.0.0.0 255.255.255.255
   deny ip 18.30.133.1 0.0.0.255 0.0.0.0 255.255.255.255
   deny ip 18.30.155.1 0.0.0.255 0.0.0.0 255.255.255.255
   permit ip 0.0.0.0 255.255.255.255 0.0.0.0 255.255.255.255
   exit
```

Figure 10-23. Example of an ACL Configured Syntax Listing

Display the ACL Assignments for an Interface

This command briefly lists the identification and type(s) of ACLs currently assigned to a particular interface (one or more ports and/or trunks) in the running-config file. (The switch allows up to one, inbound ACL assignment per interface.)

Syntax: show access-list ports < *interface* >

List the ACLs assigned to interfaces in the running config file.

N o t e

This information also appears in the **show running** display. If you executed **write memory** after configuring an ACL, it appears in the **show config** display.

For example, if you assigned a standard ACL with an ACL-ID of "1" to filter inbound traffic on port 10, you could quickly verify this assignment as follows:

```
ProCurve(config)# show access-list ports 7

Access Lists for Port 7

  Inbound  : 2
  Type     : Standard

ProCurve(config)# show access-list ports all

Access Lists for Port 3

  Inbound  : 1
  Type     : Standard


Access Lists for Port 7

  Inbound  : 2
  Type     : Standard


Access Lists for Port Trk1

  Inbound  : 2
  Type     : Standard
```

Indicates that a standard ACL with the ID of "2" is assigned to filter inbound traffic on port 7.

Indicates that a standard ACL with an ID of "1" is assigned to filter inbound traffic on port 3, and that another standard ACL with an ID of "2" is assigned to filter inbound traffic on port 7 and Trk1 (trunk 1).

Figure 10-24. Example of Listing the ACL Assignment for an Interface

Displaying the Content of a Specific ACL

This command displays a specific ACL configured in the running config file in an easy-to-read tabular format.

Note   This information also appears in the **show running** display. If you executed **write memory** after configuring an ACL, it appears in the **show config** display.

Syntax: show access-list < *acl-name-string* >

Display detailed information on the content of a specific ACL configured in the running-config file.

For example, suppose you configured the following two ACLs in the switch:

ACL ID	ACL Type	Desired Action
1	Standard	• Deny IP traffic from 18.28.236.77 and 18.29.140.107. • Permit IP traffic from all other sources.
105	Extended	• Permit any TCP traffic from 18.30.133.27 to any destination. • Deny any other IP traffic from 18.30.133.(1-255). • Permit all other IP traffic from any source to any destination.

Inspect the ACLs as follows:

```
ProCurve(config)# show access-list 1
Access Control Lists
   Name: 1                        Indicates whether the ACL
   Type: Standard                 is assigned to an interface.
   Applied: Yes

 ID   action        IP              Mask            Log
------------------------------------------------------------
 1     deny   std  10.28.236.77    0.0.0.0
 2     deny   std  10.29.140.107   0.0.0.0
 3     permit std  0.0.0.0         255.255.255.255
```

Listing for a Standard ACL

Listing for an Extended ACL

```
ProCurve(config)# show access-list 105
Access Control Lists

   Name: 105              Indicates whether the ACL is assigned to an interface.
   Type: Extended
   Applied: No            Indicates source and destination entries in the ACL.

 ID   action     IP              Mask            proto  oper  port(s)   Log
----------------------------------------------------------------------------
 1   permit  src: 10.30.133.27  0.0.0.0          TCP    none    0
             dst: 0.0.0.0       255.255.255.255  TCP    eq     23
 2   deny    src: 10.30.133.1   0.0.0.255        IP
             dst: 0.0.0.0       255.255.255.255  IP                     log
 3   permit  src: 0.0.0.0       255.255.255.255  IP
             dst: 0.0.0.0       255.255.255.255  IP
```

Indicates that the source TCP port can be any value.

Figure 10-25. Examples of Listings Showing the Content of Standard and Extended ACLs

Table 10-9.   Descriptions of Data Types Included in Show Access-List < _interface_ > Output

Field	Description
Name	The ACL identifier. Can be a number from 1 to 199, or a name.
Type	Standard or Extended. The former uses only source IP addressing. The latter uses both source and destination IP addressing and also allows TCP or UDP port specifiers.
Applied	"Yes" means the ACL has been applied to an interface. "No" means the ACL exists in the switch configuration, but has not been applied to any interfaces, and is therefore not in use.
ID	The sequential number of the Access Control Entry (ACE) in the specified ACL.
action	Permit (forward) or deny (drop) a packet when it is compared to the criteria in the applicable ACE and found to match.
IP	**In Standard ACLs:** The source IP address to which the configured mask is applied to determine whether there is a match with a packet. **In Extended ACLs:** The source and destination IP addresses to which the corresponding configured masks are applied to determine whether there is a match with a packet.
Mask	The mask configured in an ACE and applied to the corresponding IP address in the ACE to determine whether a packet matches the filtering criteria.
proto	Used only in extended ACLs to specify the packet protocol type to filter. Must be either IP, TCP, or UDP.
oper	Used only in extended ACLs where a TCP or UDP port type and number have been entered. Specifies how to compare the corresponding TCP or UDP port number in a packet to the port number in the ACE.
port(s)	Used only in extended ACLs to show any TCP or UDP port number that has been entered in the ACE.
Log	Shows the status of logging for the entry (ACE). A blank space indicates ACL logging is not enabled for that ACE.

Displaying the Current Per-Port ACL Resources

Assigning an ACL to one or more interfaces reduces the available per-port rule and mask resources for those interfaces. (An unassigned ACL does not affect the rule and mask count.) This command displays the current per-port rule and mask resources available on the switch. For more information on rule and mask usage, refer to "Planning an ACL Application on a Series 3400cl or Series 6400cl Switch" on page 10-17.

Syntax: show access-list resources

> *Displays the currently available per-port rules and ACL masks on the switch. Note that the available rules can be used by ACL assignments, QoS configurations, Rate-Limiting configurations, and enabling IGMP. For more information, refer to "ACL Resource Usage and Monitoring" on page 10-18.*

```
ProCurve(config)# show access-list resources
QoS/ACL Resource Usage

 Port | Rules Available | ACL Masks Available
 -----+-----------------+-----------------
   1  | 119             |        6
   2  | 120             |        8
   3  | 120             |        8
   4  | 120             |        8
   5  | 120             |        8
   6  | 120             |        8
   7  | 120             |        8
   8  | 120             |        8
   9  | 120             |        8
  10  | 120             |        8
  11  | 120             |        8
  12  | 120             |        8
  13  | 120             |        8
  14  | 120             |        8
  15  | 120             |        8
  16  | 120             |        8
  17  | 120             |        8
  18  | 120             |        8
  19  | 120             |        8
  20  | 120             |        8
  21  | 120             |        8
  22  | 120             |        8
  23  | 120             |        8
  24  | 120             |        8

Maximum Rules per-port : 120
Maximum Masks per-port : 8
```

Indicates that one rule and two masks have been used. All other ports show the default quantity of rules and masks, which means that there are no ACLs or QoS assigned to these other ports on the switch.

Note: Because ACLs and QoS use the same rule resources in the switch, **show access-list resources** and **show qos resources** both list the same resource table. This table indicates the combined resource use of both features (plus Rate-Limiting and IGMP (if configured). Refer to page 10-18.)

Figure 10-26. Example of a Show Access-List Resources Command Output

Display All ACLs and Their Assignments in the Switch Startup-Config File and Running-Config File

The **show config** and **show running** commands include in their listings any configured ACLs and any ACL assignments to interfaces. Refer to figure 10-16 (page 10-50) and figure 10-17 (page 10-51) for examples. Remember that **show config** lists the startup-config file and **show running** lists the running-config file.

Editing ACLs and Creating an ACL Offline

Earlier sections of this chapter describe how to use the CLI to create an ACL. Beginning with "Using the CLI To Edit ACLs", below, describes how to use the CLI to edit existing ACLs. However, you can also create or edit an ACL offline, then use a TFTP server to upload the ACL as a command file. The offline method (page 10-72) provides a useful alternative to using the CLI for creating or editing large ACLs.

Using the CLI To Edit ACLs

The switch applies individual ACEs in the order in which they occur in an ACL. You can use the CLI to delete individual ACEs from anywhere in an ACL and to append new ACEs to the end of an ACL. However, the CLI method does not allow you to insert a new ACE between two existing ACEs.

Note

Before editing an assigned ACL, you must use the **no interface < *interface* > access-group < *acl-#* > in** command to remove the ACL from all interfaces to which it is assigned.

Using the CLI To Edit a Short ACL. To insert a new ACE between existing ACEs in a short ACL, you may want to delete the ACL and then re-configure it by entering your updated list of ACEs in the correct order.

Using the CLI to Edit a Longer ACL. To insert a new ACE between existing ACEs in a longer ACL:

   a.   Delete the first ACE that is out of sequence and all following ACEs through the end of the ACL.

   b.   Re-Enter the desired ACEs in the correct sequence.

General Editing Rules

■ You can delete any ACE from an ACL by repeating the ACE's entry command, preceded by the "no" statement. When you enter a new ACE, the switch inserts it as the last entry of the specified ACL.

■ Deleting the last ACE from a *numeric* ACL, removes the ACL from the configuration. Deleting the last ACE from a *named* ACL leaves the ACL in memory. In this case, the ACL is "empty" and cannot perform any filtering tasks. (In any ACL the implicit "deny any" does not apply unless the ACL includes at least one explicit ACE.)

■ When you create a new ACL, the switch inserts it as the last ACL in the startup-config file. (Executing **write memory** saves the running-config file to the startup-config file.)

Deleting Any ACE from an ACL

You can delete an ACE from an ACL by repeating the ACE's entry command, preceded by the "**no**" statement.

Syntax: no access-list < *interface* > < permit | deny > < any | host | *ip-addr/mask-length* >

Deletes an ACE from a standard ACL. All variable parameters in the command must be an exact match with their counterparts in the ACE you want to delete.

no access-list < *interface* > < permit | deny > < ip | tcp | udp >
 < src-addr: any | host | *ip-addr/mask-length* > [operator < src-port-num >]
 < dest-addr: any | host | ip-addr-mask-length > [operator < dest-port-num
>
 [log]

Deletes an ACE from a standard ACL. All variable parameters in the command must be an exact match with their counterparts in the ACE you want to delete.

For example, the first of the following two commands creates an ACE in ACL 22 and the second deletes the same ACE:

```
                                                             Creates an ACE in ACL 22.

ProCurve(config)# access-list 22 permit host 18.28.152.64
ProCurve(config)# no access-list 22 permit host 18.28.152.64
                                                             Removes the same ACE
                                                             from ACL 22, regardless of
                                                             the ACE's position in the
                                                             ACL.
```

Figure 10-27. Example of Deleting an ACE from a Standard ACL

Figure 10-28 shows an example of deleting an ACE from an extended ACL.

```
ProCurve(config)# show config          ACL 103 Before Removing
Startup configuration:                 the Second "deny" ACE.
  .
  .
  .
ip access-list extended "103"
    deny tcp 0.0.0.0 255.255.255.255 10.10.10.2 0.0.0.0 eq 23 log      Use no access-list
    deny tcp 0.0.0.0 255.255.255.255 10.10.20.2 0.0.0.0 eq 23 log      to remove this line
    permit ip 0.0.0.0 255.255.255.255 0.0.0.0 255.255.255.255          from ACL 103.
    exit
vlan 1
    name "DEFAULT_VLAN"
    untagged A1
    .
    .
    .
ProCurve(config)# no access-list 103 deny tcp any host 10.10.20.2 eq 23 log
ProCurve(config)# write mem
ProCurve(config)# show config
Startup configuration:                 ACL 103 After Removing
  .                                     the Second "deny" ACE.
  .
ip access-list extended "103"
    deny tcp 0.0.0.0 255.255.255.255 10.10.10.2 0.0.0.0 eq 23 log
    permit ip 0.0.0.0 255.255.255.255 0.0.0.0 255.255.255.255
    exit
vlan 1
    name "DEFAULT_VLAN"
    untagged 1
```

Figure 10-28. Example of Deleting an ACE from an ACL

Working Offline To Create or Edit an ACL

Note

When creating an ACL offline, ensure that the interfaces to which you plan to assign the ACL will have adequate per-port rules and ACL masks available. Note that if you attempt to apply an ACL to multiple interfaces and one of those interfaces does not have sufficient resources to support the ACL, the command will fail for all specified interfaces. For more on per-port ACL resources, refer to "Planning an ACL Application on a Series 3400cl or Series 6400cl Switch" on page 10-17.

For longer ACLs that would be difficult or time-consuming to accurately create or edit in the CLI, you can use the offline method:

1. Begin by doing one of the following:
   - To edit one or more existing ACLs, use **copy command-output tftp** to copy the current version of the ACL configuration to a file in your TFTP server. For example, to copy the ACL configuration to a file named **acl02.txt** in the TFTP directory on a server at 10.28.227.2:

     ```
     ProCurve# copy command-output 'show access-list
     config' tftp 10.28.227.2 acl02.txt pc
     ```
   - To create a new ACL, just open a text file in the appropriate directory on a TFTP server accessible to the switch.

2. Use the text editor to create or edit the ACL(s).

3. Use **copy tftp command-file** to download the file as a list of commands to the switch.

Creating an ACL Offline

Use a text editor that allows you to create an ASCII text file (.txt).

If you are replacing an ACL on the switch with a new ACL that uses the same number or name syntax, begin the command file with a "no" command to remove the earlier version of the ACL from the switch's running-config file. Otherwise, the switch will append the new ACEs in the ACL you download to the existing ACL. For example, if you plan to use the Copy command to *replace* ACL "103", you would place this command at the beginning of the edited file:

```
no ip access-list extended 103
```

```
no ip access-list extended 103

ip access-list extended "103"

    deny tcp 0.0.0.0 255.255.255.255 10.10.10.2 0.0.0.0 eq 23 log

    permit ip 0.0.0.0 255.255.255.255 0.0.0.0 255.255.255.255

    exit
```

Removes an existing ACL and replaces it with a new version with the same identity. To append new ACEs to the ACL instead of replacing it, you would omit the first line.

Figure 10-29. Example of an Offline ACL File Designed To Replace An Existing ACL

For example, suppose that you wanted to create an extended ACL to fulfill the following requirements (Assume a subnet mask of 255.255.255.0.):

■ ID: 160

■ Deny Telnet access to a server at 10.10.10.100 from these three IP addresses on port 2 (with ACL logging):

- 10.10.20.17
- 10.10.20.23
- 10.10.20.40

■ Allow any inbound access from all other addresses on port 2:

■ Permit internet access to the following two IP addresses through port 24, but deny access to all other addresses through this port (without ACL logging).

- 10.10.20.98
- 10.10.20.21

■ Deny all traffic from port 3 to the server at 10.10.10.100 (without ACL logging).

■ Deny all traffic from port 5 to the server at 10.10.10.100 (without ACL logging), but allow any other traffic from port 5.

1. To create an ACL offline for the above requirements, you would create a **.txt** file with the content shown in figure 10-30.
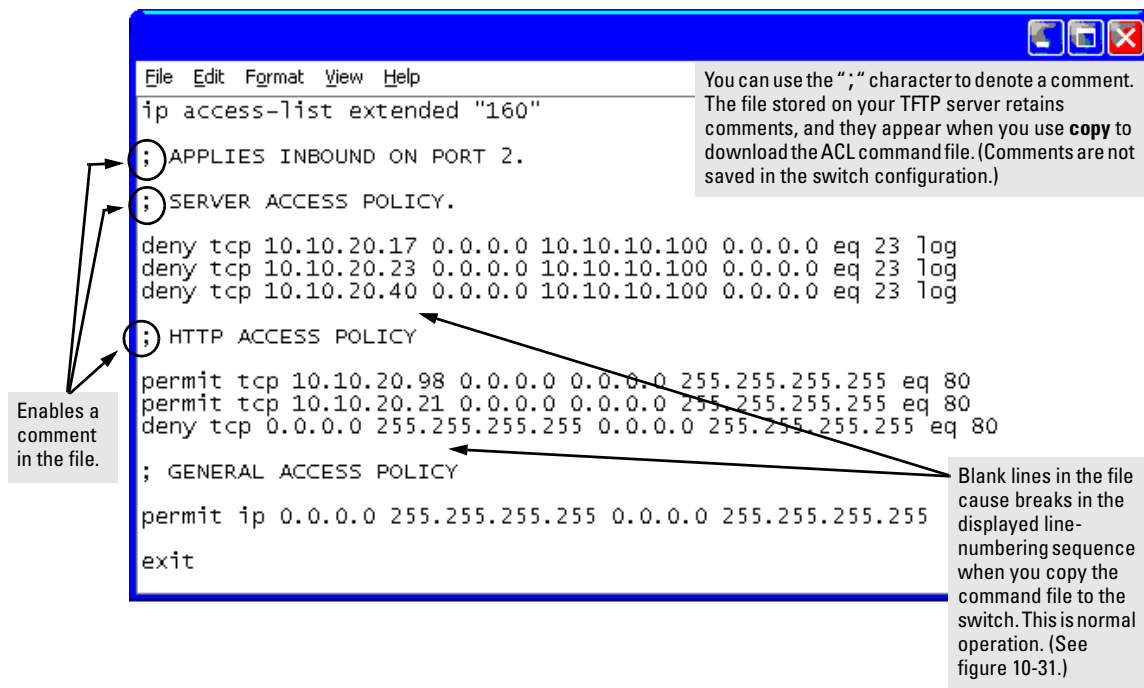
Figure 10-30. Example of a.txt File Designed for Creating an ACL

2. After you copy the above .txt file to a TFTP server the switch can access, you would then execute the following command to download the file to the switch's startup-config file:

```
ProCurve(config)# copy tftp command-file 13.28.234.180 list-160.txt
Running configuration may change, do you want to continue [y/n]?  y
   1. ip access-list extended "160"
   3. ; APPLIES INBOUND ON PORT 2.
   5. ; SERVER ACCESS POLICY.
   7. deny tcp 10.10.20.17 0.0.0.0 10.10.10.100 0.0.0.0 eq 23 log
   8. deny tcp 10.10.20.23 0.0.0.0 10.10.10.100 0.0.0.0 eq 23 log
   9. deny tcp 10.10.20.40 0.0.0.0 10.10.10.100 0.0.0.0 eq 23 log
  11. ; HTTP ACCESS POLICY
  13. permit tcp 10.10.20.98 0.0.0.0 0.0.0.0 255.255.255.255 eq 80
  14. permit tcp 10.10.20.21 0.0.0.0 0.0.0.0 255.255.255.255 eq 80
  15. deny tcp 0.0.0.0 255.255.255.255 0.0.0.0 255.255.255.255 eq 80
  17. ; GENERAL ACCESS POLICY
  19. permit ip 0.0.0.0 255.255.255.255 0.0.0.0 255.255.255.255
  21. exit
```

Figure 10-31. Example of Using "copy tftp command-file" To Configure an ACL in the Switch

Note

If a transport error occurs, the switch does not execute the command and the ACL is not configured.

3. Next, assign the new ACL to the intended interface which, in this example, is for port 2.

```
ProCurve(config)# interface 2 access-group 160 in
```

4. Inspect the effect of the ACL on the switch's per-port resources.

```
ProCurve(config)# show access-list resources
QoS/ACL Resource Usage

 Port | Rules Available | ACL Masks Available
 -----+-----------------+-----------------
   1  | 120             | 7
   2  | 114             | 3
   3  | 120             | 7
   4  | 120             | 7
   5  | 120             | 7
   .  |  .              |  .
   .  |  .              |  .
   .  |  .              |  .
```

ACL 160 used six per-port rules and 5 ACL masks on port 2. This means that ACL 160 could be replaced with a larger ACL that uses up to three more masks. The switch reserves eight masks per-port for ACL and IGMP use. (When enabled in a VLAN, IGMP uses one mask per-port on all ports on the switch.)

Figure 10-32. Inspection of Per-Port Resource Usage After Assigning an ACL

5. Inspect the new running configuration:

```
ProCurve(config)# show running
```

6. If the configuration appears satisfactory, save it to the startup-config file:

```
ProCurve(config)# write memory
```

Enable ACL "Deny" Logging

ACL logging enables the switch to generate a message when IP traffic meets the criteria for a match with an ACE that results in an explicit "deny" action. You can use ACL logging to help:

■ Test your network to ensure that your ACL configuration is detecting and denying the traffic you do not want forwarded

■ Receive notification when the switch detects attempts to transmit traffic you have designed your ACLs to reject

The switch sends ACL messages to Syslog and optionally to the current console, Telnet, or SSH session. You can configure up to six Syslog server destinations.

Requirements for Using ACL Logging

■ The switch configuration must include an ACL (1) assigned to an interface and (2) containing an ACE configured with the **deny** action and the **log** option.

■ To screen routed packets with destination IP addresses outside of the switch, IP routing must be enabled.

■ For ACL logging to a Syslog server, the server must be accessible to the switch and identified (with the **logging < *ip-addr* >** command) in the switch configuration.

■ Debug must be enabled for ACLs and one or both of the following:
   • logging (for sending messages to Syslog)
   • Session (for sending messages to the current console interface)

ACL Logging Operation

When the switch detects a packet match with an ACE and the ACE includes both the **deny** action and the optional **log** parameter, an ACL log message is sent to the designated debug destination. The first time a packet matches an ACE with **deny** and **log** configured, the message is sent immediately to the destination and the switch starts a wait-period of approximately five minutes. (The exact duration of the period depends on how the packets are internally routed.) At the end of the collection period, the switch sends a single-line summary of any additional "deny" matches for that ACE (and any other "deny" ACEs for which the switch detected a match). If no further log messages are generated in the wait-period, the switch suspends the timer and resets itself to send a message as soon as a new "deny" match occurs. The data in the message includes the information illustrated in figure 10-33.

Note: To fit this illustration on the page, the portion of the message generated by the Syslog server itself is shown in the line above the portion of the message generated by the switch. Depending on the terminal emulator you use, you may see information similar to this, which includes the date and time the log message was received, the IP address of the default VLAN on the sending switch, and the message type (ACL).

Indicates the destination IP address of the denied packet.

```
Oct 15 10:04:45 10.28.234.181 ACL:
ACL 10/15/04 10:04:45: denied ip src 10.28.234.183 (PORT 10), 1 packets
```

Begins the actual message generated by the ACL itself, and indicates message type (ACL), date and time of generation.

ACL action (denied), protocol type (IP, TCP, or UDP), source IP address of the denied packet(s) and, if applicable, the TCP or UDP port number at the source device. The port number is "0" if the protocol is "IP" or you did not specify a port number.

Indicates the port on which the ACL is assigned.

Indicates the number of packets (deny instances) detected.

Figure 10-33. Example of the Content of an ACL-Generated Message

Enabling ACL Logging on the Switch

1. Use the debug command to:
   a. Configure one or more log destinations.
   b. If you are using a Syslog server, use the **logging** command to configure the server's IP address. (You can configure up to six Syslog servers.)
   c. Ensure that the switch can access any Syslog servers you specify.

2. Configure one or more ACLs with the deny action and the log option.

For example, suppose that you want to do the following:

■ On port 10, configure an extended ACL with an ACL-ID of 143 to deny Telnet traffic from IP address 10.38.100.127.

■ Configure the switch to send an ACL log message to the console and to a Syslog server at IP address 10.38.110.54 on port 11 if the switch detects a match denying Telnet access from 10.38.100.127.
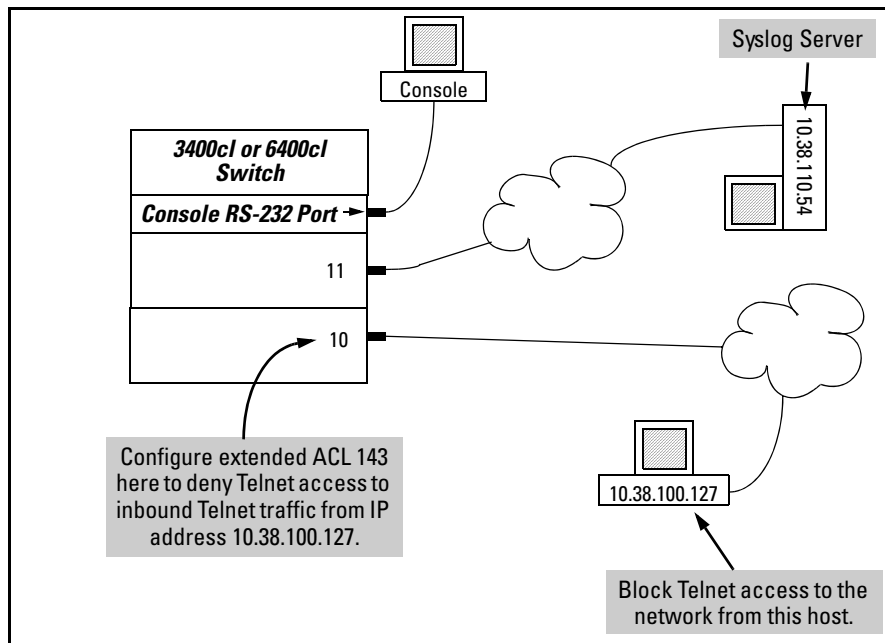
Figure 10-34. Example of an ACL Log Application

```
ProCurve(config)# access-list 143 deny tcp host 10.38.100.127 any eq telnet log
ProCurve(config)# access-list 143 permit ip any any
ProCurve(config)# interface 10 access-group 143 in
ProCurve(config)# logging 10.38.110.54
ProCurve(config)# debug acl
ProCurve(config)# debug destination logging
ProCurve(config)# debug destination session
ProCurve(config)# write memory
ProCurve(config)# show debug
Debug Logging
 Destination:
  Logging
    10.38.110.54
  Session
 Enabled debug types:
  event
  acl log
```

Figure 10-35. Commands for Applying an ACL with Logging to Figure 10-34

Operating Notes for ACL Logging

■   The ACL logging feature generates a message only when packets are explicitly denied as the result of a match, and not when explicitly permitted or implicitly denied. To help test ACL logging, configure an ACL with an explicit **deny any** and **log** statements at the end of the list, and apply the ACL to an appropriate interface.

■   Logging enables you to selectively test specific devices or groups. However, excessive logging can affect switch performance. For this reason, HP recommends that you remove the logging option from ACEs for which you do not have a present need. Also, avoid configuring logging where it does not serve an immediate purpose. (Note that ACL logging is not designed to function as an accounting method.) See also "Apparent Failure To Log All "Deny" Matches" in the section titled "ACL Problems", found in appendix C, "Troubleshooting" of the Management and Configuration Guide for your switch.

■   When configuring logging, you can reduce excessive use by configuring the appropriate ACEs to match with specific hosts instead of entire subnets.

General ACL Operating Notes

ACLs do not provide DNS hostname support.

Protocol Support: ACL criteria includes IP, TCP, and UDP. ACLs do not use these protocols:

■ TOS (Type-of-Service)

■ Precedence

■ MAC information

■ QoS

ACLs do not affect switch serial port access.

ACLs filter both Layer 2 and Layer 3 on a port.

There is no performance degradation with ACLs enabled; traffic is at line rate.

When the ACL configuration includes TCP or UDP options, the switch operates in "strict" TCP and UDP mode for increased control. The switch compares all TCP and UDP packets against the ACLs. (In the ProCurve Series 9300 Routing Switches, the Strict TCP and Strict UDP modes are optional and must be specifically invoked.)

Replacing or Adding To an Active ACL Policy.   If you assign an ACL to an interface and subsequently want to add or replace ACEs in that ACL, you must first remove the ACL from all assigned interfaces.

Note

When an ACE becomes active, it screens the packets resulting from new traffic connections. It does not screen packets resulting from currently open traffic connections. If you invoke a new ACE to screen packets in a currently open traffic connection, you must force the connection to close before the ACE can begin screening packets from that source.

ACLs Do Not Filter Traffic Generated by the Switch. Because ACLs on the 3400cl/6400cl switches filter only inbound traffic at the inbound physical port, outbound traffic from any source is not filtered by any ACL(s) configured on the switch. Filtering of such traffic must be done at a downstream device.

< *acl-list-#* >: `Unable to apply access control list.`

The indicated ACL cannot be applied to an interface because an ACL is already assigned to the interface. The command fails for all included interfaces, including any that do not already have an ACL assigned.

`Duplicate access control entry.`

The switch detects an attempt to create a duplicate ACE in the same ACL.

—This page is intentionally unused—

11

IP Routing Features

Contents

Overview of IP Routing

The Procurve Series 5300xl, 4200vl, 3400cl, and 6400cl switches offer the following IP routing features, as noted:

- **IP Static Routes** – up to 256 static routes for 5300xl, 3400/6400cl switches. Up to 16 static routes for 4200vl switches.
- **RIP** (Router Information Protocol) – supports RIP Version 1, Version 1 compatible with Version 2 (default), and Version 2 (5300xl, 3400cl, 6400cl)
- **OSPF** (Open Shortest Path First) – the standard routing protocol for handling larger routed networks (5300xl, 3400cl, 6400cl)
- **IRDP** (ICMP Router Discovery Protocol) – advertises the IP addresses of the routing interfaces on this switch to directly attached host systems
- **DHCP Relay** – allows you to extend the service range of your DHCP server beyond its single local network segment
- **Downstream Host Support** – varies for different switch models. The switches covered by this guide, when configured for IP routing, support the number of hosts indicated below:

Switch Model	Downstream Hosts	Network Subnet Addresses
5300xl Chassis (Total)	192,000	n/a
5300xl Chassis (Per Module)	128,000	n/a
4200vl Chassis (Total)	192,000	n/a
4200vl Chassis (Per Module)	128.000	n/a
3400cl and 6400cl Stackables	8,000	1,000

Throughout this chapter, the 5300xl, 4200vl, 3400cl, and 6400cl switches are referred to as "routing switches". When IP routing is enabled on your switch, it behaves just like any other IP router.

Basic IP routing configuration consists of adding IP addresses, enabling IP routing, and, enabling a route exchange protocol, such as Routing Information Protocol (RIP).

For configuring the IP addresses, refer to the chapter titled "Configuring IP Addresses" in the *Management and Configuration Guide* for your switch. The rest of this chapter describes IP routing and how to configure it in more

detail. Use the information in this chapter if you need to change some of the IP parameters from their default values or you want to view configuration information or statistics.

IP Interfaces

On the routing switches, IP addresses are associated with individual VLANs. By default, there is a single VLAN (Default_VLAN) on the routing switch. In that configuration, a single IP address serves as the management access address for the entire device. If routing is enabled on the routing switch, the IP address on the single VLAN also acts as the routing interface.

Each IP address on a routing switch must be in a different sub-net. You can have only one VLAN interface that is in a given sub-net. For example, you can configure IP addresses 192.168.1.1/24 and 192.168.2.1/24 on the same routing switch, but you cannot configure 192.168.1.1/24 and 192.168.1.2/24 on the same routing switch.

You can configure multiple IP addresses on the same VLAN.

The number of IP addresses you can configure on an individual VLAN interface is 8.

You can use any of the IP addresses you configure on the routing switch for Telnet, Web management, or SNMP access, as well as for routing.

N o t e   All ProCurve devices support configuration and display of IP address in classical sub-net format (example: 192.168.1.1 255.255.255.0) and Classless Interdomain Routing (CIDR) format (example: 192.168.1.1/24). You can use either format when configuring IP address information. IP addresses are displayed in classical sub-net format only.

IP Tables and Caches

The following sections describe the IP tables and caches:

- ARP cache table
- IP route table
- IP forwarding cache

The software enables you to display these tables.

ARP Cache Table

The ARP cache contains entries that map IP addresses to MAC addresses. Generally, the entries are for devices that are directly attached to the routing switch.

An exception is an ARP entry for an interface-based static IP route that goes to a destination that is one or more router hops away. For this type of entry, the MAC address is either the destination device's MAC address or the MAC address of the router interface that answered an ARP request on behalf of the device, using proxy ARP.

ARP Cache. The ARP cache contains dynamic (learned) entries. The software places a dynamic entry in the ARP cache when the routing switch learns a device's MAC address from an ARP request or ARP reply from the device.

The software can learn an entry when the switch or routing switch receives an ARP request from another IP forwarding device or an ARP reply. Here is an example of a dynamic entry:

```
     IP Address          MAC Address         Type         Port
1    207.95.6.102        0800.5afc.ea21      Dynamic      6
```

Each entry contains the destination device's IP address and MAC address.

To configure other ARP parameters, see "Configuring ARP Parameters" on page 11-14.

IP Route Table

The IP route table contains routing paths to IP destinations.

N o t e
The default gateway, which you specify when you configure the basic IP information on the switch, is used only when routing is not enabled on the switch.

The IP route table can receive the routing paths from the following sources:

■   A directly-connected destination, which means there are no router hops to the destination

■   A static IP route, which is a user-configured route

■   A route learned through RIP

■   A route learned through OSPF

The IP route table contains the best path to a destination.

■ When the software receives paths from more than one of the sources listed above, the software compares the administrative distance of each path and selects the path with the lowest administrative distance. The administrative distance is a protocol-independent value from 1 – 255.

The IP route table is displayed by entering the CLI command **show ip route** from any context level in the console CLI. Here is an example of an entry in the IP route table:

```
Destination    Network Mask     Gateway    Type        Sub-Type    Metric
1.1.0.0        255.255.0.0      99.1.1.2   connected               1
```

Each IP route table entry contains the destination's IP address and sub-net mask and the IP address of the next-hop router interface to the destination. Each entry also indicates route type, and for OSPF routes, the sub type, and the route's IP metric (cost). The type indicates how the IP route table received the route.

To configure a static IP route, see "Configuring a Static IP Route" on page 11-22

IP Forwarding Cache

The IP forwarding cache provides a fast-path mechanism for forwarding IP packets. The cache contains entries for IP destinations. When a ProCurve routing switch has completed processing and addressing for a packet and is ready to forward the packet, the device checks the IP forwarding cache for an entry to the packet's destination.

■ If the cache contains an entry with the destination IP address, the device uses the information in the entry to forward the packet out the ports listed in the entry. The destination IP address is the address of the packet's final destination. The port numbers are the ports through which the destination can be reached.

■ If the cache does not contain an entry, the software can create an entry in the forwarding cache.

■ The aging interval depends on the number of entries in the cache, ranging from 12 seconds (full table) to 36 seconds (empty table). Entries are only aged if they are not being utilized by traffic. If you have an entry that is always being utilized, it will never age. If there is no traffic, it will age in 12 to 36 seconds. The age timer is not configurable.

N o t e   You cannot add static entries to the IP forwarding cache.

IP Route Exchange Protocols

This feature is not available on the Series 4200vl switches.

The switch supports the following IP route exchange protocols:

■ Routing Information Protocol (RIP)

■ Open Shortest Path First (OSPF)

These protocols provide routes to the IP route table. You can use one or more of these protocols, in any combination. The protocols are disabled by default. For configuration information, see the following:

■ "Configuring RIP" on page 11-24

■ "Configuring OSPF" on page 11-37

IP Global Parameters for Routing Switches

The following table lists the IP global parameters and the page where you can find more information about each parameter.

Table 11-1.   IP Global Parameters for Routing Switches

Parameter	Description	Default	See page
Router ID	The value that routers use to identify themselves to other routers when exchanging route information. OSPF uses the router ID to identify routers. RIP does not use the router ID.	The lowest-numbered IP address configured on the lowest-numbered routing interface.	11-13
Address Resolution Protocol (ARP)	A standard IP mechanism that routers use to learn the Media Access Control (MAC) address of a device on the network. The router sends the IP address of a device in the ARP request and receives the device's MAC address in an ARP reply.	Enabled	11-14
ARP age	The amount of time the device keeps a MAC address learned through ARP in the device's ARP cache. The device resets the timer to zero each time the ARP entry is refreshed and removes the entry if the timer reaches the ARP age.	Five minutes	not configurable
Proxy ARP	An IP mechanism a router can use to answer an ARP request on behalf of a host, by replying with the router's own MAC address instead of the host's.	Disabled	11-16

Parameter	Description	Default	See page
Time to Live (TTL)	The maximum number of routers (hops) through which a packet can pass before being discarded. Each router decreases a packet's TTL by 1 before forwarding the packet. If decreasing the TTL causes the TTL to be 0, the router drops the packet instead of forwarding it.	64 hops	Refer to the chapter titled "Configuring IP Addressing" in the *Management and Configuration Guide*.
Directed broadcast forwarding	A directed broadcast is a packet containing all ones (or in some cases, all zeros) in the host portion of the destination IP address. When a router forwards such a broadcast, it sends a copy of the packet out each of its enabled IP interfaces. **Note**: You also can enable or disable this parameter on an individual interface basis. See table 11-2 on page 11-12.	Disabled	11-17
ICMP Router Discovery Protocol (IRDP)	An IP protocol that a router can use to advertise the IP addresses of its router interfaces to directly attached hosts. You can enable or disable the protocol at the Global CLI Config level. You also can enable or disable IRDP and configure the following protocol parameters on an individual VLAN interface basis at the VLAN Interface CLI Config level. • Forwarding method (broadcast or multicast) • Hold time • Maximum advertisement interval • Minimum advertisement interval • Router preference level	Disabled	11-76 11-77
Static route	An IP route you place in the IP route table.	No entries	11-20
Default network route	The router uses the default network route if the IP route table does not contain a route to the destination. Enter an explicit default route (0.0.0.0 0.0.0.0 or 0.0.0.0/0) as a static route in the IP route table.	None configured	11-23

IP Interface Parameters for Routing Switches

Table 11-2 lists the interface-level IP parameters for routing switches.

Table 11-2.   IP Interface Parameters – Routing Switches

Parameter	Description	Default	See page
IP address	A Layer 3 network interface address; separate IP addresses on individual VLAN interfaces.	None configured	chapter 7
Metric	A numeric cost the router adds to RIP routes learned on the interface. This parameter applies only to RIP routes.	1 (one)	11-26
ICMP Router Discovery Protocol (IRDP)	Locally overrides the global IRDP settings. See table 11-1 on page 11-10 for global IRDP information.	Disabled	11-77
IP helper address	The IP address of a UDP application server (such as a BootP or DHCP server) or a directed broadcast address. IP helper addresses allow the routing switch to forward requests for certain UDP applications from a client on one sub-net to a server on another subnet.	None configured	11-95

Configuring IP Parameters for Routing Switches

The following sections describe how to configure IP parameters. Some parameters can be configured globally while others can be configured on individual VLAN interfaces. Some parameters can be configured globally and overridden for individual VLAN interfaces.

N o t e   This section describes how to configure IP parameters for routing switches. For IP configuration information when routing is not enabled, see chapter 7, "Configuring IP Addressing".

Configuring IP Addresses

You can configure IP addresses on the routing switch's VLAN interfaces. Configuring IP addresses is described in detail in the chapter titled "Configuring IP Addressing" in the *Management and Configuration Guide* for your switch.

Changing the Router ID

In most configurations, a routing switch has multiple IP addresses, usually configured on different VLAN interfaces. As a result, a routing switch's identity to other devices varies depending on the interface to which the other device is attached. Some routing protocols, including Open Shortest Path First (OSPF), identify a routing switch by just one of the IP addresses configured on the routing switch, regardless of the interfaces that connect the routing switches. This IP address is the router ID.

N o t e   Routing Information Protocol (RIP) does not use the router ID.

By default, the router ID on a ProCurve routing switch is the lowest numbered IP interface configured on the device.

If you prefer, you can explicitly set the router ID to any valid IP address. The IP address cannot be in use on another device in the network.

N o t e   To display the router ID, enter the **show ip ospf** CLI command at any Manager EXEC CLI level.

To change the router ID, enter a command such as the following:

```
ProCurve(config)# ip router-id 209.157.22.26
```

Syntax: Syntax: ip router-id < *ip-addr* >

The < *ip-addr* > can be any valid, unique IP address.

N o t e	You can specify an IP address used for an interface on the ProCurve routing switch, but do not specify an IP address in use by another device.

Configuring ARP Parameters

Address Resolution Protocol (ARP) is a standard IP protocol that enables an IP routing switch to obtain the MAC address of another device's interface when the routing switch knows the IP address of the interface. ARP is enabled by default and cannot be disabled.

How ARP Works

A routing switch needs to know a destination's MAC address when forwarding traffic, because the routing switch encapsulates the IP packet in a Layer 2 packet (MAC layer packet) and sends the Layer 2 packet to a MAC interface on a device directly attached to the routing switch. The device can be the packet's final destination or the next-hop router toward the destination.

The routing switch encapsulates IP packets in Layer 2 packets regardless of whether the ultimate destination is locally attached or is multiple router hops away. Since the routing switch's IP route table and IP forwarding cache contain IP address information but not MAC address information, the routing switch cannot forward IP packets based solely on the information in the route table or forwarding cache. The routing switch needs to know the MAC address that corresponds with the IP address of either the packet's locally attached destination or the next-hop router that leads to the destination.

For example, to forward a packet whose destination is multiple router hops away, the routing switch must send the packet to the next-hop router toward its destination, or to a default route or default network route if the IP route table does not contain a route to the packet's destination. In each case, the routing switch must encapsulate the packet and address it to the MAC address of a locally attached device, the next-hop router toward the IP packet's destination.

To obtain the MAC address required for forwarding a datagram, the routing switch does the following:

■ First, the routing switch looks in the ARP cache (not the static ARP table) for an entry that lists the MAC address for the IP address. The ARP cache maps IP addresses to MAC addresses. The cache also lists the port attached to the device and, if the entry is dynamic, the age of the entry. A dynamic ARP entry enters the cache when the routing switch receives an ARP reply or receives an ARP request (which contains the sender's IP address and MAC address). A static entry enters the ARP cache from the static ARP table (which is a separate table) when the interface for the entry comes up.

To ensure the accuracy of the ARP cache, each dynamic entry has its own age timer. The timer is reset to zero each time the routing switch receives an ARP reply or ARP request containing the IP address and MAC address of the entry. If a dynamic entry reaches its maximum allowable age, the entry times out and the software removes the entry from the table. Static entries do not age out and can be removed only by you.

■ If the ARP cache does not contain an entry for the destination IP address, the routing switch broadcasts an ARP request out all its IP interfaces. The ARP request contains the IP address of the destination. If the device with the IP address is directly attached to the routing switch, the device sends an ARP response containing its MAC address. The response is a unicast packet addressed directly to the routing switch. The routing switch places the information from the ARP response into the ARP cache.

ARP requests contain the IP address and MAC address of the sender, so all devices that receive the request learn the MAC address and IP address of the sender and can update their own ARP caches accordingly.

Note: The ARP request broadcast is a MAC broadcast, which means the broadcast goes only to devices that are directly attached to the routing switch. A MAC broadcast is not routed to other networks. However, some routers, including ProCurve routing switches, can be configured to reply to ARP requests from one network on behalf of devices on another network. See "Enabling Proxy ARP" below.

N o t e   If the routing switch receives an ARP request packet that it is unable to deliver to the final destination because of the ARP time-out and no ARP response is received (the routing switch knows of no route to the destination address), the routing switch sends an ICMP Host Unreachable message to the source.

Enabling Proxy ARP

Proxy ARP allows a routing switch to answer ARP requests from devices on one network on behalf of devices in another network. Since ARP requests are MAC-layer broadcasts, they reach only the devices that are directly connected to the sender of the ARP request. Thus, ARP requests do not cross routers.

For example, if Proxy ARP is enabled on a routing switch connected to two sub-nets, 10.10.10.0/24 and 20.20.20.0/24, the routing switch can respond to an ARP request from 10.10.10.69 for the MAC address of the device with IP address 20.20.20.69. In standard ARP, a request from a device in the 10.10.10.0/24 sub-net cannot reach a device in the 20.20.20.0 sub-net if the sub-nets are on different network cables, and thus is not answered.

An ARP request from one sub-net can reach another sub-net when both sub-nets are on the same physical segment (Ethernet cable), since MAC-layer broadcasts reach all the devices on the segment.

Proxy ARP is disabled by default on ProCurve routing switches. To enable Proxy ARP, enter the following commands from the VLAN context level in the CLI:

```
ProCurve(config)# vlan 1
ProCurve(vlan-1)# ip proxy-arp
```

To again disable IP proxy ARP, enter the following command:

```
ProCurve(vlan-1)# no ip proxy-arp
```

Syntax: [no] ip proxy-arp

Configuring Forwarding Parameters

The following configurable parameters control the forwarding behavior of ProCurve routing switches:

■ Time-To-Live (TTL) threshold

■ Forwarding of directed broadcasts

All these parameters are global and thus affect all IP interfaces configured on the routing switch.

To configure these parameters, use the procedures in the following sections.

Changing the TTL Threshold

The configuration of this parameter is covered in chapter 7, "Configuring IP Addressing".

Enabling Forwarding of Directed Broadcasts

A directed broadcast is an IP broadcast to all devices within a single directly-attached network or sub-net. A net-directed broadcast goes to all devices on a given network. A sub-net-directed broadcast goes to all devices within a given sub-net.

N o t e   A less common type, the all-sub-nets broadcast, goes to all directly-attached sub-nets. Forwarding for this broadcast type also is supported, but most networks use IP multicasting instead of all-sub-net broadcasting.

Forwarding for all types of IP directed broadcasts is disabled by default. You can enable forwarding for all types if needed. You cannot enable forwarding for specific broadcast types.

To enable forwarding of IP directed broadcasts, enter the following CLI command:

```
ProCurve(config)# ip directed-broadcast
```

Syntax: [no] ip directed-broadcast

ProCurve software makes the forwarding decision based on the routing switch's knowledge of the destination network prefix. Routers cannot determine that a message is unicast or directed broadcast apart from the destination network prefix. The decision to forward or not forward the message is by definition only possible in the last hop router.

To disable the directed broadcasts, enter the following CLI command:

```
ProCurve(config)# no ip directed-broadcast
```

Configuring ICMP

You can configure the following ICMP limits:

- **Burst-Normal** – The maximum number of ICMP replies to send per second.
- **Reply Limit** – You can enable or disable ICMP reply rate limiting.

Disabling ICMP Messages

ProCurve devices are enabled to reply to ICMP echo messages and send ICMP Destination Unreachable messages by default.

You can selectively disable the following types of Internet Control Message Protocol (ICMP) messages:

- **Echo messages** (ping messages) – The routing switch replies to IP pings from other IP devices.
- **Destination Unreachable messages** – If the routing switch receives an IP packet that it cannot deliver to its destination, the routing switch discards the packet and sends a message back to the device that sent the packet to the routing switch. The message informs the device that the destination cannot be reached by the routing switch.
- **Address Mask replies** – You can enable or disable ICMP address mask replies.

Disabling Replies to Broadcast Ping Requests

By default, ProCurve devices are enabled to respond to broadcast ICMP echo packets, which are ping requests. You can disable response to ping requests on a global basis using the following CLI method.

To disable response to broadcast ICMP echo packets (ping requests), enter the following command:

```
ProCurve(config)# no ip icmp echo broadcast-request
```

Syntax: [no] ip icmp echo broadcast-request

If you need to re-enable response to ping requests, enter the following command:

```
ProCurve(config)# ip icmp echo broadcast-request
```

Disabling ICMP Destination Unreachable Messages

By default, when a ProCurve device receives an IP packet that the device cannot deliver, the device sends an ICMP Unreachable message back to the host that sent the packet. The following types of ICMP Unreachable messages are generated:

■ Administration – The packet was dropped by the ProCurve device due to a filter or ACL configured on the device.

■ Fragmentation-needed – The packet has the "Don't Fragment" bit set in the IP Flag field, but the ProCurve device cannot forward the packet without fragmenting it.

■ Host – The destination network or subnet of the packet is directly connected to the ProCurve device, but the host specified in the destination IP address of the packet is not on the network.

■ Network – The ProCurve device cannot reach the network specified in the destination IP address of the packet.

■ Port – The destination host does not have the destination TCP or UDP port specified in the packet. In this case, the host sends the ICMP Port Unreachable message to the ProCurve device, which in turn sends the message to the host that sent the packet.

■ Protocol – The TCP or UDP protocol on the destination host is not running. This message is different from the Port Unreachable message, which indicates that the protocol is running on the host but the requested protocol port is unavailable.

■ Source-route-failure – The device received a source-routed packet but cannot locate the next-hop IP address indicated in the packet's Source-Route option.

N o t e Disabling an ICMP Unreachable message type does not change the ProCurve device's ability to forward packets. Disabling ICMP Unreachable messages prevents the device from generating or forwarding the Unreachable messages.

To disable all ICMP Unreachable messages, enter the following command:

```
ProCurve(config)# no ip icmp unreachable
```

Syntax: [no] ip icmp unreachable

Disabling ICMP Redirects

You can disable ICMP redirects on the ProCurve routing switch. only on a global basis, for all the routing switch interfaces. To disable ICMP redirects globally, enter the following command at the global CONFIG level of the CLI:

```
ProCurve(config)# no ip icmp redirects
```

Syntax: [no] ip icmp redirects

Configuring Static IP Routes

The IP route table can receive routes from the following sources:

- Directly-connected networks – When you add an IP VLAN interface, the routing switch automatically creates a route for the network the interface is in.
- RIP – If RIP is enabled, the routing switch can learn about routes from the advertisements other RIP routers send to the routing switch. If the route has a lower administrative distance than any other routes from different sources to the same destination, the routing switch places the route in the IP route table.
- OSPF – See RIP, but substitute "OSPF" for "RIP".
- Statically configured route – You can add routes directly to the route table. When you add a route to the IP route table, you are creating a static IP route. This section describes how to add static routes to the IP route table.
- Default network route – This is a specific static route that the routing switch uses if other routes to the destination are not available. See "Configuring the Default Route" on page 11-23.

Static Route Types

You can configure the following types of static IP routes:

- **Standard** – the static route consists of the destination network address and network mask, and the IP address of the next-hop gateway.
- **Null (reject)** – the static route consists of the destination network address and network mask, and the **reject** parameter. Typically, the null route is configured as a backup route for discarding traffic if the primary route is unavailable.

11-17

Static IP Route Parameters

When you configure a static IP route, you must specify the following parameters:

■   The IP address and network mask for the route's destination network.

■   The route's path, which can be one of the following:

  •   The IP address of a next-hop gateway

  •   A "null" interface. The routing switch drops traffic forwarded to the null interface.

The routing switch also applies default values for the following routing parameters:

■   **The route's metric:** In the case of static routes, this is the value the routing switch uses when comparing a static route to routes in the IP route table from other sources to the same destination. The metric applies only to routes that the routing switch has already placed in the IP route table. The fixed metric for static IP routes is "1".

■   **The route's administrative distance:** In the case of static routes, this is the value the routing switch uses to compare a static route to routes from other route sources to the same destination before placing a route in the IP route table. This does not apply to routes that are already in the IP route table. The fixed administrative distance for static IP routes is "1".

The fixed metric and administrative distance values ensure that the routing switch always prefers static IP routes over routes from other sources to the same destination.

Static Route States Follow Port States

IP static routes remain in the IP route table only so long as the next-hop gateway, port, or virtual interface used by the route is available. If the gateway or port becomes unavailable, the software removes the static route from the IP route table. If the gateway or port later becomes available again, the software adds the route back to the route table.

This feature allows the routing switch to adjust to changes in network topology. The routing switch does not continue trying to use routes on unavailable paths but instead uses routes only when their paths are available.

The following command configures a static route to 207.95.7.0, using 207.95.6.157 as the next-hop gateway.

```
ProCurve(config)# ip route 207.95.7.0/24 207.95.6.157
```

When you configure a static IP route, you specify the destination address for the route and the next-hop gateway or routing switch interface through which the routing switch can reach the route. The routing switch adds the route to the IP route table. In this case, Router A knows that 207.95.6.157 is reachable through port A2, and also assumes that local interfaces within that sub-net are on the same port. Router A deduces that IP interface 207.95.7.188 is also on port A2.

The software automatically removes a static IP route from the IP route table if the port used by that route becomes unavailable. When the port becomes available again, the software automatically re-adds the route to the IP route table.

Configuring a Static IP Route

To configure an IP static route with a destination address of 192.0.0.0 255.0.0.0 and a next-hop router IP address of 195.1.1.1, enter the following commands:

```
ProCurve(config)# ip route 192.0.0.0 255.0.0.0 195.1.1.1
```

Syntax: [no] ip route < *dest-ip-addr* >/< *mask-bits* > < *next-hop-ip-addr* | reject | blackhole> [distance]

dest-ip-addr >/< mask-bits: The route destination and network mask length for the destination IP address. Alternatively, you can enter the mask itself. For example, you can enter either **10.0.0.0/24** or **10.0.0.0 255.255.255.0** for a route destination of 10.0.0.0 255.255.255.0.

next-hop-ip-addr : This IP address is the gateway for reaching the destination. The next-hop IP address is not required to be directly reachable on a local subnet. (If the next-hop IP address is not directly reachable, the route will be added to the routing table as soon as a route to this address is learned.)

reject: Specifies a null route where IP traffic for the specified destination is discarded and an ICMP error notification is returned to the sender.

blackhole: Specifies a null route where IP traffic for the specified destination is discarded and no ICMP error notification is returned to the sender.

distance: Specifies the administrative distance to associate with a static route. If not specified, this value is set to a default of 1.

The **no** form of the command deletes the specified route for the specified destination next-hop pair.

Configuring the Default Route

You can also assign the default router as the destination by entering 0.0.0.0 0.0.0.0.

Configuring a "Null" Route

You can configure the routing switch to drop IP packets to a specific network or host address by configuring a "null" (sometimes called "null0") static route for the address. When the routing switch receives a packet destined for the address, the routing switch drops the packet instead of forwarding it.

To configure a null static route to drop packets destined for network 209.157.22.x, enter the following commands:

```
ProCurve(config)# ip route 209.157.22.0 255.255.255.0
reject
ProCurve(config)# write memory
```

Syntax: ip route < *ip-addr* > < *ip-mask* > reject

or

ip route < *ip-addr* >/< *mask-bits* > reject

The **< *ip-addr* >** parameter specifies the network or host address. The routing switch will drop packets that contain this address in the destination field instead of forwarding them.

The **< *ip-mask* >** parameter specifies the network mask. Ones are significant bits and zeros allow any value. For example, the mask 255.255.255.0 matches on all hosts within the Class C sub-net address specified by <ip-addr>. Alternatively, you can specify the number of bits in the network mask. For example, you can enter 209.157.22.0/24 instead of 209.157.22.0 255.255.255.0.

The **reject** parameter indicates that this is a null route. You must specify this parameter to make this a null route.

Configuring RIP

This feature is not available on the Series 4200vl switches.

This section describes how to configure RIP using the CLI interface.

To display RIP configuration information and statistics, see "Displaying RIP Information" on page 11-31.

Overview of RIP

Routing Information Protocol (RIP) is an IP route exchange protocol that uses a ***distance vector*** (a number representing distance) to measure the cost of a given route. The cost is a distance vector because the cost often is equivalent to the number of router hops between the ProCurve routing switch and the destination network.

A ProCurve routing switch can receive multiple paths to a destination. The software evaluates the paths, selects the best path, and saves the path in the IP route table as the route to the destination. Typically, the best path is the path with the fewest hops. A hop is another router through which packets must travel to reach the destination. If the ProCurve routing switch receives a RIP update from another router that contains a path with fewer hops than the path stored in the ProCurve routing switch's route table, the routing switch replaces the older route with the newer one. The routing switch then includes the new path in the updates it sends to other RIP routers, including ProCurve routing switches.

RIP routers, including ProCurve routing switches, also can modify a route's cost, generally by adding to it, to bias the selection of a route for a given destination. In this case, the actual number of router hops may be the same, but the route has an administratively higher cost and is thus less likely to be used than other, lower-cost routes. A RIP route can have a maximum cost of 15. Any destination with a higher cost is considered unreachable. Although limiting to larger networks, the low maximum hop count prevents endless loops in the network.

The switches covered in this guide support the following RIP types:

- Version 1
- V1 compatible with V2
- Version 2 (the default)

N o t e   **ICMP Host Unreachable Message for Undeliverable ARPs.** If the
routing switch receives an ARP request packet that it is unable to deliver to
the final destination because of the ARP timeout and no ARP response is
received (the routing switch knows of no route to the destination address),
the routing switch sends an ICMP Host Unreachable message to the source.

RIP Parameters and Defaults

The following tables list the RIP parameters, their default values, and where
to find configuration information.

RIP Global Parameters

11-3 lists the global RIP parameters and their default values.

Table 11-3. RIP Global Parameters

Parameter	Description	Default
RIP state	Routing Information Protocol V2-only.	Disabled
auto-summary	Enable/Disable advertisement of summarized routes.	Enabled
metric	Default metric for imported routes.	1
redistribution	RIP can redistribute static and connected routes. (RIP redistributes connected routes by default, when RIP is enabled.)	Disabled

RIP Interface Parameters

11-4 lists the VLAN interface RIP parameters and their default values.

N o t e   RIP interface configuration is performed on VLAN interfaces in the switches
covered by this manual.

Table 11-4. RIP Interface Parameters

Parameter	Description	Default
RIP version	The version of the protocol that is supported on the interface. The version can be one of the following: • Version 1 only • Version 2 only • Version 1 compatible with version 2	V2-only
metric	A numeric cost the routing switch adds to RIP routes learned on the interface. This parameter applies only to RIP routes.	1
IP address	The routes that a routing switch learns or advertises can be controlled.	The routing switch learns and advertises all RIP routes on all RIP interfaces
loop prevention	The method the routing switch uses to prevent routing loops caused by advertising a route on the same interface as the one on which the routing switch learned the route. • **Split horizon** - the routing switch does not advertise a route on the same interface as the one on which the routing switch learned the route. • **Poison reverse** - the routing switch assigns a cost of 16 ("infinite" or "unreachable") to a route before advertising it on the same interface as the one on which the routing switch learned the route.	Poison reverse
receive	Define the RIP version for incoming packets	V2-only
send	Define the RIP version for outgoing packets	V2-only

Configuring RIP Parameters

Use the following procedures to configure RIP parameters on a system-wide and individual VLAN interface basis.

Enabling RIP

RIP is disabled by default. To enable it, use one of the following methods. When you enable RIP, the default RIP version is **RIPv2-only**. You can change the RIP version on an individual interface basis to **RIPv1** or **RIPv1-compatible-v2** if needed.

To enable RIP on a routing switch, enter the following commands:

```
ProCurve(config)# ip routing
ProCurve(config)# router rip
ProCurve(rip)# exit
ProCurve(config)# write memory
```

Syntax: [no] router rip

N o t e   IP routing must be enabled prior to enabling RIP. The first command in the preceding sequence enables IP routing.

Changing the RIP Type on a VLAN Interface

When you enable RIP on a VLAN interface, **RIPv2-only** is enabled by default. You can change the RIP type to one of the following on an individual VLAN interface basis:

- Version 1 only
- Version 2 only (the default)
- Version 1 - compatible - version 2

To change the RIP type supported on a VLAN interface, enter commands such as the following:

```
ProCurve(config)# vlan 1
ProCurve(vlan-1)# ip rip v1-only
ProCurve(vlan-1)# exit
ProCurve(config)# write memory
```

Syntax: [no] ip rip < v1-only | v1-compatible-v2 | v2-only >

Changing the Cost of Routes Learned on a VLAN Interface

By default, the switch interface increases the cost of a RIP route that is learned on the interface. The switch increases the cost by adding one to the route's metric before storing the route.

You can change the amount that an individual VLAN interface adds to the metric of RIP routes learned on the interface.

N o t e   RIP considers a route with a metric of 16 to be unreachable. Use this metric only if you do not want the route to be used. In fact, you can prevent the switch from using a specific interface for routes learned though that interface by setting its metric to 16.

To increase the cost a VLAN interface adds to RIP routes learned on that interface, enter commands such as the following:

```
ProCurve(config)# vlan 1
ProCurve(vlan-1)# ip rip metric 5
```

These commands configure vlan-1 to add 5 to the cost of each route learned on the interface.

Syntax: ip rip metric < 1-16 >

Configuring RIP Redistribution

You can configure the routing switch to redistribute connected and static routes into RIP. When you redistribute a route into RIP, the routing switch can use RIP to advertise the route to its RIP neighbors.

To configure redistribution, perform the following tasks:

1. Configure redistribution filters to permit or deny redistribution for a route based on the destination network address or interface. (optional)

2. Enable redistribution

Define RIP Redistribution Filters

Route redistribution imports and translates different protocol routes into a specified protocol type. On the switches covered in this guide, redistribution is supported for static routes and directly connected routes only. Redistribution of any other routing protocol into RIP is not currently supported. When you configure redistribution for RIP, you can specify that static or connected routes are imported into RIP routes. Likewise, OSPF redistribution supports the import of static or connected routes into OSPF routes.

To configure for redistribution, define the redistribution tables with "restrict" redistribution filters. In the CLI, use the **restrict** command for RIP at the RIP router level.

N o t e Do not enable redistribution until you have configured the redistribution filters. Otherwise, the network might get overloaded with routes that you did not intend to redistribute.

Example: To configure the switch to filter out redistribution of static or connected routes on network 10.0.0.0, enter the following commands:

```
ProCurve(config)# router rip
ProCurve(rip)# restrict 10.0.0.0 255.0.0.0
ProCurve(rip)# write memory
```

N o t e The default configuration permits redistribution for all default connected routes only.

Syntax: restrict < *ip-addr* > < *ip-mask* > | < *ip-addr* /< *prefix length* >

This command prevents any routes with a destination address that is included in the range specified by the address/mask pair from being redistributed by RIP.

Modify Default Metric for Redistribution

The default metric is a global parameter that specifies the cost applied to all RIP routes by default. The default value is 1. You can assign a cost from 1 – 15.

Example: To assign a default metric of 4 to all routes imported into RIP, enter the following commands:

```
ProCurve(config)# router rip
ProCurve(rip)# default-metric 4
```

Syntax: default-metric < *value* >

The **< *value* >** can be from 1 – 15. The default is 1.

Enable RIP Route Redistribution

N o t e Do not enable redistribution until you have configured the redistribution filters. Otherwise, the network might get overloaded with routes that you did not intend to redistribute.

To enable redistribution of connected and static IP routes into RIP, enter the following commands.

```
0(config)# router rip
ProCurve(rip)# redistribute connected
ProCurve(rip)# redistribute static
ProCurve(rip)# write memory
```

Syntax: [no] redistribute connected | static

Changing the Route Loop Prevention Method

RIP can use the following methods to prevent routing loops:

■ **Split horizon** - the routing switch does not advertise a route on the same interface as the one on which the routing switch learned the route.

■ **Poison reverse** - the routing switch assigns a cost of 16 ("infinity" or "unreachable") to a route before advertising it on the same interface as the one on which the routing switch learned the route. This is the default.

These loop prevention methods are configurable on an individual VLAN interface basis.

N o t e   These methods are in addition to RIP's maximum valid route cost of 15.

Poison reverse is enabled by default. Disabling poison reverse causes the routing switch to revert to **Split horizon**. (Poison reverse is an extension of Split horizon.) To disable Poison reverse on an interface, and thereby enable Split horizon, enter the following:

```
ProCurve(config)# vlan 1
ProCurve(vlan-1)# no ip rip poison-reverse
```

Syntax: [no] ip rip poison-reverse

Entering the command without the "no" option will re-enable Poison reverse.

Displaying RIP Information

All RIP configuration and status information is shown by the CLI command **show ip rip** and options off that command. The following RIP information can be displayed:

RIP Information Type	Page
General Information	11-31
Interface Information	11-33
Peer Information	11-34
Redistribute Information	11-36
Restrict Information	11-36

Displaying General RIP Information

To display general RIP information, enter **show ip rip** at any context level. The resulting display will appear similar to the following:

```
ProCurve(config)# show ip rip

 RIP global parameters

  RIP protocol   : enabled
  Auto-summary   : enabled
  Default Metric : 4
  Distance       : 120
  Route changes  : 0
  Queries        : 0

 RIP interface information

  IP Address      Status       Send mode      Recv mode    Metric       Auth
  --------------- ------------ -------------- ------------ ------------ ----
  100.1.0.1       enabled      V2-only        V2-only      5            none
  100.2.0.1       enabled      V2-only        V2-only      5            none
  100.3.0.1       enabled      V2-only        V2-only      5            none
  100.4.0.1       enabled      V2-only        V2-only      5            none
  100.10.0.1      enabled      V2-only        V2-only      5            none
  100.11.0.1      enabled      V2-only        V2-only      5            none
  100.12.0.1      enabled      V2-only        V2-only      5            none

 RIP peer information

  IP Address      Bad routes   Last update timeticks
  --------------- ------------ ----------------------
```

Figure 11-1.Example of General RIP Information Listing

The display is a summary of Global RIP information, information about interfaces with RIP enabled, and information about RIP peers. The following fields are displayed:

■ **RIP protocol** – Status of the RIP protocol on the router. RIP must be enabled here and on the VLAN interface for RIP to be active. The default is **disabled**.

■ **Auto-summary** – Status of Auto-summary for all interfaces running RIP. If auto-summary is enabled, then subnets will be summarized to a class network when advertising outside of the given network.

■ **Default Metric** – Sets the default metric for imported routes. This is the metric that will be advertised with the imported route to other RIP peers. A RIP metric is a measurement used to determine the 'best' path to network; 1 is the best, 15 is the worse, 16 is unreachable.

■ **Route changes** – The number of times RIP has modified the routing switch's routing table.

■ **Queries** – The number of RIP queries that have been received by the routing switch.

■ **RIP Interface Information** – RIP information on the VLAN interfaces on which RIP is enabled.

• **IP Address** – IP address of the VLAN interface running rip.

• **Status** – Status of RIP on the VLAN interface.

• **Send mode** – The format of the RIP updates: RIP 1, RIP 2, or RIP 2 version 1 compatible.

• **Recv mode** – The switch can process RIP 1, RIP 2, or RIP 2 version 1 compatible update messages.

• **Metric** – The path "cost", a measurement used to determine the 'best' RIP route path; 1 is the best, 15 is the worse, 16 is unreachable.

• **Auth** – RIP messages can be required to include an authentication key if enabled on the interface.

■ **RIP Peer Information** – RIP Peers are neighboring routers from which the routing switch has received RIP updates.

• **IP Address** – IP address of the RIP neighbor.

• **Bad routes** – The number of route entries which were not processed for any reason.

• **Last update timeticks** – How many seconds have passed since we received an update from this neighbor.

Syntax: show ip rip

Displaying RIP Interface Information

To display RIP interface information, enter the show ip rip interface command at any context level. The resulting display will appear similar to the following:

```
ProCurve# show ip rip interface

 RIP interface information

  IP Address       Status       Send mode          Recv mode    Metric        Auth
 ---------------  -----------  ----------------    ----------   -----------   ----
  100.1.0.1        enabled      V2-only            V2-only      1             none
  100.2.0.1        enabled      V2-only            V2-only      1             none
  100.3.0.1        enabled      V2-only            V2-only      1             none
  100.4.0.1        enabled      V2-only            V2-only      1             none
```

Figure 11-2.Example of Show IP RIP Interface Output

See "RIP Interface Information" on the previous page for definitions of these fields.

You can also display the information for a single RIP VLAN interface, by specifying the VLAN ID for the interface, or specifying the IP address for the interface.

Displaying RIP interface information by VLAN ID: For example, to show the RIP interface information for VLAN 1000, use the **show ip rip interface vlan < *vid* >** command.

```
ProCurve# show ip rip interface vlan 4

 RIP configuration and statistics for VLAN 4

 RIP interface information for 100.4.0.1

  IP Address : 100.4.0.1
  Status     : enabled

  Send mode  : V2-only
  Recv mode  : V2-only
  Metric : 1
  Auth : none

  Bad packets received : 0
  Bad routes received  : 0
  Sent updates : 0
```

Figure 11-3. Example of RIP Interface Output by VLAN

The information in this display includes the following fields, which are defined under ""RIP Interface Information" on page 11-32: **IP Address**, **Status**, **Send mode**, **Recv mode**, **Metric**, and **Auth**.

The information also includes the following fields:

■   **Bad packets received** – The number of packets that were received on this interface and were not processed for any reason.

■   **Bad routes received** – The number of route entries that were received on this interface and were not processed for any reason.

■   **Sent updates** – The number of RIP routing updates that have been sent on this interface.

Displaying RIP interface information by IP Address: For example, to show the RIP interface information for the interface with IP address 100.2.0.1, enter the **show ip rip interface** command as shown below:

```
ProCurve# show ip rip interface 100.2.0.1

 RIP interface information for 100.2.0.1

  IP Address : 100.2.0.1
  Status     : enabled

  Send mode  : V2-only
  Recv mode  : V2-only
  Metric : 1
  Auth : none

  Bad packets received : 0
  Bad routes received  : 0
  Sent updates : 0
```

Figure 11-4. Example of Show IP RIP Interface Output by IP Address

The information shown in this display has the same fields as for the display for a specific VLAN ID. See the previous page for the definitions of these fields.

Syntax: show ip rip interface [*ip-addr* | vlan < *vlan-id* >]

Displaying RIP Peer Information

To display RIP peer information, enter the **show ip rip peer** command at any context level.

The resulting display will appear similar to the following:

```
ProCurve# show ip rip peer

RIP peer information

 IP Address      Bad routes  Last update timeticks
 --------------- ----------- ---------------------
 100.1.0.100     0           1
 100.2.0.100     0           0
 100.3.0.100     0           2
 100.10.0.100    0           1
```

Figure 11-5. Example of Show IP RIP Peer Output

This display lists all neighboring routers from which the routing switch has received RIP updates. The following fields are displayed:

- **IP Address** – IP address of the RIP peer neighbor.
- **Bad routes** – The number of route entries that were not processed for any reason.
- **Last update timeticks** – How many seconds have passed since the routing switch received an update from this peer neighbor.

Displaying RIP information for a specific peer: For example, to show the RIP peer information for the peer with IP address 100.1.0.100, enter **show ip rip peer 100.1.0.100**.

```
ProCurve# show ip rip peer 100.0.1.100

RIP peer information for 100.0.1.100

  IP Address : 100.1.0.100

  Bad routes  : 0
  Last update timeticks : 2
```

Figure 11-6. Example of Show IP RIP Peer < *ip-addr* > Output

This display lists the following information for a specific RIP peer:

- **IP Address** – IP address of the RIP peer neighbor.
- **Bad routes** – The number of route entries which were not processed for any reason.
- **Last update timeticks** – How many seconds have passed since the routing switch received an update from this neighbor.

Displaying RIP Redistribution Information

To display RIP redistribution information, enter the **show ip rip redistribute** command at any context level:

```
ProCurve# show ip rip redistribute

RIP redistributing

 Route type Status
 ---------- --------
 connected  enabled
 static     enabled
```

Figure 11-7. Example of Show IP RIP Redistribute Output

RIP automatically redistributes connected routes which are configured on interfaces that are running RIP, and all routes that are learned via RIP. The **router rip redistribute** command, described on page 11-28, configures the routing switch to cause RIP to advertise connected routes that are not running RIP, and static routes. The display shows whether RIP redistribution is enabled or disabled for connected and static routes.

Displaying RIP Redistribution Filter (restrict) Information

To display RIP restrict filter information, enter the **show ip rip restrict** command at any context level:

```
ProCurve# show ip rip restrict

RIP restrict list

 IP Address      Mask
 -------------- ---------------


```

Figure 11-8. Example of Show IP RIP Restrict Output

The display shows if any routes, identified by the IP Address and Mask fields are being restricted from redistribution. The restrict filters are configured by the **router rip restrict** command described on page 11-28.

Configuring OSPF

This feature is not available on the Series 4200vl switches.

This section describes how to configure OSPF using the CLI interface.

To display OSPF configuration information and statistics, see "Displaying OSPF Information" on page 11-56.

Overview of OSPF

OSPF is a link-state routing protocol. The protocol uses link-state advertisements (LSA) to update neighboring routers regarding its interfaces and information on those interfaces. The switch floods these LSAs to all neighboring routers to update them regarding the interfaces. Each router maintains an identical database that describes its area topology to help a router determine the shortest path between it and any neighboring router.

The switches covered in this guide support the following types of LSAs, which are described in RFC 2328:

■ Router link

■ Network link

■ Summary link

■ Autonomous system (AS) summary link

■ AS external link

OSPF is built upon a hierarchy of network components. The highest level of the hierarchy is the ***Autonomous System (AS)***. An autonomous system is defined as a number of networks, all of which share the same routing and administration characteristics.

An AS can be divided into multiple ***areas***. Each area represents a collection of contiguous networks and hosts. Areas define the limit to which link-state advertisements are broadcast, thereby limiting the amount of flooding that occurs within the network. An area is represented in OSPF by either an IP address or a number.

You can further limit the broadcast area of flooding by defining an area range. The area range allows you to assign an aggregate value to a range of IP addresses. This aggregate value becomes the address that is advertised instead all of the individual addresses it represents being advertised. You can assign up to 8 ranges in an OSPF area.

An OSPF router can be a member of multiple areas. Routers with membership in multiple areas are known as *Area Border Routers (ABRs)*. Each ABR maintains a separate topological database for each area the router is in. Each topological database contains all of the LSA databases for each router within a given area. The routers within the same area have identical topological databases. The ABR is responsible for forwarding routing information or changes between its border areas.

An *Autonomous System Boundary Router (ASBR)* is a router that is running multiple protocols and serves as a gateway to routers outside an area and those operating with different protocols. The ASBR is able to import and translate different protocol routes into OSPF through a process known as *redistribution*. For more details on redistribution and configuration examples, see "Enabling Route Redistribution" on page 11-53.

Designated Routers in Multi-Access Networks

In a network that has multiple routers attached, OSPF elects one router to serve as the designated router (DR) and another router on the segment to act as the backup designated router (BDR). This arrangement minimizes the amount of repetitive information that is forwarded on the network by forwarding all messages to the designated router and backup designated routers responsible for forwarding the updates throughout the network.

Designated Router Election

In a network with no designated router and no backup designated router, the neighboring router with the highest priority is elected as the DR, and the router with the next largest priority is elected as the BDR.

If the DR goes off-line, the BDR automatically becomes the DR. The router with the next highest priority becomes the new BDR.

N o t e   Priority is a configurable option at the interface level. You can use this parameter to help bias one 5300xl, 3400cl, or 6400cl switch as the DR.

If two neighbors share the same priority, the router with the highest router ID is designated as the DR. The router with the next highest router ID is designated as the BDR.

N o t e   By default, the router ID is the lowest numbered IP address configured on the device. For more information or to change the router ID, see "Changing the Router ID" on page 11-13.

When multiple ProCurve switches on the same network are declaring themselves as DRs, then both priority and router ID are used to select the designated router and backup designated routers.

When only one router on the network claims the DR role despite neighboring routers with higher priorities or router IDs, this router remains the DR. This is also true for BDRs.

The DR and BDR election process is performed when one of the following events occurs:

- an interface is in a waiting state and the wait time expires
- an interface is in a waiting state and a hello packet is received that addresses the BDR
- a change in the neighbor state occurs, such as:
  - a neighbor state transitions from 2 or higher
  - communication to a neighbor is lost
  - a neighbor declares itself to be the DR or BDR for the first time

OSPF RFC 1583 and 2328 Compliance

The switches covered in this guide are configured, by default, to be compliant with the RFC 1583 OSPF V2 specification. These switches can also be configured to operate with the latest OSPF standard, RFC 2328.

N o t e   For details on how to configure the system to operate with the RFC 2328, see "Configuring OSPF" on page 11-41.

Reduction of Equivalent AS External LSAs

An OSPF ASBR uses AS External link advertisements (AS External LSAs) to originate advertisements of a route to another routing domain, such as a RIP domain. The ASBR advertises the route to the external domain by flooding AS External LSAs to all the other OSPF routers (except those inside stub networks) within the local OSPF Autonomous System (AS).

In some cases, multiple ASBRs in an AS can originate equivalent LSAs. The LSAs are equivalent when they have the same cost, the same next hop, and the same destination. The ProCurve switch optimizes OSPF by eliminating duplicate AS External LSAs in this case. The switch with the lower router ID flushes the duplicate External LSAs from its database and thus does not flood the duplicate External LSAs into the OSPF AS. AS External LSA reduction therefore reduces the size of the switch's link state database.

This enhancement implements the portion of RFC 2328 that describes AS External LSA reduction. This enhancement is enabled by default, requires no configuration, and cannot be disabled.

OSPF eliminates duplicate AS External LSAs. When two or more 5300xl, 3400cl, and/or 6400cl switches configured as ASBRs have equal-cost routes to the same next-hop router in an external routing domain, the ASBR with the highest router ID floods the AS External LSAs for the external domain into the OSPF AS, while the other ASBRs flush the equivalent AS External LSAs from their databases. As a result, the overall volume of route advertisement traffic within the AS is reduced and the switches that flush the duplicate AS External LSAs have more memory for other OSPF data.

Algorithm for AS External LSA Reduction. The AS External LSA reduction feature behavior changes under the following conditions:

■ There is one ASBR advertising (originating) a route to the external destination, but one of the following happens:

• A second ASBR comes on-line

• A second ASBR that is already on-line begins advertising an equivalent route to the same destination.

In either case above, the ProCurve switch with the higher router ID floods the AS External LSAs and the other ProCurve switch flushes its equivalent AS External LSAs.

■ One of the ASBRs starts advertising a route that is no longer equivalent to the route the other ASBR is advertising. In this case, the ASBRs each flood AS External LSAs. Since the LSAs either no longer have the same cost or no longer have the same next-hop router, the LSAs are no longer equivalent, and the LSA reduction feature no longer applies.

■ The ASBR with the higher router ID becomes unavailable or is reconfigured so that it is no longer an ASBR. In this case, the other ASBR floods the AS External LSAs.

Dynamic OSPF Activation and Configuration

OSPF is automatically activated when you enable it. The protocol does not require a software reload.

Without ever having to reset the switch, you can change and save all the OSPF configuration options, including the following:

- all OSPF interface-related parameters (for example: area, hello timer, router dead time cost, priority, re-transmission time, transit delay)
- all area parameters
- all area range parameters
- all virtual-link parameters
- all global parameters
- creation and deletion of an area, interface or virtual link
- changes to address ranges
- changes to global values for redistribution
- addition of new virtual links

The only configuration change that requires you to disable and then re-enable OSPF operation is reconfiguring the Router ID.

Configuring OSPF

To begin using OSPF on the switch, perform the steps outlined below:

1. Enable routing on the routing switch.
1. Enable OSPF on the routing switch.
2. Assign the areas to which the routing switch will be attached.
3. Assign individual VLAN interfaces to the OSPF areas.
4. Define redistribution "restrict" filters, if desired.
5. Enable redistribution, if you defined redistribution filters.
6. Modify default global and interface parameters as required.
7. Modify OSPF standard compliance, if desired.

Note   OSPF is automatically enabled without a system reset.

Configuration Rules

- If the switch is to operate as an ASBR, you must enable redistribution. When you do that, ASBR capability is automatically enabled.
- All VLAN interfaces on which you wish to run OSPF must be assigned to one of the defined areas. When a VLAN interface is assigned to an area, the IP address is automatically included in the assignment. To include additional addresses, you must enable OSPF on them separately, or use the "all" option in the assignment.

OSPF Parameters

You can modify or set the following global and interface OSPF parameters.

Global Parameters:

- Modify OSPF standard compliance setting.
- Assign an area.
- Define an area range.
- Define the area virtual link.
- Set global default metric for OSPF.
- Define redistribution metric type.
- Enable redistribution.
- Define redistribution restrict filters.
- Modify OSPF Traps generated.

Interface Parameters:

- Assign interfaces to an area.
- Select the authentication method (simple password or MD5) and the authentication key for the interface.
- Modify the cost for a link.
- Modify the dead interval.
- Modify the priority of the interface.
- Modify the retransmit interval for the interface.
- Modify the transit delay of the interface.

N o t e   When using the CLI, you set global level parameters at the OSPF CONFIG Level of the CLI. To reach that level, make sure routing is enabled and then enter the command **router ospf** at the global CONFIG Level. Interface parameters for OSPF are set at the VLAN CONFIG Level using the CLI command **ip ospf.**

Enabling OSPF

When you enable OSPF, the protocol is automatically activated. To enable OSPF, use the CLI commands:

```
ProCurve(config)# ip routing
ProCurve(config)# router ospf
```

The first command enables routing on the switch. The second command launches you into the OSPF router level where you can assign areas and modify OSPF global parameters.

N o t e   **Regarding Disabling OSPF.** If you disable OSPF, the switch retains all the configuration information for the disabled protocol in flash memory. If you subsequently restart OSPF, that previous configuration will be applied.

Assigning OSPF Areas

Once OSPF is enabled on the system, you can assign areas. Assign an IP address or number as the *area ID* for each area. The area ID is representative of only the first IP address. To include other IP addresses, you must enable OSPF on them separately, or use the "all" option in the assignment. Each VLAN interface on the switch can support 16 areas.

N o t e   You can assign subnets individually to areas. The limit on the number of areas is 16.

An area can be *normal* or a *stub*.

■ **Normal** – A switch within an OSPF normal area can send and receive External Link State Advertisements (LSAs).

■ **Stub** – A switch within an OSPF stub area cannot send or receive External LSAs. In addition, the routing switches in an OSPF stub area must use a default route to the area's Area Border Router (ABR) or Autonomous System Boundary Router (ASBR) to send traffic out of the area.

Example: Here is an example of the commands to set up several OSPF areas.

```
ProCurve(ospf)# area 192.5.1.0
ProCurve(ospf)# area 200.5.0.0
ProCurve(ospf)# area 0.0.0.0
ProCurve(ospf)# write memory
```

Syntax: area < *num* > | < *ip-addr* > [normal | stub < *cost* > [no-summary]]

The **< *num* > | < *ip-addr* >** parameter specifies the area number, which can be a number or in IP address format. If you specify a number, the number can be from 0 – 4,294,967,295.

The **< *cost* >** specifies the cost of the default route to be injected into the stub area, if this routing switch is an ABR. The value can be from 1 – 16,777,215. If you configure a stub area, you must specify the cost. There is no default. Normal areas do not use the cost parameter.

N o t e   The switch CLI requires that you enter a cost value when specifying the stub parameter, but this cost value is ignored. The actual cost is provided by the Area Border Router (ABR).

The **no-summary** parameter applies only to stub areas and disables summary LSAs from being sent into the area. See "Disabling Summary LSAs" below.

Disabling Summary LSAs

By default, the switch sends summary LSAs (LSA type 3) into stub areas. You can further reduce the number of LSAs sent into a stub area by configuring the switch to stop sending summary LSAs into the area. You can disable the summary LSAs when you are configuring the stub area or later after you have configured the area.

This feature disables origination of summary LSAs, but the switch still accepts summary LSAs from OSPF neighbors and floods them to other neighbors. The switch can form adjacencies with other routers regardless of whether summarization is enabled or disabled for areas on each switch.

When you enter a command to disable the summary LSAs, the change takes effect immediately. If you apply the option to a previously configured area, the switch flushes all of the summary LSAs it has generated (as an ABR) from the area.

N o t e   This feature applies only when the switch is configured as an Area Border Router (ABR) for the area. To completely prevent summary LSAs from being sent to the area, disable the summary LSAs on each OSPF router that is an ABR for the area.

To disable summary LSAs for a stub area, enter commands such as the following:

```
ProCurve(config-ospf-router)# area 40 stub 3 no-summary
```

Assigning an Area Range (optional)

You can assign a *range* for an area, but it is not required. Ranges allow a specific IP address and mask to represent a range of IP addresses within an area, so that only that reference range address is advertised to the network, instead of all the addresses within that range. Each area can have up to 8 range addresses.

Example.   To define an area range for sub-nets on 193.45.5.1 and 193.45.6.2, enter the following commands:

```
ProCurve(config)# router ospf
ProCurve(ospf)# area 192.45.5.1 range 193.45.0.0
255.255.0.0
ProCurve(ospf)# area 193.45.6.2 range 193.45.0.0
255.255.0.0
```

Syntax: area < *ospf-area-id* | backbone > range < *ip-addr*/ *mask-length* >
              [no-advertise]

The **<** *ospf-area-id* **>** parameter specifies the area number, which can be in IP address format.

The **<** *ip-addr* **>** parameter following **range** specifies the IP address portion of the range. The software compares the address with the significant bits in the mask. All network addresses that match this comparison are summarized in a single route advertised by the switch.

The **<** *mask-length* **>** parameter specifies the portions of the IP address that a route must contain to be summarized in the summary route. In the example above, all networks that begin with 209.157 are summarized into a single route.

Assigning VLANs to an Area

Once you define OSPF areas, you can assign VLANs to the areas. All VLANs in the switch must be assigned to one of the defined areas on an OSPF router. When a VLAN is assigned to an area, the first IP address is automatically included in the assignment. To include other IP addresses, you must enable OSPF on them separately, or use the "all" option in the assignment.

Example: To assign VLAN 8 of Switch A to area 192.5.0.0 and then save the changes, enter the following commands:

```
ProCurve(ospf)# vlan 8
RouterA(vlan-8)# ip ospf area 192.5.0.0
RouterA(vlan-8)# write memory
```

Modifying Interface Defaults

OSPF has interface parameters that you can configure. For simplicity, each of these parameters has a default value. No change to these default values is required except as needed for specific network configurations.

VLAN default values can be modified using the following CLI commands at the **VLAN interface level** of the CLI:

- **ip ospf area < *ip-addr* >**
- **ip ospf authentication-key < *password* >**
- **ip ospf md5-auth-key-chain < *chain-name-str* >**
- **ip ospf cost < *num* >**
- **ip ospf dead-interval < *value* >**
- **ip ospf hello-interval < *value* >**
- **ip ospf priority < value >**
- **ip ospf retransmit-interval <** *value* **>**
- **ip ospf transmit-delay <** *value* **>**

For a complete description of these parameters, see the summary of OSPF interface parameters in the next section.

OSPF Interface Parameters

The following parameters apply to OSPF interfaces.

Area: Assigns an interface to a specific area. You can assign either an IP address or number to represent an OSPF Area ID. If you assign a number, it can be any value from 0 – 4,294,967,295.

Authentication-key: OSPF supports two methods of authentication for each VLAN—simple password and MD5. In addition, the value can be set to none, meaning no authentication is performed. Only one method of authentication can be active on a subnet at a time. The default authentication value is none. The two authentication methods are configured by different commands:

- **Simple password** – Use the **ip ospf authentication-key <*password*>** command. The simple password method of authentication requires you to configure an alphanumeric password on an interface. The simple password setting takes effect immediately. All OSPF packets transmitted on the interface contain this password. Any OSPF packet that is received on the interface is checked for this password. If the password is not present, then the packet is dropped. The password can be up to eight characters long.

- **MD5** – Use the **ip ospf md5-auth-key-chain <*chain-name-str*>** command. The MD5 method of authentication uses key chains that you configure through the Key Management System (KMS – described in your switch Security Guide). The **<*chain-name-str*>** is the name of the key chain that you have previously configured by using the KMS commands.

Cost: Indicates the overhead required to send a packet across an interface. You can modify the cost to differentiate between 100 Mbps and 1000 Mbps (1 Gbps) links. The default cost is always 1.

Dead-interval: Indicates the number of seconds that a neighbor router waits for a hello packet from the current switch before declaring the switch down. The value can be from 1 – 2,147,483,647 seconds. The default is 40 seconds.

Hello-interval: Represents the length of time between the transmission of hello packets. The value can be from 1 – 65535 seconds. The default is 10 seconds.

Priority: Allows you to modify the priority of an OSPF router. The priority is used when selecting the designated router (DR) and backup designated routers (BDRs). The value can be from 0 – 255 (with 255 as the highest priority). The default is 1. If you set the priority to 0, the switch does not participate in DR and BDR election.

Retransmit-interval: The time between retransmissions of link-state advertisements (LSAs) to adjacent routers for this interface. The value can be from 0 – 3600 seconds. The default is 5 seconds.

Transit-delay: The time it takes to transmit Link State Update packets on this interface. The value can be from 0 – 3600 seconds. The default is 1 second.

Assigning Virtual Links

It is highly recommended that all ABRs (area border routers) have either a direct or indirect link to the OSPF backbone area (0.0.0.0 or 0). If an ABR does not have a physical link to the area backbone, the ABR can configure a *virtual link* to another router within the same area, which has a physical connection to the area backbone.

N o t e	A backbone area can be purely virtual with no physical backbone links. Also note that virtual links can be "daisy chained". If so, it may not have one end physically connected to the backbone.

The path for a virtual link is through an area shared by the neighbor ABR (router with a physical backbone connection), and the ABR requiring a logical connection to the backbone.

Two parameters must be defined for all virtual links—**transit area ID** and **neighbor router**.

- The *transit area ID* represents the shared area of the two ABRs and serves as the connection point between the two routers. This number should match the area ID value.

- The *neighbor router* field is the router ID (IP address) of the router that is physically connected to the backbone, when assigned from the router interface requiring a logical connection. When assigning the parameters from the router with the physical connection, the router ID is the IP address of the router requiring a logical connection to the backbone.

N o t e	By default, the router ID is the lowest numbered IP address configured on the device. For more information or to change the router ID, see "Changing the Router ID" on page 11-13.

N o t e	When you establish an area virtual link, you must configure it on both of the routers (both ends of the virtual link).
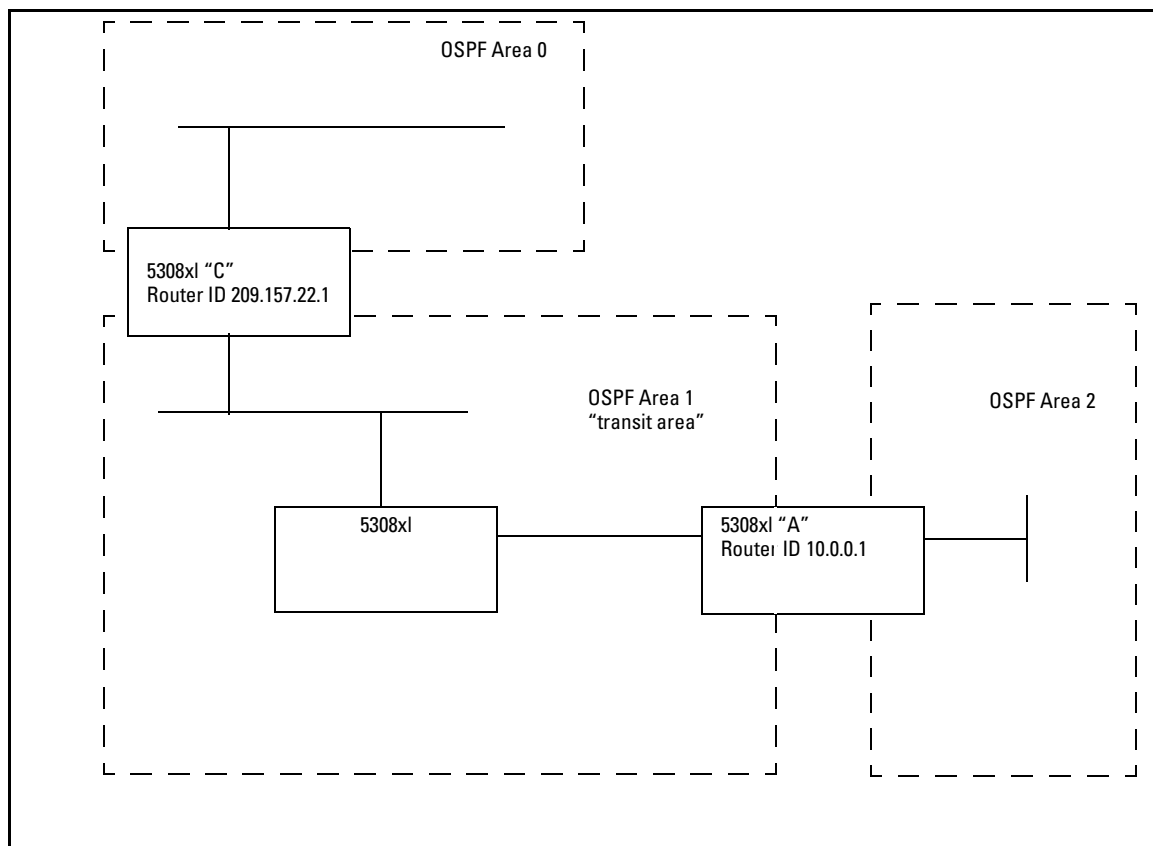
Figure 11-9. Defining OSPF virtual links within a network

Example. Figure 11-9 shows an OSPF area border router, Routing Switch-A, that is cut off from the backbone area (Area 0). To provide backbone access to Routing Switch-A, you can add a virtual link between Routing Switch-A and Routing Switch-C using Area 1 as a transit area. To configure the virtual link, you define the link on the router that is at each end of the link. No configuration for the virtual link is required on the routers in the transit area.

To configure the virtual link on Routing Switch-A, enter the following commands:

```
ProCurve(ospf)# area 1 virtual-link 209.157.22.1
ProCurve(ospf)# write memory
```

To configure the virtual link on Routing Switch-C, enter the following commands:

```
ProCurve(ospf)# area 1 virtual-link 10.0.0.1
ProCurve(ospf)# write memory
```

Syntax: area <ip-addr> | <num> virtual-link <router-id>

The **area < *ip-addr* > | < *num* >** parameter specifies the transit area.

The ***<router-id>*** parameter specifies the router ID of the OSPF router at the remote end of the virtual link. To display the router ID, enter the **show ip** command.

See "Modify Virtual Link Parameters" below for descriptions of the optional parameters.

Modifying Virtual Link Parameters

OSPF has some parameters that you can modify for virtual links. Notice that these are a subset of the parameters that you can modify for physical interfaces. **cost** is not configured for virtual links, it is calculated by route calculation.

You can modify default values for virtual links using the following CLI command at the *OSPF router level* of the CLI, as shown in the following syntax:

Syntax: area < *num* > | < *ip-addr* > virtual-link < *ip-addr* > [authentication-key < *string* >| md5-auth-key-chain < *chain-name-str* >] [dead-interval < *num* >] [hello-interval < *num* >] [retransmit-interval < *num* >] [transmit-delay < *num* >]

The parameters are described below. For syntax information, at the CLI prompt, enter the command **area help**.

Virtual Link Parameter Descriptions

You can modify the following virtual link interface parameters:

Authentication Key: OSPF supports two methods of authentication for each virtual link—**simple password** and **MD5**. In addition, the value can be set to **none**, meaning no authentication is performed. Only one method of authentication can be active on a subnet at a time. The default authentication value is none. The two authentication methods are configured by different commands:

- **Simple password** – Use the **area _\<num\>_ | _\<ip-addr\>_ virtual-link _\<ip-addr\>_ authentication-key _\<password\>_** command. The simple password method of authentication requires you to configure an alphanumeric password on an interface. The simple password setting takes effect immediately. All OSPF packets transmitted on the interface contain this password. Any OSPF packet that is received on the interface is checked for this password. If the password is not present, then the packet is dropped. The password can be up to eight characters long.

- **MD5** – Use the **area _\<num\>_ | _\<ip-addr\>_ virtual-link _\<ip-addr\>_ md5-auth-key-chain _\<chain-name-str\>_** command. The MD5 method of authentication uses key chains that you configure through the Key Management System (KMS – described in your switch Security Guide). The **_\<chain-name-str\>_** is the name of the key chain that you have previously configured by using the KMS commands.

Hello Interval: The length of time between the transmission of hello packets. The range is 1 – 65535 seconds. The default is 10 seconds.

Retransmit Interval: The interval between the re-transmission of link state advertisements to router adjacencies for this interface. The range is 0 – 3600 seconds. The default is 5 seconds.

Transmit Delay: The period of time it takes to transmit Link State Update packets on the interface. The range is 0 – 3600 seconds. The default is 1 second.

Dead Interval: The number of seconds that a neighbor router waits for a hello packet from the current routing switch before declaring the routing switch down. The range is 1 – 65535 seconds. The default is 40 seconds.

Defining Redistribution Filters

Route redistribution imports and translates different protocol routes into a specified protocol type. Redistribution is supported for only static routes and directly connected routes. Redistribution of any other routing protocol into OSPF is not currently supported. When you configure redistribution for OSPF, you can specify that static or connected routes are imported into OSPF routes. Likewise, RIP redistribution supports the import of static or connected routes into RIP routes.

To configure for redistribution, define the redistribution tables with restrict redistribution filters. In the CLI, use the **restrict** command for OSPF at the OSPF router level.

N o t e   Do not enable redistribution until you have configured the redistribution filters. Otherwise, the network might get overloaded with routes that you did not intend to redistribute.

Example: To configure the switch acting as an ASBR to filter out redistribution of static or connected routes on network 10.0.0.0, enter the following commands:

```
ProCurve(config)# router ospf
ProCurve(ospf)# restrict 10.0.0.0 255.0.0.0
ProCurve(ospf)# write memory
```

N o t e   Redistribution is permitted for all routes by default.

Syntax: restrict < *ip-addr* > < *ip-mask* > | < *ip-addr*/< *prefix length* >

This command prevents any routes with a destination address that is included in the range specified by the address/mask pair from being redistributed by OSPF.

Modifying Default Metric for Redistribution

The default metric is a global parameter that specifies the cost applied to all OSPF routes by default. The default value is 10. You can assign a cost from 1 – 16,777,215.

Example: To assign a default metric of 4 to all routes imported into OSPF, enter the following commands:

```
ProCurve(config)# router ospf
ProCurve(ospf)# default-metric 4
```

Syntax: default-metric < *value* >

The < *value* > can be from 1 – 16,777,215. The default is 10.

Enabling Route Redistribution

Do not enable redistribution until you have configured the redistribution "restrict" filters. Otherwise, the network might get overloaded with routes that you did not intend to redistribute.

To enable redistribution of connected and static IP routes into OSPF, enter the following commands.

```
ProCurve(config)# router ospf
ProCurve(ospf)# redistribution connected
ProCurve(ospf)# redistribution static
ProCurve(ospf)# write memory
```

Syntax: [no] redistribution connected | static

Modifying Redistribution Metric Type

The redistribution metric type is used by default for all routes imported into OSPF. Type 1 metrics are the same "units" as internal OSPF metrics and can be compared directly. Type 2 metrics are not directly comparable, and are treated as larger than the largest internal OSPF metric. The default value is type 2.

To modify the default value to type 1, enter the following command:

```
ProCurve(config-ospf-router)# metric-type type1
```

Syntax: metric-type type1 | type2

The default is **type2**.

Administrative Distance

The switch can learn about networks from various protocols, including RIP, and OSPF. Consequently, the routes to a network may differ depending on the protocol from which the routes were learned. For the switches covered by this guide, the administrative distance for OSPF routes is set at 110.

The switch selects one route over another based on the source of the route information. To do so, the switch can use the administrative distances assigned to the sources.

Modifying OSPF Traps Generated

OSPF traps as defined by RFC 1850 are supported on the switches covered in this guide. OSPF trap generation is enabled by default.

When using the CLI, you can disable all or specific OSPF trap generation by entering the following CLI command:

```
ProCurve(ospf)# no snmp-server trap ospf
```

To later re-enable the trap feature, enter the command:

```
ProCurve(ospf)# snmp-server trap ospf
```

To disable a specific OSPF trap, enter the following command:

```
ProCurve(ospf)# no snmp-server trap ospf < ospf-trap >.
```

These commands are at the OSPF Router Level of the CLI.

Here is a summary of OSPF traps supported on switches covered in this guide, and their associated MIB objects from RFC 1850:

Table 11-5. OSPF Traps and Associated MIB Objects

OSPF Trap Name	MIB Object
interface-state-change-trap	ospfIfstateChange
virtual-interface-state-change-trap	ospfVirtIfStateChange
neighbor-state-change-trap	ospfNbrStateChange
virtual-neighbor-state-change-trap	ospfVirtNbrStateChange
interface-config-error-trap	ospfIfConfigError
virtual-interface-config-error-trap	ospfVirtIfConfigError
interface-authentication-failure-trap	ospfIfAuthFailure
virtual-interface-authentication-failure-trap	ospfVirtIfAuthFailure
interface-receive-bad-packet-trap	ospfIfrxBadPacket
virtual-interface-receive-bad-packet-trap	ospfVirtIfRxBadPacket
interface-retransmit-packet-trap	ospfTxRetransmit
virtual-interface-retransmit-packet-trap	ospfVirtIfTxRetransmit

OSPF Trap Name	MIB Object
originate-lsa-trap	ospfOriginateLsa
originate-maxage-lsa-trap	ospfMaxAgeLsa
link-state-database-overflow-trap	ospfLsdbOverflow
link-state-database-approaching-overflow-trap	ospfLsdbApproachingOverflow

Examples:

1. To stop an OSPF trap from being collected, use the following CLI command:

    ```
    ProCurve(ospf)# no trap < ospf-trap >
    ```

2. To disable reporting of the neighbor-state-change-trap, enter the following command:

    ```
    ProCurve(ospf)#no trap neighbor-state-change-trap
    ```

3. To reinstate the trap, enter the following command:

    ```
    ProCurve(ospf)# trap neighbor-state-change-trap
    ```

Syntax: [no] snmp-server trap ospf < *ospf-trap* >

Modifying OSPF Standard Compliance Setting

N o t e   All routes in an AS should be configured with the same compliance setting. If any routers in a domain support only RFC 1583, then all routers must be configured with 1583 compatibility.

If all the routers support RFC 2178 or RFC 2328, you should disable RFC 1583 compatibility on all the routers in the domain, since these standards are more robust against routing loops on external routes.

The switch is configured, by default, to be compliant with the RFC 1583 OSPF V2 specification.

To configure a switch to operate with the latest OSPF standard, RFC 2328, enter the following commands:

```
ProCurve(config)# router ospf
ProCurve(ospf)# no rfc1583-compatibility
```

Syntax: [no] rfc1583-compatibility

Displaying OSPF Information

You can use CLI commands to display the following OSPF information:

OSPF Information Type	Page
General Information	11-56
Area information	11-58
External link state information	11-59
Interface information	11-60
Link state information	11-63
Neighbor information	11-65
Route information	11-71
Virtual Neighbor information	11-68
Virtual Link information	11-69

Displaying General OSPF Configuration Information

To display general OSPF configuration information, enter **show ip ospf general** at any CLI level:

```
ProCurve# show ip ospf general

OSPF General Status

  OSPF protocol         : enabled
  Router ID             : 10.0.8.36
  RFC 1583 compatibility : compatible

  Default import metric     : 1
  Default import metric type : external type 2

  Area Border               : yes
  AS Border                 : yes
  External LSA Count        : 9
  External LSA Checksum Sum : 408218
  Originate New LSA Count   : 24814
  Receive New LSA Count     : 14889
```

Figure 11-10.Example of Show IP OSPF General Output

Syntax: show ip ospf general

The following fields are shown in the OSPF general status display:

Table 11-6. CLI Display of OSPF General Information

This Field...	Displays...
OSPF protocol	indicates whether OSPF is currently enabled.
Router ID	the Router ID that this routing switch is currently using to identify itself
RFC 1583 compatibility	indicates whether the routing switch is currently using RFC 1583 (compatible) or RFC 2328 (non-compatible rules for calculating external routes.
Default import metric	indicates the default metric that will be used for any routes redistributed into OSPF by this routing switch
Default import metric type	indicates the metric type (type 1 or type 2) that will be used for any routes redistributed into OSPF by this routing switch
Area Border	indicates whether this routing switch is currently acting as an area border router
AS Border	indicates whether this routing switch is currently acting as an autonomous system border router (redistributing routes)
External LSA Count	indicates the total number of external LSAs currently in the routing switch's link state database
External LSA Checksum Sum	the sum of the checksums of all external LSAs currently in the routing switch's link state database (quick check for whether database is in sync with other routers in the routing domain)
Originate New LSA Count	count of the number of times this switch has originated a new LSA
Receive New LSA Count	count of the number of times this switch has received a new LSA

Displaying OSPF Area Information

To display OSPF area information, enter **show ip ospf area** at any CLI level:

```
ProCurve> show ip ospf area

OSPF Area Information

  Area ID          Type   Cost  SPFR   ABR  ASBR  LSA   Checksum

  --------------- ------ ----- ------ ---- ---- ----- ----------
  0.0.0.0          normal 0     1      0    0    1     0x0000781f
  192.147.60.0     normal 0     1      0    0    1     0x0000fee6
  192.147.80.0     stub   1     1      0    0    2     0x000181cd
```

Figure 11-11.Example of Show IP OSPF Area Output

Syntax: show ip ospf area [*ospf-area-id*]

The [***ospf-area-id***] parameter shows information for the specified area. If no area is specified, information for all the OSPF areas configured is displayed.

The OSPF area display shows the following information:

Table 11-7. CLI Display of OSPF Area Information

This Field...	Displays...
Area ID	The identifier for this area.
Type	The area type, which can be either "normal" or "stub".
Cost	The metric for the default route that the routing switch will inject into a stub area if the routing switch is an ABR for the area. This value only applies to stub areas.
SPFR	The number of times the routing switch has run the shortest path first route calculation for this area.
ABR	The number of area border routers in this area.
ASBR	The number of autonomous system border routers in this area.
LSA	The number of LSAs in the link state database for this area.
Chksum(Hex)	The sum of the checksums of all LSAs currently in the area's link state database. This value can be compared to the value for other routers in the area to verify database synchronization.

Displaying OSPF External Link State Information

To display external link state information, enter **show ip ospf external-link-state** at any CLI level. When you enter this command, an output similar to the following is displayed:

```
ProCurve# show ip ospf external-link-state

Link State ID    Router ID        Age   Sequence #   Checksum
  --------------  --------------  ----  -----------  ----------
   10.3.7.0         10.0.8.37       232   0x80000005   0x0000d99f
   10.3.8.0         10.0.8.37       232   0x80000005   0x0000cea9
   10.3.9.0         10.0.8.37       232   0x80000005   0x0000c3b3
   10.3.10.0        10.0.8.37       232   0x80000005   0x0000b8bd
   10.3.33.0        10.0.8.36      1098  0x800009cd   0x0000b9dd
```

Figure 11-12.Example of Show IP OSPF External-Link-State Output

Syntax: show ip ospf external-link-state

The OSPF external link state display shows the following information:

Table 11-8. CLI Display of OSPF External Link State Information

This Field...	Displays...
Link State ID	LSA ID for this LSA. Normally, the destination of the external route, but may have some "host" bits set.
Router ID	Router ID of the router that originated this external LSA.
Age	Current age (in seconds) of this LSA.
Sequence #	Sequence number of the current instance of this LSA.
Chksum(Hex)	LSA checksum value.

Syntax: show ip ospf external-link-state [status | advertise] [link-state-id < *link-state-id* > | router-id < *router-id* > | sequence-number < *sequence#* >]

The **status** keyword is optional and can be omitted. The output can be filtered to show a subset of the total output by specifying the **link-state-id**, **router-id**, or **sequence-number** options.

The **advertise** keyword displays the hexadecimal data in the specified LSA packet, the actual contents of the LSAs. This can also be filtered as above by including the **link-state-id**, **router-id**, or **sequence-number** options.

```
ProCurve# show ip ospf external-link-state advertise

OSPF External LSAs

 Advertisements
 ------------------------------------------------------------------------
 000302050a0307000a00082580000005d99f0024ffffff008000000a0000000000000000
 000302050a0308000a00082580000005cea90024ffffff008000000a0000000000000000
 000302050a0309000a00082580000005c3b30024ffffff008000000a0000000000000000
 000302050a030a000a00082580000005b8bd0024ffffff008000000a0000000000000000
 000002050a0321000a000824800009cdb9dd0024ffffff00800000010000000000000000
```

Figure 11-13.Example of the Output for Show IP OSPF External-Link-State Advertise

Displaying OSPF Interface Information

To display OSPF interface information, enter **show ip ospf interface** at any CLI level:

```
ProCurve# show ip ospf interface

OSPF Interface Status

 IP Address      Status    Area ID          State    Auth-type Cost   Priority
 --------------- --------  ---------------  -------  --------- ------ ---------
 10.3.18.36      enabled   10.3.16.0        BDR      none      1      1
 10.3.53.36      enabled   10.3.48.0        BDR      none      1      1
```

Figure 11-14.Example of the Output for Show IP OSPF Interface

Syntax: show ip ospf interface [vlan < *vlan-id* > | < *ip-addr* >]

The OSPF interface display shows the following information:

Table 11-9. CLI Display of OSPF Interface Information

This Field...	Displays...
IP Address	The local IP address for this interface.
Status	enabled or disabled - whether OSPF is currently enabled on this interface.
Area ID	The ID of the area that this interface is in.
State	The current state of the interface. The value will be one of the following: • DOWN - the underlying VLAN is down • WAIT - the underlying VLAN is up, but we are waiting to hear hellos from other routers on this interface before we run designated router election • DR - this switch is the designated router for this interface • BDR - this switch is the backup designated router for this interface • DROTHER - this router is not the designated router or backup designated router for this interface
Auth-type	none or simple - will be none if no authentication key is configured, simple if an authentication key is configured. All routers running OSPF on the same link must be using the same authentication type and key.
Cost	The OSPF's metric for this interface.
Priority	This routing switch's priority on this interface for use in the designated router election algorithm.

The **< *ip-addr* >** parameter displays the OSPF interface information for the specified IP address.

Displaying OSPF Interface Information for a Specific VLAN or IP Address

To display OSPF interface information for a specific VLAN or IP address, enter **show ip ospf interface < *ip-addr* >** at any CLI level. For example:

```
ProCurve# show ip ospf interface 10.3.18.36

OSPF Interface Status for 10.3.18.36

  IP Address      : 10.3.18.36          Status  : enabled
  Area ID         : 10.3.16.0

  State  : BDR                          Auth-type : none
  Cost   : 1                            Priority  : 1
  Type   : BCAST

  Transit Delay     : 1                 Retrans Interval  : 5
  Hello Interval    : 10                Rtr Dead Interval : 40
  Designated Router : 10.3.18.34        Events            : 3
  Backup Desig. Rtr : 10.3.18.36
```

Figure 11-15.Example of Show IP OSPF Interface < *ip-addr* > Output

Syntax: show ip ospf interface [vlan < *vlan-id* > | < *ip-addr* >]

The OSPF interface display for a specific VLAN or IP address has the same information as the non-specific show ip ospf interface command for the **IP Address**, **Area ID**, **Status**, **State**, **Auth-type**, **Cost**, and **Priority** fields. See the information for the general command above for definitions of these fields.

The show ip ospf interface command for a specific VLAN or IP address shows the following additional information:

Table 11-10. CLI Display of OSPF Interface Information – VLAN or IP Address

This Field...	Displays...
Type	Will always be **BCAST** for interfaces on this routing switch. Point-to-point or NBMA (frame relay or ATM) type interfaces are not supported on the 5300xl, 3400cl, and 6400cl switches.
Transit Delay	Configured transit delay for this interface.
Retrans Interval	Configured retransmit interval for this interface.

This Field...	Displays...
Hello Interval	Configured hello interval for this interface.
Rtr Dead Interval	Configured router dead interval for this interface.
Designated Router	IP address of the router that has been elected designated router on this interface.
Backup Desig. Rtr	IP address of the router that has been elected backup designated router on this interface.
Events	Number of times the interface state has changed.

If you issue a **show ip ospf interface vlan <vlan-id>** command, the information will be the same as shown in the previous table, but for the IP address on the indicated VLAN.

Displaying OSPF Link State Information

To display OSPF link state information, enter **show ip ospf link-state** at any CLI level. When you enter this command, the switch displays an output similar to the following:

```
OSPF Link State Database for Area 0.0.0.0
                          Advertising
 LSA Type    Link State ID  Router ID       Age  Sequence #  Checksum
 ---------- --------------- --------------- ---- ----------- ----------
 Router      10.0.8.32      10.0.8.32       65   0x80000281  0x0000a7b6
 Router      10.0.8.33      10.0.8.33       1638 0x80000005  0x0000a7c8
 Network     10.3.2.37      10.0.8.37       1695 0x80000006  0x00000443
 Summary     10.3.16.0      10.0.8.33       1638 0x80000007  0x0000c242
 Summary     10.3.16.0      10.0.8.35       1316 0x80000008  0x0000aa58
 Summary     10.3.17.0      10.0.8.33       1638 0x8000027b  0x0000becf
 Summary     10.3.17.0      10.0.8.35       1316 0x80000008  0x0000a957
 AsbSummary 10.0.8.36       10.0.8.33       1412 0x80000002  0x00002cba

OSPF Link State Database for Area 10.3.16.0
                          Advertising
 LSA Type    Link State ID  Router ID       Age  Sequence #  Checksum
 ---------- --------------- --------------- ---- ----------- ----------
 Router      10.0.8.33      10.0.8.33       1727 0x8000027e  0x0000d53c
 Router      10.0.8.34      10.0.8.34       1420 0x80000283  0x0000de4f
 Network     10.3.16.34     10.0.8.34       1735 0x80000005  0x00001465
```

Figure 11-16. Example of Show IP OSPF Link-State Output

Syntax: show ip ospf link-state

The OSPF link state display shows contents of the LSA database, one table for each area. The following information is shown:

Table 11-11. CLI Display of OSPF Link State Information

This Field...	Displays...
LSA Type	Type of LSA. The possible types are: Router Network Summary AsbSummary
Link State ID	LSA ID for this LSA. The meaning depends on the LSA type.
Advertised Router ID	Router ID of the router that originated this LSA.
Age	Current age (in seconds) of this LSA.
Sequence #	Sequence number of the current instance of this LSA.
Chksum(Hex)	LSA checksum value.

Other options for this command: The **status** keyword is optional and can be omitted. The output can be filtered to show a subset of the total output by specifying the **area-id**, **link-state-id**, **router-id**, **LSA type**, or **sequence-number** options.

The **advertise** keyword displays the hexadecimal data in the specified LSA packet, the actual contents of the LSAs. This can also be filtered as above by including the **area-id**, **link-state-id**, **router-id**, **LSA type**, or **sequence-number** options.

The full syntax of the command is:

Syntax: show ip ospf link-state [status | advertise] [< *area-id* > | link-state-id < *link-state-id* > | router-id < *router-id* > | type < router | network | summary | as-summary > | sequence-number < *sequence#* >]

An example of the **show ip ospf link-state advertise** is:

```
OSPF Link State Database for Area 0.0.0.0

 Advertisements
 -------------------------------------------------------------------------
 000202010a0008200a00082080000281a7b60054000000050a030e00ffffff0003000001...
 000202010a0008210a00082180000006a5c90024010000010a0008230a03112104000002
 000102010a0008230a00082380000015755d006c010000070a030600ffffff0003000001...
 000202020a0302250a000825800000070244024ffffff000a0008250a0008230a000820
 000202030a0310000a00082180000008c043001cffffff0000000002
 000102030a0310000a00082380000009a859001cffffff0000000001
 000002030a0310000a00082480000009ac53001cffffff0000000002
 000202040a0008240a000821800000032abb001c000000000000000b
 000102040a0008240a00082380000004c12a001c0000000000000002

OSPF Link State Database for Area 10.3.16.0

 Advertisements
 -------------------------------------------------------------------------
 000202010a0008210a0008218000027fd33d0054050000050a031900ffffff0003000001...
 000102010a0008220a00082280000284dc50006000000060a031500ffffff0003000001...
 000102020a0311220a0008228000027bf9080020ffffff000a0008220a000821
```

Figure 11-17. Example of the Output for Show IP OSPF Link-State Advertise

Displaying OSPF Neighbor Information

To display OSPF neighbor information, enter **show ip ospf neighbor** at any CLI level:

```
OSPF Neighbor Information

 Router ID       Pri IP Address       NbIfState State    Rxmt QLen Events
 --------------- --- --------------- --------- -------- --------- ----------
 10.0.8.34       1   10.3.18.34       DR        FULL     0         6
 10.3.53.38      1   10.3.53.38       DR        FULL     0         6
```

Figure 11-18. Example of Show IP OSPF Neighbor Output

Syntax: show ip ospf neighbor [*ip-addr*]

The **[*ip-addr*]** can be specified to retrieve detailed information for the specific neighbor only. This is the IP address of the neighbor, not the Router ID.

This display shows the following information.

Table 11-12. CLI Display of OSPF Neighbor Information

Field	Description
Router ID	The router ID of the neighbor.
Pri	The OSPF priority of the neighbor. The priority is used during election of the Designated Router (DR) and Backup designated Router (BDR).
IP Address	The IP address of this routing switch's interface with the neighbor.
NbIfState	The neighbor interface state. The possible values are: • **DR** – this neighbor is the elected designated router for the interface. • **BDR** – this neighbor is the elected backup designated router for the interface. • ***blank*** – this neighbor is neither the DR or the BDR for the interface.
State	The state of the conversation (the adjacency) between your routing switch and the neighbor. The possible values are: • **INIT** – A Hello packet has recently been seen from the neighbor. However, bidirectional communication has not yet been established with the neighbor. (The switch itself did not appear in the neighbor's Hello packet.) All neighbors in this state (or higher) are listed in the Hello packets sent from the associated interface. • **2WAY** – Communication between the two routers is bidirectional. This is the most advanced state before beginning adjacency establishment. The Designated Router and Backup Designated Router are selected from the set of neighbors in the 2Way state or greater. • **EXSTART** – The first step in creating an adjacency between the two neighboring routers. The goal of this step is to decide which router is the master, and to decide upon the initial Database Description (DD) sequence number. Neighbor conversations in this state or greater are called adjacencies. • **EXCHANGE** – The switch is describing its entire link state database by sending Database Description packets to the neighbor. Each Database Description packet has a DD sequence number, and is explicitly acknowledged. Only one Database Description packet can be outstanding at any time. In this state, Link State Request packets can also be sent asking for the neighbor's more recent advertisements. All adjacencies in Exchange state or greater are used by the flooding procedure. In fact, these adjacencies are fully capable of transmitting and receiving all types of OSPF routing protocol packets. • **LOADING** – Link State Request packets are sent to the neighbor asking for the more recent advertisements that have been discovered (but not yet received) in the Exchange state. • **FULL** – The neighboring routers are fully adjacent. These adjacencies will now appear in router links and network link advertisements.
`Rxmt QLen`	Remote transmit queue length – the number of LSAs that the routing switch has sent to this neighbor and for which the routing switch is awaiting acknowledgements.
Events	The number of times the neighbor's state has changed.

Displaying OSFPF Redistribution Information

As described under "Enabling Route Redistribution" on page 11-53, you can configure the routing switch to redistribute connected and static routes into OSPF. When you redistribute a route into OSPF, the routing switch can use OSPF to advertise the route to its OSPF neighbors.

To display the status of the OSPF redistribution, enter **show ip ospf redistribute** at any CLI context level:

```
ProCurve# show ip ospf redistribute

OSPF redistributing

  Route type Status
  ---------- --------
  connected  enabled
  static     enabled
```

Figure 11-19.Example of Output for Show IP OSPF Redistribute

The display shows whether redistribution of each of the route types, connected and static is enabled.

Displaying OSFPF Redistribution Filter (restrict) Information

As described under "Defining Redistribution Filters" on page 11-51, you can configure the redistribution filters on the routing switch to restrict route redistribution by OSPF.

To display the status of the OSPF redistribution filters, enter **show ip ospf restrict** at any CLI context level.

```
ProCurve# show ip ospf restrict

OSPF restrict list

  IP Address      Mask
  --------------- ---------------
  10.0.8.0        255.255.248.0
  15.0.0.0        255.0.0.0
```

Figure 11-20.Example of Output for Show IP OSPF Restrict

This display shows the configured restrict entries.

Displaying OSPF Virtual Neighbor Information

To display OSPF virtual neighbor information, enter **show ip ospf virtual-neighbor** at any CLI level.

```
OSPF Virtual Interface Neighbor Information

 Router ID        Area ID         State    IP Address       Events
 --------------- --------------- -------- --------------- --------
 10.0.8.33        10.3.16.0       FULL     10.3.17.33       5
 10.0.8.36        10.3.16.0       FULL     10.3.18.36       5
```

Figure 11-21.Example of Output for Show IP OSPF Virtual-Neighbor

Syntax: show ip ospf virtual-neighbor [area < *area-id* > | < *ip-address* >]

This display shows the following information.

Table 11-13. CLI Display of OSPF Virtual Neighbor Information

Field	Description
Router ID	The router ID of this virtual neighbor (configured).
Area ID	The area ID of the transit area for the virtual link to this neighbor (configured).
State	The state of the adjacency with this virtual neighbor. The possible values are the same as the OSPF neighbor states. See the State parameter definition in table 11-12 on page 11-66. Note that virtual neighbors should never stay in the 2WAY state.
IP Address	IP address of the virtual neighbor that the routing switch is using to communicate to that virtual neighbor.
Events	The number of times the virtual neighbor's state has changed.

Notice from the syntax statement that you can get OSPF virtual neighbor information for a specific area or a specific IP address.

Displaying OSPF Virtual Link Information

To display OSPF virtual link information, enter **show ip ospf virtual-link** at any CLI level.

```
ProCurve# show ip ospf virtual-link

OSPF Virtual Interface Status

  Transit AreaID   Neighbor Router Authentication  Interface State
  ---------------  --------------- ---------------  ---------------
  10.3.16.0        10.0.8.33       none             P2P
  10.3.16.0        10.0.8.36       none             P2P
```

Figure 11-22.Example of Output for Show IP OSPF Virtual-Link

Syntax: show ip ospf virtual-link [area < *area-id* > | < *ip-address* >]

This display shows the following information.

Table 11-14. CLI Display of OSPF Virtual Link Information

Field	Description
Transit Area ID	Area ID of transit area for the virtual link.
Neighbor Router	Router ID of the virtual neighbor.
Authentication	none or simple (same as for normal interface).
Interface State	The state of the virtual link to the virtual neighbor. The possible values are: • DOWN – the routing switch has not yet found a route to the virtual neighbor. • P2P – (point-to-point) the routing switch has found a route to the virtual neighbor. Virtual links are "virtual" serial links, hence the point-to-point terminology.

Notice from the syntax statement that you can get OSPF virtual link information for a specific area or a specific IP address.

Example: To get OSPF virtual link information for IP address 10.0.8.33, enter **show ip ospf virtual-link 10.0.8.33**. A display similar to the following is shown.

```
ProCurve# show ip ospf virtual-link 10.0.8.33

OSPF Virtual Interface Status for interface 10.0.8.33

  Transit AreaID  : 10.3.16.0
  Neighbor Router : 10.0.8.33

  Authentication  : none              Transit Delay  : 1
  Interface State : P2P               Rtr Interval   : 5
  Events          : 1                 Hello Interval : 10
                                      Dead Interval  : 40
```

Figure 11-23.Example of Output for Show IP OSPF Virtual-Link < *ip-addr* >

In this display, these fields show the same type of information as described for the general OSPF virtual link display: **Transit Area ID**, **Neighbor Router**, **Authentication**, and I**nterface State**. This display shows the following additional information:

Table 11-15. CLI Display of OSPF Virtual Link Information – Specific IP Address

Field	Description
Events	The number of times the virtual link interface state has changed.
Transit delay	The configured transit delay for the virtual link.
Rtr Interval	The configured retransmit interval for the virtual link.
Hello Interval	The configured hello interval for the virtual link.
Dead Interval	The configured router dead interval for the virtual link

Displaying OSPF Route Information

To display OSPF route and other OSPF configuration information, enter show ip ospf at any CLI level:

```
ProCurve# show ip ospf

OSPF Configuration Information

  OSPF protocol  : enabled
  Router ID      : 10.0.8.35

 Currently defined areas:

                        Stub          Stub          Stub
  Area ID       Type    Default Cost  Summary LSA   Metric Type
  --------------- ------ ------------- ------------ ---------------
  backbone      normal 1              don't send   ospf metric
  10.3.16.0     normal 1              don't send   ospf metric
  10.3.32.0     normal 1              don't send   ospf metric

 Currently defined address ranges:

  Area ID       LSA Type   IP Network      Network Mask    Advertise
  --------------- ---------- --------------- --------------- ---------
  10.3.16.0     Summary    10.3.16.0       255.255.255.0   yes

 OSPF interface configuration:

                                Admin         Authen
  IP Address    Area ID         Status  Type  Type   Cost  Pri
  --------------- --------------- ----- ------ ------ ----- ---
  10.3.2.35     backbone        enabled BCAST none   1     1
  10.3.3.35     backbone        enabled BCAST none   1     1
  10.3.16.35    10.3.16.0       enabled BCAST none   1     1
  10.3.32.35    10.3.32.0       enabled BCAST none   1     1

 OSPF configured interface timers:

                 Transit Retransmit Hello    Dead
  IP Address     Delay   Interval   Interval Interval
  --------------- ------- ---------- --------- ----------
  10.3.2.35      1       5          10        40
  10.3.3.35      1       5          10        40
  10.3.16.35     1       5          10        40
  10.3.32.35     1       5          10        40

 OSPF configured virtual interfaces:

                                Authen Xmit   Rxmt   Hello  Dead
  Area ID         Router ID     Type   Delay  Intvl  Intvl  Interval
  --------------- --------------- ------ ------ ------ ------ ----------
  10.3.16.0     10.0.8.33       none   1      5      10     40
  10.3.16.0     10.0.8.36       none   1      5      10     40
```

Figure 11-24.Example of Output for Show IP OSPF

Syntax: show ip ospf

This screen has a lot of information, most of it already covered in other show commands. The following table shows definitions for the fields:

Table 11-16. CLI Display of OSPF Route and Status Information

Field	Description
OSPF protocol	**enabled** or **disabled** – indicates if OSPF is currently enabled.
Router ID	The Router ID that this routing switch is currently using to identify itself.
Currently Defined Areas:	
Area ID	The identifier for this area.
Type	The type of OSPF area (normal or stub).
Stub Default Cost	The metric for any default route we will inject into a stub area if we are an ABR for the area. This value only applies to stub areas.
Stub Summary LSA	**send** or **don't send** – indicates the state of the no-summary option for the stub area. The value indicates if the area is "totally stubby" (no summaries sent from other areas) or just "stub" (summaries sent). Only applies to stub areas, and only takes effect if the routing switch is the ABR for the area.
Stub Metric Type	This value is always **ospf metric**.
Currently defined address ranges:	
Area ID	The area where the address range is configured.
LSA Type	This value is always **Summary**.
IP Network	The address part of the address range specification.
Network Mask	The mask part of the address range specification.
Advertise	Whether we are advertising (**yes**) or suppressing (**no**) this address range.

N o t e   The remaining interface and virtual link information is the same as for the previously described OSPF show commands.

OSPF Equal-Cost Multipath (ECMP) for Different Subnets Available Through the Same Next-Hop Routes

Using software prior to release E.10.*xx*, if different subnet destinations in an OSPF network are reachable through a set of equal-cost next-hop routes, the router chooses the same next-hop route for traffic to all of these destinations. Beginning with software release E.10.*xx*, 5300xl routers support optional load-sharing across redundant links where the network offers two, three, or four equal-cost next-hop routes for traffic to different subnets. (All traffic for different hosts in the same subnet goes through the same next-hop router.)

For example, in the OSPF network shown below, IP load-sharing is enabled on router "A". In this case, OSPF calculates three equal-cost next-hop routes for each of the subnets and then distributes per-subnet route assignments across these three routes.
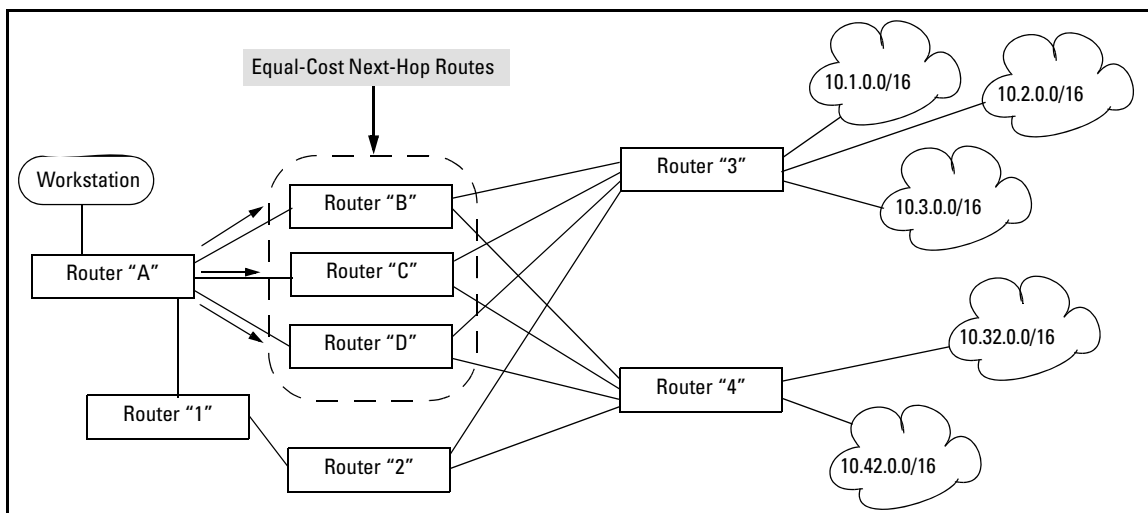
Figure 11-25.Example of Load-Sharing Traffic to Different Subnets Through Equal-Cost Next-Hop Routers

Table 11-17. Example of a Routing Table for the Network in Figure 11-25

Destination Subnet	Router "A" Next Hop
10.1.0.0/16	Router "C"
10.2.0.0/16	Router "D"
10.3.0.0/16	Router "B"
10.32.0.0/16	Router "B"
10.42.0.0/16	Router "D"

Note that IP load-sharing does not affect routed traffic to different hosts on the same subnet. That is, all traffic for different hosts on the same subnet will go through the same next-hop router. For example, if subnet 10.32.0.0 includes two servers at 10.32.0.11 and 10.32.0.22, then all traffic from router "A" to these servers will go through router "B".

Syntax: [no] ip lload-sharing < 2 - 4 >

When OSPF is enabled and multiple, equal-cost, next-hop routes are available for traffic destinations on different subnets, this feature, by default, enables load-sharing among up to four next-hop routes. The **no** *form of the command disables this load-sharing so that only one route in a group of multiple, equal-cost, next-hop routes is used for traffic that could otherwise be load-shared across multiple routes. For example, in figure11-25 on page <zBlue>73, the next-hop routers "B", "C", and "D" are available for equal-cost load-sharing of eligible traffic. Disabling IP load-sharing means that router "A" selects only one next-hop router for traffic that is actually eligible for load-sharing through different next-hop routers. (Default: Enabled with four equal-cost, next-hop routes allowed)*

__Note:__ In the default configuration, IP load-sharing is enabled by default. However, it has no effect unless IP routing and OSPF are enabled.
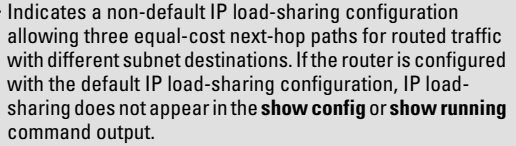
< 1 - 4 >

Specifies the maximum number of equal-cost next hop paths the router allows. (Range: __2__ - __4__; Default: __4__)

Displaying the Current IP Load-Sharing Configuration

Use the **show running** command to view the currently active IP load-sharing configuration, and **show config** to view the IP load-sharing configuration in the startup-config file. (While in its default configuration, IP load-sharing does not appear in the command output.) If IP load sharing is configured with non-default settings (disabled or configured for either two or three equal-cost next-hop paths), then the current settings are displayed in the command output.

```
ProCurve Switch 5304XL(config)# show running

Running configuration:

; J4850A Configuration Editor; Created on
release #E.10.00

hostname "ProCurve Switch 5304xl"
module 1 type J4820A
snmp-server community "public" Unrestricted
vlan 1
    name "DEFAULT_VLAN"
    untagged A1-A24
    ip address dhcp-bootp
    exit
ip load-sharing 3
access-controller vlan-base 2000
```

Indicates a non-default IP load-sharing configuration allowing three equal-cost next-hop paths for routed traffic with different subnet destinations. If the router is configured with the default IP load-sharing configuration, IP load-sharing does not appear in the **show config** or **show running** command output.

Figure 11-26.Displaying a Non-Default IP Load-Sharing Configuration

Configuring IRDP

The ICMP Router Discovery Protocol (IRDP) is used by ProCurve routing switches to advertise the IP addresses of its router interfaces to directly attached hosts. IRDP is enabled by default. You can enable the feature on a global basis or on an individual VLAN interface basis.

When IRDP is enabled, the routing switch periodically sends Router Advertisement messages out the IP interfaces on which the feature is enabled. The messages advertise the routing switch's IP addresses to directly attached hosts who listen for the messages. In addition, hosts can be configured to query the routing switch for the information by sending Router Solicitation messages.

Some types of hosts use the Router Solicitation messages to discover their default gateway. When IRDP is enabled on the ProCurve routing switch, the routing switch responds to the Router Solicitation messages. Some clients interpret this response to mean that the routing switch is the default gateway. If another router is actually the default gateway for these clients, leave IRDP disabled on the ProCurve routing switch.

IRDP uses the following parameters. If you enable IRDP on individual VLAN interfaces, you can configure these parameters on an individual VLAN interface basis.

■ **Packet type** - The routing switch can send Router Advertisement messages as IP broadcasts or as IP multicasts addressed to IP multicast group 224.0.0.1. The default packet type is IP broadcast.

■ **Hold time** - Each Router Advertisement message contains a hold time value. This value specifies the maximum about of time the host should consider an advertisement to be valid until a newer advertisement arrives. When a new advertisement arrives, the hold time is reset. The hold time is always longer than the maximum advertisement interval. Therefore, if the hold time for an advertisement expires, the host can reasonably conclude that the router interface that sent the advertisement is no longer available. The default hold time is three times the maximum message interval.

■ **Maximum message interval and minimum message interval** - when IRDP is enabled, the routing switch sends the Router Advertisement messages every 450-600 seconds by default. The time within this interval that the routing switch selects is random for each message and is not affected by traffic loads or other network factors. The random interval minimizes the probability that a host will receive Router Advertisement

messages from other routers at the same time. The interval on each IRDP-enabled routing switch interface is independent of the interval on other IRDP-enabled interfaces. The default maximum message interval is 600 seconds. The default minimum message interval is 450 seconds.

■ **Preference** - If a host receives multiple Router Advertisement messages from different routers, the host selects the router that send the message with the highest preference as the default gateway. The preference can be a number from -4294967296 to 4294967295. The default is 0.

Enabling IRDP Globally

To enable IRDP globally, enter the following command:

```
ProCurve(config)# ip irdp
```

This command enables IRDP on the IP interfaces on all ports. Each port uses the default values for the IRDP parameters.

Enabling IRDP on an Individual VLAN Interface

To enable IRDP on an individual VLAN interface and configure IRDP parameters, enter commands such as the following:

```
ProCurve(config)# vlan 1
ProCurve(vlan-1)# ip irdp maxadvertinterval 400
```

This example shows how to enable IRDP on a specific interface (VLAN 1) and change the maximum advertisement interval for Router Advertisement messages to 400 seconds.

Syntax: [no] ip irdp [broadcast | multicast] [holdtime <seconds>] [maxadvertinterval < *seconds* >] [minadvertinterval < *seconds* >] [preference < *number* >]

■ **broadcast | multicast** - This parameter specifies the packet type the routing switch uses to send the Router Advertisement.

• **broadcast** - The routing switch sends Router Advertisements as IP broadcasts.

• **multicast** - The routing switch sends Router Advertisements as multicast packets addressed to IP multicast group 224.0.0.1. This is the default.

■ **holdtime < *seconds* >** - This parameter specifies how long a host that receives a Router Advertisement from the routing switch should consider the advertisement to be valid. When a host receives a new Router Advertisement message from the routing switch, the host resets the hold time

for the routing switch to the hold time specified in the new advertisement. If the hold time of an advertisement expires, the host discards the advertisement, concluding that the router interface that sent the advertisement is no longer available. The value must be greater than the value of the maxadvertinterval parameter and cannot be greater than 9000. The default is three times the value of the maxadvertinterval parameter.

■ **maxadvertinterval** - This parameter specifies the maximum amount of time the routing switch waits between sending Router Advertisements. You can specify a value from 1 to the current value of the holdtime parameter. The default is 600 seconds.

■ **minadvertinterval** - This parameter specifies the minimum amount of time the routing switch can wait between sending Router Advertisements. The default is three-fourths (0.75) the value of the maxadvertinterval parameter. If you change the maxadvertinterval parameter, the software automatically adjusts the minadvertinterval parameter to be three-fourths the new value of the maxadvertinterval parameter. If you want to override the automatically configured value, you can specify an interval from 1 to the current value of the maxadvertinterval parameter.

■ **preference < *number* >** - This parameter specifies the IRDP preference level of this routing switch. If a host receives Router Advertisements from multiple routers, the host selects the router interface that sent the message with the highest preference as the host's default gateway. The valid range is -4294967296 to 4294967295. The default is 0.

Displaying IRDP Information

To display IRDP information, enter show ip irdp from any CLI level.

```
ProCurve# show ip irdp

Status and Counters - ICMP Router Discovery Protocol

  Global Status : Disabled

  VLAN Name       Status   Advertising   Min int Max int Holdtime Preference
                            Address      (sec)   (sec)   (sec)
  --------------- -------- ------------ ------- ------- -------- -----------
  DEFAULT_VLAN    Enabled  multicast     450     600     1800     0
  VLAN20          Enabled  multicast     450     600     1800     0
  VLAN30          Enabled  multicast     450     600     1800     0
```

Figure 11-27.Example of Output for Show IP IRDP

Configuring DHCP Relay

Overview

The Dynamic Host Configuration Protocol (DHCP) is used for configuring hosts with IP address and other configuration parameters without human intervention. The protocol is composed of three components: the DHCP client, the DHCP server, and the DHCP relay agent. The DHCP client sends broadcast request packets to the network, the DHCP servers respond with broadcast packets that offer IP parameters, such as an IP address for the client. After the client chooses the IP parameters, communication between the client and server is by unicast packets.

The function of the DHCP relay agent is to forward the DHCP messages to other subnets so that the DHCP server doesn't have to be on the same subnet as the DHCP clients. The DHCP relay agent transfers the DHCP messages from DHCP clients located on a subnet without DHCP server, to other subnets. It also relays answers from DHCP servers to DHCP clients.

DHCP Option 82

Introduction

Option 82 is called the Relay Agent Information option and is inserted by the DHCP relay agent when forwarding client-originated DHCP packets to a DHCP server. Servers recognizing the Relay Agent Information option may use the information to implement IP address or other parameter assignment policies. The DHCP Server echoes the option back verbatim to the relay agent in server-to-client replies, and the relay agent strips the option before forwarding the reply to the client.

 The "Relay Agent Information" option is organized as a single DHCP option that contains one or more "sub-options" that convey information known by the relay agent. The initial sub-options are defined for a relay agent that is co-located in a public circuit access unit. These include a "circuit ID" for the incoming circuit, and a "remote ID" which provides a trusted identifier for the remote high-speed modem.

The routing switch can operate as a DHCP relay agent to enable communication between a client and a DHCP server on a different subnet. Without Option 82, DHCP operation modifies client IP address request packets to the extent needed to forward the packets to a DHCP server. Option 82 enhances this

operation by enabling the routing switch to append an *Option 82 field* to such client requests. This field includes two suboptions for identifying the routing switch (by MAC address or IP address) and the routing switch port the client is using to access the network. A DHCP server with Option 82 capability can read the appended field and use this data as criteria for selecting the IP addressing it will return to the client through the usual DHCP server response packet. This operation provides several advantages over DHCP without Option 82:

■   An Option 82 DHCP server can use a relay agent's identity and client source port information to administer IP addressing policies based on client and relay agent location within the network, regardless of whether the relay agent is the client's primary relay agent or a secondary agent.

■   A routing switch operating as a primary Option 82 relay agent for DHCP clients requesting an IP address can enhance network access protection by blocking attempts to use an invalid Option 82 field to imitate an authorized client, or by blocking attempts to use response packets with missing or invalid Option 82 suboptions to imitate valid response packets from an authorized DHCP server.

■   An Option 82 relay agent can also eliminate unnecessary broadcast traffic by forwarding an Option 82 DHCP server response only to the port on which the requesting client is connected, instead of broadcasting the DHCP response to all ports on the VLAN.

N o t e   The routing switch's DHCP Relay Information (Option 82) feature can be used in networks where the DHCP server(s) are compliant with RFC 3046 Option 82 operation. DHCP Servers that are not compliant with Option 82 operation ignore Option 82 fields. For information on configuring an Option 82 DHCP server, refer to the documentation provided with the server application.

Some client applications can append an Option 82 field to their DHCP requests. Refer to the documentation provided for your client application.

It is not necessary for all relay agents on the path between a DHCP client and the server to support Option 82, and a relay agent without Option 82 should forward DHCP packets regardless of whether they include Option 82 fields. However, Option 82 relay agents should be positioned at the DHCP policy boundaries in a network to provide maximum support and security for the IP addressing policies configured in the server.

Option 82 Server Support

To apply DHCP Option 82, the routing switch must operate in conjunction with a server that supports Option 82. (DHCP servers that do not support Option 82 typically ignore Option 82 fields.) Also, the routing switch applies Option 82 functionality only to client request packets being *routed* to a DHCP server. DHCP relay with Option 82 does not apply to *switched* (non-routed) client requests.

For information on configuring policies on a server running DHCP Option 82, refer to the documentation provided for that application.
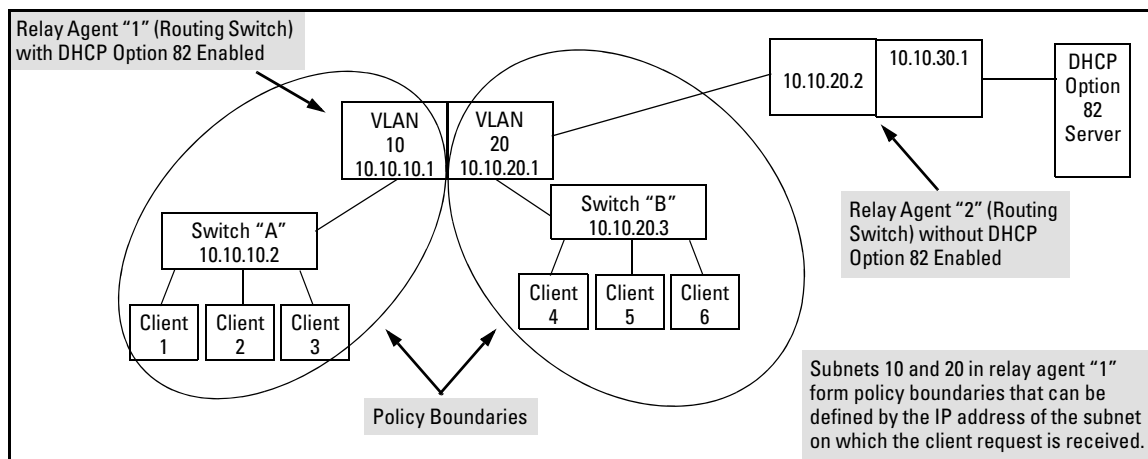
Figure 11-28.Example of a DHCP Option 82 Application

Terminology

Circuit ID: In Option 82 applications, the number of the port through which the routing switch receives a DHCP client request. On ProCurve fixed-port switches, the Circuit ID of a given port corresponds to the port number appearing on the front of the switch for that port. On ProCurve chassis switches, the port number for a given port corresponds to the internal if Index number for that port. This value is included as a suboption in an Option 82 field that the relay agent appends to a Client DHCP request before forwarding the request toward a DHCP server. (For more on Circuit ID, refer to "Circuit ID" in the bulleted list on page 11-84.)

DHCP Policy Boundary: For Option 82 applications, an area of a network as defined by connection to a given routing switch or subnet and/or a specific port belonging to the routing switch or subnet.

DHCP relay agent: See Relay Agent.

Forwarding Policy: The Option 82 method the routing switch uses to process incoming client DHCP requests. For a given inbound DHCP client request, the forwarding policy determines whether the routing switch will add Option 82 information, replace existing Option 82 information, or leave any existing information unchanged. The policy also determines whether the routing switch will forward the client request toward a DHCP server or drop the request. For a DHCP server response to an Option 82 client request, the routing switch can optionally perform a validation check to determine whether to forward or drop the response. Each Option 82 relay agent in the path between a DHCP client and an Option 82 DHCP server can be configured with a unique forwarding policy, which enhances DHCP policy control over discrete areas of a network.

Primary Relay Agent: In the path between a DHCP client and a DHCP server, the first routing switch (configured to support DHCP operation) that a client DHCP request encounters in the path from the client to a DHCP server.

Relay Agent: A routing switch that is configured to support DHCP operation.

Remote ID: In Option 82 applications on ProCurve switches, either the MAC address of a relay agent, or the IP address of a VLAN or subnet configured on a relay agent. This value is included as a suboption in an Option 82 field that the relay agent appends to a Client DHCP request before forwarding the request toward a DHCP server. (For more on Remote ID, refer to "Remote ID" in the bulleted list on page 11-84.)

Secondary Relay Agent: In the path between a DHCP client and a DHCP server, any routing switch (configured to support DHCP operation) other than the primary relay agent.

General DHCP Option 82 Requirements and Operation

Requirements. DHCP Option 82 operation is configured at the global config level and requires the following:

■ IP routing enabled on the switch

■ DHCP-Relay Option 82 enabled (global command level)

■ routing switch access to an Option 82 DHCP server on a different subnet than the clients requesting DHCP Option 82 support

■ one IP Helper address configured on each VLAN supporting DHCP clients

General DHCP-Relay Operation with Option 82. Typically, the first (primary) Option 82 relay agent to receive a client's DHCP request packet appends an Option 82 field to the packet and forwards it toward the DHCP server identified by the IP Helper address configured on the VLAN in which the client packet was received. Other, upstream relay agents used to forward the packet may append their own Option 82 fields, replace the Option 82 field(s) they find in the packet, forward the packet without adding another field, or drop the packet. (Intermediate next-hop routing switches without Option 82 capability can be used to forward—route—client request packets with Option 82 fields.) Response packets from an Option 82 server are routed back to the primary relay agent (routing switch), and include an IP addressing assignment for the requesting client and an exact copy of the Option 82 data the server received with the client request. The relay agent strips off the Option 82 data and forwards the response packet out the port indicated in the response as the Circuit ID (client access port). Under certain validation conditions described later in this section, a relay agent detecting invalid Option 82 data in a response packet may drop the packet.
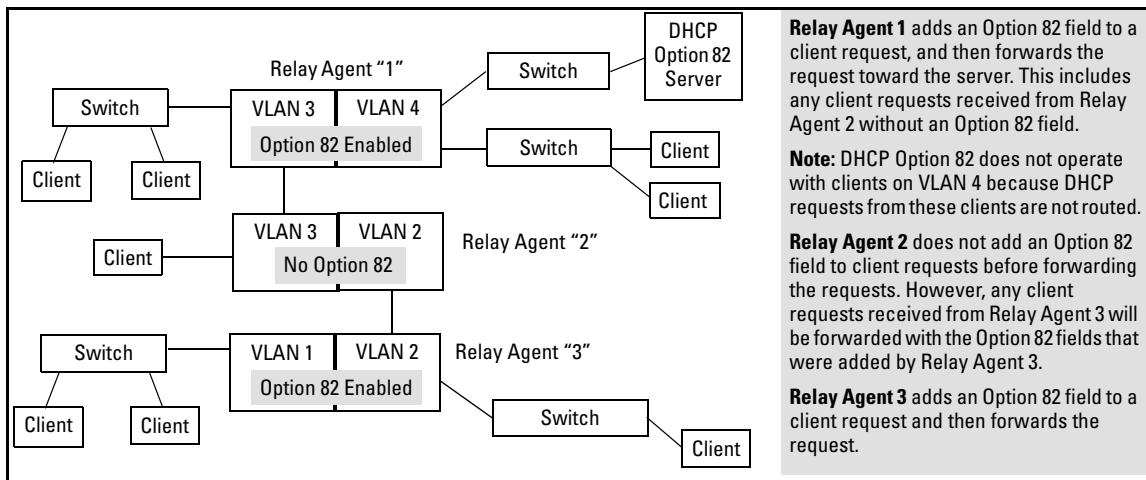
Figure 11-29.Example of DHCP Option 82 Operation in a Network with a Non-Compliant Relay Agent

Option 82 Field Content

The Remote ID and Circuit ID subfields comprise the Option 82 field a relay agent appends to client requests. A DHCP server configured to apply a different IP addressing policy to different areas of a network uses the values in these subfields to determine which DHCP policy to apply to a given client request.

- **Remote ID:** This configurable subfield identifies a policy area that comprises either the routing switch as a whole (by using the routing switch MAC address) or an individual VLAN configured on the routing switch (by using the IP address of the VLAN receiving the client request).

    - Use the IP address option if the server will apply different IP addressing policies to DHCP client requests from ports in different VLANs on the same routing switch.

    - Use the MAC address option if, on a given routing switch, it does not matter to the DHCP server which VLAN is the source of a client request (that is, use the MAC address option if the IP addressing policies supported by the target DHCP server do not distinguish between client requests from ports in different VLANs in the same routing switch)

        To view the MAC address for a given routing switch, execute the **show system-information** command in the CLI.
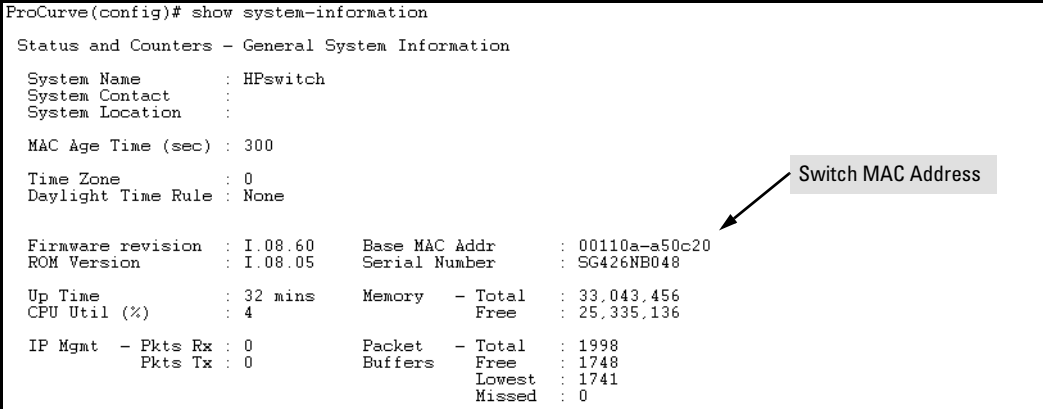
```
ProCurve(config)# show system-information

 Status and Counters - General System Information

  System Name        : HPswitch
  System Contact     :
  System Location    :

  MAC Age Time (sec) : 300

  Time Zone          : 0                                        Switch MAC Address
  Daylight Time Rule : None


  Firmware revision  : I.08.60     Base MAC Addr    : 00110a-a50c20
  ROM Version        : I.08.05     Serial Number    : SG426NB048

  Up Time            : 32 mins     Memory   - Total : 33,043,456
  CPU Util (%)       : 4                    Free    : 25,335,136

  IP Mgmt  - Pkts Rx : 0           Packet   - Total : 1998
             Pkts Tx : 0           Buffers    Free  : 1748
                                              Lowest : 1741
                                              Missed : 0
```

Figure 11-30.Using the CLI To View the Switch MAC Address

- **Circuit ID:** This non-configurable subfield identifies the port number of the physical port through which the routing switch received a given DHCP client request, and is necessary to identify if you want to configure an Option 82 DHCP server to use the Circuit ID to select a DHCP policy to assign to clients connected to the port. This number is the identity of the inbound port. On ProCurve fixed-port switches, the port number used for the Circuit ID is always the same as the physical port number shown on the front of the switch. On ProCurve chassis switches, where a dedicated, sequential block of internal port numbers are reserved for each slot, regardless of whether a slot is occupied, the circuit ID for a given port is

the sequential index number for that port position in the slot. (To view the Index number assignments for ports in the routing switch, use the **walkmib ifname** command.)

For example, the circuit ID for a client connected to port 11 on a ProCurve 2650-PWR (J8165A) switch is "11". However, the Circuit ID for port B11 on a ProCurve 5304xl (J4850A) is "37". (See Figure 11-31, below.)

```
ProCurve(config)# walkmib ifname
ifName.1 = A1
ifName.2 = A2
ifName.3 = A3
ifName.4 = A4
ifName.27 = B1
ifName.28 = B2
ifName.29 = B3
ifName.30 = B4
ifName.31 = B5
ifName.32 = B6
ifName.33 = B7
ifName.34 = B8
ifName.35 = B9
ifName.36 = B10
ifName.37 = B11
ifName.38 = B12
ifName.39 = B13
ifName.40 = B14
ifName.41 = B15
ifName.42 = B16
ifName.43 = B17
ifName.44 = B18
ifName.45 = B19
-- MORE --, next page: Space, next line: Enter, quit: Control-C
```

In this example, the 5304xl has a 4-port module installed in slot "A" and a 24-port module installed in slot "B". Thus, the first port numbers in the listing are the Index numbers reserved for slot "A". The first Index port number for slot "B" is "27", and the Index port number for port B11 (and therefore the Circuit ID number) is "37".

The Index (and Circuit ID) number for port B11 on a 5304xl routing switch.

Figure 11-31.Using Walkmib To Determine the Circuit ID for a Port on a ProCurve Chassis

For example, suppose you wanted port 10 on a given relay agent to support no more than five DHCP clients simultaneously, you could configure the server to allow only five IP addressing assignments at any one time for the circuit ID (port) and remote ID (MAC address) corresponding to port 10 on the selected relay agent.

Similarly, if you wanted to define specific ranges of addresses for clients on different ports in the same VLAN, you could configure the server with the range of IP addresses allowed for each circuit ID (port) associated with the remote ID (IP address) for the selected VLAN.

Forwarding Policies

DHCP Option 82 on ProCurve switches offers four forwarding policies, with an optional validation of server responses for three of the policy types (**append**, **replace**, or **drop**).

Table 11-18. Configuration Options for Managing DHCP Client Request Packets

Option 82 Configuration	DHCP Client Request Packet Inbound to the Routing Switch	
	Packet Has No Option 82 Field	Packet Includes an Option 82 Field
Append	Append an Option 82 Field	**Append** allows the most detail in defining DHCP policy boundaries. For example, where the path from a client to the DHCP Option 82 server includes multiple relay agents with Option 82 capability, each relay agent can define a DHCP policy boundary and append its own Option 82 field to the client request packet. The server can then determine in detail the agent hops the packet took, and can be configured with a policy appropriate for any policy boundary on the path. **Note:** In networks with multiple relay agents between a client and an Option 82 server, **append** can be used only if the server supports multiple Option 82 fields in a client request. If the server supports only one Option 82 field in a request, consider using the **keep** option.
Keep	Append an Option 82 Field	If the relay agent receives a client request that already has one or more Option 82 fields, **keep** causes the relay agent to retain such fields and forward the request without adding another Option 82 field. But if the incoming client request does not already have any Option 82 fields, the relay agent appends an Option 82 field before forwarding the request. Some applications for **keep** include: • The DHCP server does not support multiple Option 82 packets in a client request and there are multiple Option 82 relay agents in the path to the server. • The unusual case where DHCP clients in the network add their own Option 82 fields to their request packets and you do not want any additional fields added by relay agents. This policy does not include the **validate** option (described in the next section) and allows forwarding of all server response packets arriving inbound on the routing switch (except those without a primary relay agent identifier.)
Replace	Append an Option 82 Field	**Replace** replaces any existing Option 82 fields from downstream relay agents (and/or the originating client) with an Option 82 field for the current relay agent. Some applications for **replace** include: • The relay agent is located at a point in the network that is a DHCP policy boundary and you want to replace any Option 82 fields appended by down-stream devices with an Option 82 field from the relay agent at the boundary. (This eliminates downstream Option 82 fields you do not want the server to use when determining which IP addressing policy to apply to a client request.) • In applications where the routing switch is the primary relay agent for clients that may append their own Option 82 field, you can use **replace** to delete these fields if you do not want them included in client requests reaching the server.

| Option 82 Configuration | DHCP Client Request Packet Inbound to the Routing Switch | |
	Packet Has No Option 82 Field	Packet Includes an Option 82 Field
Drop	Append an Option 82 Field	**Drop** causes the routing switch to drop an inbound client request with an Option 82 field already appended. If no Option 82 fields are present, **drop** causes the routing switch to add an Option 82 field and forward the request. As a general guideline, configure **drop** on relay agents at the edge of a network, where an inbound client request with an appended Option 82 field may be unauthorized, a security risk, or for some other reason, should not be allowed.

Multiple Option 82 Relay Agents in a Client Request Path

Where the client is one router hop away from the DHCP server, only the Option 82 field from the first (and only) relay agent is used to determine the policy boundary for the server response. Where there are multiple Option 82 router hops between the client and the server, you can use different configuration options on different relay agents to achieve the results you want. This includes configuring the relay agents so that the client request arrives at the server with either one Option 82 field or multiple fields. (Using multiple Option 82 fields assumes that the server supports multiple fields and is configured to assign IP addressing policies based on the content of multiple fields.)
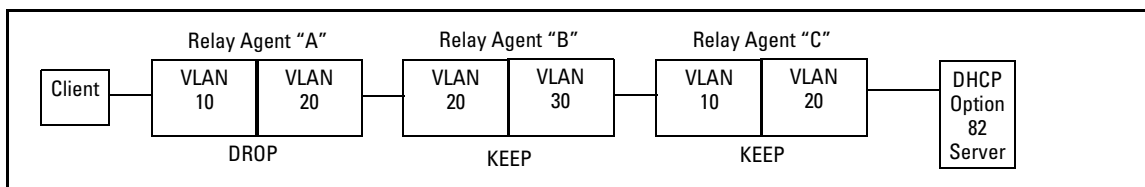
Figure 11-32.Example Configured To Allow Only the Primary Relay Agent To Contribute an Option 82 Field

The above combination allows for detection and dropping of client requests with spurious Option 82 fields. If none are found, then the drop policy on the first relay agent adds an Option 82 field, which is then kept unchanged over the next two relay agent hops ("B" and "C"). The server can then enforce an IP addressing policy based on the Option 82 field generated by the edge relay agent ("A"). In this example, the DHCP policy boundary is at relay agent 1.
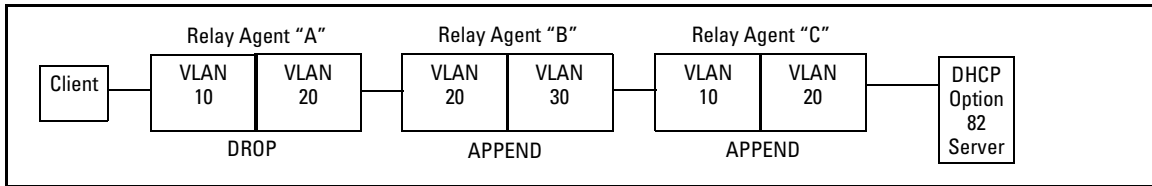
Figure 11-33.Example Configured To Allow Multiple Relay Agents To Contribute an Option 82 Field

This is an enhancement of the previous example. In this case, each hop for an accepted client request adds a new Option 82 field to the request. A DHCP server capable of using multiple Option 82 fields can be configured to use this approach to keep a more detailed control over leased IP addresses. In this example, the primary DHCP policy boundary is at relay agent "A", but more global policy boundaries can exist at relay agents "B" and "C".
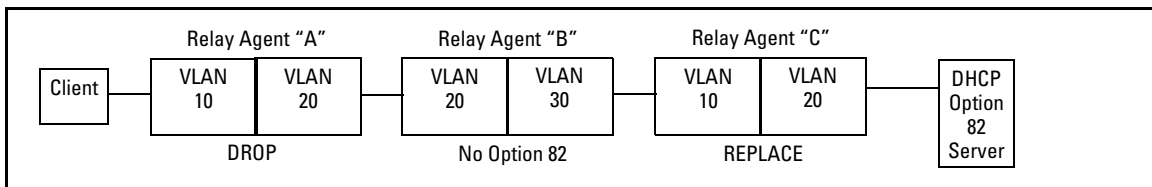
Figure 11-34.Example Allowing Only an Upstream Relay Agent To Contribute an Option 82 Field

Like the first example, above, this configuration drops client requests with spurious Option 82 fields from clients on the edge relay agent. However, in this case, only the Option 82 field from the last relay agent is retained for use by the DHCP server. In this case the DHCP policy boundary is at relay agent "C". In the previous two examples the boundary was with relay "A".

Validation of Server Response Packets

A valid Option 82 server response to a client request packet includes a copy of the Option 82 field(s) the server received with the request. With validation disabled, most variations of Option 82 information are allowed, and the corresponding server response packets are forwarded.

Server response validation is an option you can specify when configuring Option 82 DHCP for **append**, **replace**, or **drop** operation. (Refer to "Forwarding Policies" on page 11-86.) Enabling validation on the routing switch can enhance protection against DHCP server responses that are either from untrusted sources or are carrying invalid Option 82 information.

With validation enabled, the relay agent applies stricter rules to variations in the Option 82 field(s) of incoming server responses to determine whether to forward the response to a downstream device or to drop the response due to invalid (or missing) Option 82 information. Table 11-18, below, illustrates relay agent management of DHCP server responses with optional validation enabled and disabled

Table 11-19. Relay Agent Management of DHCP Server Response Packets.

Response Packet Content	Option 82 Configuration	Validation Enabled on the Relay Agent	Validation Disabled (The Default)
Valid DHCP server response packet without an Option 82 field.	**append**, **replace**, or **drop**[1]	Drop the server response packet.	Forward server response packet to a downstream device.
	keep[2]	Forward server response packet to a downstream device.	Forward server response packet to a downstream device.
The server response packet carries data indicating a given routing switch is the primary relay agent for the original client request, but the associated Option 82 field in the response contains a *Remote ID* and *Circuit ID* combination that did not originate with the given relay agent.	**append**	Drop the server response packet.	Forward server response packet to a downstream device.
	replace or **drop**[1]	Drop the server response packet.	Drop the server response packet.
	keep[2]	Forward server response packet to a downstream device.	Forward server response packet to a downstream device.
The server response packet carries data indicating a given routing switch is the primary relay agent for the original client request, but the associated Option 82 field in the response contains a *Remote ID* that did not originate with the relay agent.	**append**	Drop the server response packet.	Forward server response packet to a downstream device.
	replace or **drop**[1]	Drop the server response packet.	Drop the server response packet.
	keep[2]	Forward server response packet to a downstream device.	Forward server response packet to a downstream device.
All other server response packets[3]	**append**, **keep**[2], **replace**, or **drop**[1]	Forward server response packet to a downstream device.	Forward server response packet to a downstream device.

[1]Drop is the recommended choice because it protects against an unauthorized client inserting its own Option 82 field for an incoming request.

[2]A routing switch with DHCP Option 82 enabled with the **keep** option forwards all DHCP server response packets except those that are not valid for either Option 82 DHCP operation (compliant with RFC 3046) or DHCP operation without Option 82 support (compliant with RFC 2131).

[3]A routing switch with DHCP Option 82 enabled drops an inbound server response packet if the packet does not have any device identified as the primary relay agent (*giaddr* = null; refer to RFC 2131).

Multinetted VLANs

On a multinetted VLAN, each interface can form an Option 82 policy boundary within that VLAN if the routing switch is configured to use IP for the remote ID suboption. That is, if the routing switch is configured with IP as the remote ID option and a DHCP client request packet is received on a multinetted VLAN, the IP address used in the Option 82 field will identify the subnet on which the packet was received instead of the IP address for the VLAN. This enables an Option 82 DHCP server to support more narrowly defined DHCP policy boundaries instead of defining the boundaries at the VLAN or whole routing switch levels. If the MAC address option (the default) is configured instead, then the routing switch MAC address will be used regardless of which subnet was the source of the client request. (The MAC address is the same for all VLANs configured on the routing switch.)

Note that all request packets from DHCP clients in the different subnets in the VLAN must be able to reach any DHCP server identified by the IP Helper Address(es) configured on that VLAN.

Configuring Option 82 Operation on the Routing Switch

Syntax: dhcp-relay option 82 < append [validate] | replace [validate] | drop [validate] | keep > [ip | mac]

> **append:** *Configures the routing switch to append an Option 82 field to the client DHCP packet. If the client packet has any existing Option 82 field(s) assigned by another device, then the new field is appended to the existing field(s).*
>
> *The appended Option 82 field includes the switch Circuit ID (inbound port number*) associated with the client DHCP packet, and the switch Remote ID. The default switch remote ID is the MAC address of the switch on which the packet was received from the client. To use the incoming VLAN's IP address instead of the switch MAC address for the remote ID, use the* **ip** *option (below).*

> **replace:** *Configures the routing switch to replace any existing Option 82 field(s) in an inbound client DHCP packet with one Option 82 field for the current routing switch.*
>
> *The replacement Option 82 field includes the switch circuit ID (inbound port number*) associated with the client DHCP packet, and the switch remote ID. The default switch remote ID is the MAC address of the switch on which the packet was received from the client. To use the incoming VLAN's IP address instead of the switch MAC address for the remote ID, use the* **ip** *option (below).*

> **drop:** *Configures the routing switch to unconditionally drop any client DHCP packet received with existing Option 82 field(s). This means that such packets will not be forwarded. Use this option where access to the routing switch by untrusted clients is possible.*
>
> *If the routing switch receives a client DHCP packet without an Option 82 field, it adds an Option 82 field to the client and forwards the packet. The added Option 82 field includes the switch circuit ID (inbound port number*) associated with the client DHCP packet, and the switch remote ID. The default switch remote ID is the MAC address of the switch on which the packet was received from the client. To use the incoming VLAN's IP address instead of the switch MAC address for the remote ID, use the IP option (below).*

keep: *For any client DHCP packet received with existing Option 82 field(s), configures the routing switch to forward the packet as-is, without replacing or adding to the existing Option 82 field(s).*

[validate]: *This option operates when the routing switch is configured with append, replace, or drop as a forwarding policy. With validate enabled, the routing switch applies stricter rules to an incoming Option 82 server response to determine whether to forward or drop the response. For more information, refer to "Validation of Server Response Packets" on page 11-88.*

[ip | mac]

This option specifies the remote ID suboption the routing switch will use in Option 82 fields added or appended to DHCP client packets. The choice of type depends on how you want to define DHCP policy areas in the client requests sent to the DHCP server. (Refer to "Option 82 Field Content" on page 11-83.)

ip: *Specifies the IP address of the VLAN on which the client DHCP packet enters the switch.*

mac: *Specifies the routing switch's MAC address. (The MAC address used is the same MAC address that is assigned to all VLANs configured on the routing switch.) This is the default setting.*

Notes on Default Remote ID Selection: *Executing the Option 82 command without specifying either* **ip** *or* **mac** *configures the remote ID as the MAC address of the switch on which the packet was received from the client. The command options for viewing the routing switch MAC address are listed at the end of the "Remote ID" description that begins on page 11-83.*

Operating Notes

■ This implementation of DHCP relay with Option 82 complies with the following RFCs:

- RFC 2131
- RFC 3046

■ Moving a client to a different port allows the client to continue operating as long as the port is a member of the same VLAN as the port through which the client received its IP address. However, rebooting the client after it moves to a different port can alter the IP addressing policy the client receives if the DHCP server is configured to provide different policies to clients accessing the network through different ports.

■ The IP address of the primary DHCP relay agent receiving a client request packet is automatically added to the packet, and is identified as the *giaddr* (gateway interface address). (That is, the giaddr is the IP address of the VLAN on which the request packet was received from the client.) For more information, refer to RFC 2131 and RFC 3046.

■ DHCP request packets from multiple DHCP clients on the same relay agent port will be routed to the same DHCP server(s). Note that when using 802.1X on a 5300xl switch running software release E.09.xx or greater, a port's VLAN membership may be changed by a RADIUS server responding to a client authentication request. In this case the DHCP server(s) accessible from the port may change if the VLAN assigned by the RADIUS server has different DHCP helper addresses than the VLAN used by unauthenticated clients.

■ Where multiple DHCP servers are assigned to a VLAN, a DHCP client request cannot be directed to a specific server. Thus, where a given VLAN is configured for multiple DHCP servers, all of these servers should be configured with the same IP addressing policy.

■ Where routing switch "A" is configured to insert its MAC address as the Remote ID in the Option 82 fields appended to DHCP client requests, and upstream DHCP servers use that MAC address as a policy boundary for assigning an IP addressing policy, then replacing switch "A" makes it necessary to reconfigure the upstream DHCP server(s) to recognize the MAC address of the replacement switch. This does not apply in the case where an upstream relay agent "B" is configured with **option 82 replace**, which removes the Option 82 field originally inserted by switch "A".

■ Relay agents without Option 82 can exist in the path between Option 82 relay agents and an Option 82 server. The agents without Option 82 will forward client requests and server responses without any effect on Option 82 fields in the packets.

■ If the routing switch is not able to add an Option 82 field to a client's DHCP request due to the message size exceeding the MTU (Maximum Transmission Unit) size, then the request is forwarded to the DHCP server without Option 82 information and an error message is logged in the switch's Event Log.

DHCP Packet Forwarding

The DHCP relay agent on the routing switch forwards DHCP client packets to all DHCP servers that are configured in the table administrated for each VLAN.

Unicast Forwarding

The packets are forwarded using unicast forwarding if the IP address of the DHCP server is a specific host address. The DHCP relay agent sets the destination IP address of the packet to the IP address of the DHCP server and forwards the message.

Broadcast Forwarding

The packets are forwarded using broadcast forwarding if the IP address of the DHCP server is a subnet address or IP broadcast address (255.255.255.255). The DHCP relay agent sets the DHCP server IP address to broadcast IP address and will be forwarded to all VLANs with configured IP interfaces (except the source VLAN).

Minimum Requirements for DHCP Relay Operation

For the DHCP Relay agent to work, the following steps must be completed:

1. DHCP Relay is enabled on the routing switch (the default setting)

2. A DHCP server is servicing the routing switch

3. IP Routing is enabled on the routing switch

4. There is a route from the DHCP server to the routing switch and back

5. An IP Helper address is configured on the routing switch, set to the IP address of the DHCP server on the VLAN connected to the DHCP Client.

Enabling DHCP Relay

The factory-default configuration enables DHCP. However, if DHCP has been disabled, you can re-enable it at the Config CLI context level by entering this command:

```
ProCurve(config)# dhcp-relay
```

To disable the DHCP Relay function, enter the command:

```
ProCurve(config)# no dhcp-relay
```

Configuring a Helper Address

At the VLAN configuration CLI context level, enter the commands to add the DHCP server's IP address to the VLANs list. For example, to configure a helper address for VLAN 1, enter these commands:

```
ProCurve(config)# vlan 1
ProCurve(vlan-1)# ip helper-address < ip-addr >
```

To remove the DHCP server helper address, enter this command:

```
ProCurve(vlan-1)# no ip helper-address < ip-addr >
```

You can configure up to 256 IP helper addresses in the switch.

N o t e   For the Series 5300xl switches only, software release E.08.07 or greater is required to support 256 IP helper addresses. Earlier releases support only 128 IP helper addresses. All other switch models covered by this guide allow 256 IP helper addresses, regardless of the software version running on the switch.

Viewing the Current DHCP Relay Configuration

Determining the DHCP Relay Setting. Use **show config** (or **show running** for the running-config file) to list the current DHCP Relay setting. Note that because DHCP Relay is enabled in the default configuration, it does not appear in these listings unless it is disabled.

```
ProCurve(config)# show config
Startup configuration:

; J4850A Configuration Editor; Created on release #E.07.21

hostname " ProCurve"
cdp run
module 1 type J4820A
ip default-gateway 18.30.240.1
snmp-server community "public" Unrestricted
vlan 1
    name "DEFAULT_VLAN"
    untagged A1
    ip address 18.30.240.180 255.255.248.0
    no untagged A2-A24
    exit
no dhcp-relay          ◄────── Non-Default DHCP-Relay Setting
```

Figure 11-35.Example of Startup-Config Listing with DHCP-Relay Disabled

Listing the Currently Configured DHCP Helper Addresses.

Syntax: show ip helper-address < *vlan-id* >

This command shows the currently configured IP Helper addresses, regard-less of whether DHCP-Relay is enabled. For example:

```
ProCurve(config)# show ip helper-address vlan 1
 IP Helper Addresses
  IP Helper Address
  ----------------
  10.28.227.97
  10.29.227.53
```

Figure 11-36.Example of Listing for IP Helper Addresses

UDP Broadcast Forwarding on 5300xl, 4200vl and 3400cl Switches

Overview

Some applications rely on client requests sent as limited IP broadcasts addressed to a UDP application port. If a server for the application receives such a broadcast, the server can reply to the client. Since typical router behavior, by default, does not allow broadcast forwarding, a client's UDP broadcast requests cannot reach a target server on a different subnet unless the router is configured to forward client UDP broadcasts to that server.

Series 3400cl switches, Series 4200vl switches, and Series 5300xl switches with software release E.09.xx and later, that have routing enabled, include optional per-VLAN UDP broadcast forwarding that allows up to 256 server and/or subnet entries on the switch (16 entries per-VLAN). If an entry for a particular UDP port number is configured on a VLAN and an inbound UDP broadcast packet with that port number is received on the VLAN, then the switch routes the packet to the appropriate subnet. (Each entry can designate either a single device or a single subnet. The switch ignores any entry that designates multiple subnets.)

N o t e   The number of UDP broadcast forwarding entries supported is affected by the number of IP helper addresses configured to support DHCP Relay. Refer to "Operating Notes for UDP Broadcast Forwarding" on page 11-102.

A UDP forwarding entry includes the desired UDP port number, and can be either an IP unicast address or an IP subnet broadcast address for the subnet the server is in. Thus, an incoming UDP packet carrying the configured port number will be:

■ Forwarded to a specific host if a unicast server address is configured for that port number.

■ Broadcast on the appropriate destination subnet if a subnet address is configured for that port number.

Note that a UDP forwarding entry for a particular UDP port number is always configured in a specific VLAN and applies only to client UDP broadcast requests received inbound on that VLAN. If the VLAN includes multiple subnets, then the entry applies to client broadcasts with that port number from any subnet in the VLAN.

For example, VLAN 1 (15.75.10.1) is configured to forward inbound UDP packets as shown in table 11-20:

Table 11-20. Example of a UDP Packet-Forwarding Environment

Interface	IP Address	Subnet Mask	Forwarding Address	UDP Port	Notes
VLAN 1	15.75.10.1	255.255.255.0	15.75.11.43	1188	Unicast address for forwarding inbound UDP packets with UDP port 1188 to a specific device on VLAN 2.
			15.75.11.255	1812	Broadcast address for forwarding inbound UDP packets with UDP port 1812 to any device in the 15.75.11.0 network.
			15.75.12.255	1813	Broadcast address for forwarding inbound UDP packets with UDP port 1813 to any device in the 15.75.12.0 network.
VLAN 2	15.75.11.1	255.255.255.0	*None*	*N/A*	Destination VLAN for UDP 1188 broadcasts from clients on VLAN 1. The device identified in the unicast forwarding address configured in VLAN 1 must be on this VLAN. Also the destination VLAN for UDP 1812 from clients on VLAN 1.
VLAN 3	15.75.12.1	255.255.255.0	*None*	*N/A*	Destination VLAN for UDP 1813 broadcasts from clients on VLAN 1.

N o t e   If an IP server or subnet entry is invalid, a switch will not try to forward UDP packets to the configured device or subnet address.

Subnet Masking for UDP Forwarding Addresses

The subnet mask for a UDP forwarding address is the same as the mask applied to the subnet on which the inbound UDP broadcast packet is received. To forward inbound UDP broadcast packets as limited broadcasts to other subnets, use the broadcast address that covers the subnet you want to reach. For example, if VLAN 1 has an IP address of 15.75.10.1/24 (15.75.10.1 255.255.255.0), then you can configure the following unicast and limited broadcast addresses for UDP packet forwarding to subnet 15.75.11.0:

Forwarding Destination Type	IP Address
UDP Unicast to a Single Device in the 15.75.11.0 Subnet	15.75.11.X
UDP Broadcast to Subnet 15.75.11.0	15.75.11.255

Configuring and Enabling UDP Broadcast Forwarding

To configure and enable UDP broadcast forwarding on the switch:

1. Enable routing.

2. Globally enable UDP broadcast forwarding.

3. On a per-VLAN basis, configure a forwarding address and UDP port type for each type of incoming UDP broadcast you want routed to other VLANs.

Globally Enabling UDP Broadcast Forwarding

Syntax [no] ip udp-bcast-forward

> *Enables or disables UDP broadcast forwarding on the router. Routing must be enabled before executing this command. Using the* **no** *form of this command disables any* **ip forward protocol udp** *commands configured in VLANs on the switch. (Default: Disabled)*

Configuring UDP Broadcast Forwarding on Individual VLANs

This command routes an inbound UDP broadcast packet received from a client on the VLAN to the unicast or broadcast address configured for the UDP port type.

Syntax [no] ip forward-protocol udp < *ip-address* > < *port-number | port-name* >

> *Used in a VLAN context to configure or remove a server or broadcast address and its associated UDP port number. You can configure a maximum of 16* **forward-protocol udp** *assignments in a given VLAN. The switch allows a total of 256* **forward-protocol udp** *assignments across all VLANs. You can configure UDP broadcast forwarding addresses regardless of whether UDP broadcast forwarding is globally enabled on the switch. However, the feature does not operate unless globally enabled.*
>
> *— Continued on the next page. —*

— Continued from the preceding page. —

< ip-address >: *This can be either of the following:*
- *The unicast address of a destination server on another subnet. For example: 15.75.10.43.*
- *The broadcast address of the subnet on which a destination server operates. For example, the following address directs broadcasts to All hosts in the 15.75.11.0 subnet: 15.75.11.255.*

Note: The subnet mask for a forwarded UDP packet is the same as the subnet mask for the VLAN (or subnet on a multinetted VLAN) on which the UDP broadcast packet was received from a client.

< udp-port-# >: *Any UDP port number corresponding to a UDP application supported on a device at the specified unicast address or in the subnet at the specified broadcast address. For more information on UDP port numbers, refer to "TCP/UDP Port Number Ranges" on page 11-102.*

< port-name >: Allows use of common names for certain well-known UDP port numbers. You can type in the specific name instead of having to recall the corresponding number:

dns: Domain Name Service (53)
ntp: Network Time Protocol (123)
netbios-ns: NetBIOS Name Service (137)
netbios-dgm: NetBIOS Datagram Service (138)
radius: Remote Authentication Dial-In User Service (1812)
radius-old: Remote Authentication Dial-In User Service (1645)
rip: Routing Information Protocol (520)
snmp: Simple Network Management Protocol (161)
snmp-trap: Simple Network Management Protocol (162)
tftp: Trivial File Transfer Protocol (69)
timep: Time Protocol (37)

For example, the following command configures the router to forward UDP broadcasts from a client on VLAN 1 for a time protocol server:

```
ProCurve(config)# ip forward-protocol udp 15.75.11.155
timep
```

Displaying the Current IP Forward-Protocol Configuration

Syntax   show ip forward-protocol [vlan < *vid* >]

Displays the current status of UDP broadcast forwarding and lists the UDP forwarding address(es) configured on all static VLANS in the switch or on a specific VLAN.

```
WorkingConfig(config)# show ip forward-protocol

 IP Forwarder Addresses

    UDP Broadcast Forwarding: Disabled

VLAN: 1
 IP Forward Addresses UDP Port
 -------------------- --------
 15.75.11.43          37
 15.75.11.255         53
 15.75.12.255         1813

VLAN: 2
 IP Forward Addresses UDP Port
 -------------------- --------
 15.75.12.255         1812

VLAN: 3
 IP Forward Addresses UDP Port
 -------------------- --------
 15.75.10.155         162
```

Global Display Showing UDP Broadcast Forwarding Status and Configured Forwarding Addresses for Inbound UDP Broadcast Traffic for All VLANs Configured on the Router.

Figure 11-37.Displaying Global IP Forward-Protocol Status and Configuration

```
ProCurve(config)# show ip forward-protocol vlan 1

 IP Forwarder Addresses

    UDP Broadcast Forwarding: Disabled

 IP Forward Addresses UDP Port
 -------------------- --------
 15.75.11.43          37
 15.75.11.255         53
 15.75.12.255         1813
```

Display Showing UDP Broadcast Forwarding Status and the Configured Forwarding Addresses for inbound UDP Broadcast Traffic on VLAN 1

Figure 11-38.Displaying IP Forward-Protocol Status and Per-VLAN Configuration

Operating Notes for UDP Broadcast Forwarding

Maximum Number of Entries. The number of UDP broadcast entries and IP helper addresses combined can be up to 16 per VLAN, with an overall maximum of 256 on the switch. (IP helper addresses are used with the switch's DHCP Relay operation. For more information, refer to "Configuring DHCP Relay" on page 11-79.) For example, if VLAN 1 has 2 IP helper addresses configured, you can add up to 14 UDP forwarding entries in the same VLAN.

TCP/UDP Port Number Ranges. There are three ranges:

- Well-Known Ports: 0 - 1023
- Registered Ports: 1024 - 49151
- Dynamic and/or Private Ports: 49152 - 65535

For more information, including a listing of UDP/TCP port numbers, go to the *Internet Assigned Numbers Authority* (IANA) web site at:

www.iana.org

Then click on:

Protocol Number Assignment Services

P (Under "Directory of General Assigned Numbers" heading)

Port Numbers

Messages Related to UDP Broadcast Forwarding

Message	Meaning
`udp-bcast-forward: IP Routing support must be enabled first.`	Appears in the CLI if an attempt to enable UDP broadcast forwarding has been made without IP routing being enabled first. Enable IP routing, then enable UDP broadcast forwarding.
`UDP broadcast forwarder feature enabled`	UDP broadcast forwarding has been globally enabled on the router. Appears in the Event Log and, if configured, in SNMP traps.
`UDP broadcast forwarder feature disabled`	UDP broadcast forwarding has been globally disabled on the router. This action does not prevent you from configuring UDP broadcast forwarding addresses, but does prevent UDP broadcast forwarding operation. Appears in the Event Log and, if configured, in SNMP traps.
`UDP broadcast forwarder must be disabled first.`	Appears in the CLI if you attempt to disable routing while UDP forwarding is enabled on the switch.

Configuring Static Network Address Translation (NAT) for Intranet Applications on the 5300xl Switches

This section applies only to the ProCurve Series 5300xl switches.

Static NAT is useful in applications where you want to conceal a "private", or hidden region of your network from the general population of users in the "public" region, but allow access from the "public" region to selected devices in the hidden region. NAT performs this function by translating the IP addresses of the selected devices so that they appear to logically reside in the public region of your network instead of in a hidden region. This is done by mapping a virtual, public IP address to the actual, private IP address of the device you want to make accessible to clients in the public region. For example:

Static NAT Table		
Client Name	IP Seen in Corporate (Public) Region:	Configured (Private) IP:
A	15.33.235.10	10.10.10.11
B	15.33.235.32	10.10.10.12
C	15.33.235.38	10.10.10.13
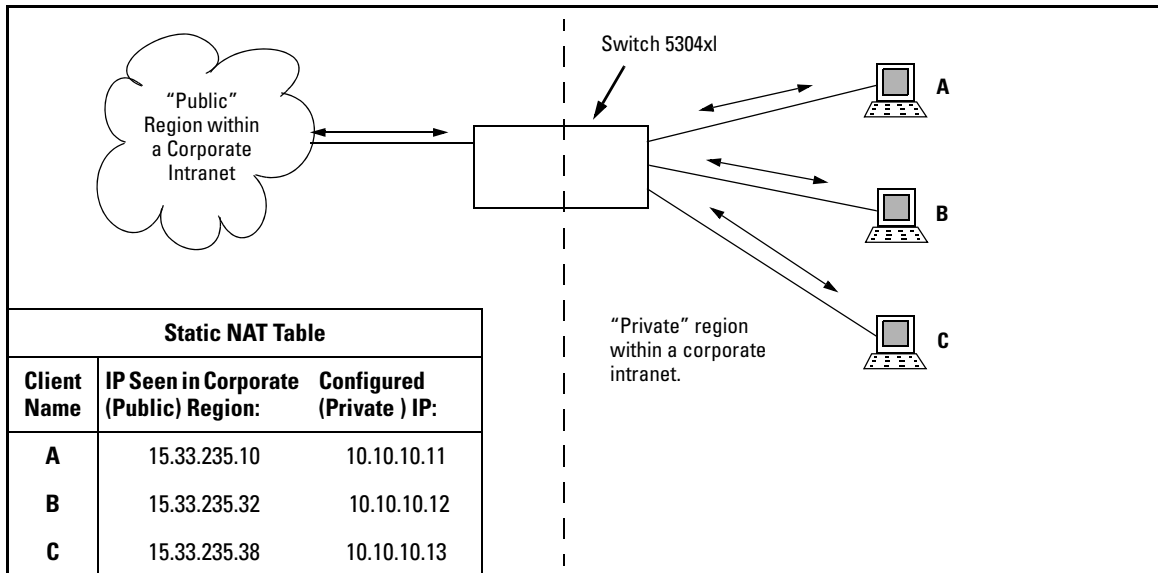
Figure 11-39.Example of a Static NAT Application in an Intranet

Static NAT operates globally, on a per-switch basis and evaluates all incoming and outgoing packets on all ports. NAT performs IP address mapping only on packets having a source or destination IP address appearing in the static NAT

table the switch maintains when NAT is configured. Note also that static NAT operates at the layer 3 level. IP addresses embedded in layers 4 - 7, as is the case with some applications, are not translated by static NAT.

Static NAT Operating Rules

■   Uses one-to-one IP address mapping. That is, with each "private" device IP address you configure for static NAT, there must be a corresponding virtual, "public" IP address.

■   Allows up to 32 client IP addresses per switch, which requires an equal number of virtual IP address assignments. Note that increasing the number of NAT mappings can reduce overall NAT performance.

■   Requires routing to be enabled on the switch.

Configuring Static NAT

Syntax: [no] ip nat static < *private-ip* > < *public-ip* >

Configures the switch to map a virtual IP address over the actual IP address for a device residing in a region of your network that is hidden from general network users.

*< **private-ip** > : This is the IP address of a device in a region of your network that you want to remain hidden from general network users. (This address is the actual IP address configured on the device.)*

*< **public-ip** > :   This is the virtual IP address you want to use to access (from the public region of the network) a specific device residing in the hidden portion of the network.*

*With NAT configured, the switch intercepts the traffic requesting the < **public-ip** > address and redirects it to the corresponding < **private-ip** > address. In this case, the switch translates the destination ip address to the < **private-ip** > address and then forwards the traffic normally. In the opposite direction, the switch intercepts the traffic from a configured < **private-ip** > address destined to the public network and translates the < **private-ip** > address to its corresponding < **public-ip** > address before forwarding the traffic.*

You can configure up to 32 IP NAT static mappings on the switch, which means you can map the configured IP addresses of 32 devices to corresponding virtual IP addresses.

*The [**no**] form of the command removes the specified static NAT assignment from the switch's running configuration.*

Example. This example uses the topology in figure 11-39 on page 11-103:

■ The switch is connected to the corporate intranet through VLAN 100 (IP address: 15.33.235.1).

■ The three devices are configured on VLAN 101 in the corporation's "private" region (IP address: 10.10.10.1) with these IP addresses:

A. 10.10.10.11

B. 10.10.10.12

C. 10.10.10.13

■ The system administrator selects these virtual IP addresses to make the three devices appear to reside in the corporation's "public" region:

A. 15.33.235.10

B. 15.33.235.32

C. 15.33.235.38

To configure the static NAT mapping between the actual IP addresses configured on the devices and the corresponding virtual IP addresses:

```
ProCurve(config)# ip nat static 10.10.10.11 15.33.235.10
ProCurve(config)# ip nat static 10.10.10.12 15.33.235.32
ProCurve(config)# ip nat static 10.10.10.13 15.33.235.38
```

The above commands create the virtual IP address mappings in figure 11-40:
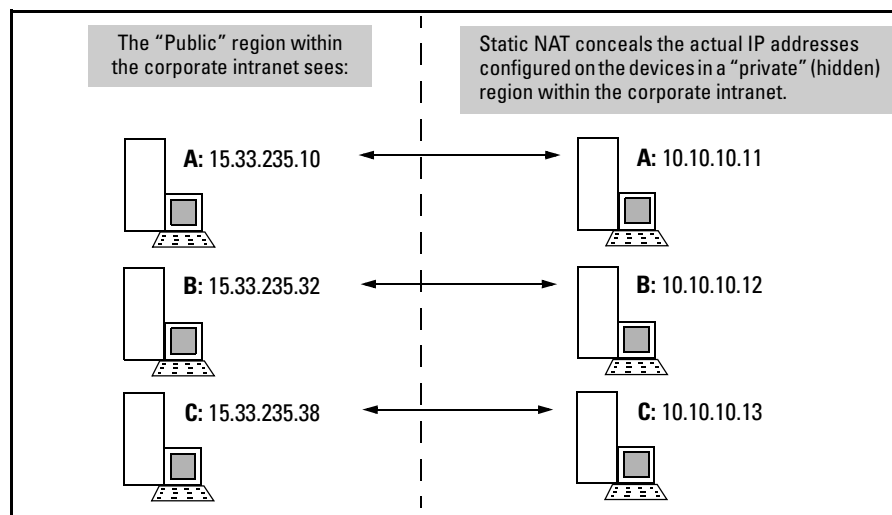
Figure 11-40.Example of Static NAT Mapping of Virtual IP Addressing

Displaying Static NAT Statistics and Configuration

Syntax: show ip nat

> *Displays the current IP NAT static configuration in the running-config file and the current IP NAT counters.*
>
> **Total Translations***: Displays a 32-bit counter showing the number of packets in which IP NAT has translated the source or destination IP address from a private address to a public address or from a public address to a private address.*

For example, the following shows a sample of **show ip nat** output for the example on page 11-105.

```
ProCurve# show ip nat
Total translations: 26784


    Private            Public
  _____    _____
    10.10.10.11        15.33.235.10
    10.10.10.12        15.33.235.32
    10.10.10.13        15.33.235.38
```

Figure 11-41.Example of Displaying NAT Mappings

Static NAT Operating Notes

■ Static NAT on the 5300xl switches is a method for accessing a private region within an intranet. It is not the dynamic NAT often used for IP address translation from private IP addresses to registered, global IP addresses on the internet, and is not supported for Internet NAT applications.

■ Non-NAT hosts in the same subnet (VLAN) as NAT hosts will be routed normally. That is, the IP addresses of hosts without a static NAT entry will not be translated.

■ Static NAT is not intended for bandwidth-intensive or high-traffic applications, and such environments can degrade NAT performance.

■ For a given virtual IP address, static NAT applies the subnet mask that is configured in the corresponding actual IP addressing.

- Static NAT does not provide TCP/UDP port number translation.

- Static NAT is not intended to support a large number of clients.

- Static NAT is not a security application and should not be considered as a substitute for a firewall.

12

Router Redundancy Using XRRP

Contents

Introduction to XRRP

XRRP does not apply to the Series 4200vl switches.

XRRP (XL Router Redundancy Protocol) provides router redundancy, or failover, to a backup router in case one fails. XRRP is similar to the industry standard VRRP (Virtual Router Redundancy Protocol), although the details of the operation are different.

Throughout this discussion, the switches covered by this manual are considered as routers and the term "router" is used to identify them.

Terminology

The following terms are used in the description of XRRP:

■ **Protection Domain** – A pair of physical routers that are running XRRP and are configured to provide fail-over protection for each other.

■ **Virtual Router** – A virtual routing device that provides a router interface to host computers that are accessing it. Each physical router in the Protection Domain can own a virtual router. On fail-over, one physical router may own all the virtual routers in the Protection Domain. Movement of the virtual router responsibility, as part of the XRRP operation, is transparent to the host computers.

■ **Master** – The physical router that is currently providing the virtual router interface to the host computers.

■ **Owner** – The physical router that is configured with the IP address that is involved in the XRRP operation.

■ **Advertisement Interval** – The time interval at which the Master router sends out XRRP packets on each virtual router interface.

■ **Fail-Over:** When the access to a VLAN from one of the routers in the Protection Domain fails, the routing function of that router is automatically transferred to the other router in the Protection Domain.

■ **Fail-Back Router:** In a given protection domain, this is the XRRP-enabled router that takes over the routing functions transferred from its XRRP peer in the domain when the peer loses access to one or more of its XRRP VLANs. The fail-back router must have access to all of its XRRP VLANs at the time of the fail-over. See also **Fail-Over Router**.

■ **Fail-Over Router:** In a given protection domain, this is an XRRP-enabled router that loses access to one or more of its XRRP VLANs, causing a fail-over of its routing functions to the other XRRP router (peer) in the domain. The peer must have access to all of its XRRP VLANs at the time of the fail-over, and is designated as the "fail-back" router. See also **Fail-Back Router**.

■ **Infinite Fail-Back:** The operating mode of an XRRP router in which the router does not automatically allow fail-back when its peer in the protection domain recovers from a fail-over condition.

■ **Primary Control:** The mode in which the XRRP VLANs configured on a router in a protection domain are controlled by that router.

■ **Secondary Control:** The mode in which the XRRP VLANs configured on one router in a protection domain are controlled by the other (fail-back) XXRP router in the domain. A fail-back router advertising XRRP packets for the failed (peer) router's backed-up IP addresses has both primary and secondary control of the XRRP VLANs in the domain.

■ **Permanent Control:** The mode associated with infinite fail-back in which a fail-over has occurred in a protection domain and the resulting fail-back router has both primary control and secondary control of the XRRP VLANs in the domain. See also **Primary Control** and **Secondary Control**.

Overview of XRRP Operation

XRRP allows you to configure pairs of switches to behave as backup routers for each other. Each pair of routers configured to operate this way is defined as a protection domain. (You can use the switches covered in this manual in any combination to create a protection domain.) If either router in the protection domain fails for whatever reason, the other router automatically takes over the routing function of the failed router. This transfer of the routing function is transparent to the host computers that are using the routers.

N o t e   To accomplish this transfer, both routers in the protection domain must have identical network access so that each can get to all the same subnets and the same end nodes without going through each other.

Figure 12-1 shows an example of a Protection Domain being used to provide redundant connectivity between some clients and the network servers that the hosts need to access. As part of the XRRP configuration, you define the identity of the protection domain. In figure 1, it is Domain 2. See "Configuring XRRP" on page 12-20 for information on how to configure XRRP.
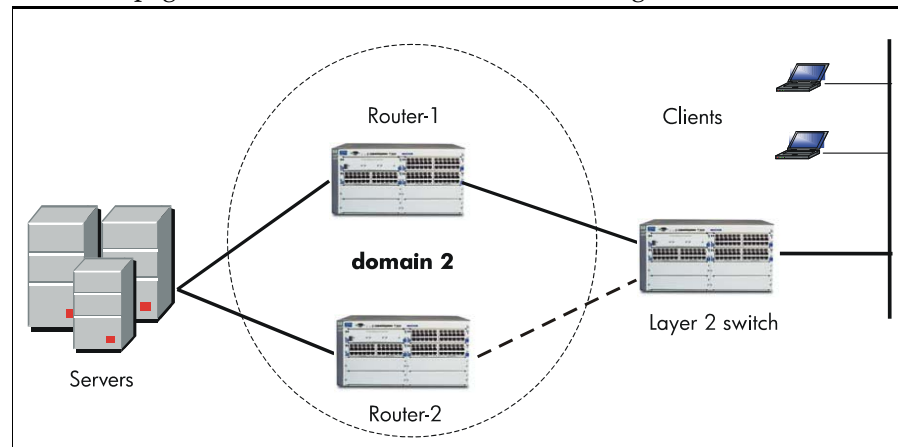
Figure 12-1. XRRP Protection Domain

The clients are connected to the routers through a Layer 2 switch (in this case a Procurve Switch 4108gl).

XRRP During Normal Router Operation

For each router, XRRP defines a virtual router, using the IP address that you have configured on the router interface, and for which XRRP assigns a virtual MAC address based on the Protection Domain ID and the XRRP router number of the router that owns the interface. The configuration is done for each VLAN on which you wish to use XRRP for router redundancy, so the router interfaces for each virtual router must be in the VLAN. Each Protection Domain contains two routers, but within a single VLAN, up to 16 Protection Domains (16 pairs of routers) can be configured.

In the situation in which both routers in the Protection Domain are operating normally, none of the VLANs are down, each physical router behaves as the Master of all of its XRRP virtual router interfaces. The Master and Owner of each interface is the same.

In the example shown in figure 12-2, the XRRP configuration is done in VLAN 5. For Domain 2, Router-1 is given the IP address of 10.1.1.1 and Router-2 is given the address 10.1.1.2. XRRP assigns MAC addresses MAC-A to Router-1 and MAC-B to Router-2. Note that the clients in figure 2 use both of the virtual routers as their default gateways. Client 10.1.1.48 is configured to use virtual router 10.1.1.1 as its default gateway, and client 10.1.1.49 is configured to use virtual router 10.1.1.2. In this way XRRP can be used to provide load balancing as long as both virtual routers are operating normally. The virtual routers will route packets passed to them, respond to IP ARP requests and PING packets, and perform the other router functions.
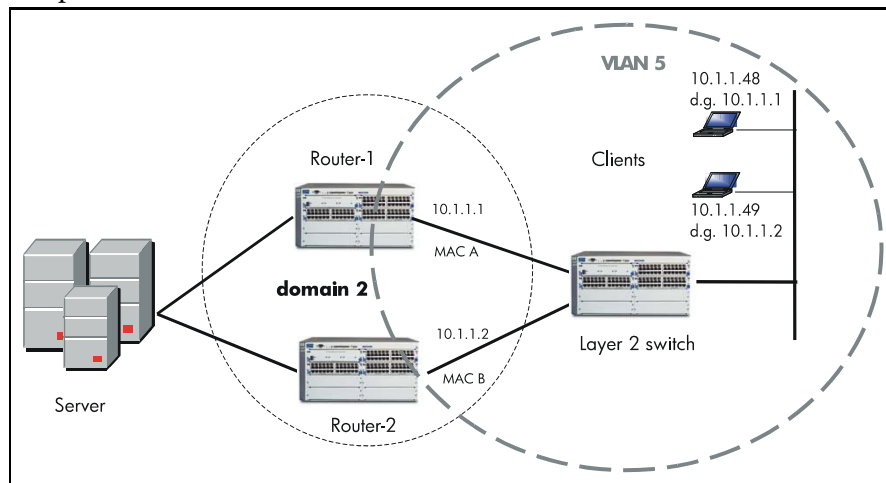
Figure 12-2. XRRP During Normal Router Operation

XRRP Fail-Over Operation

If all access to a VLAN from one of the routers in the Protection Domain fails, the routing function of that router is automatically transferred to the other router in the Protection Domain. The master of the virtual router in the Protection Domain sends out multicast advertisements at the XRRP advertisement interval (every 5 seconds by default). If the other router in the Protection Domain does not hear an advertisement packet within 3 advertisement intervals, this other router will become the master router, and it takes control of the IP address and the MAC address of the failed router.

Single VLAN Operation

In figure 12-3, the link between the layer-2 switch and Router-2 fails. As a result, Router-2 no longer hears any link signals on VLAN 5 and the communication between Router-2 and Router-1 is disabled. Router-1, after not hearing XRRP packets from Router-2, will take over the IP addresses from Router-2 for the VLAN 5 interfaces and it will take over the XRRP MAC address for Router-2. Now Router-1 is the Master for its own IP addresses *and* the IP addresses for Router-2 for VLAN 5, and it is the Master of its own XRRP MAC address and the XRRP MAC address for Router-2. As far as the clients are concerned, the transfer of router functionality is transparent – they can still get to the servers using the same IP addresses and MAC addresses as before.
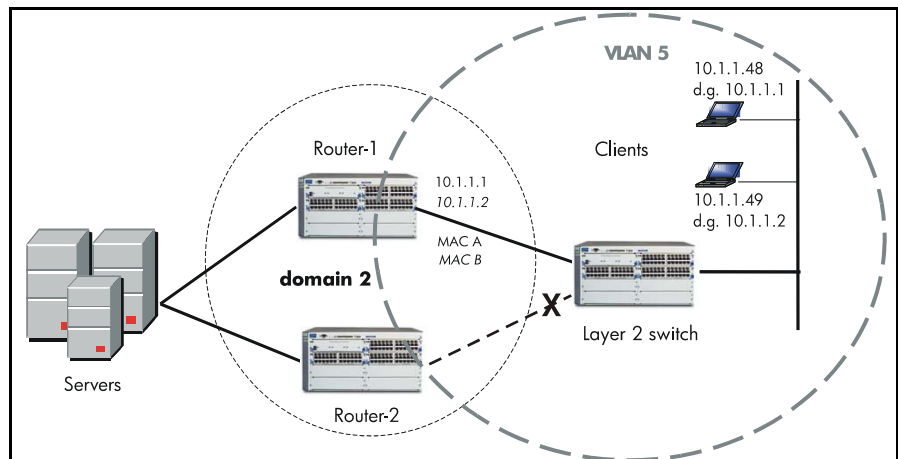
Figure 12-3. XRRP Fail-Over Example

N o t e   Figure 12-3 shows a single interface on VLAN 5, but multiple interfaces could exist. For the fail-over to occur, Router-2 must have lost communication on all the VLAN 5 interfaces.

When the fail-over occurs, Router-1 would take over as the Master of the IP address for Router-2 on VLAN 5. If Router-2 has multiple IP addresses on VLAN 5, a multinet situation, Router-1 takes over all the IP addresses for Router-2 on VLAN 5.

Multiple VLAN Operation

If a router has XRRP interfaces in multiple VLANs, there are some additional details in the way that XRRP operates. For each VLAN on which you wish to run XRRP, a virtual router interface is created.

Total Router Fail-Over. In the multiple VLAN case, fail-over again occurs when XRRP packets are not heard on at least one of the VLANs from the other virtual router in the Protection Domain. Even if one or more VLANs are still operating correctly, when one VLAN fails (a link signal is no longer detected by the router from any device in the VLAN), the router with the failed VLAN will stop its operation as the Master of its owned virtual router interfaces. The fail-over is a "total router" fail-over. The router with the failed VLAN stops routing on all of its XRRP virtual interfaces and the other router in the Protection Domain takes control of all the XRRP IP and MAC addresses.

Depending on whether the routers can maintain communication through at least one of the XRRP VLANs (the VLAN continues to operate correctly for both routers in the Protection Domain), there are two causes of fail-over:

■   If communication is maintained, the router with the failed VLAN can execute what is called a "fast fail-over". This situation is depicted in figure 12-4.

■   If *all* XRRP communication is lost between the routers in the Protection Domain, the normal fail-over occurs after 3 advertisement intervals, as shown in figure 12-5.

Fast Fail-Over. As shown in figure 12-4, if the same link goes down as was shown in figure 12-3, the standard fail-over does not occur. As soon as Router-2 detects the loss of link signal from any device in VLAN 5, it immediately requests, through VLAN 6, that Router-1 to take over all of its virtual router resources. This function is referred to as "fast fail-over". Because it occurs as soon as link signal is lost, the fail-over can take as little as one second to complete.
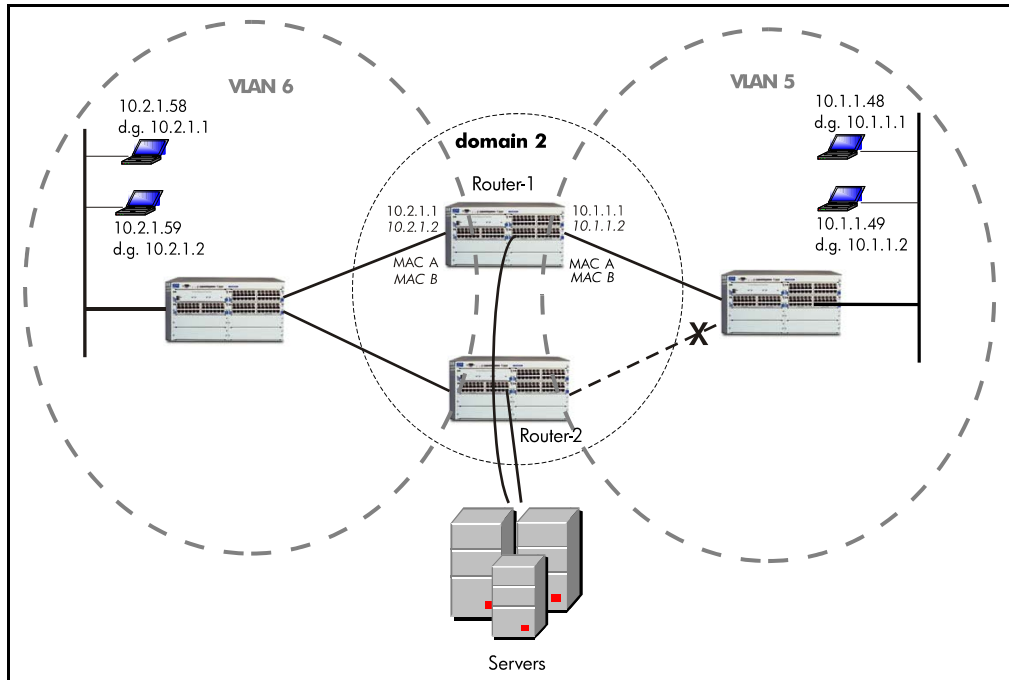
Figure 12-4. Fast Fail-Over with Partial VLAN Failure

When Router-2 makes the fast fail-over request, if Router-1 has no failed VLANs, then it will take control of Router-2's virtual interfaces. If Router-1 also has one or more failed VLANs, then it will not take control and both routers will continue to control only their owned IP addresses.

Standard Fail-Over. In the multiple-VLAN situation in which all communi-
cation between the routers in the Protection Domain is lost, the standard
XRRP fail-over occurs. As shown in figure 6, Router-2 has lost communica-
tions on all of its XRRP virtual router interfaces. In this case, Router-1 will no
longer hear XRRP packets coming from Router-2. If that condition persists for
3 advertisement intervals, Router-1 then takes over all of the virtual routers
from Router-2.
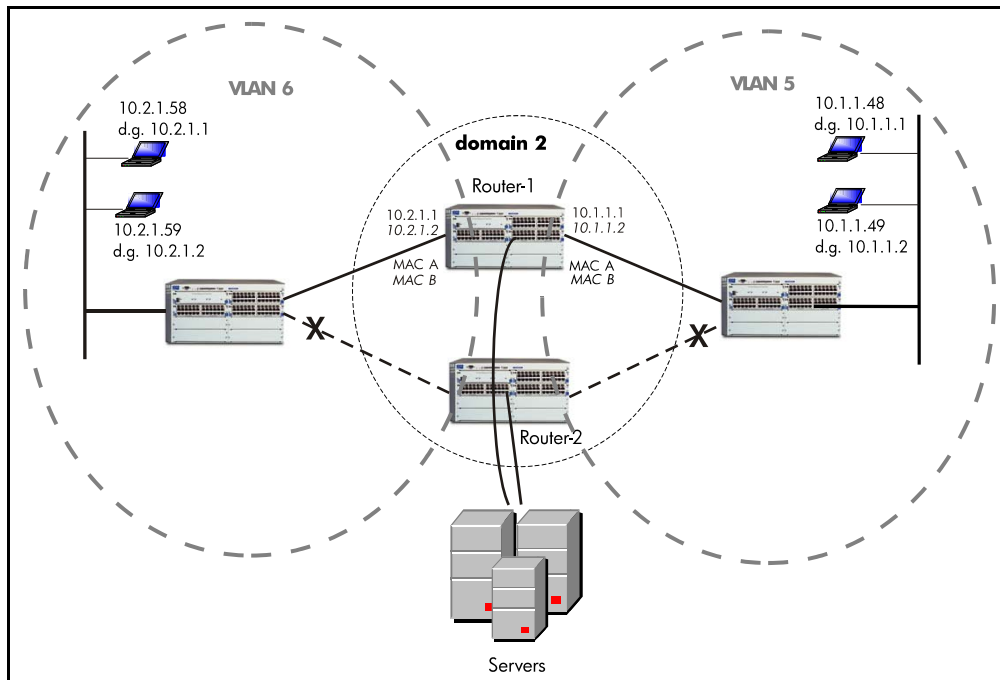
Figure 12-5.Standard XRRP Fail-Over with Total VLAN Failure

If the cause of the total VLAN access failure, as shown in figure 17-5, is because
of a complete router failure (due to building power loss, for example), the
router that remains active will wait for the three XRRP advertisement intervals
and will then take control of the failed router's IP and MAC addresses. If both
routers are still active but all network connections between them have been
severed, then both routers will take over for each other. This means that
identical IP and MAC addresses will exist on both routers, but in a completely
severed network, there will be no duplicated MAC or duplicate IP address
errors.

If Communication is Maintained Through Non-XRRP Interfaces. In some cases, it may be possible that all connectivity is lost between the routers on all their XRRP virtual router interfaces, in which case XRRP operates and both routers try to take control of all the virtual routers in the Protection Domain, but if connectivity still exists on non-XRRP VLANs, a situation could occur in which both routers allow and use the same MAC addresses on the non-XRRP VLAN(s). This could create a situation in which a switch connected between the two routers will see continuous move interrupts and potential duplication of inbound packets if that switch floods. To prevent this condition, a simple XRRP protocol packet is exchanged between the two routers on the non-XRRP VLAN to inform each other of their uses of the MAC addresses. This exchange prevents the routers from taking over each other's MAC addresses. Note that this protocol is used only when one router attempts to take over control of the other router's virtual router interfaces.

XRRP Infinite Fail-Back for the 5300xl Switches

Introduction

XRRP infinite failback is an optional enhancement to the Series 5300xl XRRP routing feature, and is designed to reduce network disruption due to peer router fail-backs occurring automatically. This is accomplished by configuring XRRP with the infinite fail-back option and then using a manual fail-back command that can be executed at the discretion of a system operator.

Overview of Infinite Fail-Back Operation

If a fail-over event occurs in an XRRP protection domain, the peer for the failed router automatically takes over the routing function for the failed router. This peer router, which is already the master for its own XRRP VLAN(s) and XRRP MAC address (primary address control), also becomes the master for the XRRP VLAN(s) and XRRP MAC address for the failed router (secondary address control).

The Problem. Prior to software release E.09.05, if a failed XRRP router recovers access to all of its XRRP VLANs, then a fail-back automatically occurs, which removes secondary address control from the fail-back router and restores control of these addresses to the recovered router. (The fail-back router ceases to advertise XRRP packets for the failed router's backed-up IP addresses.) This automatic fail-back can cause a network slowdown due to a disruption of the TCP connections during the fail-back. In cases where a fail-over/fail-back cycle occurs repeatedly, frequent network disruptions can occur.

The Solution. Beginning with software release E.09.05, you can optionally configure *XRRP infinite fail-back*, which blocks automatic fail-back as long as the fail-back router continues XRRP operation with at least one of its XRRP VLANs remaining up. In this mode, the fail-back router maintains "permanent" primary and secondary address control. This means that recovery of the fail-over router does not automatically result in a fail-back from its peer, and can only occur when either a system operator uses the CLI to force fail-back or there is a system change affecting the fail-back router. (Events that can cause a fail-back to occur are described under "Router Operation in the Infinite Fail-Back Mode" on page 12-14.)

Causes of Fail-Over and Fail-Back

Fail-Over. An XRRP router fail-over to its peer occurs in these instances:

- The router loses connectivity on *any* XRRP VLAN (and the peer router has maintained connectivity with *all* of its XRRP VLANs).

- None of an XRRP router's multicast advertisements are detected by the peer router within three advertisement intervals (and the peer router has maintained connectivity with at least one of its XRRP VLANs).

N o t e   If a peer router is unable to support a fail-over due to its own failures, the fail-over does not occur and routing will cease in at least some areas of the network.

Fail-Back. Without infinite fail-back enabled, an XRRP peer router with both primary and secondary control in a protection domain automatically initiates a fail-back when it detects XRRP advertisements indicating that the failed router has recovered access to all of its XRRP VLANs.

With infinite fail-back enabled in the protection domain, a change in the state of the failed router does not initiate a fail-back.

Fail-Over Operation with Infinite Fail-Back Enabled
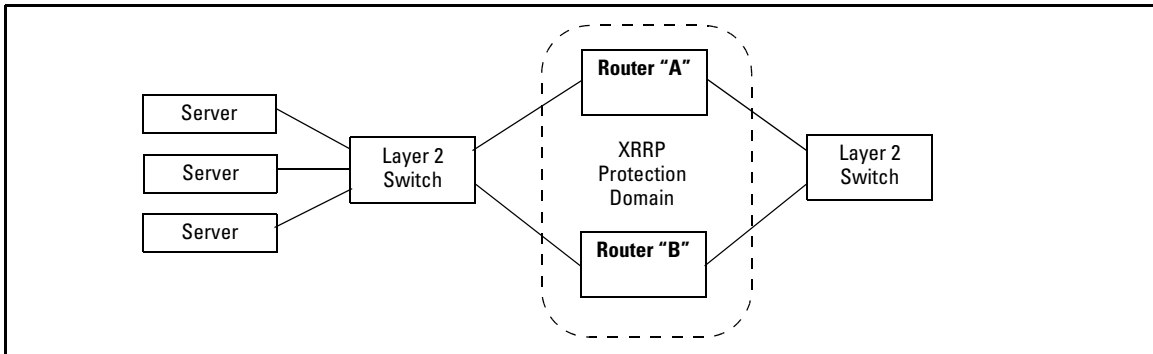
Figure 12-6. Example of XRRP Routers in a Protection Domain

When XRRP is enabled on router "A" with infinite fail-back already enabled in the configuration, the router immediately waits for a period equal to three times the XRRP advertisement interval to determine the current fail-back mode of its XRRP peer router "B". Depending on the result of the waiting period, router "A" assumes one of the following modes:

■ **primary control only:** occurs if router "B" is doing one of the following:
  • running with only primary control mode.
  • running with primary and secondary control mode, but *has not* detected a prior fail-over on router "A". (This can occur if a failed router is replaced with another XRRP router.)

■ **primary and secondary control:** occurs if router "B" is in a fail-over mode.

■ **fail-over:** occurs if router "B" *has* detected a prior fail-over on router "A" and is in the primary and secondary control mode as a result.

Router Operation in the Fail-Over Mode

An XRRP router that has failed-over to a peer (fail-back router) in a protection domain (where both routers are configured with infinite fail-back), and then recovers, remains in the fail-over mode unless the status of the fail-back router changes as described under <zBlue>"Router Operation in the Infinite Fail-Back Mode", below. This means that changes solely in the status of the failed-over router, such as recovery of all its XRRP VLANs, cannot initiate a fail-back from the peer router.

Router Operation in the Infinite Fail-Back Mode

An XRRP router with permanent (primary and secondary) address control and infinite fail-back enabled will not surrender permanent control (fail-back) to a recovered fail-over router except as described below.

■ XRRP traffic is moving between the two routers and a system operator initiates fail-back by using the **xrrp ctrl-transfer** CLI command in either of the following cases:

  • on the fail-back router to force a fail-back to the fail-over router

  • on the fail-over router, provided that traffic can move between the fail-back router and the fail-over router

  (This command allows the fail-over router to resume primary control of its XRRP VLANs, which causes the fail-back router to stop advertising XRRP packets for the fail-over router IP addresses.)

N o t e   If a system operator uses **xrrp ctrl-transfer** to force a fail-back from a fail-back router to a fail-over router that has not recovered access to all of its XRRP VLANs, the fail-back is blocked and the fail-back router continues to advertise XRRP packets for the failed router's backed-up IP addresses.

■ Fail-Back is triggered by an event in which all of the fail-back router's XRRP VLANs go down due to network problems or by the fail-back router rebooting. (Contrary to operation without infinite fail-back enabled, the fail-back router retains permanent control as long as at least one of its XRRP VLANs is up.)

■ A system operator initiates fail-back by disabling and re-enabling XRRP on the peer router.

■ Fail-Back is triggered by replacing the failed router in a protection domain with an XRRP router that is *not* configured with infinite fail-back.

N o t e   In a protection domain where XRRP router "A" has permanent control and peer router "B" has all of its XRRP VLANs up, replacing router "A" causes a fail-back that restores router "B" to primary control of its XRRP VLANs.

Enabling Infinite Fail-Back in a Protection Domain

As described in the chapter titled "Router Redundancy Using XRRP" in the *Advanced Traffic Management Guide* for your router, both router peers in a protection domain must have identical network access so that each can get to all the same subnets and the same end nodes without going through each other.

1. Before enabling infinite fail-back, configure XRRP on both routers in the protection domain. (Refer to the chapter mentioned in the above paragraph.)

2. Configure infinite failback on both routers in the domain.

Syntax: [no] xrrp inf-failback

> *Enables or disables infinite fail-back on a router running XRRP. In a given protection domain, infinite fail-back must be individually enabled on both XRRP routers to provide full fail-back control. For information on configuring general XRRP operation, refer to the chapter titled "Router Redundancy Using XRRP" in the <u>Advanced Traffic Management Guide</u> for your Series 5300xl router. (Default: Disabled)*

3. Enable XRRP operation in the protection domain by executing the **xrrp** command in the CLI of both routers.

Initiating a Fail-Back When Infinite Fail-Back Is Enabled

Syntax: xrrp ctrl-transfer

> *In a protection domain where infinite fail-back has already been enabled, this command enables a system operator to manually initiate fail-back where the fail-over router has already regained access to all of its XRRP VLANs. Successful execution of this command leaves both routers in a protection domain with primary control of their configured XRRP VLANs. The command can be executed in the CLI of either router. Note that if the fail-over router has not regained access to all of its XRRP VLANs, the fail-back is blocked and the fail-back router continues to maintain permanent control of all XRRP VLANs in the domain.*

Displaying the Infinite Fail-Back Configuration

Syntax: show xrrp config global

Indicates the router's global XRRP configuration, including infinite fail-back status.

```
HP ProCurve Switch 5304XL(config)# show xrrp config global

 Status and Counters - XRRP Global Configuration Information

  XRRP Enabled   : Yes
  Domain Number  : 2
  Router Number  : 1
  Failback Delay : 10
  Infinite Failback : Enabled
```

Infinite failback status in a router configured for XRRP operation. When this shows **Enabled**, automatic fail-back does not operate and it is necessary to use the **xrrp ctrl-transfer** command to initiate a fail-back.

Note: Events in the fail-back router, such as a reboot, can also cause a fail-back. Refer to "Router Operation in the Infinite Fail-Back Mode" on page 12-14.

Figure 12-7. Example of Displaying XRRP Infinite Fail-Back Configuration Status

XRRP Failback Log Messages

Message	Meaning
Infinite failback has been enabled on this router.	Infinite fail-back is enabled on the router.
Infinite failback is not active on this router.	Infinite fail-back has been disabled on the router.
Peer router has permanent control	The router is not taking primary address control because its peer has permanent (primary and secondary) address control for the protection domain.
User has triggered failback to router < *router-name* >	A system operator has executed the **xrrp ctrl-transfer** command to force a fail-back from the router currently having permanent (primary and secondary) control to its peer (that is, to the fail-over router).

XRRP Operating Notes

- **Reserved Multicast MAC Address** – XRRP uses the following multicast MAC address for its protocol packets: **0101-E794-0640**

- **Use of Proxy ARP on non-XRRP VLANs** – Although it is not disallowed, you should not configure Proxy ARP on non-XRRP VLANs on a router running XRRP. To do so will potentially cause loss connectivity on those non-XRRP VLANs should the router fail-over to the other router in the Protection Domain.

  The non-XRRP VLANs will not fail-over, however the XRRP-assigned MAC address, which were used while the router was operating as an XRRP router, were used on *all* the router interfaces, XRRP and non-XRRP. When the router fails-over its XRRP interfaces, it stops operating as an XRRP router and reverts back to using the default factory-assigned MAC address on all the interfaces. Any hosts that rely on proxy ARP will only receive updated ARPs for the router MAC address not for all the possible IP addresses that the router had previously responded too as a proxy ARP interface. Note: this is not a problem on the XRRP interfaces because the XRRP-assigned MAC address will have moved over to the other router and proxy ARP learned routes will still be valid. (See also "Router connectivity" on the next page).

- **Static and Default route usage** – You should never set up a default or static route that points to the peer XRRP router as the path. Should fail-over occur, this path is no longer valid and connectivity on that path will be lost.

- **Router connectivity** – In general peer routers using XRRP must have identical connectivity. That is, they must have the same access to all remote subnets, and the route costs of the access must be the same. This will prevent the routing protocols from using the peer XRRP router as the best path to get to a given subnet.

  If this is not done, then fail-over may have to wait until the routing protocols converge before full connectivity is restored. Should one router have exclusive access to a given subnet, (that is, the only way one of the XRRP routers can get to a given subnet is though its peer) connectivity to those exclusive subnets may be lost when fail-over occurs.

- **SNMP Requests** – SNMP requests on an XRRP router interface follow the virtual interface, which may be different from the physical interface in a fail-over situation. Alternately, you can ensure that the SNMP requests are made on the management VLAN or other non-XRRP interface.

■ **Multiple VLAN Considerations** – When using multiple VLANs, some
consideration must be given to whether the router interfaces are
connected to devices that have a **multiple forwarding database** (a MAC
address table for each VLAN):

• If the switch at the other end of a router interface connection *has* a
multiple forwarding database, you can use a separate interface for
each VLAN. Since the switch at the other end has a separate MAC
address table for each VLAN, the fact that the router uses the same
MAC address on all interfaces causes no problems.
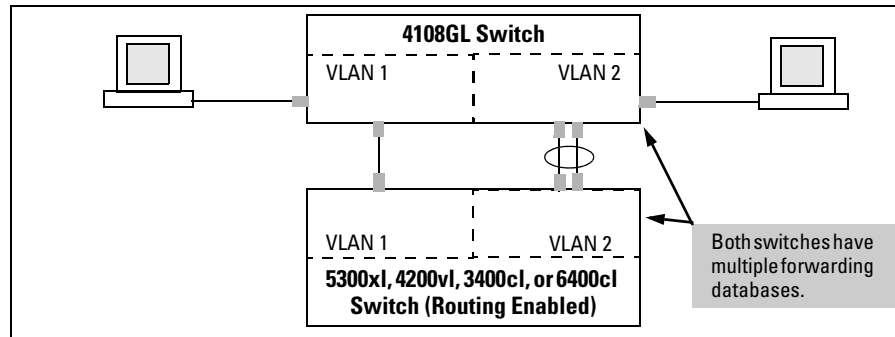
**Figure 12-8. Example of a Valid Topology for Devices Having Multiple Forwarding
Databases in a Multiple VLAN Environment**

The ProCurve switches having a multiple forwarding database are:
shown in Table 12-1.

Table 12-1. Forwarding Database Structure for Managed ProCurve Switches

Multiple-Forwarding Databases*
Series 6400cl switches
Series 6200yl switches
Switch 6108
Series 5400zl switches
Series 5300xl switches
Series 4200 vl switches
Series 4100gl switches
Series 3500yl switches
Series 3400cl switches
Switch 2900yl
Switch 2810
Series 2800 switches
Series 2600/2600-PWR switches
Series 2510 switches
Series 1800 switches

*To determine whether other vendors' devices use single-forwarding or multiple-forwarding database architectures, refer to the documentation provided for those devices.

- If the switch at the other end of the router interface connection *does not have* a multiple forwarding database, you can use only a single interface for the connection. For multiple VLANs, use VLAN tagging.

  To increase the network bandwidth of the connection between the router and the switch, use a trunk of multiple physical links rather than a single physical link.
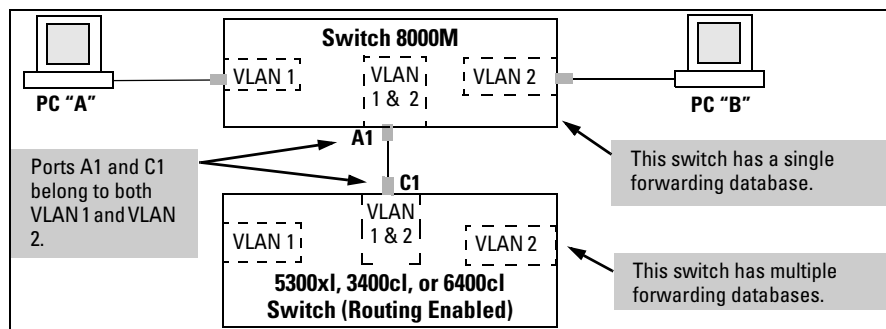
Figure 12-9. Example of a Solution for Single-Forwarding to Multiple-Forwarding Database Devices in a Multiple VLAN Environment

As of this printing, the ProCurve switches that do not have a multiple forwarding database include:

- 1600M, 2400M, 2424M, 4000M, and 8000M switches
- Some older AdvanceStack switches
- Series 2500 switches

For more information, refer to "Multiple VLAN Considerations" on page 2-18.

Configuring XRRP

Configuring XRRP is performed through the switch console CLI at the global configuration level by using the **xrrp** command. Use the **xrrp ?** command to see a list of possible options. You define which VLANs have XRRP configured through the **xrrp instance** command described on page 12-22.

You should first customize the XRRP configuration, as described below, and then enable XRRP, as described on page 12-23. Some of the configuration parameters cannot be changed while XRRP is operational. These are identified in the parameter descriptions below.

Customizing the XRRP Configuration

To customize the XRRP configuration, use any of the following XRRP command options at the CLI global configuration level:

Syntax: xrrp domain < 1-16 >
no xrrp

xrrp [router < 1-2 >]

xrrp failback < 10-999 >

xrrp trap < trap-name | all >

xrrp instance < *owner-router-number* > < *vlan-id* > [advertise < 1-60 > |
authentication < *auth-string* > | ip < *ip-addr*/*mask-length* >]

xrrp domain < 1-16 >

This command sets the XRRP Protection Domain that the router is in. The
router can be in only one domain. The default value is 1. This value cannot
be changed if there is at least one virtual router instance running on the
router. To change the value after XRRP is operating, you must first disable
XRRP (use the **no xrrp** command).

xrrp router < 1-2 >

This command sets the unique number for the router within a given
Protection Domain. No two routers in the same Protection Domain can
have the same router number. The default value is 1.

This value cannot be changed if there is at least one virtual router instance
running on the router. To change the value after XRRP is operating, you
must first disable XRRP (use the **no xrrp** command).

xrrp failback < 10-999 >

This command sets the XRRP fail back time in seconds. The fail back time
is the delay that a router will wait before trying to take back control of all
the XRRP virtual routers it owns after its VLANs come back up. The
default time is 10 seconds.

[no] xrrp trap < *trap-name* | all >

This command enables or disables the generation of SNMP traps for XRRP
events on the router. The following trap names are available:

state-change – signifies that the router has a experienced a state change.
The trap sent would contain the ***domain-number***, ***router-number***, and state
information.

master-transition – signifies that the router state has changed specifically to the master state. The trap sent would contain the ***domain-number***, ***router-number***, and state information.

authentication-failure – signifies that the virtual router instance has received an XRRP packet with an authentication mismatch. The trap sent would contain the ***domain-number***, ***router-number***, and virtual router instance ID (virtual router owner number and VLAN ID) of the virtual router that detected the error.

To enable all the traps, use the command **xrrp trap all**.

To disable the traps, use the **no** form of the command, with the trap name to disable a specific trap or with **all** to disable all the traps. By default, all the traps are disabled.

[no] xrrp instance < *owner-router-number* > < *vlan-id* >

This command configures the virtual router interface on the router. The virtual router interface (XRRP instance) is identified by the ***owner-router-number*** and the ***vlan-id***. The ***owner-router-number*** is the XRRP router number of the router that owns the IP address(es). The ***vlan-id*** identifies the VLAN on which the XRRP instance is running.

Required Parameters – For each router in the Protection Domain, an **xrrp instance** command must be entered for each of the following:

- To create each virtual router interface for the physical router being configured, you would enter an **xrrp instance** command with the router number and the VLAN ID for that interface. For example, to create a virtual router interface in VLAN 5 for the router that has the XRRP router number 1, you would enter the following command:

    xrrp instance 1 5

- To specifically identify the virtual router interfaces on the other router in the Protection Domain, you would enter an **xrrp instance** command with the **ip** parameter. For example, on Router-1 in VLAN 5, to identify the virtual router interface on Router-2 that has the IP address 10.1.1.2 and mask length 24, you would enter the following command:

    xrrp instance 2 5 ip 10.1.1.2/24

For the instance command that creates the virtual router interface on the router being configured, the **ip** parameter must *not* be specified. These XRRP instances, which are being configured on the router that owns the IP address, automatically use the IP address of the VLAN on the router being configured.

Please see the configuration examples on page 12-25 to help clarify these concepts.

- If a VLAN has multiple IP addresses (a multinet situation), an individual IP address can be removed from the XRRP configuration. To remove an IP address from fail-over protection by the router being configured, use the **no** version of the instance command. For example, to remove the virtual interface in the above example from the fail-over protection provided by Router-1, you would enter the following command:

      no xrrp instance 2 5 ip 10.1.1.2/24

    You cannot remove an individual IP address if it is the only IP address associated with the backup router.

Variable Parameters – In addition, the following variable parameters can be specified by the **xrrp instance** command:

- **advertise < 1-60 >** – this parameter determines how quickly standard failover occurs by specifying the frequency, in seconds, that the XRRP Master sends out XRRP advertisement packets. The default is 5, indicating that the Master sends out a packet every 5 seconds. (The standard failover is three times the **advertise** interval.) The default standard failover is 15 seconds.

- **authentication < *auth-string* >** – this parameter sets the string that is used by the virtual router instance for the authentication of the received XRRP packets. The string can be up to 8 characters long. This same string must be configured on all the virtual routers in the Protection Domain that wish to use authentication.

    By default, there is no authentication. Use the **no** version of the command to disable the authentication that was previously enabled on the virtual router interface.

N o t e   For every VLAN on which you wish to run XRRP, you must first configure the VLAN with an IP address.

Enabling and Disabling XRRP

Syntax: [no] xrrp

Once you have completed the XRRP customization, as described in the previous section, use the **xrrp** command by itself to enable XRRP operation on the switch for all VLANs on which XRRP has been configured. Use the **no xrrp** command to disable all XRRP operation on the switch.

Configuration Rules

■   XRRP can be configured only on statically configured IP VLANs. VLANs automatically created by GVRP cannot be used.

■   XRRP cannot be configured on the management VLAN or on any VLAN that gets its IP address through DHCP or Bootp.

■   XRRP must be disabled before the Protection Domain number or the router number configuration can be changed. Use the **no xrrp** command to disable XRRP.

■   Dynamic reconfiguration – You should be aware that although XRRP can be reconfigured while it is running, dynamic configurations can lead to inconsistency between the two routers while the configuration changes are in progress. This will potentially result in error log messages until the configurations are consistent (for example, matched IP addresses for primary on one side and secondary on the other). To avoid these logs, disable XRRP while changing its configuration.

Use the **no xrrp** command to disable XRRP.

Configuration Examples

The following configuration examples create the XRRP setups in the single VLAN and multiple VLAN environments shown in the figures earlier in this chapter.

Configuration for Figure 12-2 – Single VLAN Example

See the figure on page 12-6.

Router-1 Configuration	Explanation
ProCurve (vlan-5)# ip address 10.1.1.1/24	Configures the IP address of the router interface in VLAN 5.
ProCurve (config)# xrrp domain 2	Sets the identity of the Protection Domain.
ProCurve (config)# xrrp router 1	Sets the XRRP router number.
ProCurve (config)# xrrp instance 1 5	Creates the XRRP virtual router interface.
ProCurve (config)# xrrp instance 2 5 ip 10.1.1.2/24	Identifies the virtual router interface on Router-2 for which Router-1 is providing fail-over protection.
ProCurve (config)# xrrp	Enables XRRP operation on Router-1.
ProCurve (config)# write memory	Saves this configuration to startup memory.

Router-2 Configuration	(the explanation is the same as for Router-1)
ProCurve (vlan-5)# ip address 10.1.1.2/24	
ProCurve (config)# xrrp domain 2	
ProCurve (config)# xrrp router 2	
ProCurve (config)# xrrp instance 2 5	
ProCurve (config)# xrrp instance 1 5 ip 10.1.1.1/24	
ProCurve (config)# xrrp	
ProCurve (config)# write memory	

Configuration for Figure 12-4 – Multiple VLANs

See the figure on page 12-9.

Router-1 Configuration	Explanation
ProCurve (vlan-5)# ip address 10.1.1.1/24	Configures the IP address of the router interface in VLAN 5.
ProCurve (vlan-6)# ip address 10.2.1.1/24	Configures the IP address of the router interface in VLAN 6.
ProCurve (config)# xrrp domain 2	Sets the identity of the Protection Domain.
ProCurve (config)# xrrp router 1	Sets the XRRP router number.
ProCurve (config)# xrrp instance 1 5	Creates the XRRP virtual router interface in VLAN 5.
ProCurve (config)# xrrp instance 2 5 ip 10.1.1.2/24	Identifies the virtual router interface on Router-2 for which Router-1 is providing fail-over protection in VLAN 5.
ProCurve (config)# xrrp instance 1 6	Creates the XRRP virtual router interface in VLAN 6.
ProCurve (config)# xrrp instance 2 6 ip 10.2.1.2/24	Identifies the virtual router interface on Router-2 for which Router-1 is providing fail-over protection in VLAN 6.
ProCurve (config)# xrrp	Enables XRRP operation on Router-1.
ProCurve (config)# write memory	Saves this configuration to startup memory.

Router-2 Configuration	(the explanation is the same as for Router-1)
ProCurve (vlan-5)# ip address 10.1.1.2/24	
ProCurve (vlan-6)# ip address 10.2.1.2/24	
ProCurve (config)# xrrp domain 2	
ProCurve (config)# xrrp router 2	
ProCurve (config)# xrrp instance 2 5	
ProCurve (config)# xrrp instance 1 5 ip 10.1.1.1/24	
ProCurve (config)# xrrp instance 2 6	
ProCurve (config)# xrrp instance 1 6 ip 10.2.1.1/24	
ProCurve (config)# xrrp	
ProCurve (config)# write memory	

Displaying XRRP Data

To verify XRRP configuration and for XRRP status and statistics information display, use the following CLI **show xrrp** commands at either the Manager level or the global configuration level:

Syntax: show xrrp traps

This command displays the information on the configured XRRP traps.

```
ProCurve(config)# show xrrp traps

 Status and Counters - XRRP Traps Information

 Trap Name             | Status
 --------------------- + --------
 state-change          | Enabled
 master-transition     | Disabled
 authentication-failure | Disabled
```

Figure 12-10.Example of Output for Show XRRP Traps

Syntax: show xrrp config [global | instance [< *owner-router-num* > < *vlan-id* >]]

This command displays XRRP configuration information. Invoked without parameters it shows global and virtual routers configuration on the switch.

If the **global** keyword is specified, then the generic configuration information is displayed. (instances).

```
ProCurve(config)# show xrrp config global

 Status and Counters - XRRP Global Configuration Information

  XRRP Enabled   : Yes
  Domain Number  : 2
  Router Number  : 1
  Failback Delay : 11
  Infinite Failback: Disabled
```

Figure 12-11.Example of Output for Show XRRP Config Global

The display also includes all owner router information (instances). Global information for XRRP is excluded.

The keyword **instance** can be used to display configuration information for the virtual router instance(s). If no parameters are specified after this keyword, the information for all virtual routers is displayed, otherwise the information for the particular virtual router is displayed by specifying the *owner-router-number* and the *vlan-id* in the command. In the example below, the configuration information for the virtual router number 1 on VLAN 5 is requested.

```
ProCurve(config)# show xrrp config instance 1 5

 Status and Counters - XRRP Virtual Router Configuration Information

  Owner Router Number : 1
  VLAN ID             : 5
  Authentication Type : Simple Text Password
  Authentication Key  : password
  Advertise Interval  : 5

  IP Address      Subnet Mask
  --------------- ---------------
  10.1.1.1        255.255.248.0
  10.2.1.1        255.255.248.0
```

Figure 12-12.Example of Output for Show XRRP Config Instance [<*owner-router-number*> <*vlan-id*>]

Syntax: show xrrp statistics [global |
　　　　　　　　instance [< *owner-router-num* > < *vlan-id* >] |
　　　　　　　　router < *router-num* >]

This command displays XRRP status and statistics information.

If the keyword **global** is used, then generic information is displayed:

```
ProCurve(config)# show xrrp statistics global

 Status and Counters - XRRP Global Statistics Information

  XRRP Enabled       : Yes
  This Domain Number : 2
  This Router Number : 1
  XRRP MAC Addr      : 0001e7-940601
  XRRP AND Mask      : ffffff-ffffff
  XRRP Up Time       : 46 hours

  Pkts Rx      Corrupt Pkts Bad Version  Bad Chksum   Not Domain
  ------------ ------------ ------------ ------------ -------------
  7            0            0            0            0
```

Figure 12-13.Example of Output for Show XRRP Statistics Global

The keyword **instance** can be used to display statistics information for the virtual router instance(s) on the switch. If no parameters are specified after this keyword, the information for all virtual routers is displayed, otherwise the information for the particular virtual router is displayed by specifying the *owner-router-number* and the *vlan-id* in the command. In the example below, the statistics information for the virtual router number 1 on VLAN 5 is requested.

```
ProCurve(config)# show xrrp statistics instance 1 5

 Status and Counters - XRRP Virtual Router Statistics Information

  Owner Router Number : 1
  VLAN ID             : 5
  Operational State   : Master
  Up Time             : 64 mins

  Pkts Rx            : 0              Pkts Tx                : 780
  Zero Priority Rx   : 0              Zero Priority Tx       : 0
  Bad Version Pkts   : 0              Mismatched Pswd Pkts   : 0
  Mismatched IP Pkts : 0              Mismatched Interval Pkts : 0
```

Figure 12-14.Example of Output for Show XRRP Statistics Instance [<*owner-router-number*> <*vlan-id*>]

The keyword **router** can be used to display statistics information for the specific router coordinator operating in the XRRP domain as shown in the next example.

```
ProCurve(config)# show xrrp statistics router 2

 Status and Counters - XRRP Router Coordinator Statistics Information

  Router Number : 2
  Become Master : 1
  Master Time   : 76 mins

  Type1        Type1        Type2        Type2        Unknown
  Pkts Rx      Pkts Tx      Pkts Rx      Pkts Tx      VLAN ID
  ------------ ------------ ------------ ------------ ------------
  0            924          0            0            0
```

Figure 12-15.Example of Output for Show XRRP Statistics Router *<router-num>*

Comparison Between XRRP and VRRP

The following information compares the characteristics of XRRP and the industry standard VRRP.

■ XRRP will allow a router to respond to SNMP requests on the virtual router IP address even if it is not the owner. VRRP does not. This would allow you to still access the failed router on VLANs that are accessible on that router.

■ XRRP uses the same MAC address for each virtual router owned by a given physical router. VRRP uses a separate MAC address per virtual router.

■ XRRP uses a fail-over domain concept with up to 2 routers in the fail-over domain and up to 16 domains connected to a given VLAN. VRRP uses a flat space with up to 255 virtual routers in a level 2 switch fabric. However these 255 virtual routers can be used over on every VLAN with VRRP.

■ XRRP will warn you of mismatched configurations between the routers but will attempt to use the current master configuration whenever possible when these mismatches occur.

■ VRRP fails over at the virtual router level allowing a given physical router to continue to route packets on those virtual routers that it still owns. XRRP will fail-over at the router level. If one of the virtual routers controlled by a physical router fails, then all the virtual routers that it owns will be taken over by the other router in the same XRRP Protection Domain.

■ XRRP has fast fail-over. VRRP does not.

Messages Related to XRRP Operation

These messages appear in the Event Log and, if Syslog Debug is configured, in the designated Debug destinations.

Message	Meaning
`Unable to alloc a msg buffer from routine` < *routine name* >	Indicates that a message buffer could not be allocated. Although XRRP can handle this event in this case, it does indicate that the system is critically low on message buffers and will probably crash soon.
`Duplicate IP address` < *ip-addr* > `detected`	Indicates that a duplicate IP address has been detected on the network for an enabled secondary address. XRRP will relinquish control of the address to correct the duplicate IP address situation. This can happen when XRRP is disabled on one router, but not disabled on its peer.
`IP address` < *ip-addr* > `in use, XRRP cannot take over`	Indicates that some other device on the network currently has the IP address configured and XRRP wanted to take control of this secondary address. XRRP will not take over the address until this situation has been corrected so as to avoid the creation of a duplicate IP address on the network. It will however check at the failover period (3 times the advertisement interval) if the situation had been corrected. **Note:** this is only done for secondary addresses.
`Cannot add MCAST address` < *xx-xx-xx-xx-xx-xx* >	Indicates that XRRP initialization has failed since the Multicast receive address could not be configured. This is a critical internal error and should not occur on a healthy switch.
`No IP address configured for VR on rtr` < *router-num* >, `vid` < *vid-number* >	Indicates that the Virtual router interface on the displayed router and VID does not have an IP address configured. This can happen to virtual router interfaces for secondary addresses, when the primary address has been changed. This error will force the remote miss-configuration flag to be set so fail-over will only occur when a complete router failure occurs. (This indicates router miss-configuration.)

Message	Meaning
`Failed to alloc a pkt buf for an XRRP pkt from` < *routine-name* >	Indicates that XRRP was not able to allocate a packet for transmission. This indicates that the system is critically low on resources.
`Pkt rcvd that was too short, len =` < *packet-len* >`, min =` < *min-length-allowed* >	Indicates that XRRP received an XRRP packet that was too short. This event increments the global corrupt packet counter.
`Pkt rcvd with a checksum error from rtr` < *router-num* >	Indicates that XRRP received an XRRP packet that contained a checksum error
`Pkt rcvd with an illegal domain number of` < *domain-num* >	Indicates that XRRP received an XRRP packet that contained an illegal domain number (domain must be between 1 and 16).
`Pkt rcvd with an illegal rtr number of < router-num > in domain` < *domain-num* >`.`	This indicates that XRRP received an XRRP packet that contained an illegal router number (router must be between 1 and 2)
`Pkt rcvd with dup number of` < *router-num* > `in domain` < *domain-num* >	Indicates that XRRP received an XRRP packet that contained a the local routers router number. (This indicates router miss-configuration.)
`Rcvd its own pkt back (network loop) on rtr` < *router-num* >	Indicates that XRRP received a packet sent out by this router (network loop problem).
`Rcvd pkt with version number` < *version-num* > `expected` < *version-num* >	Indicates that XRRP received a packet with the wrong XRRP version number.
`Rcvd a pkt from rtr` < *router-num* >`, owner` < *router-num* >`, vid` < *vid-num* >`, no such VR`	Indicates that XRRP received a packet that doesn't reference any virtual routers on this router. This error also occurs if the VID received is not configured on this router which would also not have the virtual router configured. This error will not force the remote miss-configuration flag. Fail-over can still occur from the router where the VR is not configured to the router that has the VR configured so long as a duplicate IP address doesn't exist.(This indicates a configuration error.)
`Rcvd a pkt from rtr` < *router-num* >`, with unknown pkt type` < *packet-type* >	Indicates that XRRP received a packet with an unknown XRRP packet type.
`Rcvd a pkt from rtr` < *router-num* >`, with ver/type =` < *version-num* >`, expected` < *version-num* >	Indicates that XRRP received a packet with an unknown VRRP version or type number in the VRRP portion of the packet.

Message	Meaning
`No local IP addr <`*IP-address-in-hex*`> from rtr <`*router-num*`>, on <`*vid-num*`>.`	Indicates that XRRP received a packet with an IP address that doesn't match any of the configured IP addresses on the associated virtual router. This error will force the remote miss-configuration flag to be set so fail-over will only occur when a complete router failure occurs. (Indicates a configuration error.)
`Rtr <`*router-num*`>, has <`*ip-address-count*`> IP addrs on vid <`*vid-num*`/>,local rtr has <`*ip-address-count*`>`	Indicates that XRRP received a packet with an IP address count that doesn't match the local configuration. This error will not force the remote miss-configuration flag to be set so fail-over can still occur. (Indicates a configuration error.)
`Rcvd a pkt from rtr <`*router-num*`>on vid <`*vid-num*`>with the wrong password`	This indicates that XRRP received a packet with an authentication error. This error will not force the remote miss-configuration flag to be set so fail-over can still occur. (Indicates a configuration error.)
`Rcvd pkt from rtr <`*router-num*`>on vid <`*vid-num*`>with a msmtchd ad int`	Indicates that XRRP received a packet with a mismatched advertisement interval. This error will not force the remote miss-configuration flag to be set so fail-over can still occur. (Indicates a configuration error.)
`No VR instances for the local rtr <`*router-num*`>`	Indicates that XRRP does not have the local router configured with any virtual router instances. No Fail-over on this router is possible. (Indicates a configuration error.)
`No VR instances for the remote rtr <`*router-num*`>`	Indicates that XRRP does not have the remote (non-owner) router configured with any virtual router instances. This error will not force the remote miss-configuration flag to be set however this router has no information on how to fail-over for the remote router. (Indicates a configuration error.)
`Rtr <`*router-num*`> has taken <`*primary/ secondary*`>IP address control`	This informational message indicates that XRRP has taken control of either the primary or secondary address.
`Rtr <`*router-num*`>has relinquished <`*primary/secondary*`> IP address control`	This informational message indicates that XRRP has taken released control of either the primary or secondary address.

Message	Meaning
Remote rtr < *router-num* > domain < *domain-num* > is miss-configured	Indicates that the remote router is miss-configured relative to the local router. This condition will prevent fail-over except when complete router failure has occurred. Both routers must agree on the configuration. (**Note:** if one router never comes up then remote miss-configuration is not detected until the remote router does come up.)
PKT received on VID < *vid-#* > sent on VID < *vid-#* >	This message indicates that an XRRP message was received on the wrong VLAN. This implies that the wires have been cross connected so that two untagged but different VLANs have been interconnected.
< *local/remote* > rtr < *router-num* > configuration corrected	Indicates that the router misconfiguration has been corrected. The message only occurs if a miss-configuration had been previously detected. If a local miss-configuration been corrected then the message will start with **Local**. If the remote router configuration has been corrected then the message will start with **Remote**. All miss-configuration errors must be corrected (Local or Remote) for this message to occur. **Note:** Detection of **Remote** miss-configurations will time out if the remote router stops sending XRRP packets.
Local rtr missing secondary VR on Vid < *vid-#* >	Indicates that an XRRP router has a primary VR without a matching secondary VR. This configuration error would prevent the router from taking control of the indicated VLAN in the event of a failover. This error may occur if the user dynamically adds a primary VR without adding a secondary VR before the primary sends out a packet, or if a secondary VR is removed and the primary is left configured. To avoid this message, disable XRRP before adding and or deleting VRs. **Note:** As long as the configuration is corrected so that each primary VR has a secondary, the protocol will correct any temporary miss-configuration. This error will force the remote miss-configuration flag to be set so failover will only occur when a complete router failure occurs. (Indicates a configuration error.)

— This page is intentionally unused—

13

Stack Management for the Series 3400cl, 6400cl, and 4200vl Switches

Contents

Introduction to Stack Management on Series 3400cl, 6400cl and 4200vl Switches

ProCurve Stack Management (*stacking*) enables you to use a single IP address and standard network cabling to manage a group of up to 16 total switches in the same IP subnet (broadcast domain). Using stacking, you can:

■   Reduce the number of IP addresses needed in your network.

■   Simplify management of small workgroups or wiring closets while scaling your network to handle increased bandwidth demand.

■   Eliminate any specialized cables for stacking connectivity and remove the distance barriers that typically limit your topology options when using other stacking technologies.

■   Add switches to your network without having to first perform IP addressing tasks.

Stacking Support on ProCurve Switches

As of August 2006, the following ProCurve switches include stacking:

■ ProCurve Series 6400cl   ■ ProCurve Series 2800

■ ProCurve Series 6200yl   ■ ProCurve Series 2600

■ ProCurve Switch 6108   ■ ProCurve Switch 2510

■ ProCurve Series 5400zl   ■ ProCurve Series 2500

■ ProCurve Series 5300xl   ■ ProCurve Switch 8000M[1, 2]

■ ProCurve Series 4200vl   ■ ProCurve Switch 4000M[1, 2]

■ ProCurve Series 4100gl   ■ ProCurve Switch 2424M1[1, 2]

■ ProCurve Series 3400cl   ■ ProCurve Switch 2400M[1, 2]

■ ProCurve Switch 2900yl   ■ ProCurve Switch 1600M[1, 2]

■ ProCurve Switch 2810

[1]Requires software release C.08.03 or later, which is included with the 8000M, 4000M, 2424M, and 1600M models as of July, 2000. Release C.08.03 or a later version is also available on the ProCurve Networking web site at **www.procurve.com**. (Click on **Software updates**.)

[2]Discontinued product.

N o t e

Stacking and meshing cannot both be enabled at the same time on a Series 3400cl or Series 6400cl switch.

In the default configuration, stacking is enabled on the 3400cl, 6400cl and 4200vl switches.

Summary of Stacking Features

Feature	Default	Menu	CLI	Web
view stack status				
view status of a single switch	n/a	page 13-26 thru page 13-28	page 13-31	Refer to Online Help
view candidate status	n/a		page 13-31	
view status of commander and its stack	n/a		page 13-32	
view status of all stacking-enabled switches in the ip subnet	n/a		page 13-32	
configure stacking				
enable/disable candidate Auto-Join	enabled/Yes	page 13-15	page 13-37	
"push" a candidate into a stack	n/a	page 13-15	page 13-37	
configure a switch to be a commander	n/a	page 13-13	page 13-33	
"push" a member into another stack	n/a	page 13-24	page 13-39	
remove a member from a stack	n/a	page 13-21	page 13-40 or page 13-41	
"pull" a candidate into a stack	n/a	page 13-17	page 13-36	
"pull" a member from another stack	n/a	page 13-19	page 13-38	
convert a commander or member to a member of another stack	n/a	page 13-24	page 13-39	
access member switches for configuration and traffic monitoring	n/a	page 13-23	page 13-42	
disable stacking	enabled	page 13-15	page 13-44	
transmission interval	60 seconds	page 13-13	page 13-44	

Components of ProCurve Stack Management

Table 13-1. Stacking Definitions

Stack	Consists of a Commander switch and any Member switches belonging to that Commander's stack.
Commander	A switch that has been manually configured as the controlling device for a stack. When this occurs, the switch's stacking configuration appears as **Commander**.
Candidate	A switch that is ready to join (become a Member of) a stack through either automatic or manual methods. A switch configured as a Candidate is not in a stack.
Member	A switch that has joined a stack and is accessible from the stack Commander.

Before: Stack named "Engineering" consists of Commander and Switch "C". Switch "B" is a Candidate eligible to join the stack.

After: Switch "B" joins the stack, thus changing from a Candidate to a Member of the stack.

Stack

Stack Name:
Engineering

Commander: Switch A

Member: Switch C

Candidate: Switch B

Stack Name:
Engineering

Commander: Switch A

Member: Switch C
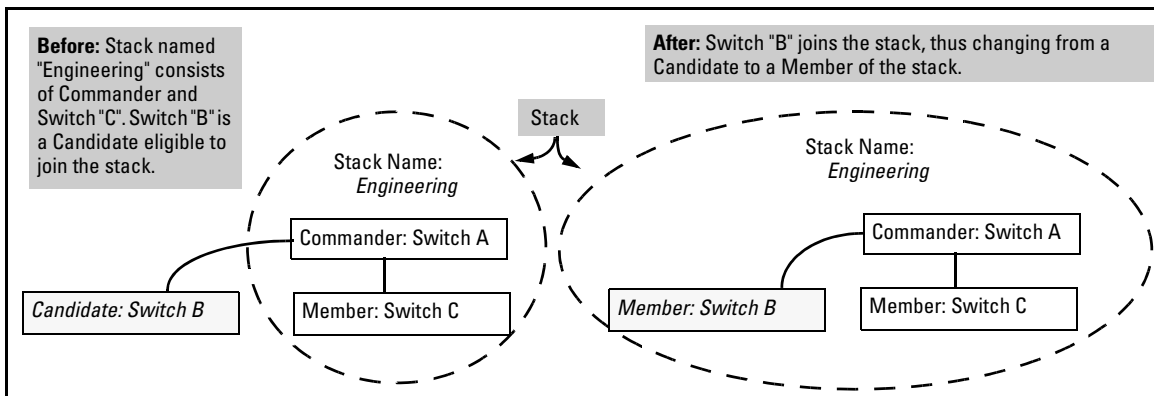
Member: Switch B

Figure 13-1. Illustration of a Switch Moving from Candidate to Member

General Stacking Operation

After you configure one switch to operate as the Commander of a stack, additional switches can join the stack by either automatic or manual methods. After a switch becomes a Member, you can work through the Commander switch to further configure the Member switch as necessary for all of the additional software features available in the switch.

The Commander switch serves as the in-band entry point for access to the Member switches. For example, the Commander's IP address becomes the path to all stack Members and the Commander's Manager password controls access to all stack Members.

Use the Commander's console or web browser interface to access the user interface on any Member switch in the same stack.	**Wiring Closet "A"**	
	Member Switch 1 IP Address: *None Assigned* Manager Password: leader	**Candidate Switch** IP Address: *None Assigned* Manager Password: francois
Network Backbone		
	Wiring Closet "B"	
Commander Switch 0 IP Address: 10.28.227.100 Manager Password: leader	**Non-Member Switch** IP Address: 10.28.227.105 Manager Password: donald	**Member Switch 2** IP Address: *None Assigned* Manager Password: leader

Figure 13-2. Example of Stacking with One Commander Controlling Access to Wiring Closet Switches

Interface Options. You can configure stacking through the switch's menu interface, CLI, or the web browser interface. For information on how to use the web browser interface to configure stacking, see the online Help for the web browser interface.

Web Browser Interface Window for Commander Switches. The web browser interface window for a Commander switch differs in appearance from the same window for non-commander switches.

Operating Rules for Stacking

General Rules

- Stacking is an optional feature (enabled in the default configuration) and can easily be disabled. Stacking has no effect on the normal operation of the switch in your network.

- A stack requires one Commander switch. (Only one Commander allowed per stack.)

- All switches in a particular stack must be in the same IP subnet (broadcast domain). A stack cannot cross a router.

- A stack accepts up to 16 switches (numbered 0-15), including the Commander (always numbered 0).

- The stacking feature supports up to 100 switches in the same IP subnet (broadcast domain). A switch can belong to only one stack. In the event that the 100 switch limit is exceeded, it may take multiple attempts to add or move a member to any given stack. Once a member is added to a stack, it is not "forgotten" by the Commander.

- Viewing Stack Status (all) only displays up to 100 devices. Devices that are not members of a given stack may periodically "drop out" of the list when viewing Stack Status (all).

- If multiple VLANs are configured, stacking uses only the primary VLAN on any switch. In the factory-default configuration, the DEFAULT_VLAN is the primary VLAN. (See "Stacking Operation with Multiple VLANs Configured" on page 13-44 and "The Primary VLAN" on page 2-45.)

- Stacking allows intermediate devices that do not support stacking. This enables you to include switches that are distant from the Commander.
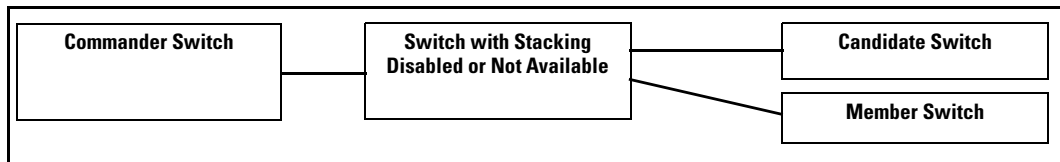
Figure 13-3. Example of a Non-Stacking Device Used in a Stacking Environment

Specific Rules

Table 13-2. Specific Rules for Commander, Candidate, and Member Switch

	IP Addressing and Stack Name	Number Allowed Per Stack	Passwords	SNMP Communities
Commander	**IP Addr:** Requires an assigned IP address and mask for access via the network. **Stack Name:** Required	Only one Commander switch is allowed per stack.	The Commander's Manager and Operator passwords are assigned to any switch becoming a Member of the stack. If you change the Commander's passwords, the Commander propagates the new passwords to all stack Members.	Standard SNMP community operation. The Commander also operates as an SNMP proxy to Members for all SNMP communities configured in the Commander.
Candidate	**IP Addr:** Optional. Configuring an IP address allows access via Telnet or web browser interface while the switch is not a stack member. In the factory default configuration the switch automatically acquires an IP address if your network includes DHCP service. **Stack Name:** N/A	n/a	Passwords optional. If the Candidate becomes a stack Member, it assumes the Commander's Manager and Operator passwords. If a candidate has a password, it cannot be automatically added to a stack. In this case, if you want the Candidate in a stack, you must manually add it to the stack.	Uses standard SNMP community operation if the Candidate has its own IP addressing.
Member	**IP Addr:** Optional. Configuring an IP address allows access via Telnet or web browser interface without going through the Commander switch. This is useful, for example, if the stack Commander fails and you need to convert a Member switch to operate as a replacement Commander. **Stack Name:** N/A	Up to 15 Members per stack.	When the switch joins the stack, it automatically assumes the Commander's Manager and Operator passwords and discards any passwords it may have had while a Candidate. **Note:** If a Member leaves a stack for any reason, it retains the passwords assigned to the stack Commander at the time of departure from the stack.	Belongs to the same SNMP communities as the Commander (which serves as an SNMP proxy to the Member for communities to which the Commander belongs). To join other communities that *exclude* the Commander, the Member must have its own IP address. Loss of stack membership means loss of membership in any community that is configured only in the Commander. See "SNMP Community Operation in a Stack" on page 13-43.

N o t e   In the default stack configuration, the Candidate **Auto Join** parameter is
enabled, but the Commander **Auto Grab** parameter is disabled. This prevents
Candidates from automatically joining a stack prematurely or joining the
wrong stack (if more than one stack Commander is configured in a subnet or
broadcast domain). If you plan to install more than one stack in a subnet, HP
recommends that you leave **Auto Grab** disabled on all Commander switches
and manually add Members to their stacks. Similarly, if you plan to install a
stack in a subnet (broadcast domain) where stacking-capable switches are
not intended for stack membership, you should set the **Stack State** parameter
(in the Stack Configuration screen) to **Disabled** on those particular switches.

Configuring Stack Management

Overview of Configuring and Bringing Up a Stack

This process assumes that:

■   All switches you want to include in a stack are connected to the same
        subnet (broadcast domain).

■   If VLANs are enabled on the switches you want to include in the stack,
        then the ports linking the stacked switches must be on the primary
        VLAN in each switch (which, in the default configuration, is the
        default VLAN). If the primary VLAN is tagged, then each switch in the
        stack must use the same VLAN ID (VID) for the primary VLAN. (Refer
        to "The Primary VLAN" on page 2-45, and "Stacking Operation with
        Multiple VLANs Configured" on page 13-44.)

■   *If you are including a ProCurve Switch 8000M, 4000M, 2424M,
        2400M, or 1600M in a stack, you must first update all such devices
        to software version C.08.03 or later.* (You can get a copy of the latest
        software version from the ProCurve Networking web site and/or copy
        it from one switch to another. For downloading instructions, see
        appendix A, "File Transfers", in the *Management and Configuration
        Guide* for these switch models.)

Options for Configuring a Commander and Candidates. Depending on how Commander and Candidate switches are configured, Candidates can join a stack either automatically or by a Commander manually adding ("pulling") them into the stack. In the default configuration, a Candidate joins only when *manually* pulled by a Commander. You can reconfigure a Commander to *automatically* pull in Candidates that are in the default stacking configuration. You can also reconfigure a Candidate switch to either "push" itself into a particular Commander's stack, convert the Candidate to a Commander (for a stack that does not already have a Commander), or to operate as a standalone switch without stacking. The following table shows your control options for adding Members to a stack.

Table 13-3. Stacking Configuration Guide

Join Method[1]	Commander (IP Addressing Required)	Candidate (IP Addressing Optional)	
	Auto Grab	Auto Join	Passwords
Automatically add Candidate to Stack (Causes the first 15 eligible, discovered switches in the subnet to automatically join a stack.)	**Yes**	**Yes** *(default)*	No *(default)*[*]
Manually add Candidate to Stack (Prevent automatic joining of switches you don't want in the stack)	**No** *(default)*	**Yes** *(default)*	Optional[*]
	Yes	**No**	Optional[*]
	Yes	**Yes** *(default)* or **No**	Configured
Prevent a switch from being a Candidate	**N/A**	**Disabled**	Optional

[*]The Commander's Manager and Operator passwords propagate to the candidate when it joins the stack.

The easiest way to *automatically* create a stack is to:

1. Configure a switch as a Commander.

2. Configure IP addressing and a stack name on the Commander.

3. Set the Commander's **Auto Grab** parameter to **Yes**.

4. Connect Candidate switches (in their factory default configuration) to the network.

This approach automatically creates a stack of up to 16 switches (including the Commander). However this replaces manual control with an automatic process that may bring switches into the stack that you did not intend to include. With the Commander's **Auto Grab** parameter set to **Yes**, *any switch* conforming to all four of the following factors automatically becomes a stack Member:

■   Default stacking configuration (**Stack State** set to **Candidate**, and **Auto Join** set to **Yes**)

■   Same subnet (broadcast domain) and default VLAN as the Commander (If VLANs are used in the stack environment, see "Stacking Operation with a Tagged VLAN" on page 13-44.)

■   No Manager password

■   14 or fewer stack members at the moment

General Steps for Creating a Stack

This section describes the general stack creation process. For the detailed configuration processes, see pages 13-13 through 13-36 for the menu interface and pages 13-29 through 13-41 for the CLI.

1.   Determine the naming conventions for the stack. You will need a stack name. Also, to help distinguish one switch from another in the stack, you can configure a unique system name for each switch. Otherwise, the system name for a switch appearing in the Stacking Status screen appears as the stack name plus an automatically assigned switch number. For example:

```
                            Pacific Ocean
=========================- CONSOLE - MANAGER MODE -=============================
                     Stacking - Stacking Status (All)

        Stack Name             MAC Address      System Name          Status
   --------------------     ------------   ----------------   --------------------
   Big Waters                 0060b0-880a80  Pacific Ocean      Commander Up
                              0060b0-df1a00  Coral Sea          Member Up

   Online                     0060b0-df7680  online-0           Commander Up
                              001083-3c7480  online-1           Member Up
                              0060b0-312f00  online-2           Member Up
                              001083-3c09c0  online-3           Member Up


   Actions->   Back      Next page     Prev page     Help

 Return to previous screen.
 Use up/down arrow keys to scroll to other entries, left/right arrow keys to
 change action selection, and <Enter> to execute action.
```

For status descriptions, see the table on page 13-45.

Stack with unique system name for each switch.

Stack named "Online" with no previously configured system names assigned to individual switches.

Figure 13-4.   Using the System Name to Help Identify Individual Switches

2. Configure the Commander switch. Doing this first helps to establish consistency in your stack configuration, which can help prevent startup problems.

   • A stack requires one Commander switch. If you plan to implement more than one stack in a subnet (broadcast domain), the easiest way to avoid unintentionally adding a Candidate to the wrong stack is to manually control the joining process by leaving the Commander's **Auto Grab** parameter set to **No** (the default).

   • The Commander assigns its Manager and Operator passwords to any Candidate switch that joins the stack.

   • The Commander's SNMP community names apply to members.

3. For automatically or manually pulling Candidate switches into a stack, you can leave such switches in their default stacking configuration. If you need to access Candidate switches through your network before they join the stack, assign IP addresses to these devices. Otherwise, IP addressing is optional for Candidates and Members. (Note that once a Candidate becomes a member, you can access it through the Commander to assign IP addressing or make other configuration changes.)

4. Make a record of any Manager passwords assigned to the switches (intended for your stack) that are not currently members. (You will use these passwords to enable the protected switches to join the stack.)

5. If you are using VLANs in the stacking environment, you must use the default VLAN for stacking links. For more information, see "Stacking Operation with a Tagged VLAN" on page 13-44.

6. Ensure that all switches intended for the stack are connected to the same subnet (broadcast domain). As soon as you connect the Commander, it will begin discovering the available Candidates in the subnet.

   • If you configured the Commander to automatically add Members (**Auto Grab** = **Yes**), the first fifteen discovered Candidates meeting both of the following criteria will automatically join the stack:

     – **Auto Join** parameter set to **Yes** (the default)

     – Manager password not configured

   • If you configured the Commander to manually add Members (**Auto Grab** set to **No**—the default), you can begin the process of selecting and adding the desired Candidates.

7. Ensure that all switches intended for the stack have joined.

8. If you need to do specific configuration or monitoring tasks on a Member, use the console interface on the Commander to access the Member.

Using the Menu Interface To View Stack Status and Configure Stacking

Using the Menu Interface To View and Configure a Commander Switch

1. Configure an IP address and subnet mask on the Commander switch. (Refer to the *Management and Configuration Guide* for your switch.)

2. Display the Stacking Menu by selecting **Stacking** in the Main Menu.

```
                              DEFAULT_CONFIG

=========================- CONSOLE - MANAGER MODE -=========================
                             Stacking Menu

      1. Stacking Status (This Switch)
      2. Stacking Status (All)
      3. Stack Configuration
      0. Return to Main Menu...



Shows the status of Stack.
To select menu item, press item number, or highlight item and press <Enter>.
```

Figure 13-5. The Default Stacking Menu

3. Display the Stack Configuration menu by pressing [**3**] to select **Stack Configuration**.

```
                              DEFAULT_CONFIG


=========================- CONSOLE - MANAGER MODE -=========================
                      Stacking - Stack Configuration


   Stack State : Candidate
   Auto Join [Yes] : Yes
   Transmission Interval [60] : 60



  Actions->    Cancel      Edit     Save     Help

Cancel changes and return to previous screen.
Use arrow keys to change action selection and <Enter> to execute action.
```

Figure 13-6. The Default Stack Configuration Screen

4. Move the cursor to the Stack State field by pressing **[E]** (for **Edit**). Then use the Space bar to select the **Commander** option.

5. Press the downarrow key to display the Commander configuration fields in the Stack Configuration screen.

```
                              DEFAULT_CONFIG

==========================- CONSOLE - MANAGER MODE -============================
                        Stacking - Stack Configuration

   Stack State : Commander
   Stack Name :
   Auto Grab [No] : No
   Transmission Interval [60] : 60


  Actions->    Cancel      Edit      Save      Help



Use arrow keys to change field selection, <Space> to toggle field choices,
and <Enter> to go to Actions.
```

Figure 13-7. The Default Commander Configuration in the Stack Configuration Screen

6. Enter a unique stack name (up to 15 characters; no spaces) and press the downarrow key.

7. Ensure that the Commander has the desired **Auto Grab** setting, then press the downarrow key:

   • **No** (the default) prevents automatic joining of Candidates that have their **Auto Join** set to **Yes**.

   • **Yes** enables the Commander to automatically take a Candidate into the stack as a Member if the Candidate has **Auto Join** set to **Yes** (the default Candidate setting) and does not have a previously configured password.

8. Accept or change the transmission interval (default: 60 seconds), then press **[Enter]** to return the cursor to the **Actions** line.

9. Press **[S]** (for **Save**) to save your configuration changes and return to the Stacking menu.

Your Commander switch should now be ready to automatically or manually acquire Member switches from the list of discovered Candidates, depending on your configuration choices.

Using the Menu To Manage a Candidate Switch

Using the menu interface, you can perform these actions on a Candidate switch:

■ Add ("push") the Candidate into an existing stack

■ Modify the Candidate's stacking configuration (**Auto Join** and **Transmission Interval**)

■ Convert the Candidate to a Commander

■ Disable stacking on the Candidate so that it operates as a standalone switch

In its default stacking configuration, a Candidate switch can either automatically join a stack or be manually added ("pulled") into a stack by a Commander, depending on the Commander's **Auto Grab** setting. The following table lists the Candidate's configuration options:

Table 13-4. Candidate Configuration Options in the Menu Interface

Parameter	Default Setting	Other Settings
Stack State	Candidate	Commander, Member, or Disabled
Auto Join	Yes	No
Transmission Interval	60 Seconds	Range: 1 to 300 seconds

Using the Menu To "Push" a Switch Into a Stack, Modify the Switch's Configuration, or Disable Stacking on the Switch. Use Telnet or the web browser interface to access the Candidate if it has an IP address. Otherwise, use a direct connection from a terminal device to the switch's console port. (For information on how to use the web browser interface, see the online Help provided for the browser.)

1. Display the Stacking Menu by selecting **Stacking** in the console Main Menu.

2. Display the Stack Configuration menu by pressing **[3]** to select **Stack Configuration**.

```
                             DEFAULT_CONFIG

=========================- CONSOLE - MANAGER MODE -=============================
                       Stacking - Stack Configuration

  Stack State : Candidate
  Auto Join [Yes] : Yes
  Transmission Interval [60] : 60



  Actions->   Cancel      Edit      Save      Help

 Cancel changes and return to previous screen.
 Use arrow keys to change action selection and <Enter> to execute action.
```
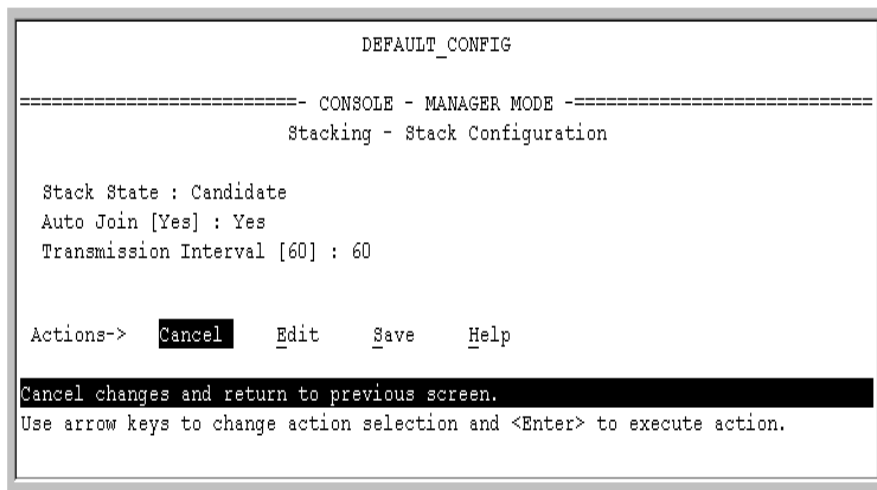
Figure 13-8. The Default Stack Configuration Screen

3. Move the cursor to the Stack State field by pressing **[E]** (for **Edit**).

4. Do one of the following:

    - To disable stacking on the Candidate, use the Space bar to select the **Disabled** option, then go to step 5.

      Note: Using the menu interface to disable stacking on a Candidate removes the Candidate from all stacking menus.

    - To insert the Candidate into a specific Commander's stack:

      i. Use the space bar to select Member.

      ii. Press **[Tab]** once to display the **Commander MAC Address** parameter, then enter the MAC address of the desired Commander.

    - To change **Auto Join** or **Transmission Interval**, use **[Tab]** to select the desired parameter, and:

      – To change **Auto Join**, use the Space bar.

      – To change **Transmission Interval**, type in the new value in the range of 1 to 300 seconds.

         Note: All switches in the stack must be set to the same transmission interval to help ensure proper stacking operation. HP recommends that you leave this parameter set to the default 60 seconds.

    Then go to step 5.

5. press **[Enter]** to return the cursor to the **Actions** line.

6. Press **[S]** (for **Save**) to save your configuration changes and return to the Stacking menu.

Using the Commander To Manage The Stack

The Commander normally operates as your stack manager and point of entry into other switches in the stack. This typically includes:

■ Adding new stack members

■ Moving members between stacks

■ Removing members from a stack

■ Accessing stack members for individual configuration changes and traffic monitoring

The Commander also imposes its passwords on all stack members and provides SNMP community membership to the stack. (See "SNMP Community Operation in a Stack" on page 13-43.)

Using the Commander's Menu To Manually Add a Candidate to a Stack. In the default configuration, you must manually add stack Members from the Candidate pool. Reasons for a switch remaining a Candidate instead of becoming a Member include any of the following:

■ **Auto Grab** in the Commander is set to **No** (the default).

■ **Auto Join** in the Candidate is set to **No**.

    Note: When a switch leaves a stack and returns to Candidate status, its **Auto Join** parameter resets to **No** so that it will not immediately rejoin a stack from which it has just departed.

■ A Manager password is set in the Candidate.

■ The stack is full.

Unless the stack is already full, you can use the Stack Management screen to manually convert a Candidate to a Member. If the Candidate has a Manager password, you will need to use it to make the Candidate a Member of the stack.

1. To add a Member, start at the Main Menu and select:

    9. Stacking...

        4. Stack Management

    You will then see the Stack Management screen:

For status descriptions, see the table on page 13-45.

```
                              Pacific Ocean

==========================- CONSOLE - MANAGER MODE -==============================
                        Stacking - Stack Management

  SN    MAC Address       System Name       Device Type          Status
  --    -------------     ----------------   -----------     -----------------------
  1     0060b0-df1a00     Coral Sea          3400cl-48G      Member Up
  2     080009-8c5080     North Atlantic     HP 2824         Member Up



  Actions->    Back      Add      Edit      Delete      Help

Return to previous screen.
Use up/down arrow keys to change record selection, left/right arrow keys to
change action selection, and <Enter> to execute action.
```

Figure 13-9. Example of the Stack Management Screen

2. Press **[A]** (for **Add**) to add a Candidate. You will then see this screen listing the available Candidates:

```
                              Pacific Ocean
==========================- CONSOLE - MANAGER MODE -==============================
                        Stacking - Stack Management

  Switch Number : 3             The Commander automatically selects an
  MAC Address :                 available switch number (SN). You have the
  Candidate Password :          option of assigning any other available number.

  Candidate MAC     System Name        Device Type
  -------------     ----------------   -----------          Candidate List
  0060b0-e94300     DEFAULT_CONFIG     3400cl-48G
  080009-918f80     DEFAULT_CONFIG     HP 2824



  Actions->   Cancel     Edit      Save      Help


Use arrow keys to change field selection, <Space> to toggle field choices,
and <Enter> to go to Actions.
```

Figure 13-10. Example of Candidate List in Stack Management Screen

3. Either accept the displayed switch number or enter another available number. (The range is 0 - 15, with 0 reserved for the Commander.)

4. Use the downarrow key to move the cursor to the MAC Address field, then type the MAC address of the desired Candidate from the Candidate list in the lower part of the screen.

5. Do one of the following:

- If the desired Candidate has a Manager password, press the downarrow key to move the cursor to the Candidate Password field, then type the password.

- If the desired Candidate does not have a password, go to step 6.

6. Press **[Enter]** to return to the Actions line, then press **[S]** (for **Save**) to complete the Add process for the selected Candidate. You will then see a screen similar to the one in figure 13-11, below, with the newly added Member listed.

   Note: If the message **Unable to add stack member: Invalid Password** appears in the console menu's Help line, then you either omitted the Candidate's Manager password or incorrectly entered the Manager password.

For status descriptions, see the table on page 13-45.

```
                         Pacific Ocean

=========================- CONSOLE - MANAGER MODE -=============================
                     Stacking - Stack Management

  SN    MAC Address      System Name      Device Type         Status
  --    -------------    ---------------  -----------     --------------------
  1     0060b0-df1a00    Coral Sea        3400cl-48G      Member Up
  2     080009-8c5080    North Atlantic   3400cl-48G      Member Up
  3     0060b0-e94300    Big_Waters-3     3400cl-48G      Member Up
```

New Member added in step 6.
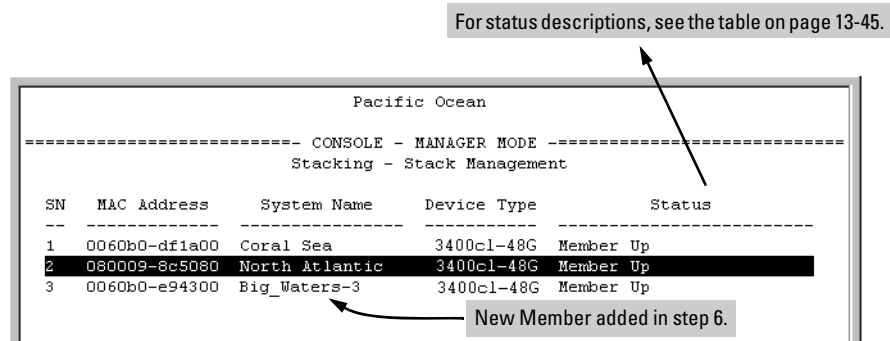
Figure 13-11. Example of Stack Management Screen After New Member Added

Using the Commander's Menu To Move a Member From One Stack to Another. Where two or more stacks exist in the same subnet (broadcast domain), you can easily move a Member of one stack to another stack if the destination stack is not full. (If you are using VLANs in your stack environment, see "Stacking Operation with a Tagged VLAN" on page 13-44.) This procedure is nearly identical to manually adding a Candidate to a stack (page 13-17). (If the stack from which you want to move the Member has a Manager password, you will need to know the password to make the move.)

1. To move a Member from one stack to another, go to the Main Menu of the Commander in the destination stack and display the Stacking Menu by selecting

   9. Stacking...

2. To learn or verify the MAC address of the Member you want to move, display a listing of all Commanders, Members, and Candidates in the subnet by selecting:

    2. Stacking Status (All)

You will then see the Stacking Status (All) screen:

For status descriptions, see the table on page 13-45.

```
                           Pacific Ocean

==========================- CONSOLE - MANAGER MODE -===================================
                    Stacking - Stacking Status (All)

       Stack Name          MAC Address      System Name           Status
    ------------------    -------------    ----------------    --------------------
    Big Waters            0060b0-880a80    Pacific Ocean       Commander Up
                          0060b0-df1a00    Coral Sea           Member Up
                          080009-8c5080    North Atlantic      Member Up
    Newstack              001083-c3fc00    Newstack-0          Commander Up
                          080009-918f80    Newstack-1          Member Up
                          0060b0-df2a00    Newstack-2          Member Up
    Others:               001083-3c09c0    DEFAULT_CONFIG      Candidate
                          0060b0-e94300    DEFAULT_CONFIG      Candidate
                          080009-918f80    DEFAULT_CONFIG      Candidate




    Actions->    Back     Next page     Prev page      Help

Return to previous screen.
Use up/down arrow keys to scroll to other entries, left/right arrow keys to
change action selection, and <Enter> to execute action.
```

This column lists the MAC Addresses for switches discovered (in the local subnet) that are configured for Stacking.

Using the MAC addresses for these Members, you can move them between stacks in the same subnet.

Figure 13-12. Example of How the Stacking Status (All) Screen Helps You Find Member MAC Addresses

3. In the Stacking Status (All) screen, find the Member switch that you want to move and note its MAC address, then press **[B]** (for **Back**) to return to the Stacking Menu.

4. Display the Commander's Stack Management screen by selecting

    4. Stack Management

    (For an example of this screen, see figure 13-9 on page 13-18.)

5. Press **[A]** (for **Add**) to add the Member. You will then see a screen listing any available candidates. (See figure 13-10 on page 13-18.) Note that you will not see the switch you want to add because it is a Member of another stack and not a Candidate.)

6. Either accept the displayed switch number or enter another available number. (The range is 0 - 15, with 0 reserved for the Commander.)

7. Use the downarrow key to move the cursor to the MAC Address field, then type the MAC address of the desired Member you want to move from another stack.

8. Do one of the following:

    • If the stack containing the Member you are moving has a Manager password, press the downarrow key to select the Candidate Password field, then type the password.

    • If the stack containing the Member you want to move does not have a password, go to step 9.

9. Press **[Enter]** to return to the Actions line, then press **[S]** (for **Save**) to complete the Add process for the selected Member. You will then see a screen similar to the one in figure 13-9 on page 13-18, with the newly added Member listed.

N o t e :   If the message **Unable to add stack member: Invalid Password** appears in the console menu's Help line, then you either omitted the Manager password for the stack containing the Member or incorrectly entered the Manager password.

You can "push" a Member from one stack to another by going to the Member's interface and entering the MAC address of the destination stack Commander in the Member's **Commander MAC Address** field. Using this method moves the Member to another stack without a need for knowing the Manager password in that stack, but also blocks access to the Member from the original Commander.

Using the Commander's Menu To Remove a Stack Member. These rules affect removals from a stack:

■ When a Candidate becomes a Member, its **Auto Join** parameter is automatically set to **No**. This prevents the switch from automatically rejoining a stack as soon as you remove it from the stack.

■ When you use the Commander to remove a switch from a stack, the switch rejoins the Candidate pool for your IP subnet (broadcast domain), with **Auto Join** set to **No**.

■ When you remove a Member from a stack, it frees the previously assigned switch number (**SN**), which then becomes available for assignment to another switch that you may subsequently add to the stack. The default switch number used for an add is the lowest unassigned number in the Member range (1 - 15; 0 is reserved for the Commander).

To remove a Member from a stack, use the Stack Management screen.

1. From the Main Menu, select:

    9. Stacking...

        4. Stack Management

    You will then see the Stack Management screen:
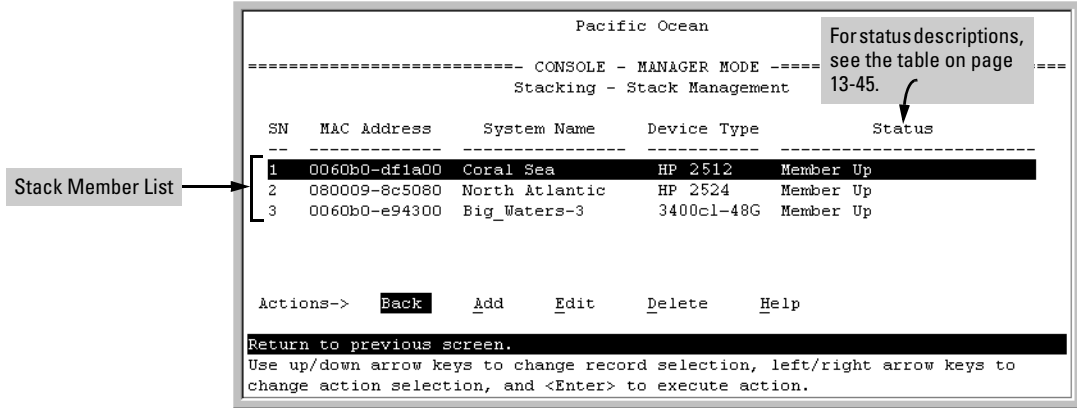
Figure 13-13. Example of Stack Management Screen with Stack Members Listed

2. Use the downarrow key to select the Member you want to remove from the stack.
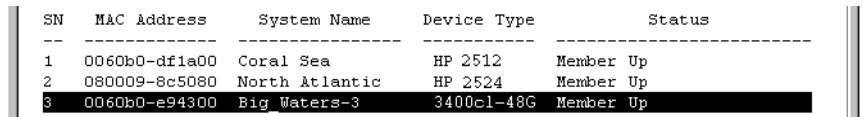
Figure 13-14. Example of Selecting a Member for Removal from the Stack

3. Type **[D]** (for **Delete**) to remove the selected Member from the stack. You will then see the following prompt:
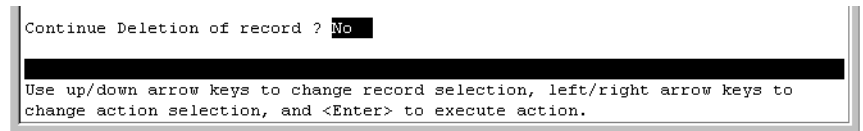
Figure 13-15. The Prompt for Completing the Deletion of a Member from the Stack

4.   To continue deleting the selected Member, press the Space bar once to
     select **Yes** for the prompt, then press **[Enter]** to complete the deletion. The
     Stack Management screen updates to show the new stack Member list.

Using the Commander To Access Member Switches for Configuration Changes and Monitoring Traffic

After a Candidate becomes a stack Member, you can use that stack's
Commander to access the Member's console interface for the same configu-
ration and monitoring that you would do through a Telnet or direct-connect
access.
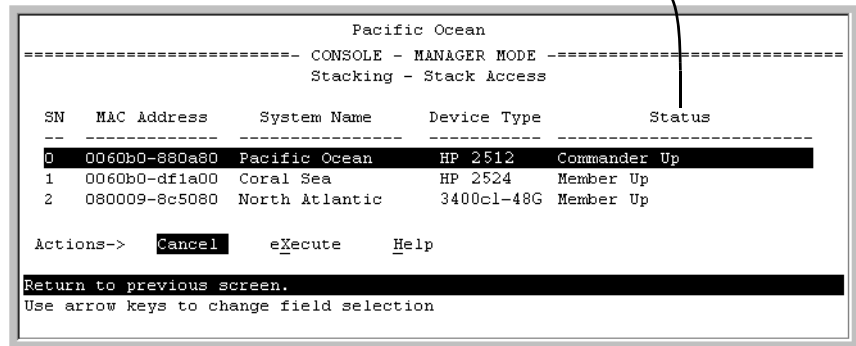
1.   From the Main Menu, select:

     9. Stacking...
          5. Stack Access

     You will then see the Stack Access screen:

For status descriptions, see the table on page 13-45.

```
                              Pacific Ocean
=========================- CONSOLE - MANAGER MODE -=========================
                         Stacking - Stack Access

  SN    MAC Address      System Name      Device Type         Status
  --   -------------   ----------------   -----------   --------------------------
  0    0060b0-880a80   Pacific Ocean      HP 2512       Commander Up
  1    0060b0-df1a00   Coral Sea          HP 2524       Member Up
  2    080009-8c5080   North Atlantic     3400cl-48G    Member Up


  Actions->    Cancel      eXecute      Help

  Return to previous screen.
  Use arrow keys to change field selection
```

Figure 13-16. Example of the Stack Access Screen

Use the down arrow key to select the stack Member you want to access, then
press **[X]** (for **eXecute**) to display the console interface for the selected Member.
For example, if you selected switch number 1 (system name: **Coral Sea**) in figure
13-16 and then pressed **[X]**, you would see the Main Menu for the switch named
Coral Sea.

```
                                             Coral Sea ◄─────────┐
                                                                 │
============================ TELNET - MANAGER MODE ========================
                               Main Menu

      1. Status and Counters...
      2. Switch Configuration...
      3. Console Passwords...
      4. Event Log
      5. Command Line (CLI)
      6. Reboot Switch                        ┌─────────────────────────┐
      7. Download OS                          │ Main Menu for stack      │
      8. Run Setup                            │ Member named "Coral Sea" │
      9. Stacking...                          │ (SN = 1 from figure 13-16)│
      0. Logout                               └─────────────────────────┘
    Provides the menu to display configuration, status, and counters.
    To select menu item, press item number, or highlight item and press <Enter>.
```

Figure 13-17. The eXecute Command Displays the Console Main Menu for the Selected Stack Member

2. You can now make configuration changes and/or view status data for the selected Member in the same way that you would if you were directly connected or telnetted into the switch.

3. When you are finished accessing the selected Member, do the following to return to the Commander's Stack Access screen:

   a. Return to the Member's Main Menu.

   b. Press **[0]** (for Logout), then **[Y]** (for Yes).

   c. Press **[Return]**.

   You should now see the Commander's Stack Access screen. (For an example, see figure 13-16 on page 13-23.)

Converting a Commander or Member to a Member of Another Stack

When moving a commander, the following procedure returns the stack members to Candidate status (with Auto-Join set to "**No**") and converts the stack Commander to a Member of another stack. When moving a member, the procedure simply pulls a Member out of one stack and pushes it into another.

1. From the Main Menu of the switch you want to move, select

   9. Stacking

2. To determine the MAC address of the destination Commander, select

   2. Stacking Status (All)

3. Press **[B]** (for **B**ack) to return to the Stacking Menu.

4. To display Stack Configuration menu for the switch you are moving, select

   3. Stack Configuration

5. Press **[E]** (for **E**dit) to select the Stack State parameter.

6. Use the Space bar to select **Member**, then press ⬇ to move to the **Commander MAC Address** field.

7. Enter the MAC address of the destination Commander and press **[Enter]**.

8. Press **[S]** (for **S**ave).

Monitoring Stack Status

Using the stacking options in the menu interface for any switch in a stack, you can view stacking data for that switch or for all stacks in the subnet (broadcast domain). (If you are using VLANs in your stack environment, see "Stacking Operation with a Tagged VLAN" on page 13-44.) This can help you in such ways as determining the stacking configuration for individual switches, identifying stack Members and Candidates, and determining the status of individual switches in a stack. See table 13-5 on page 13-25.

Table 13-5. Stack Status Environments

Screen Name	Commander	Member	Candidate

Stack Status (This Switch)	• Commander's stacking configuration
	• Data on stack Members:
	  – Switch Number
	  – MAC Address
	  – System Name
	  – Device Type
	  – Status
		• Member's stacking configuration
		• Member Status
		• Data identifying Member's Commander:
		  – Commander Status
		  – Commander IP Address
		  – Commander MAC Address
			Candidate's stacking configuration

Stack Status (All)	Lists devices by stack name or Candidate status (if device is not a stack Member). Includes: • Stack Name • MAC Address • System Name • Status	Same as for Commander.	Same as for Commander.

Using Any Stacked Switch To View the Status for All Switches with Stacking Enabled. This procedure displays the general status of all switches in the IP subnet (broadcast domain) that have stacking enabled.

1. Go to the console Main Menu for any switch configured for stacking and select:

    9. Stacking ...

    2. Stacking Status (All)

    You will then see a Stacking Status screen similar to the following:

For status descriptions, see the table on page 13-45.

```
                              Pacific Ocean

========================- CONSOLE - MANAGER MODE -===============-============
                   Stacking - Stacking Status (All)

        Stack Name          MAC Address      System Name              Status
   -------------------     ------------    ----------------     --------------------
   Big_Waters              0060b0-880a80   Pacific Ocean        Commander Up
                           0060b0-df1a00   Coral Sea            Member Up
                           080009-8c5080   North Atlantic       Member Up
   Newstack                001083-c3fc00   Newstack-0           Commander Up
                           080009-918f80   Newstack-1           Member Up
                           0060b0-df2a00   Newstack-2           Member Up
   Others:                 001083-3c09c0   DEFAULT_CONFIG       Candidate
                           0060b0-e94300   DEFAULT_CONFIG       Candidate
                           080009-918f80   DEFAULT_CONFIG       Candidate



   Actions->    Back      Next page     Prev page     Help
  ------------------------------------------------------------------------
  Return to previous screen.
  Use up/down arrow keys to scroll to other entries, left/right arrow keys to
  change action selection, and <Enter> to execute action.
```

Figure 13-18. Example of Stacking Status for All Detected Switches Configured for Stacking

Viewing Commander Status. This procedure displays the Commander and stack configuration, plus information identifying each stack member.

To display the status for a Commander, go to the console Main Menu for the switch and select:

9. Stacking ...

1. Stacking Status (This Switch)

You will then see the Commander's Stacking Status screen:

```
                               Pacific Ocean

=========================== CONSOLE - MANAGER MODE -===========================
                    Stacking - Stacking Status (This Switch)

   Stack State         : Commander
   Transmission Interval : 60
   Stack Name          : Big_Waters Number of members      : 2
   Auto Grab           : No         Members unreachable     : 0

   SN   MAC Address     System Name     Device Type        Status
   --   -------------   ---------------  -----------   --------------------------
   0    0060b0-880a80  Pacific Ocean    HP 2512       Commander Up
   1    0060b0-df1a00  Coral Sea        HP 2524       Member Up
   2    080009-8c5080  North Atlantic   3400cl-48G    Member Up


   Actions->   Back      Help

 Return to previous screen.
 Use arrow keys to change action selection and <Enter> to execute action.
```

Figure 13-19. Example of the Commander's Stacking Status Screen

Viewing Member Status. This procedure displays the Member's stacking information plus the Commander's status, IP address, and MAC address.

To display the status for a Member:

1. Go to the console Main Menu of the Commander switch and select

    9. Stacking ...

    5. Stack Access

2. Use the downarrow key to select the Member switch whose status you want to view, then press **[X]** (for **eXecute**). You will then see the Main Menu for the selected Member switch.

3. In the Member's Main Menu screen, select

    9. Stacking ...

    1. Stacking Status (This Switch)

    You will then see the Member's Stacking Status screen:

```
                              Coral Sea

=========================- TELNET - MANAGER MODE -=============================
                  Stacking - Stacking Status (This Switch)

    Stack State              : Member
    Transmission Interval    : 60
    Switch Number            : 1
    Stack Name               : Big_Waters
    Member Status            : Joined Successfully
    Commander Status         : Commander Up
    Commander IP Address     : 10.28.227.102
    Commander MAC Address    : 0060b0-880a80


  Actions->   Back      Help

 Return to previous screen.
 Use arrow keys to change action selection and <Enter> to execute action.
```

Figure 13-20. Example of a Member's Stacking Status Screen

Viewing Candidate Status. This procedure displays the Candidate's
stacking configuration.

To display the status for a Candidate:

1. Use Telnet (if the Candidate has a valid IP address for your network) or
    a direct serial port connection to access the menu interface Main Menu
    for the Candidate switch and select

    9. Stacking ...

        1. Stacking Status (This Switch)

    You will then see the Candidate's Stacking Status screen:

```
                              Coral Sea

=========================- TELNET - MANAGER MODE -=============================
                  Stacking - Stacking Status (This Switch)

    Stack State           : Candidate
    Transmission Interval : 60
    Auto Join             : No


  Actions->   Back      Help

 Return to previous screen.
 Use arrow keys to change action selection and <Enter> to execute action.
```

Figure 13-21. Example of a Candidate's Stacking Screen

Using the CLI To View Stack Status and Configure Stacking

The CLI enables you to do all of the stacking tasks available through the menu interface.)

Table 13-6. CLI Commands for Configuring Stacking on a Switch

CLI Command	Operation

show stack
[candidates | view | all]
	Commander: Shows Commander's stacking configuration and lists the stack members and their individual status.
	Member: Lists Member's stacking configuration and status, and the status and the IP address and subnet mask of the stack Commander.
	
	Options:
	    candidates: (Commander only) Lists stack Candidates.
	    view: (Commander only) Lists current stack Members and their individual status.
	    all: Lists all stack Commanders, Members and Candidates, with their individual status.

[no] stack	**Any Stacking-Capable Switch:** Enables or disables stacking on the switch. **Default:** Stacking Enabled
[no] stack commander *<stack name>*	**Candidate or Commander:** Converts a Candidate to a Commander or changes the stack name of an existing commander. "**No**" form eliminates named stack and returns Commander and stack Members to Candidate status with **Auto Join** set to **No**. "**No**" form prevents the switch from being discovered as a stacking-capable switch. **Default:** Switch Configured as a Candidate
[no] stack auto-grab	**Commander:** Causes Commander to automatically add to its stack any discovered Candidate in the subnet that does not have a Manager password and has **Auto-Join** set to **Yes**. **Default:** Disabled **Note:** If the Commander's stack already has 15 members, the Candidate cannot join until an existing member leaves the stack.

CLI Command	Operation
[no] stack member *<switch-num>* mac-address *<mac-addr>* [password *<password-str>*]	**Commander:** Adds a Candidate to stack membership. "No" form removes a Member from stack membership. To easily determine the MAC address of a Candidate, use the **show stack candidates** command. To determine the MAC address of a Member you want to remove, use the **show stack view** command. The password (*password-str*) is required only when adding a Candidate that has a Manager password.
telnet *<1..15>* *Used In:* Commander Only	**Commander:** Uses the **SN** (switch number— assigned by the stack Commander) to access the console interface (menu interface or CLI) of a stack member. To view the list of **SN** assignments for a stack, execute the **show stack** command in the Commander's CLI.
[no] stack join *<mac-addr>*	**Candidate:** Causes the Candidate to join the stack whose Commander has the indicated MAC address. "No" form is used in a Member to remove it from the stack of the Commander having the specified address. **Member:** "Pushes" the member to another stack whose Commander has the indicated MAC address.
[no] stack auto-join	**Candidate:** Enables Candidate to automatically join the stack of any Commander in the IP subnet that has **Auto Grab** enabled, or disables **Auto-Join** in the candidate. **Default: Auto Join** enabled. **Note:** If the Candidate has a Manager password or if the available stack(s) already have the maximum of 15 Members, the automatic join will not occur.
stack transmission-interval	**All Stack Members:** specifies the interval in seconds for transmitting stacking discovery packets. **Default:** 60 seconds

Using the CLI To View Stack Status

You can list the stack status for an individual switch and for other switches that have been discovered in the same subnet.

Syntax: show stack [candidates | view | all]

Viewing the Status of an Individual Switch. The following example illustrates how to use the CLI in a to display the stack status for that switch. In this case, the switch is in the default stacking configuration.

Syntax: show stack

```
ProCurve(config)# show stack
 Stacking - Stacking Status (This Switch)

  Stack State          : Commander
  Transmission Interval : 60
  Stack Name           : Big_Waters      Number of members       : 1
  Auto Grab            : Yes             Members unreachable     : 0

  SN MAC Address     System Name       Device Type Status
  -- -------------   ----------------  ----------- ------------------------
  0  0030c1-7fcc40 HP4108               HP 4108    Commander Up
  1  0030c1-7fec40 piles-1              3400cl-48G Member Up
```

Figure 13-22. Example of Using the Show Stack Command To List the Stacking Configuration for an Individual Switch

Viewing the Status of Candidates the Commander Has Detected.

This example illustrates how to list stack candidates the Commander has discovered in the ip subnet (broadcast domain).

Syntax: show stack candidates

```
ProCurve (config)# show stack candidates
 Stack Candidates

  Candidate MAC System Name                 Device Type
  ------------- -------------------------   -----------
  0060b0-889e00 DEFAULT_CONFIG              3400cl-48G
```

Figure 13-23. Example of Using the Show Stack Candidates Command To List Candidates

Viewing the Status of all Stack-Enabled Switches Discovered in the IP Subnet. The next example lists all the stack-configured switches discovered in the IP subnet. Because the switch on which the **show stack all** command was executed is a candidate, it is included in the "Others" category.

Syntax: show stack all

```
ProCurve (config)# show stack all

 Stacking - Stacking Status (All)

  Stack Name       MAC Address    System Name                Status
  --------------   -------------  -------------------------  -------------
  Big_Waters       0030c1-7fcc40  HP4108:                    Commander Up
                   0030c1-7fec40  Big_Waters-1               Member Up
  Others:          0060b0-889e00  DEFAULT_CONFIG             Candidate
```

Figure 13-24. Result of Using the Show Stack All Command To List Discovered Switches in the IP Subnet

Viewing the Status of the Commander and Current Members of the Commander's Stack. The next example lists all switches in the stack of the selected switch.

Syntax: show stack view

```
ProCurve(config)# show stack view
 Stack Members

  SN MAC Address    System Name        Device Type Status
  -- -------------  ---------------    ----------- --------------
  0  0030c1-7fcc40  HP4108             HP 4108     Commander Up
  1  0030c1-7fec40  Big_Waters-1       3400cl-48G  Member Up
```

Figure 13-25. Example of the Show Stack View Command To List the Stack Assigned to the Selected Commander

Using the CLI To Configure a Commander Switch

You can configure any stacking-enabled switch to be a Commander as long as the intended stack name does not already exist on the broadcast domain. (When you configure a Commander, you automatically create a corresponding stack.)

Before you begin configuring stacking parameters:

1. Configure IP addressing on the switch intended for stack commander and, if not already configured, on the primary VLAN. (For more on configuring IP addressing, refer to the Management and Configuration Guide for your switch.)

N o t e	The primary VLAN must have an IP address in order for stacking to operate properly. For more on the primary VLAN, see "The Primary VLAN" on page 2-45.

2. Configure a Manager password on the switch intended for commander. (The Commander's Manager password controls access to stack Members.) For more on passwords, see the local manager and operator password information in the *Access Security Guide* for your switch.

Configure the Stack Commander.   Assigning a stack name to a switch makes it a Commander and automatically creates a stack.

Syntax: stack commander *< name-str >*

This example creates a Commander switch with a stack name of **Big_Waters**. (Note that if stacking was previously disabled on the switch, this command also enables stacking.)

```
ProCurve(config)# stack commander Big_Waters
```

As the following **show stack** display shows, the Commander switch is now ready to add members to the stack.

```
ProCurve(config)# show stack                        The stack commander command
 Stacking - Stacking Status (This Switch)           configures the Commander and names
  Stack State           : Commander                 the stack.
  Transmission Interval : 60
  Stack Name            : Big_Waters    Number of members          : 0
  Auto Grab             : No            Members unreachable        : 0

  SN MAC Address    System Name      Device Type Status
  -- ------------- ---------------- ----------- -------------------------
  0   0030c1-b24ac0 HP 4108          HP 4108     Commander Up
```

The Commander appears in the stack as Switch
Number (SN) 0.

Figure 13-26. Example of the Commander's Show Stack Screen with Only the Commander Discovered

Using a Member's CLI to Convert the Member to the Commander of a New Stack. This procedure requires that you first remove the Member from its current stack, then create the new stack. If you do not know the MAC address for the Commander of the current stack, use **show stack** to list it.

Syntax: no stack
stack commander < *stack name* >

Suppose, for example, that a ProCurve switch named "Bering Sea" is a Member of a stack named "Big_Waters". To use the switch's CLI to convert it from a stack Member to the Commander of a new stack named "Lakes", you would use the following commands:

The output from this command tells you the MAC address of the current stack Commander.

```
Bering Sea(config)# show stack
  Stacking - Stacking Status (This Switch)

    Stack State                    : Member
    Transmission Interval          : 60
    Switch Number                  : 1
    Stack Commander                : Big_Waters
    Member Status                  : Joined Successfully
    Commander Status               : Commander Up
    Commander IP Address           : 10.28.227.104
    Commander MAC Address          : 0030c1-7fc700

Bering Sea(config)# no stack join 0030c1-7fc700
Bering Sea(config)# stack name Lakes
```

Removes the Member from the "Big_Waters" stack.

Converts the former Member to the Commander of the new "Lakes" stack.

Figure 13-27. Example of Using a Member's CLI To Convert the Member to the Commander of a New Stack

Adding to a Stack or Moving Switches Between Stacks

You can add switches to a stack by adding discovered Candidates or by moving switches from other stacks that may exist in the same subnet. (You cannot add a Candidate that the Commander has not discovered.)

In its default configuration, the Commander's **Auto-Grab** parameter is set to **No** to give you manual control over which switches join the stack and when they join. This prevents the Commander from automatically trying to add every Candidate it finds that has **Auto Join** set to **Yes** (the default for the Candidate).

(If you want any eligible Candidate to automatically join the stack when the Commander discovers it, configure **Auto Grab** in the Commander to **Yes**. When you do so, *any* Candidate discovered with **Auto Join** set to **Yes** (the default) and no Manager password will join the stack, up to the limit of 15 Members.)

Using the Commander's CLI To Manually Add a Candidate to the Stack. To manually add a candidate, you will use:

■ A switch number (**SN**) to assign to the new member. Member SNs range from 1 to 15. To see which SNs are already assigned to Members, use **show stack view**. You can use any SN not included in the listing. (SNs are viewable only on a Commander switch.)

■ The MAC address of the discovered Candidate you are adding to the stack. To see this data, use the **show stack candidates** listing .

For example:

```
ProCurve (config)# show stack view
  Stack Members

  SN MAC Address      System Name       Device Type Status
  -- -------------    ----------------  ----------- -------------------------
  0  0030c1-7fec40    HP4108            HP 4108     Commander Up
  1  0060b0-880a80    Indian Ocean      3400cl-48G  Member Up
```

In this stack, the only SNs in use are 0 and 1, so you can use any SN number from 2 through 15 for new Members. (The SN of "0" is always reserved for the stack Commander.)

Note: When manually adding a switch, you must assign an SN. However, if the Commander automatically adds a new Member, it assigns an SN from the available pool of unused SNs.

Figure 13-28. Example of How To Determine Available Switch Numbers (SNs)

To display all discovered Candidates with their MAC addresses, execute **show stack candidates** from the Commander's CLI. For example, to list the discovered candidates for the above Commander:

```
                    ProCurve (config)# show stack candidates
                       Stack Candidates
                         Candidate MAC System Name             Device Type
                         ------------  -------------------------  -----------
                         0030c1-b24ac0 North Sea                  HP 4108
                         0060b0-df1a00 DEFAULT_CONFIG             3400cl-48G
```

MAC addresses of discovered Candidates.

Figure 13-29. Example of How To Determine MAC Addresses of Discovered Candidates

Knowing the available switch numbers (**SN**s) and Candidate MAC addresses, you can proceed to manually assign a Candidate to be a Member of the stack:

Syntax: stack member < *switch-number* > mac-address < *mac-addr* >
　　　　　　[password < *password-str* >]

For example, if the 3400cl-48 in the above listing did not have a Manager password and you wanted to make it a stack Member with an **SN** of **2**, you would execute the following command:

```
ProCurve(config)# stack member 2 mac-address 0060b0-
df1a00
```

The **show stack view** command then lists the Member added by the above command:

```
ProCurve(config)# show stack view
 Stack Members

 SN MAC Address    System Name      Device Type Status
 -- -------------  ---------------  ----------- ------------------------
 0  0030c1-7fec40 HP2512            HP 4108     Commander Up
 1  0060b0-880a80 Indian Ocean      3400cl-48G  Member Up
 2  0060b0-df1a00 Big_Waters-2      3400cl-48G  Member Up
```

SN (Switch Number) 2 is the new Member added by the **stack member** command.

The new member did not have a System Name configured prior to joining the stack, and so receives a System Name composed of the stack name (assigned in the Commander) with its SN number as a suffix.

Figure 13-30. Example Showing the Stack After Adding a New Member

Using Auto Join on a Candidate. In the default configuration, a Candidate's Auto Join parameter is set to "Yes", meaning that it will automatically join a stack if the stack's Commander detects the Candidate and the Commander's Auto Grab parameter is set to "Yes". You can disable Auto Join on a Candidate if you want to prevent automatic joining in this case. There is also the instance where a Candidate's Auto Join is disabled, for example, when a Commander leaves a stack and its members automatically return to Candidate status, or if you manually remove a Member from a stack. In this case, you may want to reset Auto Join to "Yes".

Status: **[no] stack auto-join**

```
ProCurve(config)# no stack auto-join
                        Disables Auto Join on a Candidate.

ProCurve(config)# stack auto-join
                        Enables Auto Join on a Candidate.
```

Using a Candidate CLI To Manually "Push" the Candidate Into a Stack . Use this method if any of the following apply:

■   The Candidate's **Auto Join** is set to **Yes** (and you do not want to enable **Auto Grab** on the Commander) or the Candidate's **Auto Join** is set to **No**.

■   Either you know the MAC address of the Commander for the stack into which you want to insert the Candidate, or the Candidate has a valid IP address and is operating in your network.

Syntax: stack join *< mac-addr >*

*where: < **mac-addr >** is the MAC address of the Commander in the destination stack.*

Use Telnet (if the Candidate has an IP address valid for your network) or a direct serial port connection to access the CLI for the Candidate switch. For example, suppose that a Candidate named "North Sea" with **Auto Join** off and a valid IP address of 10.28.227.104 is running on a network. You could Telnet to the Candidate, use **show stack all** to determine the Commander's MAC address, and then "push" the Candidate into the desired stack.

```
                                              1. Telnet to the Candidate named "North Sea".
                                              2. Use show stack all to display the Commander's
                                                 MAC address.
ProCurve# telnet 10.28.227.104
North Sea# show stack all                       MAC Address for
  Stacking - Stacking Status (All)              Stack Commander
   Stack Name       MAC Address   System Name               Status
   ------------     -----------   ----------------------    ------------
   Big_Waters       0030c1-7fec40 HP4108                    Commander Up
                    0060b0-880a80 Indian Ocean              Member Up
                    0060b0-df1a00 Bering Sea                Member Up
   Others:          0030c1-7fc700 North Sea                 Candidate

North Sea# config                             3. Set the Candidate CLI to Config mode.
North Sea(config)# stack join 0030c1-7fec40   4. Execute stack join with the
                                                 Commander's MAC address to "push"
                                                 the Candidate into the stack.
```

Figure 13-31. Example of "Pushing" a Candidate Into a Stack

To verify that the Candidate successfully joined the stack, execute **show stack all** again to view the stacking status.

Using the Destination Commander CLI To "Pull" a Member from Another Stack. This method uses the Commander in the destination stack to "pull" the Member from the source stack.

Syntax: stack member < *switch-number* >
        mac-address < *mac-addr* >
        [password < *password-str* >]

In the destination Commander, use **show stack all** to find the MAC address of the Member you want to pull into the destination stack. For example, suppose you created a new Commander with a stack name of "Cold_Waters" and you wanted to move a switch named "Bering Sea" into the new stack:

```
   ProCurve(config)# show stack all
    Stacking - Stacking Status (All)
     Stack Name       MAC Address    System Name                  Status
     ---------------  -------------  ---------------------------  -------------
     Big_Waters       0030c1-7fec40  HP4108                       Commander Up
                      0060b0-880a80  Indian Ocean                 Member Up
                      0060b0-df1a00  Bering Sea                   Member Up
     Cold_Waters      0030c1-7fc700  HP4108                       Commander Up
```
Move this switch into the "Cold Waters" stack.

Figure 13-32. Example of Stack Listing with Two Stacks in the Subnet

You would then execute the following command to pull the desired switch into the new stack:

```
ProCurve(config)# stack member 1 mac-address 0060b0-
df1a00
```

Where **1** is an unused switch number (**SN**).

Since a password is not set on the Candidate, a password is not needed in this example.

You could then use **show stack all** again to verify that the move took place.

Using a Member CLI To "Push" the Member into Another Stack. You can use the Member's CLI to "push" a stack Member into a destination stack if you know the MAC address of the destination Commander.

Syntax: stack join *<mac-addr>*

where: *< mac-addr >* is the MAC address of the Commander for the destination stack.

Converting a Commander to a Member of Another Stack. Removing the Commander from a stack eliminates the stack and returns its Members to the Candidate pool with **Auto Join** disabled.

Syntax: no stack name *< stack name>*
                 stack join *< mac-address >*

If you don't know the MAC address of the destination Commander, you can use **show stack all** to identify it.

For example, suppose you have a switch operating as the Commander for a temporary stack named "Test". When it is time to eliminate the temporary "Test" stack and convert the switch into a member of an existing stack named "Big_Waters", you would execute the following commands in the switch's CLI:

```
                                              Eliminates the "Test" stack and converts
                                              the Commander to a Candidate.
ProCurve(config)# no stack name Test
ProCurve(config)# show stack all              Helps you to identify the MAC address of the
                                              Commander for the "Big_Waters" stack.
 Stacking - Stacking Status (All)

  Stack Commander   MAC Address   System Name                   Status
  ---------------   ------------- ------------------------ ---------------
  Big_Waters        0030c1-7fc700 HP4108                         Commander Up
                    0060b0-889e00 Big_Waters-1                   Member Up
  Others:           0030c1-7fec40 HP4108                         Candidate

ProCurve(config)# stack join 0030c1-7fc700

                                              Adds the former "Test" Commander to the
                                              "Big_Waters" stack.
```

Figure 13-33. Example of Command Sequence for Converting a Commander to a Member

Using the CLI To Remove a Member from a Stack

You can remove a Member from a stack using the CLI of either the Commander or the Member.

N o t e   When you remove a Member from a stack, the Member's **Auto Join** parameter is set to **No**.

Using the Commander CLI To Remove a Stack Member. This option requires the switch number (SN) and the MAC address of the switch to remove. (Because the Commander propagates its Manager password to all stack members, knowing the Manager password is necessary only for gaining access to the Commander.)

Syntax: [no] stack member *<switch-num>* mac-address *<mac-addr>*

Use **show stack view** to list the stack Members. For example, suppose that you wanted to use the Commander to remove the "North Sea" Member from the following stack:

```
ProCurve (config)# show stack view
 Stack Members

 SN MAC Address     System Name         Device Type Status
 -- -------------   -----------------   ----------- -------------
 0  0030c1-7fec40   HP4108              HP 4108     Commander Up
 1  0060b0-880a80   Indian Ocean        3400cl-48G  Member Up
 2  0060b0-df1a00   Bering Sea          3400cl-48G  Member Up
 3  0030c1-7fc700   North Sea           HP 4108     Member Up
```

Remove this Member from the stack.

Figure 13-34. Example of a Commander and Three Switches in a Stack

You would then execute this command to remove the "North Sea" switch from the stack:

```
ProCurve(config)# no stack member 3 mac-address 0030c1-
7fc700
```

where:

- **3** is the "North Sea" Member's switch number **(SN)**
- **0030c1-7fc700** is the "North Sea" Member's MAC address

Using the Member's CLI To Remove the Member from a Stack.

Syntax: no stack join *<mac-addr>*

To use this method, you need the Commander's MAC address, which is available using the show stack command in the Member's CLI. For example:

```
North Sea(config)# show stack
  Stacking - Stacking Status (This Switch)
    Stack State                  : Member
    Transmission Interval        : 10
    Switch Number                : 3
    Stack Name                   : Big_Waters
    Member Status                : Joined Successfully
    Commander Status             : Commander Up
    Commander IP Address         : 10.28.227.103
    Commander MAC Address        : 0030c1-7fec40
```

CLI for "North Sea" Stack Member

MAC Address of the Commander for the Stack to Which the "North Sea" Switch Belongs

Figure 13-35. Example of How To Identify the Commander's MAC Address from a Member Switch

You would then execute this command in the "North Sea" switch's CLI to remove the switch from the stack:

```
North Sea(config)# no stack join 0030c1-7fec40
```
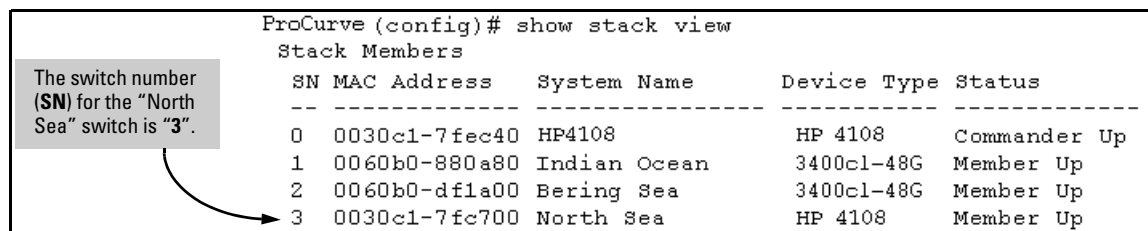
Using the CLI To Access Member Switches for Configuration Changes and Traffic Monitoring

After a Candidate becomes a Member, you can use the telnet command from the Commander to access the Member's CLI or console interface for the same configuration and monitoring that you would do through a Telnet or direct-connect access from a terminal.

Syntax: telnet *<switch-number>*

> *where:* unsigned integer is the switch number (**SN**) assigned by the Commander to each member (range: **1** - **15**).

To find the switch number for the Member you want to access, execute the **show stack view** command in the Commander's CLI. For example, suppose that you wanted to configure a port trunk on the switch named "North Sea" in the stack named "Big_Waters". Do do so you would go to the CLI for the "Big_Waters" Commander and execute show stack view to find the switch number for the "North Sea" switch:

```
ProCurve(config)# show stack view
 Stack Members

 SN MAC Address     System Name      Device Type Status
 -- -------------   ---------------  ----------- -------------
 0  0030c1-7fec40 HP4108            HP 4108      Commander Up
 1  0060b0-880a80 Indian Ocean      3400cl-48G   Member Up
 2  0060b0-df1a00 Bering Sea        3400cl-48G   Member Up
 3  0030c1-7fc700 North Sea         HP 4108      Member Up
```

The switch number (**SN**) for the "North Sea" switch is "**3**".

Figure 13-36. Example of a Stack Showing Switch Number (SN) Assignments

To access the "North Sea" console, you would then execute the following **telnet** command:

```
ProCurve(config)# telnet 3
```

You would then see the CLI prompt for the "North Sea" switch, allowing you to configure or monitor the switch as if you were directly connected to the console.

SNMP Community Operation in a Stack

Community Membership

In the default stacking configuration, when a Candidate joins a stack, it automatically becomes a Member of any SNMP community to which the Commander belongs, even though any community names configured in the Commander are not propagated to the Member's SNMP Communities listing. However, if a Member has its own (optional) IP addressing, it can belong to SNMP communities to which other switches in the stack, including the Commander, do not belong. For example:

Commander Switch
IP Addr: 10.31.29.100
Community Names:
  – blue
  – red

Member Switch 1
IP Addr: 10.31.29.18
Community Names:
  – public (the default)

Member Switch 3
IP Addr: 10.31.29.15
Community Names:
  – public (the default)
  – gray

Member Switch 2
IP Addr: None
Community Names:
  – none

- The Commander and all Members of the stack belong to the blue and red communities. Only switch 3 belongs to the gray community. Switches 1, 2, and 3 belong to the public community

- If Member Switch 1 ceases to be a stack Member, it still belongs to the public SNMP community because it has IP addressing of its own. But, with the loss of stack Membership, Switch 1 loses membership in the blue and red communities because they are not specifically configured in the switch.

- If Member Switch 2 ceases to be a stack Member, it loses membership in all SNMP communities.

- If Member Switch 3 ceases to be a stack Member, it loses membership in the blue and red communities, but—because it has its own IP addressing—retains membership in the public and gray communities.

Figure 13-37. Example of SNMP Community Operation with Stacking

SNMP Management Station Access to Members Via the Commander.

To use a management station for SNMP Get or Set access through the Commander's IP address to a Member, you must append **@sw<_switch number_>** to the community name. For example, in figure 13-37, you would use the following command in your management station to access Switch 1's MIB using the blue community:

    snmpget < _MIB variable_ > 10.31.29.100 blue@sw1

Note that because the gray community is only on switch 3, you could not use the Commander IP address for gray community access from the management station. Instead, you would access switch 3 directly using the switch's own IP address. For example:

    snmpget < _MIB variable_ > 10.31.29.15 gray

Note that in the above example (figure 13-37) you cannot use the public community through the Commander to access any of the Member switches. For example, you can use the public community to access the MIB in switches 1 and 3 by using their unique IP addresses. However, you must use the red or blue community to access the MIB for switch 2.

```
snmpget < MIB variable > 10.31.29.100 blue@sw2
```

Using the CLI To Disable or Re-Enable Stacking

In the default configuration, stacking is enabled on the switch. You can use the CLI to disable stacking on the switch at any time. Disabling stacking has the following effects:

- **Disabling a Commander:** Eliminates the stack, returns the stack Members to Candidates with **Auto Join** disabled, and changes the Commander to a stand-alone (nonstacking) switch. You must re-enable stacking on the switch before it can become a Candidate, Member, or Commander.

- **Disabling a Member:** Removes the Member from the stack and changes it to a stand-alone (nonstacking) switch. You must re-enable stacking on the switch before it can become a Candidate, Member, or Commander.

- **Disabling a Candidate:** Changes the Candidate to a stand-alone (non-stacking) switch.

Syntax: no stack   (*Disables stacking on the switch.*)
              stack   (*Enables stacking on the switch.*)

Transmission Interval

All switches in the stack must be set to the same transmission interval to help ensure proper stacking operation. HP recommends that you leave this parameter set to the default 60 seconds.

Syntax: stack transmission-interval < *seconds* >

Stacking Operation with Multiple VLANs Configured

Stacking uses the primary VLAN in a switch. In the factory-default configuration, the DEFAULT_VLAN is the primary VLAN. However, you can designate any VLAN configured in the switch as the primary VLAN. (See "The Primary VLAN" on page 2-45.)

When using stacking in a multiple-VLAN environment, the following criteria applies:

■ Stacking uses only the primary VLAN on each switch in a stack.

■ The primary VLAN can be tagged or untagged as needed in the stacking path from switch to switch.

■ The same VLAN ID (VID) must be assigned to the primary VLAN in each stacked switch.

Status Messages

Stacking screens and listings display these status messages:

Message	Condition	Action or Remedy
Candidate Auto-join	Indicates a switch configured with Stack State set to **Candidate, Auto Join** set to **Yes** (the default), and no Manager password.	None required
Candidate	Candidate cannot automatically join the stack because one or both of the following conditions apply: • Candidate has **Auto Join** set to **No**. • Candidate has a Manager password.	Manually add the candidate to the stack.
Commander Down	Member has lost connectivity to its Commander.	Check connectivity between the Commander and the Member.
Commander Up	The Member has stacking connectivity with the Commander.	None required.
Mismatch	This may be a temporary condition while a Candidate is trying to join a stack. If the Candidate does not join, then stack configuration is inconsistent.	Initially, wait for an update. If condition persists, reconfigure the Commander or the Member.
Member Down	A Member has become detached from the stack. A possible cause is an interruption to the link between the Member and the Commander.	Check the connectivity between the Commander and the Member.
Member Up	The Commander has stacking connectivity to the Member.	None required.
Rejected	The Candidate has failed to be added to the stack.	The candidate may have a password. In this case, manually add the candidate. Otherwise, the stack may already be full. A stack can hold up to 15 Members (plus the Commander).

—This page is intentionally unused—

Index

Numerics

A

P

ProCurve Networking
HP Innovation