Release Notes

Software Version 02.3.00c

for the ProCurve 9408sl Routing Switch

June 2007

The 02.3.00c release notes provide information on the following items:

- New hardware and software enhancements introduced with this release
- Procedure for upgrading the software on ProCurve 9408sl Routing Switches
- Software fixes in this release
- Known issues in this release

NOTE: ProCurve periodically updates the ProCurve 9300/9400 Series Routing Switch documentation. For the latest version of any of these publications, visit the ProCurve website at:

http://www.procurve.com

Click on **Technical Support**, then **Product manuals**.

Publication Number

5991-4721
June 2007

Applicable Products

ProCurve Routing Switch 9408sl . . . . . (J8680A)
ProCurve 9400sl Redundant
Management Module . . . . . (J8681A)
ProCurve 9400sl 4-Port 10-GbE Module . . . . (J8682A)
ProCurve 9400sl 40-Port Mini-GBIC Module . (J8684A)
ProCurve 9400sl 40-Port 10/100/1000-T
Module . . . . . (J8685A)
ProCurve 9400sl Redundant Power Supply . . (J8686A)
ProCurve 9400sl 60-Port 10/100/1000-T
Module . . . . . (J8688A)

Trademark Credits

Microsoft®, Windows®, and Windows NT® are trademarks of Microsoft Corporation. Adobe® and Acrobat® are trademarks of Adobe Systems Incorporated. SuperSpan® is a trademark of Foundry Networks, Inc.

Disclaimer

The information contained in this document is subject to change without notice.

HEWLETT-PACKARD COMPANY MAKES NO WARRANTY OF ANY KIND WITH REGARD TO THIS MATERIAL, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE. Hewlett-Packard shall not be liable for errors contained herein or for incidental or consequential damages in connection with the furnishing, performance, or use of this material.

The only warranties for HP products and services are set forth in the express warranty statements accompanying such products and services. Nothing herein should be construed as constituting an additional warranty. HP shall not be liable for technical or editorial errors or omissions contained herein.

Hewlett-Packard assumes no responsibility for the use or reliability of its software on equipment that is not furnished by Hewlett-Packard.

Warranty

See the Customer Support/Warranty booklet included with the product.

A copy of the specific warranty terms applicable to your Hewlett-Packard products and replacement parts can be obtained from your HP Sales and Service Office or authorized dealer.

Contents

Software Version 02.3.00c on the ProCurve 9408sl Routing Switch

These release notes describe the ProCurve 9408sl Routing Switch software version 02.3.00c.

The follow table lists the software enhancements introduced in releases 02.0.02, 02.1.00, 02.2.01, and 02.3.00. These enhancements differ from those described in the ProCurve 9408sl product documentation set, which is based on pre-release software version 01.0.02.

Summary of Enhancements in Software Releases 02.0.02 to 02.3.00

Software Release	Enhancement	Description	See Page
02.3.00	ACL logging of denied packets	A new logging method is implemented to prevent the Syslog buffer from being overloaded with entries that are logged each time a packet is denied access by an ACL entry.	105
02.3.00	Reordering ACL entries	An additional method of reordering the entries in a large ACL is introduced. The new method allows you to upload an ACL to a TFTP server, edit the ACL on the server, and then download and save the modified ACL entries to the startup-config file on a routing switch.	106
02.3.00	OSPF Non-Broadcast Multi-Access (NBMA) networks	Non-broadcast multi-access (NBMA) networks are supported on OSPF routers to use unicast (instead of broadcast) packets to establish neighbor relationships and broadcast route updates on Ethernet and virtual routing (VE) interfaces. An NBMA subnet is useful when using multicast traffic to send broadcast packets is not possible.	116
02.3.00	IPv6 DHCP Relay Agent	A DHCP relay agent is supported for IPv6 traffic to relay messages between a DHCP client and the DHCP server.	117
02.2.01	Reload command now causes power-cycle of interface modules (for future upgrades beyond 02.2.01)	To complete the 9408sl software update procedure, each interface module must be power-cycled so that the FPGA images for that module are loaded from flash. The “boot system” command has always caused a power-cycle of the interface modules. Now, a 9408sl that is running software version 02.2.01 or later can use the “reload” command as well as the “boot system” command to accomplish the interface module power-cycle.	N/A
02.2.01	Ability to download a new image from a PCMCIA card, using a script (for future upgrades beyond 02.2.01)	A script has been available to download 9408sl software images from a TFTP server, beginning with software version 02.1.00. Now, a 9408sl that is running software version 02.2.01 or later can download software images with a script from a PCMCIA card. Example script syntax is: src:slot1; (instead of src:tftp:<ipaddress>;). Example command syntax is: copy slot1 system <script-filename>.	N/A
02.2.01	Multi-device port authentication	Multi-device port authentication is now supported on the 9408sl.	73
02.2.01	802.1X port security	This release allows you to enable 802.1X port security and multi-device port authentication on the same interface.	74

Software Release	Enhancement(Continued)	Description	See Page
02.2.01	VLAN Byte Accounting	With this release, you can configure a VLAN to account for the number of bytes received by all the member ports.	74
02.2.01	Graceful Restart	With this release, you can enable Graceful Restart for OSPF and BGP.	77
02.2.01	802.1s Multiple Spanning Tree Protocol (MSTP)	With this release, you can configure multiple STP instances using MSTP protocol, as defined in IEEE 802.1s-2002.	82
02.2.01	BGP Null0 Routing	With this release, BGP can use null0 to resolve the next hop and install null0 BGP routes to the routing table	92
02.2.01	Port Security MAC Deny	With this release, you can configure deny mac addresses on a global level or on a per port level.	96
02.2.01	IPv6 Over IPv4 Tunnels	This release allows you to configure IPv6 over IPv4 tunnels through the following mechanism: Manually configured tunnels	97
02.2.01	Configuring Egress Priority Merging	This feature allows you to preserve the incoming 802.1p priority.	100
02.2.01	IP receive ACLs	You can use IPv4 ACLs to filter the packets intended for the management process to protect the management module from being overloaded with heavy traffic that was sent to one of the Layer 3 Switch IP interfaces.	101
02.2.01	New rule for creating trunk groups	In this release, trunks can be formed from any number of ports, as long as they contain at least 2 ports and no more than 8 ports.	101
02.2.01	OSPF point-to-point	OSPF point-to-point eliminates the need for Designated and Backup Designated routers, allowing for faster convergence of the network.	101
02.2.01	IP fragmentation protection	Fragmented IP packets with undersized fragments and overlapping fragments are dropped.	103
02.2.01	IP option attack prevention	Packets with IP options in their header are automatically dropped. Enabling the **ip ip-option-process** command allows the device to process packets that use IP options.	103
02.2.01	Static Route Tagging	Static routes can be configured with tag values.	104
02.2.01	MTU enhancements for IPv4 and IPv6	In this release, you can configure IPv4 MTU to be greater than 1500 bytes. MTU value for IPv6 is still limited to 1500 bytes.	104
02.2.01	Enhancement to the **lacp system-priority** command	The **lacp system-priority** command has been moved from the interface configuration level to the global configuration level.	105
02.2.01	Enhancement to an SSH command	The **ip ssh rsa-authentication no** command has been changed is **ip ssh key-authentication no**.	105

Software Release	Enhancement(Continued)	Description	See Page
02.2.01	Neighbor Local AS	Neighbor Local Autonomous System (AS) feature allows a router that is a member of one AS to appear to be a member of another AS.	105
02.2.01	Multi-protocol Border Gateway Protocol (MBGP)	With this release, you can configure MBGP, an extension to BGP that allows a router to support separate unicast and multicast topologies. See the "Configuring MBGP (9300 Series only)" chapter of the *Advanced Configuration and Management Guide for ProCurve 9300/9400 Series Routing Switches.* Although the chapter title states "9300 Series Only", this feature is also supported on the 9408sl, beginning with software version 02.2.01.	N/A
02.2.01	MSDP Mesh Groups	With this release, the 9408sl supports MSDP Mesh Groups. The MSDP cache size is 32 K. See the "Configuring IP Multicast Protocols (9300 Series Only)" chapter of the *Advanced Configuration and Management Guide for ProCurve 9300/9400 Series Routing Switches.* Although the chapter title states "9300 Series Only", this feature is also supported on the 9408sl, beginning with software version 02.2.01.	N/A
02.2.01	Full AS Path information in sFlow	In this release, sFlow packets now contain full AS Path information.	N/A
02.1.00	Layer-2 ACLs	This release supports Layer 2 ACLs to use the **etype** argument to filter on the following etypes (Ethertype): • IPv4-15 (Etype=0x0800, IPv4, HeaderLength 20 bytes) • ARP (Etype=0x0806, IP ARP) • IPv6 (Etype=0x86dd, IP version 6)	47
02.1.00	VSRP and MRP Signaling	This release supports VSRP and MRP signaling to provide a redundant path between a a device and an MRP ring.	49
02.1.00	VSRP Fast Start	This release supports VSRP fast start to enable the port on a VSRP master to restart when a VSRP failover occurs.	51
02.1.00	Secure Shell (SSH) Version 2	With this release, Secure Shell (SSH) Version 2 is supported on the ProCurve 9408sl as described. Note: This release supports SSH v2 only. Other versions of SSH are not supported.	53
02.1.00	Enabling support for more ACL entries	This release provides support for additional ACL entries as described: • The 9408sl routing switch supports 4K ACL entries	55

Software Release	Enhancement(Continued)	Description	See Page
02.1.00	Maximum Frame Size Support	With this release, maximum frame size per port is changed as described: Untagged Ports – 1518 bytes Tagged Ports – 1522 bytes Super-aggregated VLAN ports – 1526 bytes	55
02.1.00	Support for IPv6 on Management Port	This release allows you to configure a management port to automatically obtain an IPv6 address.	55
02.1.00	Enhancements to Rate-Limiting	This release provides several enhancements to the rate-limiting function. See the section referenced for details.	55
02.1.00	Support for Network-based ECMP Load Sharing for IPv6	While in previous releases ECMP load sharing by host was supported, this release also supports ECMP load sharing by network.	65
02.1.00	Fast Direct Routing (FDR)	Fast Direct Routing (FDR), also known as IP static cam mode, enables very large routing/forwarding tables (up to twice the published Internet routes) to be maintained at the interface module level so that all packet forwarding is done at wire speed without the need to learn the best routes in real-time. This release provides detailed instructions for enabling and operating this feature.	65
02.1.00	SSL Security for the Web Management Interface	This release supports use of the https protocol for secure management of a ProCurve 9408sl.	69
02.1.00	Setting Maximum Frame Size Per PPCR	In this release when you set the maximum frame size for a port, it applies to all other ports that are associated with the same PPCR.	70
02.1.00	MSDP	This release supports Multicast Source Discovery Protocol (MSDP) as described in the *Advanced Configuration and Management Guide for ProCurve 9300/9400 Series Routing Switches.*	N/A
02.1.00	New Command for Setting Fan Speed	A new command has been introduced to set fan speed.	71
02.1.00	Downloading a New Image Using a Script	In this release, the capability to download a new image using a script is added. See the section referenced for instructions.	71
02.0.02	CAM Partitioning	This release supports CAM Partitioning by blocks for the following CAM entries: • session-mac • ip-mac • out-session • ipv6 • ipv6-session	32

Software Release	Enhancement(Continued)	Description	See Page
02.0.02	Support for IPv6	This release supports the following additional IPv6 features: • sFlow for IPv6 • Trunk Server for Ipv6 • SNTP for IPv6 • The following IPv6 MIBs: 2452 - TCP 2454 - UDP 2465 - Textual Conventions and General Group 2466 - ICMPv6 Group For information about configuring IPv6 on the 9408sl, see the *IPv6 Configuration Guide for the ProCurve 9408sl Routing Switch*.	N/A
02.0.02	Protocol-based VLANs	This release introduces support for protocol-based VLANs. VLANs can be created for the following protocols: • AppleTalk • IPX • IPv4 • IPv6 • Other These VLANs can be static and can exclude ports.	33
02.0.02	Unicast Flooding on VLAN Ports	This feature allows devices to perform hardware flooding for Layer 2 unknown unicast packets on all ports that belong to a VLAN.	34
02.0.02	VLAN Translation	This release supports VLAN translation. VLAN Translation allows traffic from one VLAN to be transported across a different VLAN.	34
02.0.02	Layer 2 Hitless Failover	This feature allows failover from an active management module to a redundant management module with no loss of Layer 2 connectivity.	38
02.0.02	Support for IPv6 and Outbound ACLs	This release provides support for both IPv6 and Outbound ACLs on most interface modules. All ProCurve 9408sl interface modules support simultaneous IPv4 and IPv6 and outbound IPv4 ACLs.	38
02.0.02	Maximum number of server trunks increased	The maximum number of server trunks supported is increased from 16 to 64.	N/A
02.0.02	New **show ip vrrp statistics** output	In this release, more statistics are available with this show command.	39
02.0.02	New interface modules	This release supports the following new interface modules: • 60-port 1 Gigabit Ethernet Copper module • 4-port 10 Gigabit Ethernet interface module • 40-port 1 Gigabit Ethernet interface module	40

Software Release	Enhancement(Continued)	Description	See Page
02.0.02	802.1Q Tag-type Translation - Per-port Regions	This is not a new feature. The 802.1Q Tag-type translation feature has been supported in all versions of the 9408sl software. The port regions that can have tag-types assigned to them are defined for each interface module in these release notes.	40
02.0.02	Change in Default and Recommended fan threshold values.	The default and recommended low and high temperature thresholds for fan speeds on interface modules are changed with this release. The new values are shown in Table 10 at the page referenced.	40
02.0.02	Clarification to the multicast limit, broadcast limit, and unknown-unicast limit commands.	This is not a new feature but a clarification of a feature in the *Installation and Basic Configuration Guide for the ProCurve 9408sl Routing Switch.* The following limit commands are only supported at the slot level and not at the interface level: • `multicast limit` • `broadcast limit` • `unknown-unicast limit`	N/A
02.0.02	Clarification of VSRP support.	Layer-3 VSRP is not supported. Consequently, the router VSRP commands are not available and VRRP can be run concurrently with VSRP.	N/A

Feature Highlights

The ProCurve 9408sl supports many of the applicable system-level, Layer 2 and Layer 3 features supported on the ProCurve 9304M, 9308M, and 9315M Chassis devices. Configuration for most of the features is the same on the ProCurve 9408sl as on the 9300 Chassis devices.

Table 1 lists the highlights of the software features that are supported in this release.

Table 1: Feature Highlights

Category	Feature Description
System Level Features	
• 802.1X Port Security	Port Security Port Security Violation MAC Deny (Release 02.2.01) - Protection from violations at the MAC address level.
• CAM Partitioning	For session MAC, IP MAC, Out session, IPv6, IPv6 sessions
• Denial of Service (DoS) protection	Protection from SYN attacks Protection from Smurf attacks
• Management Options	Serial and Telnet access to industry-standard Command Line interface (CLI) Secure Sockets Layer (SSL) for the Web-based Graphical User Interface (GUI) Web-based GUI SNMP versions 1, 2, and 3 ProCurve Manager (PCM) and PCM+, beginning with version 2.0
• Multi-device port authentication (Release 02.2.01)	
• Reverse Path Forwarding (RPF)	Note: RPF is supported with only IPv4 enabled on the router.
• Security	AAA Authentication Local passwords RADIUS Secure Shell (SSH) version 1.5 and 2 Secure Copy (SCP) TACACS/TACACS+ User accounts
• sFlow	sFLow version 5
• SysLogD Server Logging	Multiple SysLogD server logging
Layer 2 Features	
• 802.1d	Spanning Tree Protocol (STP) and Single Spanning Tree Protocol (SSTP)
• 802.1p	Quality of Service (QoS) queue mapping Egress Priority Merging (release 02.2.01)
• 802.1q	see VLANs, below

Table 1: Feature Highlights (Continued)

Category(Continued)	Feature Description
• 802.1s (Release 02.2.01)	Multiple Spanning Tree Protocol (MSTP)
• 802.1w	Rapid Spanning Tree Protocol (RSTP) and Single Spanning Tree Protocol (SSTP)
• 802.3ad	Dynamic Link Aggregation on untagged trunks
• GVRP	Generic Attribute Registration Protocol (GARP) VLAN Registration Protocol
• Jumbo packets	Layer 2 jumbo packet support
• Layer 2 Hitless Failover	
• MAC Filtering	MAC filtering and address-lock filters to enhance network security
• MRP	Metro Ring Protocol (MRP) Phase 1 and Phase 2
• MSTP (Release 02.2.01)	
• Port Mirroring	
• Port Monitoring	
• PVST / PVST+	Per-VLAN Spanning Tree (PVST)
• Rate Limiting	Port-based, port-and-priority based, port-and-vlan-based, and port-and-ACL-based rate limiting on inbound ports are supported beginning with Release 01.1.00. Support for outbound ports is available beginning with release 02.2.00. Uses the following algorithms: Credit = (Average rate in bits per second)/(8*64453) Maximum credit total = (Maximum burst in bits)/8
• SuperSpan	
• Topology Groups	
• Trunk Groups	
• VLANs	802.1Q tagging Port-based VLANs Super Aggregated VLANs (SAV) Unicast Flooding on VLAN Ports VLAN Byte Accounting (Release 02.2.01) VLAN Translation Protocol-based VLANs for the following protocols: AppleTalk, IPX, IPv4, IPv6, Other Dual-mode VLAN ports (**Note:** This feature is automatically available on the 9408sl; you don't need the "dual-mode" command.)
• VSRP	Virtual Switch Redundancy Protocol (VSRP) VSRP and MRP Signaling VSRP Fast Start (Layer 3 VSRP is not supported)

Table 1: Feature Highlights (Continued)

Category(Continued)	Feature Description
Layer 3 Features	
• ACLs	Standard or Extended Layer 2 ACLs IPv4 Outbound ACLs and IPv6 on same interface module
• BGP	BGP routes BGP peers BGP dampening Graceful Restart (Release 02.2.01)
• BGP Null0 Routing (Release 02.2.01)	
• Fast Direct Routing (FDR)	
• Graceful Restart (Release 02.2.01)	BGP routing peers avoid changes to their forwarding paths following a switchover
• IP Forwarding	Route table
• IP Static entries	Routes ARPs Virtual interfaces Secondary addresses
• IPv6	IPv6 ACLs Forwarding OSPF BGP RIP IPv6 over IPv4 tunneling IPv6 Stack IPv6 Multicast PIM SSM sFlow for IPv6 SNTP for IPv6 Tunneling Trunk Server for IPv6
• MBGP (Release 02.2.01)	Multi-protocol Border Gateway Protocol
• MSDP	Mesh Group (Release 02.2.01)
• Multicast Routing	Multicast cache L2 IGMP table DVMRP routes PIM-DM PIM-SM MSDP
• OSPF	OSPF routes OSPF adjacencies - Dynamic OFPF LSAs OSPF filtering of advertised routes Graceful Restart (Release 02.2.01)

Table 1: Feature Highlights (Continued)

Category(Continued)	Feature Description
• PBR	Policy-Based Routing
• RIP version 2	RIP routes
• VRRP and VRRPE	Virtual Router Redundancy Protocol (VRRP) and VRRP Extended (VRRPE)

Unsupported Features

The following features are not supported in software release 02.3.00c on the 9408sl. Although commands exist to configure some of these features, they are not supported and should not be used on the 9408sl with software version 02.3.00c:

- AppleTalk
- Control packets ACL/RL
- IGMPv3
- IGMPv3 snooping
- IPv6 MLD and MLD snooping
- IPv6 PIM-SM
- IPv6 PIM-DM
- IPX
- MD5 for NTP
- NAT
- OSPF Non Broadcast support
- Private VLANs
- RARP
- VSRP at Layer 3

Feature Documentation

For feature descriptions and configuration information, see the remaining sections in these release notes and the ProCurve product manuals listed in the “Organization of Documentation” on page 118.

Software Image Files for Release 02.3.00c

To use the features in this release, you need to run the software listed in Table 2 for the ProCurve 9408sl.

Table 2: Software Image Files for the ProCurve 9408sl

Module	Boot and Monitor Images	ProCurve Software Image
Management	mb02300.bin – This file contains both the boot and monitor images for the management module.	mpr02300.bin
Interface	lb02300.bin – This file contains both the boot and monitor images for the interface module.	lp02300.bin

The interface modules require Field-Programmable Gate Array (FPGA) software. ProCurve appends the software version number to the FPGA filename. This indicates that the FPGA file contains the appropriate FPGA image versions for each of the interface modules that is running the named software version. Here are the FPGA filenames for software version 02.3.00c:

- PBIF pbif02300.bin
- XTM xtm02300.bin
- XPP xpp02300.bin
- XBR xbridge02300.bin (for only the 60-port 10/100/1000-T module)

FPGA Version Information

A new FPGA file does not necessarily contain a different version of FPGA image for all types of interface module. For example, the FPGA PBIF version might change for some of the interface modules but not all. At the end of the software upgrade process, the user is cautioned to use the “show version” command to ensure that the routing switch is running the new software versions, and also the new FPGA image versions for each interface module. For that reason, the FPGA versions are listed here for both the previous 9408sl software (version 02.1.00c), and also for the current 9408sl software (version 02.3.00c).

(Previous software) 02.1.00c FPGA versions

Type of Interface Module	02.1.00c PBIF	02.1.00c XTM	02.1.00c XPP	02.1.00c XBRIDGE
4-Port 10-GbE	ver 41	ver 89.1	ver 88.8	N/A
40-Port Mini-GBIC	ver 21	ver 89.1	ver 91.8	N/A
40-Port 10/100/1000-T	ver 21	ver 89.1	ver 91.8	N/A
60-Port 10/100/1000-T	ver 9	ver 89.1	ver 91.8	ver 25

(Previous software) 02.2.01h FPGA versions

Type of Interface Module	02.2.01h PBIF	02.2.01h XTM	02.2.01h XPP	02.2.01h XBRIDGE
4-Port 10-GbE	ver 42	ver 89.1	ver 88.9	N/A
40-Port Mini-GBIC	ver 22	ver 89.1	ver 91.9	N/A
40-Port 10/100/1000-T	ver 22	ver 89.1	ver 91.9	N/A
60-Port 10/100/1000-T	ver 10	ver 89.1	ver 91.9	ver 34

(Current software) 02.3.00c FPGA versions

Type of Interface Module	02.3.00c PBIF	02.3.00c XTM	02.3.00c XPP	02.3.00c XBRIDGE
4-Port 10-GbE	ver 42	ver 89.1	ver 88.9	N/A
40-Port Mini-GBIC	ver 22	ver 89.1	ver 91.9	N/A
40-Port 10/100/1000-T	ver 22	ver 89.1	ver 91.9	N/A
60-Port 10/100/1000-T	ver 10	ver 89.1	ver 91.9	ver 34

Flash

Each management module and interface module includes a boot flash and a code flash. The boot flash stores the boot image for the respective module. The code flash stores the monitor image, the primary and/or secondary ProCurve software image, and configuration data for the respective module.

Each interface module includes an additional code flash that stores field-programmable gate array (FPGA) images.

Table 3 provides the size of the boot flash and code flash for each module.

Table 3: Boot and Code Flash Sizes

Module	Boot Flash Size	Code Flash Size
Management	512K	32M
Interface	512K	16M (for monitor, primary and secondary software images, and configuration data) 8M (for FPGA images)

Upgradable Software Images

You must upgrade several software images on the management and interface modules. Table 4 describes the upgradable images, their location, and what they contain.

Table 4: Upgradable Software Images

Module	Location	Image/Contents
Management	Boot flash	• boot – The image from which the management module boots.
	Code flash	• monitor –This image stores the management module's Real Time Operating System (RTOS) and a development-debugging agent. After the initial startup, the ProCurve 9408sl system loads the RTOS from this image, if present, or from the boot image, if not present. • primary ProCurve software – This image contains the management module's primary ProCurve software. • secondary ProCurve software – This image contains the management module's secondary ProCurve software. If you copy the monitor and/or primary and/or secondary ProCurve software image to all interface modules using the **copy** command with the **all** keyword, the management module makes a copy of the image and stores it in its code flash as follows: • lp-monitor-0 – This file contains the interface module's monitor image. • lp-primary-0 – This file contains the interface module's primary ProCurve software. • lp-secondary-0 – This file contains the interface module's secondary ProCurve software. **NOTE:** The management module stores this software for the interface modules; it does not run this software. If you copy the monitor and/or primary and/or secondary ProCurve software image to a specified chassis slot using the **copy** command with the <chassis-slot-number> parameter, the management module does not make a copy of the image.

Table 4: Upgradable Software Images (Continued)

Module	Location	Image/Contents
Interface	Boot flash	• boot – The image from which the management module boots. Upon first startup, the interface module loads its RTOS from the boot image or the monitor image in the interface module's code flash.
	Code flash	• monitor –This image contains the interface module's RTOS. Upon subsequent startups, the interface module loads its RTOS from this image, if present, or the interface module's boot image, if not present. • primary ProCurve software – This image contains the interface module's primary ProCurve software. • secondary ProCurve software – This image contains the interface module's secondary ProCurve software. The following are field-programmable gate array (FPGA) images: • Peripheral Bus Interface FPGA (PBIF) • 10 Gigabit Traffic Manager (XTM) • 10 Gig Packet Processor (XPP) • XBR (used only for 60-port 10/100/1000-T module)

Overview of the Tasks in the Software Upgrade Process

To upgrade all or some of the ProCurve 9408sl software images, you must perform the following general steps:

1. Determine the versions of the software images currently installed and running on your system.
2. Copy the new software image from a source to a destination.

   The source from which to copy the new image is usually a TFTP server to which the 9408sl system has access or a PCMCIA flash card inserted in the management module's slot 1 or 2. The destination to which to copy the new image is either the management module's flash memory or a flash card inserted in slot 1 or 2, or the flash memory on an interface module.
3. Reboot the upgraded module(s).

Determining the Currently Installed and Running Software

To determine the currently installed and the currently running software, use the following commands:

- **show flash** – This command displays the images currently installed in the management and interface modules' code flash and boot flash.
- **show version** – This command displays the images currently running. They may be different than the currently installed images.

ProCurve recommends using both **show flash** and **show version** commands before and after upgrading the software images.

To determine the software versions currently installed in code flash and boot flash, enter the **show flash** command at any level of the CLI, as shown in the following example:

```
9408sl# show flash
~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
Active Managment Module (Top Slot)
Code Flash - Type MT28F128J3, Size 32 MB
  o Application Image (Primary)
    Version 2.3.0cT103, Size 4839020 bytes, Check Sum 25c6
    Compiled on Feb 23 2007 at 21:13:56 labeled as mpr02300c
  o Application Image (Secondary)
    Version 2.3.0cT103, Size 4839020 bytes, Check Sum 25c6
    Compiled on Feb 23 2007 at 21:13:56 labeled as mpr02300c
  o LP Kernel Image (Monitor for LP Image Type 0)
    Version 2.3.0cT115, Size 381375 bytes, Check Sum c6d5
    Compiled on Feb 23 2007 at 20:46:14 labeled as lb02300c
  o LP Application Image (Primary for LP Image Type 0)
    Version 2.3.0cT117, Size 1669687 bytes, Check Sum 1d67
    Compiled on Feb 23 2007 at 20:45:34 labeled as lp02300c
  o LP Application Image (Secondary for LP Image Type 0)
    Version 2.3.0cT117, Size 1669687 bytes, Check Sum 1d67
    Compiled on Feb 23 2007 at 20:45:34 labeled as lp02300c
  o Boot-Monitor Image
    Version 2.3.0cT105, Size 439321 bytes, Check Sum 4d7d
    Compiled on Feb 23 2007 at 21:15:32 labeled as mb02300c
  o Startup Configuration
    Size 1815 bytes, Check Sum 6dd8
    Modified on 13:38:16 GMT+00 Mon May 14 2007

Boot Flash - Type AM29LV040B, Size 512 KB
  o Boot-Monitor Image
    Version 2.3.0cT105, Size 439321 bytes, Check Sum 4d7d
    Compiled on Feb 23 2007 at 21:15:32 labeled as mb02300c
~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
Standby Management Module (Bottom Slot)
Code Flash: Type MT28F128J3, Size 32 MB
  o Application Image (Primary)
    Version 2.3.0cT103, Size 4839020 bytes, Check Sum 25c6
    Compiled on Feb 23 2007 at 21:13:56 labeled as mpr02300c
  o Application Image (Secondary)
    Version 2.3.0cT103, Size 4839020 bytes, Check Sum 25c6
    Compiled on Feb 23 2007 at 21:13:56 labeled as mpr02300c
  o LP Kernel Image (Monitor for LP Image Type 0)
    Version 2.3.0cT115, Size 381375 bytes, Check Sum c6d5
    Compiled on Feb 23 2007 at 20:46:14 labeled as lb02300c
  o LP Application Image (Primary for LP Image Type 0)
    Version 2.3.0cT117, Size 1669687 bytes, Check Sum 1d67
    Compiled on Feb 23 2007 at 20:45:34 labeled as lp02300c
  o LP Application Image (Secondary for LP Image Type 0)
    Version 2.3.0cT117, Size 1669687 bytes, Check Sum 1d67
    Compiled on Feb 23 2007 at 20:45:34 labeled as lp02300c
```

```
  o Boot-Monitor Image
    Version 2.3.0cT105, Size 439321 bytes, Check Sum 4d7d
    Compiled on Feb 23 2007 at 21:15:32 labeled as mb02300c
  o Startup Configuration
    Size 1815 bytes, Check Sum 6dd8
    Modified on 13:38:17 GMT+00 Mon May 14 2007
Boot Flash: Type AM29LV040B, Size 512 KB
  o Boot-Monitor Image
    Version 2.3.0cT105, Size 439321 bytes, Check Sum 4d7d
    Compiled on Feb 23 2007 at 21:15:32 labeled as mb02300c
~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
Line Card Slot 4
Code Flash: Type MT28F640J3, Size 16 MB
  o Application Image (Primary)
    Version 2.3.0cT117, Size 1669687 bytes, Check Sum 1d67
    Compiled on Feb 23 2007 at 20:45:34 labeled as lp02300c
  o Application Image (Secondary)
    Version 2.3.0cT117, Size 1669687 bytes, Check Sum 1d67
    Compiled on Feb 23 2007 at 20:45:34 labeled as lp02300c
  o Boot-Monitor Image
    Version 2.3.0cT115, Size 381375 bytes, Check Sum c6d5
    Compiled on Feb 23 2007 at 20:46:14 labeled as lb02300c
Boot Flash: Type AM29LV040B, Size 512 KB
  o Boot-Monitor Image
    Version 2.3.0cT115, Size 381375 bytes, Check Sum c6d5
    Compiled on Feb 23 2007 at 20:46:14 labeled as lb02300c
FPGA Version: PBIF Ver 42 XTM Ver 89.1 XPP Ver 88.6
XTM File name: 10ge_xtm_top.ncd, Compile time: 2004/09/02 12:47:13
XPP File name: 10ge_xppe_top_6k.ncd, Compile time: 2004/10/15 10:15: 6

~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
Line Card Slot 5
Code Flash: Type MT28F640J3, Size 16 MB
  o Application Image (Primary)
    Version 2.3.0cT117, Size 1669687 bytes, Check Sum 1d67
    Compiled on Feb 23 2007 at 20:45:34 labeled as lp02300c
  o Application Image (Secondary)
    Version 2.3.0cT117, Size 1669687 bytes, Check Sum 1d67
    Compiled on Feb 23 2007 at 20:45:34 labeled as lp02300c
  o Boot-Monitor Image
    Version 2.3.0cT115, Size 381375 bytes, Check Sum c6d5
    Compiled on Feb 23 2007 at 20:46:14 labeled as lb02300c
Boot Flash: Type AM29LV040B, Size 512 KB
  o Boot-Monitor Image
    Version 2.3.0cT115, Size 381375 bytes, Check Sum c6d5
    Compiled on Feb 23 2007 at 20:46:14 labeled as lb02300c
FPGA Version: PBIF Ver 10 XTM Ver 89.1 XPP Ver 91.9 XBridge Ver 34
XTM File name: 10ge_xtm_top.ncd, Compile time: 2004/09/02 12:47:13
XPP File name: 10ge_xppf_top_6k.ncd, Compile time: 2005/11/29 17:59:20
XBR File Name: routed.ncd,  Compile time: 2006/ 2/ 2 13:42:20
```

```
~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
Line Card Slot 7
Code Flash: Type MT28F640J3, Size 16 MB
  o Application Image (Primary)
    Version 2.3.0cT117, Size 1669687 bytes, Check Sum 1d67
    Compiled on Feb 23 2007 at 20:45:34 labeled as lp02300c
  o Application Image (Secondary)
    Version 2.3.0cT117, Size 1669687 bytes, Check Sum 1d67
    Compiled on Feb 23 2007 at 20:45:34 labeled as lp02300c
  o Boot-Monitor Image
    Version 2.3.0cT115, Size 381375 bytes, Check Sum c6d5
    Compiled on Feb 23 2007 at 20:46:14 labeled as lb02300c
Boot Flash: Type AM29LV040B, Size 512 KB
  o Boot-Monitor Image
    Version 2.3.0cT115, Size 381375 bytes, Check Sum c6d5
    Compiled on Feb 23 2007 at 20:46:14 labeled as lb02300c
FPGA Version: PBIF Ver 22 XTM Ver 89.1 XPP Ver 91.9
XTM File name: 10ge_xtm_top.ncd, Compile time: 2004/09/02 12:47:13
XPP File name: 10ge_xppf_top_6k_DCI.ncd, Compile time: 2006/ 4/18 17:23:42

~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
Line Card Slot 8
Code Flash: Type M58LW064D, Size 16 MB
  o Application Image (Primary)
    Version 2.3.0cT117, Size 1669687 bytes, Check Sum 1d67
    Compiled on Feb 23 2007 at 20:45:34 labeled as lp02300c
  o Application Image (Secondary)
    Version 2.3.0cT117, Size 1669687 bytes, Check Sum 1d67
    Compiled on Feb 23 2007 at 20:45:34 labeled as lp02300c
  o Boot-Monitor Image
    Version 2.3.0cT115, Size 381375 bytes, Check Sum c6d5
    Compiled on Feb 23 2007 at 20:46:14 labeled as lb02300c
Boot Flash: Type AM29LV040B, Size 512 KB
  o Boot-Monitor Image
    Version 2.3.0cT115, Size 381375 bytes, Check Sum c6d5
    Compiled on Feb 23 2007 at 20:46:14 labeled as lb02300c
FPGA Version: PBIF Ver 42 XTM Ver 89.1 XPP Ver 88.6
XTM File name: 10ge_xtm_top.ncd, Compile time: 2004/09/02 12:47:13
XPP File name: 10ge_xppe_top_6k.ncd, Compile time: 2004/10/15 10:15: 6

~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
All show flash done
```

Table 5 explains the information provided by the **show flash** commands. For the image information, note the following:

- "Version 2.1.0Txxy" indicates the image version number. The “Txxy” is used by ProCurve for record keeping. The “xx” indicates the hardware type, while the “y” indicates the image type.
- "Size <number> bytes" indicates the size, in bytes, of the image.
- "Check Sum <value>" indicates a unique ID for the image. If the contents of the image change, the check sum value changes also.
- “Compiled on <date> at <time>” indicates the date and time that ProCurve compiled the image.
- "labeled as <name>" indicates the name of the image:
  - mb<xxxxx> indicates the boot-and-monitor image name for the management module
  - mpr<xxxxx> indicates the ProCurve software image name for the management module
  - lb<xxxxx> indicates the boot-and-monitor image name for the interface module
  - lp<xxxxx> indicates the ProCurve software image name for the interface module

Table 5: Code Flash and Boot Flash Information

This Field...	Displays...
Management Modules	
<type> Management Module (<location>)	The management module for which flash information is displayed. The <type> parameter indicates an active or standby management module. The <location> parameter indicates the top or bottom slot (M1 or M2, respectively).
Code Flash	The model number and size of the management module’s code flash.
Application Image (Primary or Secondary)	Indicates the ProCurve software image installed in the primary or secondary location in the management module’s code flash.
LP Kernel Image (Monitor for LP Image Type 0)	Indicates the interface module’s monitor image stored in the management module’s code flash if you copied the boot-and-monitor image to all interface modules using the **copy** command with the **all** keyword. The management module stores these images only; it does not run the images.
LP Application (Primary or Secondary for LP Image Type 0)	Indicates the interface modules’ primary and/or secondary ProCurve software image stored in the management module’s code flash if you copied the primary and/or secondary ProCurve software image to all interface modules using the **copy** command with the **all** keyword. The management module stores these images only; it does not run the images.
Boot-Monitor Image	Indicates the monitor image installed in the management module’s code flash.
Startup Configuration	The output displays the following information about the startup configuration, which is saved in the management module’s code flash: • Size – Size, in bytes, of the startup configuration. • Check sum – A unique ID for the file. If the contents of the file change, the check sum changes also. • Modification date and time – Date and time that the startup configuration was last saved.
Boot Flash	The model number and size of the management module’s boot flash.

Table 5: Code Flash and Boot Flash Information (Continued)

This Field...	Displays...
Boot-Monitor Image	Indicates the boot image installed in the management module's boot flash.
Interface Modules	
Line Card Slot <number>	The interface module for which flash information is displayed. The <number> parameter indicates the number of the chassis slot, 1 – 8, in which the interface module is installed.
Code Flash	The model number and size of the interface module's code flash.
Application Image (Primary or Secondary)	Indicates the ProCurve software image installed in the primary or secondary location in the interface module's code flash.
Boot-Monitor Image	Indicates the monitor image installed in the interface module's code flash.
Boot Flash	The model number and size of the interface module's boot flash.
Boot-Monitor Image	Indicates the boot image installed in the interface module's boot flash.
FPGA image Information	The output displays the following information about the field-programmable gate array (FPGA) images, which are installed on the interface module: • FPGA Version – The version number of the PBIF, XTM, XPP, and XBRIDGE (for the 60-port module) images. • XTM image information – The engineering filename and compilation date and time of the XTM image. • XPP image information – The engineering filename and compilation date and time of the XPP image. • XBR image information (for 60-port module only) – The engineering filename and compilation date and time of the XBR image.

To display the software versions currently running in code flash and boot flash in all management and interface modules, enter the **show version** command at any level of the CLI:

```
ProCurve 9408sl# show version
HW: ProCurve 9408sl Router
Backplane (Serial #: SA22040010,  Part #: 31144-001.)
Switch Fabric Module (Serial #: SA22040359,  Part #: 31300-003P,  FPGA Version: 7)

======================================================================
SL M1: J8681A Redundant Management Module Active (Serial #: SA21040248, Part #:
31148-005.):

Boot     : Version 2.3.0cT105 Copyright (c) 1996-2003 Hewlett-Packard, Inc.
Compiled on Feb 23 2007 at 21:15:32 labeled as mb02300c
 (439321 bytes) from boot flash
Monitor  : Version 2.3.0cT105 Copyright (c) 1996-2003 Hewlett-Packard, Inc.
Compiled on Feb 23 2007 at 21:15:32 labeled as mb02300c
 (439321 bytes) from code flash
Applicat : Version 2.3.0cT103 Copyright (c) 1996-2003 Hewlett-Packard, Inc.
Compiled on Feb 23 2007 at 21:13:56 labeled as mpr02300c
 (4839020 bytes) from Primary
```

```
Board ID : 61 CPLD Version : 8
800 MHz Power PC processor 750FX (version 7000/0202) 133 MHz bus
512 KB Boot Flash (AM29LV040B), 32 MB Code Flash (MT28F128J3)
512 MB DRAM
Active Management uptime is 4 minutes 18 seconds

================================================================================
SL M2: J8681A Redundant Management Module Standby (Serial #: SA21040257, Part #:
31148-005.):

Boot     : Version 2.3.0cT105 Copyright (c) 1996-2003 Hewlett-Packard, Inc.
Compiled on Feb 23 2007 at 21:15:32 labeled as mb02300c
 (439321 bytes) from boot flash
Monitor  : Version 2.3.0cT105 Copyright (c) 1996-2003 Hewlett-Packard, Inc.
Compiled on Feb 23 2007 at 21:15:32 labeled as mb02300c
 (439321 bytes) from code flash
Applicat : Version 2.3.0cT103 Copyright (c) 1996-2003 Hewlett-Packard, Inc.
Compiled on Feb 23 2007 at 21:13:56 labeled as mpr02300c
 (4839020 bytes) from Primary

Board ID: 61 CPLD Version: 8
800 MHz Power PC processor 750FX (version 7000/0202) 133 MHz bus
512 KB Boot Flash (AM29LV040B), 32 MB Code Flash (MT28F128J3)
512 MB DRAM
Standby Management uptime is 4 minutes 3 seconds

================================================================================
SL 4: J8682A 4 port 10Gig Module (IPv6) (Serial #: SA35040648, Part #: 31183-105A)

Boot     : Version 2.3.0cT115 Copyright (c) 1996-2003 Hewlett-Packard, Inc.
Compiled on Feb 23 2007 at 20:46:14 labeled as lb02300c
 (381375 bytes) from boot flash
Monitor  : Version 2.3.0cT115 Copyright (c) 1996-2003 Hewlett-Packard, Inc.
Compiled on Feb 23 2007 at 20:46:14 labeled as lb02300c
 (381375 bytes) from code flash
Applicat : Version 2.3.0cT117 Copyright (c) 1996-2003 Hewlett-Packard, Inc.
Compiled on Feb 23 2007 at 20:45:34 labeled as lp02300c
 (1669687 bytes) from Primary

--More--, next page: Space, next line: Return key, quit: Control-cWARN: Module
temperature (50.78C) in slot 5 is above threshold (50 C), need to increase fan
speed.
Set fan speed to 75%.
FPGA versions: PBIF Ver 42 XTM Ver 89.1 XPP Ver 88.6
XTM File name: 10ge_xtm_top.ncd, Compile time: 2004/09/02 12:47:13
XPP File name: 10ge_xppe_top_6k.ncd, Compile time: 2004/10/15 10:15: 6

SBIA version: 129
XGMAC 0 version: N/A
XGMAC 1 version: N/A
```

```
XGMAC 2 version: N/A
XGMAC 3 version: N/A
400 MHz Power PC processor 440GP (version 4012/0481) 133 MHz bus
512 KB Boot Flash (AM29LV040B), 16 MB Code Flash (MT28F640J3)
256 MB DRAM, 8 KB SRAM, 8 MB BRAM
PPCR0: 128K entries CAM, 4096K entries PRAM, 1024K entries AGE RAM
PPCR1: 128K entries CAM, 4096K entries PRAM, 1024K entries AGE RAM
PPCR2: 128K entries CAM, 4096K entries PRAM, 1024K entries AGE RAM
PPCR3: 128K entries CAM, 4096K entries PRAM, 1024K entries AGE RAM
LP Slot 4 uptime is 3 minutes 51 seconds

==============================================================================
SL 5: J8688A 60 port 10/100/1000-T Module (IPv6+OACL) (Serial #: SA48040042, Part #:
31507-105C)

Boot     : Version 2.3.0cT115 Copyright (c) 1996-2003 Hewlett-Packard, Inc.
Compiled on Feb 23 2007 at 20:46:14 labeled as lb02300c
 (381375 bytes) from boot flash
Monitor  : Version 2.3.0cT115 Copyright (c) 1996-2003 Hewlett-Packard, Inc.
Compiled on Feb 23 2007 at 20:46:14 labeled as lb02300c
 (381375 bytes) from code flash
Applicat : Version 2.3.0cT117 Copyright (c) 1996-2003 Hewlett-Packard, Inc.
Compiled on Feb 23 2007 at 20:45:34 labeled as lp02300c
 (1669687 bytes) from Primary

FPGA versions: PBIF Ver 10 XTM Ver 89.1 XPP Ver 91.9 XBridge Ver 34
XTM File name: 10ge_xtm_top.ncd, Compile time: 2004/09/02 12:47:13
XPP File name: 10ge_xppf_top_6k.ncd, Compile time: 2005/11/29 17:59:20
XBR File Name: routed.ncd,  Compile time: 2006/ 2/ 2 13:42:20

SBIA version: 129
GMAC 0 version: N/A
GMAC 1 version: N/A
GMAC 2 version: N/A
GMAC 3 version: N/A
400 MHz Power PC processor 440GP (version 4012/0481) 133 MHz bus
512 KB Boot Flash (AM29LV040B), 16 MB Code Flash (MT28F640J3)
256 MB DRAM, 8 KB SRAM, 8 MB BRAM
PPCR0: 128K entries CAM, 4096K entries PRAM, 1024K entries AGE RAM
PPCR1: 128K entries CAM, 4096K entries PRAM, 1024K entries AGE RAM
PPCR2: 128K entries CAM, 4096K entries PRAM, 1024K entries AGE RAM
LP Slot 5 uptime is 3 minutes 51 seconds

==============================================================================
SL 7: J8685A 40 port 10/100/1000-T Module (IPv6+OACL) (Serial #: SA21040162, Part #:
31514-001P)

Boot     : Version 2.3.0cT115 Copyright (c) 1996-2003 Hewlett-Packard, Inc.
Compiled on Feb 23 2007 at 20:46:14 labeled as lb02300c
 (381375 bytes) from boot flash
```

```
Monitor  : Version 2.3.0cT115 Copyright (c) 1996-2003 Hewlett-Packard, Inc.
Compiled on Feb 23 2007 at 20:46:14 labeled as lb02300c
 (381375 bytes) from code flash
Applicat : Version 2.3.0cT117 Copyright (c) 1996-2003 Hewlett-Packard, Inc.
Compiled on Feb 23 2007 at 20:45:34 labeled as lp02300c
 (1669687 bytes) from Primary

FPGA versions: PBIF Ver 22 XTM Ver 89.1 XPP Ver 91.9
XTM File name: 10ge_xtm_top.ncd, Compile time: 2004/09/02 12:47:13
XPP File name: 10ge_xppf_top_6k_DCI.ncd, Compile time: 2006/ 4/18 17:23:42

SBIA version: 129
GMAC 0 version: N/A
GMAC 1 version: N/A
GMAC 2 version: N/A
GMAC 3 version: N/A
400 MHz Power PC processor 440GP (version 4012/0481) 133 MHz bus
512 KB Boot Flash (AM29LV040B), 16 MB Code Flash (MT28F640J3)
256 MB DRAM, 8 KB SRAM, 8 MB BRAM
PPCR0: 128K entries CAM, 4096K entries PRAM, 1024K entries AGE RAM
PPCR1: 128K entries CAM, 4096K entries PRAM, 1024K entries AGE RAM
PPCR2: 128K entries CAM, 4096K entries PRAM, 1024K entries AGE RAM
PPCR3: 128K entries CAM, 4096K entries PRAM, 1024K entries AGE RAM
LP Slot 7 uptime is 3 minutes 51 seconds

================================================================================
SL 8: J8682A 4 port 10Gig Module (IPv6) (Serial #: PR26030094, Part #: 31183-105A)

Boot     : Version 2.3.0cT115 Copyright (c) 1996-2003 Hewlett-Packard, Inc.
Compiled on Feb 23 2007 at 20:46:14 labeled as lb02300c
 (381375 bytes) from boot flash
Monitor  : Version 2.3.0cT115 Copyright (c) 1996-2003 Hewlett-Packard, Inc.
Compiled on Feb 23 2007 at 20:46:14 labeled as lb02300c
 (381375 bytes) from code flash
Applicat : Version 2.3.0cT117 Copyright (c) 1996-2003 Hewlett-Packard, Inc.
Compiled on Feb 23 2007 at 20:45:34 labeled as lp02300c
 (1669687 bytes) from Primary

FPGA versions: PBIF Ver 42 XTM Ver 89.1 XPP Ver 88.6
XTM File name: 10ge_xtm_top.ncd, Compile time: 2004/09/02 12:47:13
XPP File name: 10ge_xppe_top_6k.ncd, Compile time: 2004/10/15 10:15: 6

SBIA version: 129
XGMAC 0 version: N/A
XGMAC 1 version: N/A
XGMAC 2 version: N/A
XGMAC 3 version: N/A
400 MHz Power PC processor 440GP (version 4012/0481) 133 MHz bus
512 KB Boot Flash (AM29LV040B), 16 MB Code Flash (M58LW064D)
256 MB DRAM, 8 KB SRAM, 8 MB BRAM
```

```
PPCR0: 128K entries CAM, 8192K entries PRAM, 1024K entries AGE RAM
PPCR1: 128K entries CAM, 8192K entries PRAM, 1024K entries AGE RAM
PPCR2: 128K entries CAM, 8192K entries PRAM, 1024K entries AGE RAM
PPCR3: 128K entries CAM, 8192K entries PRAM, 1024K entries AGE RAM
LP Slot 8 uptime is 3 minutes 51 seconds
================================================================================
All show version done
```

Syntax: show version

The highlighted lines in the output indicate the currently running boot, monitor, and ProCurve software ("Application") versions for the management and interface modules. In general, note the following:

- "2.1.0Txxy" indicates the image version number. The "Txxy" is used by ProCurve for record keeping. The "xx" indicates the hardware type, while the "y" indicates the image type.
- "Compiled on <date> at <time>" indicates the date and time that ProCurve compiled the image.
- mb<xxxxx> indicates the boot-and-monitor image name for the management module.
- mpr<xxxxx> indicates the ProCurve software image name for the management module.
- lb<xxxxx> indicates the boot-and-monitor image name for the interface module.
- lp<xxxxx> indicates the ProCurve software image name for the interface module.
- "(<number> bytes)" indicates the size, in bytes, of the image.
- "from <location>" indicates the location from which the specified image was loaded.

Verifying the Status of Upgraded Modules

To display the operational status of all management and interface modules installed in a 9408sl routing switch, enter the **show module** command at any level of the CLI.

ProCurve recommends using the **show module** command after you upgrade software images and reboot the management module, which in turn boots the interface modules.

```
ProCurve 9408sl# show module
        Module                                Status      Ports  Starting MAC
M1 (upper): J8681A Redundant Management Module     Active
M2 (lower): J8681A Redundant Management Module     Standby (Ready State)
F1: Switch Fabric Module         Active
S1: Configured as J8688A 60 port 10/100/1000-T Module
S2:
S3:
S4: J8682A 4 port 10Gig Module (IPv6)  CARD_STATE_UP    4   000c.db20.d0c0
S5: J8688A 60 port 10/100/1000-T Module (IPv6+OACL)  CARD_STATE_UP    60
000c.db20.d000
S6: Configured as J8684A 40 port mini-GBIC Module
S7: J8685A 40 port 10/100/1000-T Module (IPv6+OACL)  CARD_STATE_UP    40
000c.db20.d000
S8: J8682A 4 port 10Gig Module (IPv6)  CARD_STATE_UP    4   000c.db20.d1c0
```

Syntax: show module

Software Upgrade Procedures

NOTE: Software release 02.2.01 requires 02.2.01 boot code and monitor images.

This section explains how to upgrade the following software images on the management and interface modules:

- Monitor
- Boot
- ProCurve Software
- Field-Programmable Gate Array (FPGA) (interface modules only)

The sequence for a complete system upgrade is:

A. Upgrade the management module's monitor and boot images

B. Upgrade the management module's ProCurve software image

C. Upgrade the interface module's monitor and boot images

D. Upgrade the interface module's ProCurve software image

E. Upgrade the interface module's FPGA images

F. Reboot using the **boot system** command

G. Extra Step: Use the **sync-standby** command to regain visibility of the standby management module's spare LP primary application image. (Bug workaround)

NOTE: Steps A through E can be performed in a single step using a script to copy files from a TFTP server. A sample script is included with the software files on the web. See "Downloading a New Image Using a Script" on page 71 for general script syntax and information.

A. Upgrading the Management Module's Monitor and Boot Images

Software releases 02.0.00a and later enable you to upgrade the management module's monitor and boot images simultaneously. Both images are contained in a single file, which is placed in both the boot flash and the code flash.

To upgrade the management module's monitor and boot images simultaneously, perform the following steps:

1. Place the new monitor-and-boot-image file on a TFTP server to which the system has access or on a PCMCIA flash card inserted in slot 1 or 2.
2. Copy the new monitor-and-boot-image file to the ProCurve 9408sl management module. Enter one of the following commands at the Privileged EXEC level of the CLI:

Table 6: New Command Syntax for Upgrading Monitor and Boot Images on the Management Module

Command Syntax[a]	Description
copy tftp flash <ip-addr> <image-name> **mon copy-boot**	Copies the "image-name" file from a TFTP server at "ip-addr" to both the monitor file in code flash and the boot file in boot flash.

Table 6: New Command Syntax for Upgrading Monitor and Boot Images on the Management Module (Continued)

Command Syntax[a]	Description
copy slot1 | slot2 flash <image-name> **mon copy-boot**	Copies the "image-name" file from a flash card in slot 1 or 2 to both the monitor file in code flash and the boot file in boot flash.

a.These commands are supported in software releases 02.0.00 and later.

For software version 02.3.00c, the “image-name” filename is “mb02300.bin”.

3. Verify that the new monitor and boot images have been successfully copied to flash by using the **show flash** command. Check for the boot image, monitor image, and the date and time at which the new images were built.

B. Upgrading the Management Module’s ProCurve Software Image

To upgrade the management module’s ProCurve software image (primary or secondary), perform the following steps:

1. Place the new ProCurve software image on a TFTP server to which the ProCurve 9408sl system has access or on a PCMCIA flash card inserted in slot 1 or 2.
2. Copy the new ProCurve software image from the TFTP server or a flash card in slot 1 or 2 to the management module’s code flash. To perform this step, enter one of the following commands at the Privileged EXEC level of the CLI:
   - **copy tftp flash** <ip-addr> <image-name> **primary | secondary**
   - **copy slot1 | slot2 flash** <image-name> **primary | secondary**

   For software version 02.3.00c, the “image-name” filename is “mpr02300.bin”.
3. Verify that the new ProCurve software image has been successfully copied to the specified destination by using the **show flash** command. Check for the primary or secondary image ("Application Image") and the time that the image was built.

C. Upgrading the Interface Module’s Monitor and Boot Images

Software releases 02.0.00a and later enable you to upgrade an interface module’s monitor and boot images simultaneously. Both images are contained in a single file, which is placed in both the boot flash and the code flash.

To upgrade an interface module’s monitor and boot images simultaneously, perform the following steps:

1. Place the new monitor-and-boot-image file on a TFTP server to which the system has access or on a PCMCIA flash card inserted in slot 1 or 2.

2. Copy the new monitor-and-boot-image file to the interface module(s). Enter one of the following commands at the Privileged EXEC level of the CLI:

Table 7: New Command Syntax for Upgrading the Monitor and Boot Images on the Interface Module

Command Syntax[a]	Description
copy tftp lp <ip-addr> <image-name> **monitor copy-boot all | <slot-number>**	Copies the “image-name” file from a TFTP server at “ip-addr” to all interface modules or to the specified interface module (slot-number), placing it as both the monitor file in code flash and the boot file in boot flash.
copy slot1 | slot2 lp <image-name> **monitor copy-boot all | <slot-number>**	Copies the "image-name" file from a flash card in slot 1 or 2 to all interface modules or to the specified interface module (slot-number), placing it as both the monitor file in code flash and the boot file in boot flash.

a.These commands are supported in software releases 02.0.00 and later.

For software version 02.3.00c, the “image-name” filename is “lb02300.bin”.

NOTE: If you copy the new monitor-and-boot image to all interface modules using the all keyword, the management module makes a copy of the image (called lp-monitor-0) and stores it in its code flash. If you copy the new monitor-and-boot image to a specified chassis slot, the management module does not make a copy of the image.

3. Verify that the new monitor and boot images were successfully copied to flash by using the **show flash** command. Check for the monitor image, boot image, and the date and time at which the new images were built.

D. Upgrading the Interface Module’s ProCurve Software Image

To upgrade the ProCurve software image (primary or secondary) on all interface modules or an interface module in a specified chassis slot, perform the following steps:

1. Place the new ProCurve software image on a TFTP server to which the system has access or on a PCMCIA flash card inserted in slot 1 or 2.
2. Copy the new ProCurve software image from the TFTP server or a flash card in slot 1 or 2 to all interface modules or an interface module in a specified chassis slot. To perform this step, enter one of the following commands at the Privileged EXEC level of the CLI:
   - **copy tftp lp** <ip-addr> <image-name> **primary | secondary all | <chassis-slot-number>**
   - **copy slot1 | slot2 lp** <image-name> **primary | secondary all | <chassis-slot-number>**

For software version 02.3.00c, the “image-name” filename is “lp02300.bin”.

NOTE: If you copy the new ProCurve software image to all interface modules using the **all** keyword, the management module makes a copy of the image (called lp-primary-0 or lp-secondary-0) and stores it in its code flash. If you copy the new ProCurve software image to a specified chassis slot, the management module does not make a copy of the image.

3. Verify that the new ProCurve software image has been successfully copied by entering the following command at any level of the CLI:

   show flash

   Check for the ProCurve software image ("Application Image") and the date and time at which the image was built.

E. Upgrading an FPGA for a Gigabit Ethernet Module

The Gigabit Ethernet modules contain the following upgradable field-programmable gate array (FPGA) images:

- PBIF
- XTM
- XPP
- XBRIDGE (60-port module only)

When you upgrade the ProCurve 9408sl software, it is important to also upgrade all FPGA images.

Determining the FPGA Image Versions

Normally, the **show flash** output identifies the currently-installed images, and the **show version** output identifies the currently-running images. However, the FPGA versions that are currently installed and currently running on an interface module are not correctly displayed until the interface module is power-cycled! The power-cycle of the interface modules is accomplished by one of these procedures:

- reboot the 9408sl using the **boot system** command
- power-cycle each interface module using the **lp power-off <slot>** and **lp power-on <slot>** commands
- physically power-cycle the 9408sl routing switch

If you are not sure if the interface modules were power-cycled since installing FPGA images, you may want to perform one of the listed procedures now. After that, you can use the **show flash** and **show version** commands to determine the FPGA versions currently installed and currently running on the interface modules.

NOTE: Not all FPGA versions are necessarily updated with each new software release for the ProCurve 9408sl routing switch. Also, FPGA versions are not necessarily the same for all interface modules. ProCurve indicates the set of FPGA files applicable for each software release by appending the software version to the filename. FPGA versions for each interface module are documented on page 11 of these Release Notes.

Upgrading the FPGA Images

To upgrade the FPGA images on a Gigabit Ethernet module, perform the following steps:

1. Place the new FPGA image(s) on a TFTP server to which the system has access or on a PCMCIA flash card inserted in slot 1 or 2.
2. Copy the PBIF image from the TFTP server or a flash card in slot 1 or 2 to all interface modules or an interface module in a specified chassis slot. To perform this step, enter one of the following commands at the Privileged EXEC level of the CLI:
   - **copy tftp lp** <ip-addr> <image-name> **fpga-pbif all [<module-type>]**
   - **copy tftp lp** <ip-addr> <image-name> **fpga-pbif** <chassis-slot-number>
   - **copy slot1 | slot2 lp** <image-name> **fpga-pbif all [<module-type>]**

- **copy slot1 | slot2 lp** <image-name> **fpga-pbif** <chassis-slot-number>

If you specify the module-type (e.g., 4x10g), the ProCurve 9408sl copies the PBIF images for that particular module only. If you specify **all** without a module-type, the system copies the appropriate PBIF images to their corresponding modules.

For software version 02.3.00c, the “image-name” filename is “pbif02300.bin”.

3. Copy the XTM image from the TFTP server or a flash card in slot 1 or 2 to all interface modules or an interface module in a specified chassis slot. To perform this step, enter one of the following commands at the Privileged EXEC level of the CLI:

   - **copy tftp lp** <ip-addr> <image-name> **fpga-xtm all**
   - **copy tftp lp** <ip-addr> <image-name> **fpga-xtm** <chassis-slot-number>
   - **copy slot1 | slot2 lp** <image-name> **fpga-xtm all**
   - **copy slot1 | slot2 lp** <image-name> **fpga-xtm** <chassis-slot-number>

   For the XTM image, there is no option to specify “module-type”.

   For software version 02.3.00c, the “image-name” filename is “xtm02300.bin”.

4. Copy the XPP image from the TFTP server or a flash card in slot 1 or 2 to all interface modules or an interface module in a specified chassis slot. To perform this step, enter one of the following commands at the Privileged EXEC level of the CLI:

   - **copy tftp lp** <ip-addr> <image-name> **fpga-xpp all [<module-type>]**
   - **copy tftp lp** <ip-addr> <image-name> **fpga-xpp** <chassis-slot-number>
   - **copy slot1 | slot2 lp** <image-name> **fpga-xpp all [<module-type>]**
   - **copy slot1 | slot2 lp** <image-name> **fpga-xpp** <chassis-slot-number>

   If you specify the module-type (e.g., 4x10g), the ProCurve 9408sl copies the XPP images for that particular module only. If you specify **all** without a module-type, the ProCurve 9408sl copies the appropriate XPP images to their corresponding modules.

   For software version 02.3.00c, the “image-name” filename is “xpp02300.bin”.

5. Copy the XBRIDGE image from the TFTP server or a flash card in slot 1 or 2 to all interface modules or an interface module in a specified chassis slot. To perform this step, enter one of the following commands at the Privileged EXEC level of the CLI:

   - **copy tftp lp** <ip-addr> <image-name> **fpga-xbridge all [<module-type>]**
   - **copy tftp lp** <ip-addr> <image-name> **fpga-xbridge** <chassis-slot-number>
   - **copy slot1 | slot2 lp** <image-name> **fpga-xbridge all [<module-type>]**
   - **copy slot1 | slot2 lp** <image-name> **fpga-xbridge** <chassis-slot-number>

   If you specify the module-type (e.g., 1gx60-gc-v6), the ProCurve 9408sl copies the xbridge images for that particular module only. If you specify **all** without a module-type, the ProCurve 9408sl copies the appropriate xbridge images to their corresponding modules.

   For software version 02.3.00c, the “image-name” filename is “xbridge02300.bin”.

F. Rebooting the Management Module

After upgrading the software images on the management and interface modules, you must reboot the management module. After the management module reboots, it in turn reboots the interface modules.

Furthermore, each interface module must be power-cycled in order for the new FPGA images to be loaded. Therefore, you must reboot the system using the **boot system** command (not the **reload** command). Use this command to reboot the management module, specifying **primary** or **secondary** to correspond with where you placed the new software images:

- boot system flash primary | secondary

During the management module reboot, the following synchronization events occur:

- If you have a standby management module, the active management module compares the standby module's monitor, primary, and secondary images to its own. If you have updated these images on the active module, the active module automatically synchronizes the standby module's images with its own.
- If you copied the primary and/or secondary ProCurve software image and/or monitor-and-boot image to all interface modules using the **copy** command with the **all** keyword, the management module made a copy of the image and stored it in its code flash under the names lp-primary-0, lp-secondary-0 or lp-monitor-0. By default, the system checks the interface modules' ProCurve software images, which reside in the code flash of the interface modules and the management module to make sure they are the same in both locations. (The interface module images are retained on the management module for storage only, and are not run by the management or interface modules.) If the images stored on the interface and management modules are different, the system automatically enters "interactive mode" and prompts you to do the following:
  - If you want to update the ProCurve software images in the interface module's code flash with the images in the management module's code flash, enter the **lp cont-boot sync <slot-number> | all** command at the Privileged EXEC prompt.
  - If you want to retain the ProCurve software images in the interface module's code flash, enter the **lp cont-boot no-sync <slot-number> | all** command at the Privileged EXEC prompt.

NOTE: If you do not enter a command within 60 seconds, the synchronization proceeds automatically.

After the management module finishes booting, do the following:

- Enter the **show module** command at any CLI level, and verify that the status of all interface modules is CARD_STATE_UP.
- Enter the **show version** command at any CLI level, and verify that all management and interface modules are running the new software image version.

If you find that an interface module is in a waiting state or is running an older software image, then you may have forgotten to enter the **lp cont-boot sync <slot-number>** command at the Privileged EXEC prompt.

G. Extra Step!

After upgrading from 02.1.00c to 02.3.00c, the **show flash** output does not display the LP primary application image information for the standby management module. As a workaround to this bug, issue the **sync-standby** command, which forces the standby management module to synchronize with the active management module, and solves the display issue.

Diagnostic Error Codes and Remedies for TFTP Transfers

If an error occurs with a TFTP transfer to or from a ProCurve 9408sl routing switch, one of the following error codes are displayed.

Error code	Message	Explanation and action
1	Flash read preparation failed.	A flash error occurred during the download. Retry the download. If it fails again, contact customer support
2	Flash read failed.	
3	Flash write preparation failed.	
4	Flash write failed.	
5	TFTP session timeout.	TFTP failed because of a time out. Check IP connectivity and make sure the TFTP server is running.
6	TFTP out of buffer space.	The file is larger than the amount of space on the device or TFTP server. If you are copying an image file to flash, first copy the other image to your TFTP server, then delete it from flash. (Use the **erase flash...** CLI command at the Privileged EXEC level to erase the image in the flash.) If you are copying a configuration file to flash, edit the file to remove unneeded information, then try again.
7	TFTP busy, only one TFTP session can be active.	Another TFTP transfer is active on another CLI session, SNMP, or Web management session. Wait, then retry the transfer.
8	File type check failed.	You accidentally attempted to copy the incorrect image code into the system. Retry the transfer using the correct image.
16	TFTP remote - general error.	The TFTP configuration has an error. The specific error message describes the error. Correct the error, then retry the transfer.
17	TFTP remote - no such file.	
18	TFTP remote - access violation.	
19	TFTP remote - disk full.	
20	TFTP remote - illegal operation.	
21	TFTP remote - unknown transfer ID.	
22	TFTP remote - file already exists.	
23	TFTP remote - no such user.	

Important! Required Fan Threshold Settings

Users Must Change SFM Defaults in 02.3.00c

Software version 02.3.00c has incorrect default fan temperature threshold values for the Switch Fabric Module (SFM). Users are advised to modify the SFM fan temperature thresholds and to save that setting in the config file. ProCurve recommends the same thresholds for the SFM that are defaults for the interface modules. Use these commands to apply and save the recommended SFM fan temperature thresholds:

```
ProCurveRS(config)# fan-threshold switch-fabric low 50 med 46 55 med-hi 51 60 hi
56 85

ProCurveRS(config)# write mem
```

Explanation: Figure 1 below shows the recommended fan temperature thresholds, and visually demonstrates the relationship between one speed's high threshold and the next higher speed's low threshold. (Some of the default SFM thresholds in 02.3.00c violate the rule that the low temperature threshold of a higher fan speed must be lower than the high temperature threshold of the lower fan speed.)

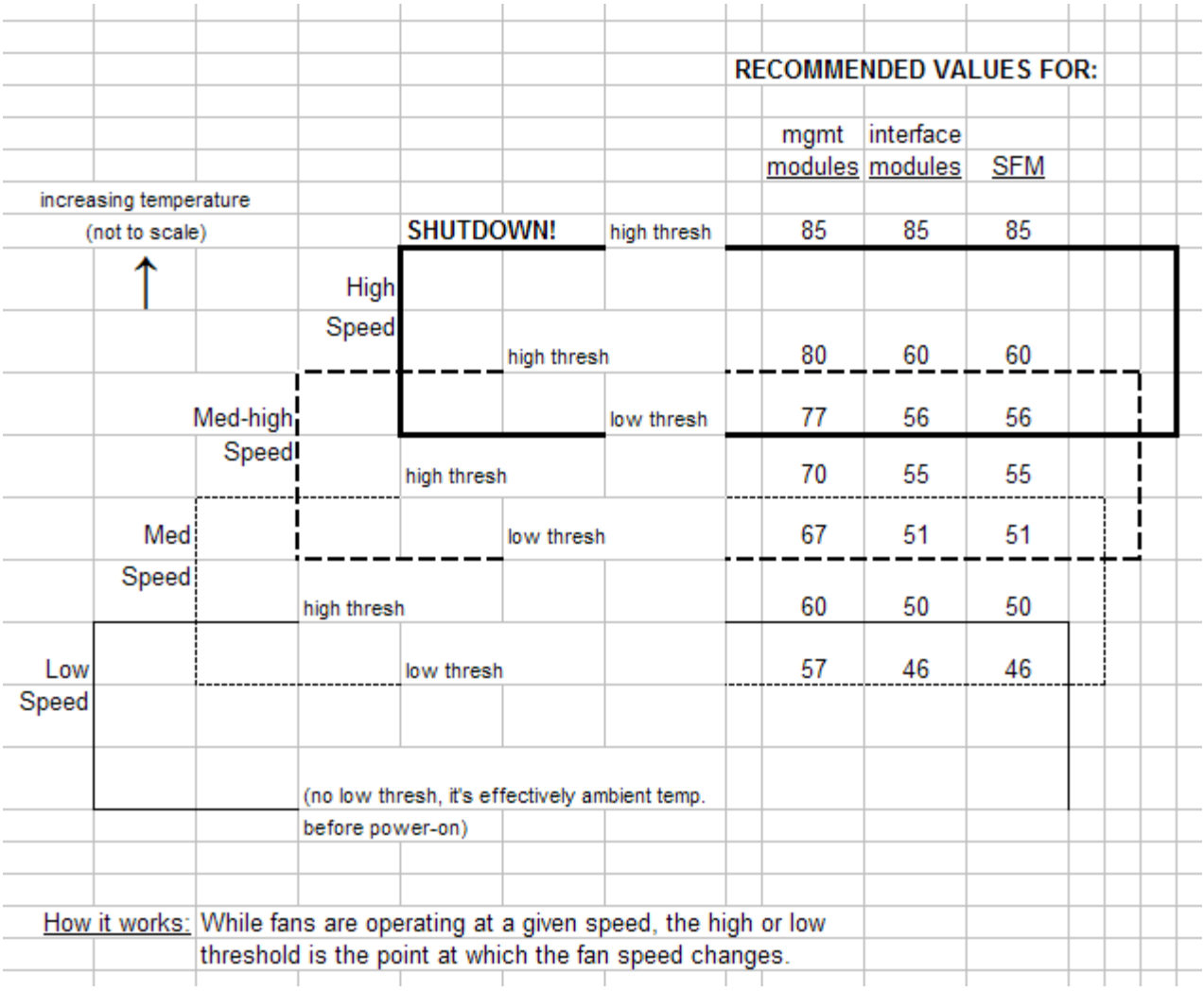


Figure 1 Fan temperature thresholds in the 9408sl

Recovering from a Lost Password

By default, the CLI does not require passwords. However, if someone has configured a password for the device but the password has been lost, you can regain super-user access to the device using the following procedure.

NOTE: Recovery from a lost password requires direct access to the serial port and a system reset.

To recover from a lost password:

1. Start a CLI session over the serial interface to the ProCurve 9408sl Switch.
2. Reboot the device.
3. While the system is booting, before the initial system prompt appears, enter **b** to enter the boot monitor mode.
4. Enter **no password** at the prompt. (You cannot abbreviate this command.)
5. Enter **boot system flash primary** at the prompt. This command causes the device to bypass the system password check.
6. After the console prompt reappears, assign a new password.

Enhancements and Configuration Notes in Release 02.0.00a to 02.0.04

This section provides details about the enhancements and configuration differences in releases 02.0.00a to 02.0.04 for the ProCurve 9408sl.

CAM Partitioning by Block

Content Addressable Memory (CAM) is a component of ProCurve modules that facilitates hardware forwarding. As packets flow through the 9408sl from source to destination, the management processor records forwarding information about the flow in CAM entries. In the 9408sl, the CAM is allocated to maintain forwarding information in separate CAM blocks for each of the following applications:

- session-mac — The Layer 4 + source MAC partition.
- ip-mac — The Layer 3 + destination MAC partition.
- out-session — The Layer 4 CAM partition.
- ipv6 — The IPv6 Layer 3 CAM partition.
- ipv6-session — The IPv6 Layer 4 CAM partition.

NOTE: Beginning with software release 02.0.02, CAM can only be partitioned globally, not on a per-slot basis. If you try to partition it by slot, it will be interpreted globally. If this software release is installed on a system that has an old configuration that specifies a per-slot CAM configuration, the last configuration on the last port will become the global one.

CAM partition block allocations

The default allocations are listed in Table 8. In most cases, this will be adequate for your needs. You can however, change this allocation to better suit your application. For example, if you are not running IPv6, you could reduce your CAM allocation for IPv6 applications to 0 and use the additional CAM available for another purpose. The next section describes how to allocate CAM partition blocks using the CLI.

Table 8: Default CAM partition allocation

9 meg module	18 meg modules	Allocation Parameter
1 block	2 blocks	session-mac
1 block	2 blocks	ip-mac

Table 8: Default CAM partition allocation (Continued)

9 meg module	18 meg modules	Allocation Parameter
	1 block	out-session
1 block	2 blocks	ipv6
1 block	1 block	ipv6-session

CLI commands for CAM partitioning

The **cam-partition** block command used to allocate a block of CAM is described in the following:

```
9408sl(config)#cam-partition block
```

Syntax: cam-partition block session-mac <blocks_allocated> ip-mac <blocks_allocated> out-session <blocks_allocated> ipv6 <blocks_allocated> ipv6-session <blocks_allocated>

<blocks_allocated> specifies the number of blocks allocated to the specified allocation parameter. A total of 4 blocks are available for 9 meg interface modules (called LV or low value) and 8 blocks for 19 meg interface (called HV or high value) modules.

EXAMPLE:

If you are not running IPv6 and want to use the 2 blocks allocated to it by default to increase the allocation for ip-mac, use the following command:

```
9408sl(config)#cam-partition block session-mac 1 ip-mac 2 out-session 1 ipv6 0 ipv6-
session 0
```

NOTE: You must define a value for each allocation parameter. If you don't want to allocate a block of CAM to a specific parameter, assign it a value of 0.

From the CLI, you will be presented with cam-partition options of ip, ipv6, mac, and session in addition to block as shown in the following:

```
9408sl(config)#cam-partition ?
  block     Block entry partition
  ip        IP entry partition
  ipv6      IP entry partition
  mac       MAC entry partition
  session   Session entry partition
```

Protocol-Based VLANs

Protocol-based VLANs provide the ability to define separate broadcast domains for several unique Layer 3 protocols within a single Layer 2 broadcast domain. Some applications for this feature might include security between departments with unique protocol requirements. This feature enables you to limit the amount of broadcast traffic to end-stations, servers, and routers.

ProCurve software release 02.0.02 provides support for the following protocol-based protocols:

- AppleTalk – The device sends AppleTalk broadcasts to all ports within the AppleTalk protocol VLAN.
- IPv4 – The device sends IPv4 broadcasts to all ports within the IP protocol VLAN.
- IPv6 – The device sends IPv6 broadcasts to all ports within the IPv6 protocol VLAN.
- IPX – The device sends IPX broadcasts to all ports within the IPX protocol VLAN.
- Other – For all other protocols that have not been configured as protocol-VLANs under this VLAN.

Protocol-based VLANs can have the following membership types:

- Static ports – Static ports are permanent members of the protocol-based VLAN and remain active members

of the VLAN regardless of whether the ports receive traffic for the VLAN's protocol.

- Exclude ports – Prevents a port in a port-based VLAN from ever becoming a member of a protocol-based VLAN.

For details on protocol-based VLANs in ProCurve devices, refer to the *Installation and Basic Configuration Guide for ProCurve 9300 Series Routing Switches*.

Configuration Considerations

Note the following configuration limitations for this feature:

- The dynamic protocol VLAN option is not supported.
- The other-protocol option defines a protocol-based VLAN for protocols that do not require a singular protocol broadcast domain or are not currently supported on the ProCurve device. It is used as a catch-all rule to mean all other protocols in addition to those already assigned. For example in the following VLAN configuration IP protocol is defined and the "other-proto" option is set to become operational when a non-IPv4 packet is received.

```
9408sl(config)#vlan 5
9408sl(config-vlan-5)#ip-proto
9408sl(config-vlan-5)#other-proto
```

Unicast Flooding on VLAN Ports

Software release 02.0.02 allows 9408sl devices to perform hardware flooding for Layer 2 unknown unicast packets on all ports on a VLAN. When this feature is enabled on a VLAN a "catch-all" CAM entry is added for the VLAN entry.

This CAM entry matches all unicast packets that have not been matched in other CAM entries. This CAM entry forces the packet to be flooded in hardware to the VLAN broadcast domain. In order for software to add CAM entries for MAC addresses that are eventually learned, a few packets need to be sent to the CPU from time to time. This is done by removing and adding the match-all CAM entry at fixed intervals.

To enable unicast flooding on a VLAN ports, enter commands such as the following:

```
9408sl(config)# vlan 2
9408sl(config-vlan-2)# unknown-unicast-flooding
9408sl(config-vlan-2)# exit
9408sl(config)# reload
```

Syntax: [no] unknown-unicast-flooding

You must reboot the 9408sl to activate the feature.

Configuration Considerations

Note the following configuration limitations for this feature:

- This feature is not supported on Layer 3 protocol-based VLANs.
- You cannot enable this feature on the designated management VLAN for the device.
- The **system-max vlan-multicast-flooding** command needs to be set to reserve CAM space for the unknown-unicast flooding CAM entries. Only when this is done can the configuration proceed.

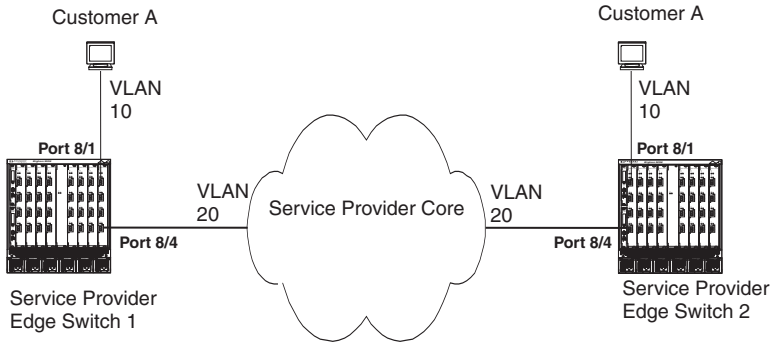
VLAN Translation

VLAN Translation allows traffic from one VLAN to be transported across a different VLAN. Under this feature, packets from the original VLAN have their VLAN ID changed at the ingress port of the VLAN that is performing the translation. When they reach the egress point on the VLAN that performed the translation, the VLAN ID is translated back to its original ID.

This feature is useful for service providers who need to carry traffic from different customers across their network while preserving the VLAN ID and priority information of the customer's network. For instance, in the following example Customer A has two geographically divided networks in the same IP subnet that are both in VLAN 10.

The service provider uses VLAN 20 to route the traffic between these two geographically divided portions of VLAN 10. Each of the service provider edge switches performs VLAN translation to translate the VLAN ID between VLAN 10 and VLAN 20.

Figure 2 VLAN Translation Example



Configuration Considerations

1. A port must be a member of the translated VLAN before it can be used in its VLAN translation group.
2. A port-VLAN pair can only be used in one VLAN translation group.
3. Up to 4096 VLAN translation groups can be configured on a switch.
4. VLAN translation should not be combined on the same port with any Layer 4 features such as ACLs, policy-based routing, or ACL-based rate-limiting.
5. Only the primary port of a trunk group can be added to a VLAN translation group. Other ports are then automatically included in the VLAN translation group.
6. If VLAN translation is enabled on a port, hardware forwarding of unknown unicast packets should not be enabled on that port.
7. This feature is currently only supported on 40-port modules.
8. VLAN translation cannot be configured on virtual ports.

CLI Command for VLAN Translation

The following command required for VLAN Translation configures a VLAN Translation group and assigns interfaces to it.

This command creates a VLAN Translation Group. Packets that arrive on a port that is configured to be in a VLAN translation group are forwarded based on the destination MAC address. First the destination MAC address in the translated VLAN is used. If the port on which the destination MAC address is learned is a member of the translated VLAN and configured in the same VLAN translation group, then the packet is forwarded to that port. The VLAN ID is replaced with the translated VLAN ID. If the port is not part of the VLAN translation group then the destination MAC address in the ingress port's VLAN is used for packet forwarding. If the destination MAC address does not exist, then the packet is flooded to the ingress port's VLAN as well as the translated VLAN.

Syntax:

```
vlan-translate-group <number>

(config-vlan-translate-group)#port <port_id> vlan-id <vlan_id>
```

number is the decimal number that you assign to a VLAN translation group.

port_id is the slot/port number that you want to configure in the VLAN translation group.

vlan_id is the VLAN number that the port specified in port_id is assigned to. The port must be separately configured in this VLAN.

EXAMPLE:

The following example creates vlan-translate-group number 1 and adds port 1/2 in VLAN 10 and port 1/5 in VLAN 20 to it.

```
9408sl(config)# vlan-translate-group 1
9408sl(config-vlan-translate-group)# port 1/2 vlan-id 10
9408sl(config-vlan-translate-group)# port 1/5 vlan-id 20
```

Configuration Example

This section describes the configuration required to enable the configuration described in Figure 2 for Service provider edge switches 1 and 2.

Service Provider Edge Switch 1 Configuration

Each port used for the VLAN translation must first be configured in its VLAN as shown below.

```
9408sl(config)# vlan 10
9408sl(config-vlan-10)# untagged ethernet 8/1
9408sl(config)# vlan 20
9408sl(config-vlan-20)# tagged ethernet 8/4
```

Each port used for VLAN translation must be added to a VLAN Translate group as shown below.

```
9408sl(config)# vlan-translate-group 1
9408sl(config-vlan-translate-group-1)# port 8/1 vlan-id 10
9408sl(config-vlan-translate-group-1)# port 8/4 vlan-id 20
```

Service Provider Edge Switch 2 Configuration

Each port used for the VLAN translation must first be configured in its VLAN as shown below.

```
9408sl(config)# vlan 10
9408sl(config-vlan-10)# untagged ethernet 8/1
9408sl(config)# vlan 20
9408sl(config-vlan-20)# tagged ethernet 8/4
```

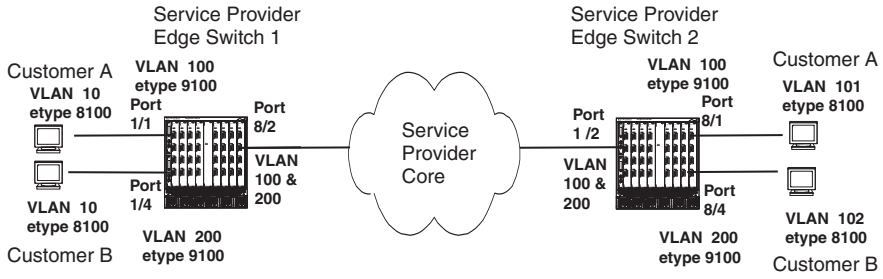
Each port used for VLAN translation must be added to a VLAN Translate group as shown below.

```
9408sl(config)# vlan-translate-group 1
9408sl(config-vlan-translate-group-1)# port 8/1 vlan-id 10
9408sl(config-vlan-translate-group-1)# port 8/4 vlan-id 20
```

Inner VLAN Translation with Super Aggregated VLANs

Inner VLAN translation is supported for packets with two VLAN tags. VLAN translation can be performed on the inner VLAN tag. In the following example, packets from customers A and B are tagged with VLAN 10 and etype 8100. Packets from customer A enter Service Provider Edge Switch 1 in VLAN 100, and packets from customer B enter Service Provider Edge Switch 1 in VLAN 200. The etype of both the ingress ports is set to 9100. The egress port on Service Provider Edge Switch 1 is contained within both VLANs 100 and 200 with etype set to 9100. Packets sent out on the egress port have two VLAN tags. On the ingress port of Service Provider Edge Switch 2 inner VLAN translation is set to translate traffic tagged with an outer VLAN tag of 100 and an inner tag of VLAN 10 to VLAN 101. Inner VLAN translation is also set to translate traffic tagged with an outer VLAN tag of 200 and an inner tag of VLAN 10 to VLAN 102. When the traffic from Service Provider Edge Switch 1 arrives at Service Provider Edge Switch 2, packets with outer VLAN tag 100 and inner VLAN tag 10 are translated to inner VLAN tag 101. Packets with outer VLAN tag 200 and inner VLAN tag 10 are translated to inner VLAN tag 102. The outer tag remains unchanged in both cases. The packet forwarding is done based on the outer VLAN tag.

Figure 3 VLAN Translation with Super Aggregated VLANs Example



Configuration Considerations

1. Inner-VLAN translation cannot be configured on virtual ports.
2. The port on which the inner-VLAN transition is configured must be a member of the outer VLAN.
3. VLAN translation and inner-VLAN translation cannot be enabled on a port at the same time.
4. If inner-VLAN translation is enabled on a port, hardware forwarding of unknown unicast packets should not be enabled on that port.
5. For a given interface, the (outer-VLAN, inner-VLAN) pair in the translation rule must be unique.
6. For trunk ports, inner-VLAN translation can be configured on the primary ports only. The configuration then applies to all ports of the trunk port.
7. There is no limit on the number of inner VLAN translation policies that can be applied to a port.
8. The trunk is rejected if any of the trunk's have VLAN or inner-VLAN translation configured.

CLI Command to Configure an Interface for VLAN Translation on a Super Aggregated VLAN

The following command is required to apply VLAN Translation for a Super Aggregated VLAN.

This command creates a VLAN translation rule on an interface used in a Super Aggregated VLAN.

Syntax:

```
inner-vlan-translate <outer_vlan_tag> <inner_vlan_tag> <translation_vlan_tag>
```

outer_vlan_tag specifies outer vlan tag of the packet with two VLAN tags. This VLAN tag is maintained with the packets through the translation process.

inner_vlan_tag specifies inner vlan tag of the packet that needs to be translated.

translation_vlan_tag specifies vlan tag that the inner VLAN tag will be translated to.

EXAMPLE:

The following example applies a VLAN translation rule to interface 1/2 to translate traffic with an outer VLAN tag of 100 and an inner VLAN tag of 10 to an outer VLAN tag of 101.

```
9408sl(config)# interface ethernet 1/2
9408sl(config-if-e1000-1/2)# inner-vlan-translate 100 10 101
```

Configuration Example

This section describes the syntax required to enable the configuration described in Figure 3 for Service provider edge switches 1 and 2.

Service Provider Edge Switch 1 Configuration

Each port used for the VLAN translation must first be configured in its VLAN as shown below.

```
9408sl(config)# vlan 100
9408sl(config-vlan-100)# untagged ethernet 1/1
9408sl(config-vlan-100)# tagged ethernet 8/2
9408sl(config)# vlan 200
9408sl(config-vlan-200)# untagged ethernet 1/4
9408sl(config-vlan-200)# tagged ethernet 8/2
```

For Super Aggregated VLANs (SAV), VLAN translation is configured under an interface as an inbound feature. For SAVs, the outer VLAN, inner VLAN and translation VLAN must be configured. The configuration for interface 8/2 in the example in Figure 3 is shown below.

```
9408sl(config)# interface ethernet 8/2
9408sl(config-if-e1000-8/2)# inner-vlan-translate 100 101 10
9408sl(config-if-e1000-8/2)# inner-vlan-translate 200 102 10
```

Service Provider Edge Switch 2 Configuration

Each port used for the VLAN translation must first be configured in its VLAN as shown below.

```
9408sl(config)# vlan 100 by port
9408sl(config-vlan-100)# untagged ethernet 8/1
9408sl(config-vlan-100)# tagged ethernet 1/2
9408sl(config)# vlan 200 by port
9408sl(config-vlan-200)# untagged ethernet 8/4
9408sl(config-vlan-200)# tagged ethernet 1/2
```

For Super Aggregated VLANs (SAV), VLAN translation is configured under an interface as an inbound feature. For SAVs, the outer VLAN, inner VLAN and translation VLAN must be configured. The configuration for interface 1/2 in the example in Figure 3 is shown below.

```
9408sl(config)# interface ethernet 1/2
9408sl(config-if-e1000-1/2)# inner-vlan-translate 100 10 101
9408sl(config-if-e1000-1/2)# inner-vlan-translate 200 10 102
```

CAM Partitioning for VLAN Translation

By default, there is no CAM space allocated for VLAN translation. To perform VLAN translation in hardware, allocate CAM space by using the following CAM partition command:

```
cam-partition block vlan-session 20% mac-session 30% flow-percent 90%
```

The above command reserves 20% of CAM space allocated for IPV6 for inner VLAN translation and 30% of CAM space allocated for IPV6 for VLAN translation and Layer 2 ACLs. The flow-percent parameter further divides this space into two parts: 90% for VLAN translation, and 10% for Layer 2 ACLs. Depending on your requirements, these percentages can be adjusted. A reload is required after a CAM partition command is configured for the CAM partition to take effect.

Support for Outbound ACLs and IPv6

All ProCurve 9408sl interface modules support simultaneous IPv4 and IPv6 and outbound IPv4 ACLs beginning with software version 02.0.02.

Layer 2 Hitless Failover

The Layer 2 Hitless Failover feature provides automatic failover from the active management module to the standby management module without interrupting operation of any interface modules in the chassis. Configuration changes made from the CLI to the active management module are also written to the standby management module even if they are not written to flash memory.

NOTE: Since both the standby and active management modules run the same code, a command that brings down the active management module will most likely bring down the standby management module. Because all configuration commands are synchronized from active to standby management module in real time, both management modules will crash at almost the same time. This in turn causes the system to reset all interface modules (similar to the behavior when the 'reboot' command is executed) and causes packet loss associated with a system reboot.

Once booted, the redundant management module keeps up-to-date copies of the active module's running configuration. Layer 2 protocols such as STP, RSTP, MRP, and VSRP are run concurrently on both the active and standby management modules. Upon the failover of the active management module, the standby module takes over as the active management module and picks up where the active module left off, without interrupting any Layer 2 traffic.

The interface modules are not reset, as they are with the previous cold-restart redundancy feature. The interface modules continue to forward traffic while the standby management module takes over operation of the system. The new now-active management module receives updates from the interface modules and sends verification information to the interface modules to ensure that they are synchronized.

If the new active management module becomes out-of-sync with an interface module, information on the interface module can be overwritten in some cases which can cause an interruption of traffic forwarding. Layer 3 hitless failover is not supported in software release 02.0.02. Consequently, a failover will result in a re-synchronization of Layer 3 data structures.

NOTE: The Redundancy CONFIG level command **running-config-sync-period** is removed beginning with software release 02.0.02, because with the Hitless Failover feature, CLI configuration is synced immediately.

New show ip vrrp statistics Output

The show ip **vrrp statistics** command displays more statistics beginning with software release 02.0.02. An example of this is shown in the following:

```
mg1#show ip vrrp statistics

Global VRRP statistics
----------------------
- received vrrp packets with checksum errors = 0
- received vrrp packets with invalid version number = 0
- received vrrp packets with unknown or inactive vrid = 0

Interface 1/1
---------------
VRID 1
- number of transitions to backup state = 2
- number of transitions to master state = 1
- total number of vrrp packets received = 129
  . received backup advertisements = 0
  . received packets with zero priority = 1
  . received packets with invalid type = 0
  . received packets with invalid authentication type = 0
  . received packets with authentication type mismatch = 0
  . received packets with authentication failures = 0
  . received packets dropped by owner = 0
  . received packets with ip ttl errors = 0
  . received packets with ip address mismatch = 0
  . received packets with advertisement interval mismatch = 0
  . received packets with invalid length = 0
- total number of vrrp packets sent = 2018
  . sent backup advertisements = 0
  . sent packets with zero priority = 0
```

```
- received arp packets dropped = 0
- received proxy arp packets dropped = 0
- received ip packets dropped = 0
```

802.1Q Tag-type Translation - Per-port Regions

The 802.1Q feature is implemented in all releases of the 9408sl. It is documented in detail in the "*Installation and Basic Configuration Guide for ProCurve 9300 Series Routing Switches*". See the *Configuring 802.1q Tag-type Translation* section of the *Configuring Virtual LANs* chapter. On the 9408sl multiple 802.1Q tag types can be assigned to an interface module. Depending on the module, an 802.1Q tag can be assigned to an individual port or to a group of ports. Table 9 describes the granularity at which each of the 9408sl interface modules can have 802.1Q tag-types assigned.

Table 9: 802.1Q tag-type assignments by module

module type	802.1Q tag-type assignment
4 x 10G	per port
40 x 1G	per 10 ports: 1 - 10, 11 - 20, 21 - 30, 31 - 40
60 x 1G	per 20 ports: 1 - 20, 21 - 40, 41 - 60

New Interface Module Temperature Threshold Values

The default and recommended low and high temperature thresholds for fan speeds on interface modules is changed beginning with software release 02.0.02. Table 10 provides the new default low and high temperature thresholds for each fan speed on ProCurve 9408sl interface modules.

Table 10: Default and Recommended Low and High Temperature Thresholds for Interface Modules and Fan Speeds

Fan Speed	Low Temperature Threshold	High Temperature Threshold
Interface modules		
High	56° C	85° C
Medium-high	51° C	60° C
Medium	46° C	55° C
Low	–	50° C

New Gigabit Ethernet Interface Modules

Release 02.0.02 for the ProCurve 9408sl introduces the following new interface module:

- 60-port 1 Gigabit Ethernet interface Module (copper)

You can install up to eight interface modules in the chassis slots of the ProCurve 9408sl.

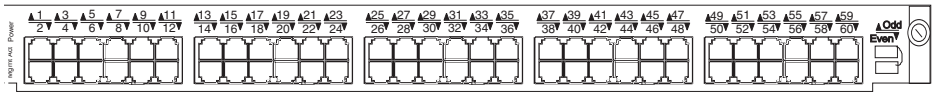
The interface modules are hot swappable, which means you can remove and replace them without powering down the system.

60-port 1 Gigabit Ethernet Interface Module (copper)

Release 02.0.02 for the ProCurve 9408sl introduces the 60-port Gigabit over Copper (GoC) interface module.

Figure 4 shows the 60-port GoC interface module's front panel.

Figure 4 60-port GoC Interface Module Front Panel



The front panel includes the following control features:

- LEDs
- 10/100/1000 Gigabit Ethernet ports with RJ-45 copper connectors

Gigabit Ethernet Ports

The 60-port GoC interface module contains 60 physical ports, through which you can connect your ProCurve routing switch to other network devices at a maximum speed of 1 Gigabit.

LEDs on the 60-port GoC Interface Module

The front panel on the 60-port GoC interface module includes two LEDs that indicate the general status of the module and two LEDs that indicate the status of each port. Table 11 describes the LEDs on the front panel of the 60-port GoC interface module.

Table 11: LEDs for 10/100/1000 Mbps Ports

LED	Position	State	Meaning
Pwr	Top right	On	The module is receiving power.
		Off	The module is not receiving power.
Mgmt Act	Top left	During initialization: steady blinking. After initialization: occasional blinking.	The active management module's processor and the interface module's processor are communicating.
		Off for an extended period.	The interface module is not being managed by the active management module.
■	Upper left corner of upper copper connector	Off	No copper port connection exists on upper copper connector.
		Green	Copper port is connected on upper copper connector.
		Amber	Traffic is being transmitted and received on upper copper connector.

Table 11: LEDs for 10/100/1000 Mbps Ports(Continued)

LED	Position	State	Meaning
■	Upper right corner of upper copper connector	Off	No copper port connection exists on lower copper connector.
		Green	Copper port is connected on lower copper connector.
		Amber	Traffic is being transmitted and received on lower copper connector.

ProCurve 9408sl Trunk Forming Rules

Trunks can be formed in the 9408sl chassis from multiple interface modules of either 2, 4, or 8 ports. Within this limit, not all combinations of ports can form a trunk. Also, all ports of a trunk must have the same attributes, and each network layer may have its specific attribute requirements. This section describes the port architecture and provides procedures to pick valid ports when forming a trunk.

Interface Module Packet Processor to Port Architecture

As stated earlier, not all combinations of ports can be used to form a trunk. This is because of how the ports are associated with Packet Processors (PPCRs) within each module. Each port on an interface Module is associated with a PPCR. Each PPCR can interface to either 1, 10, or 20 ports depending on the interface module. There can be between one and four PPCRs on an interface module. Table 12 describes the number of PPCRs on each of the interface Modules supported by the 9408sl and the interface module ports supported by each PPCR.

Table 12: PPCR to Port layouts

module type	Number of Packet Processors (PPCR)	Module Port Range Belonging to each PPCR PPCR 1	PPCR 2	PPCR 3	PPCR 4
4 x 10G	4	1	2	3	4
40 x 1G	4	1 - 10	11 - 20	21 -30	31 - 40
60 x 1G	3	1 - 20	21 - 40	41 - 60	N/A

Figure 5 shows a 40 x 1G interface Module containing 4 PPCRs. The first PPCR is shown associated with 10 ports and the last 6 ports of the fourth PPCR are shown.

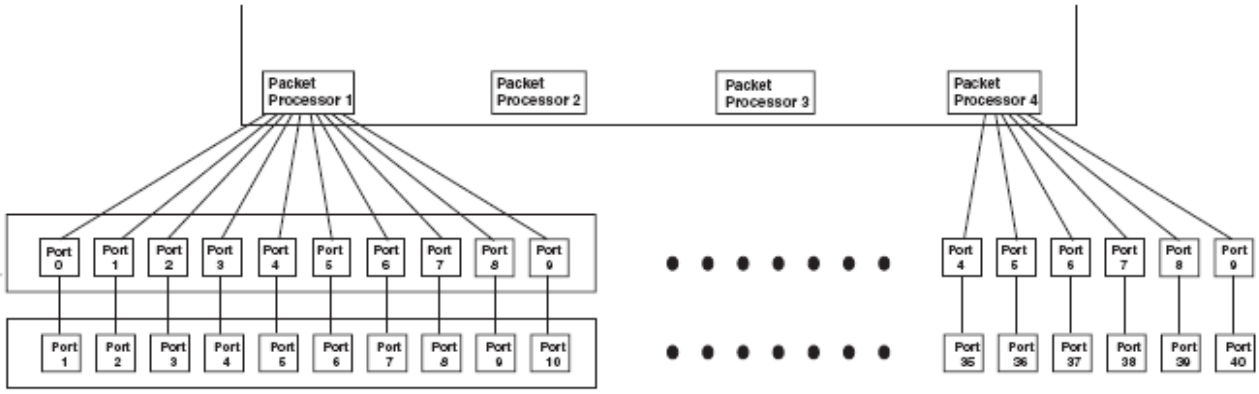


Figure 5 Packet Processors on 1G x 40 Interface Module

The ten ports associated with PPCR 1 have an identity as interface Module ports and as PPCR ports. The interface Module ports (shown in this figure) are identified as ports 1 to 10 in the first PPCR and 35 to 40 in the fourth. The PPCR ports are numbered from 0 to 9 for each PPCR. For example, interface port 1 of the module has a PPCR port number of 0, and interface port 40 of the module, which is the last interface module port, (associated with PPCR 4), has a PPCR port number of 9. The interface Module port numbers are how you identify them for mechanical connection and system configuration. The PPCR port numbers are essential to determining whether a port can be validly used in a trunk.

Either 2, 4, or 8 ports from an individual packet processor can be used in constructing a trunk. If 8 ports are used from a specific packet processor, this will constitute all the ports in an 8 port trunk (the largest possible). If either 2 ports or 4 ports are used from a specific packet processor, they may be combined with ports from another port-set or another PPCR (on the same interface module or from a different interface module) to create a bigger trunk. Since each 10 Gbps port has its own PPCR, ports for 10 Gbps modules can be used in any order as long as you follow all of the other rules for creating a trunk.

Either of the following sections; “Determining Valid Ports Using the Trunk Mask Test” and “Determining valid ports using Valid Port Tables” can be used to determine valid port-sets for use in forming 9408sl trunks.

Determining Valid Ports for Trunking

As described earlier, either 2, 4, or 8 ports from an individual PPCR can be used in constructing a trunk. Depending on the number of ports used, you can obtain all of the trunk ports you need from a single packet processor or use ports from up to 4 PPCRs on one or multiple interface Modules. If 8 ports are used from a specific PPCR, this will constitute all the ports in an 8 port trunk (the largest possible). If either 2 ports or 4 ports are used from a specific PPCR, they may be combined with ports from another PPCR (on the same interface module or from a different interface module) to create a bigger trunk. For example, from the 40 x 1 Gbps interface Module shown in Figure 5, you could use ports 1 and 2 (PPCR ports 0 and 1) from PPCR 1 and ports 39 and 40 (PPCR ports 8 and 9) from PPCR 4 to construct a 4 port trunk.

Trunk ports from a particular PPCR must be added to a trunk by following a very specific set of rules. As shown in Figure 6, five different two-port sets can be used from a PPCR as long as they start with an even PPCR port

number. For example, you can use PPCR ports 6 and 7. Two different four-port sets can be used: PPCR ports 0 - 3 or 4 - 7. If you use 8 ports from a single processor, you must use PPCR ports 0 - 7.

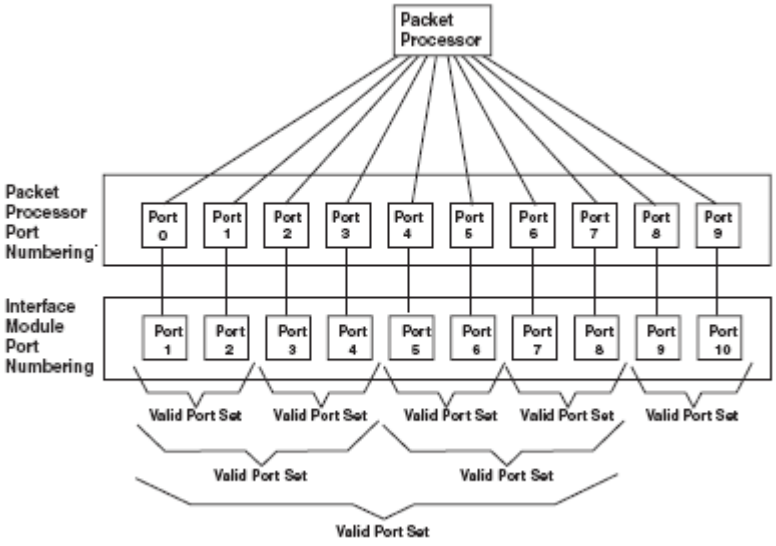


Figure 6 Port Numbering Detail

The following sections: “Determining Valid Ports Using the Trunk Mask Test” and “Determining valid ports using Valid Port Tables” provide methods for determining valid port sets that can be used to form a trunk.

NOTE: Because each 10 Gbps port is associated with a single packet processor, you can use any combination of 10 Gbps ports within an 9408sl chassis to form a trunk of 2, 4, or 8 ports.

Determining Valid Ports Using the Trunk Mask Test

As described in “Determining Valid Ports for Trunking”, you must use particular PPCR ports when forming a trunk. You can do this by applying a Trunk Mask Test as described in this section or by using the valid port tables as described in “Determining valid ports using Valid Port Tables”.

Use the following formula to determine if a set of ports can be used to form a trunk. The keywords “AND” and “NOT” are Boolean logic operators.

```
<PPCR_Port_Number> AND NOT<trunk_mask> == First PPCR port in range
```

PPCR_Port_Number - This is the PPCR port number as described earlier in this section. You must account for the fact that while the interface Module ports start at 1 and end with the last port on the Module, PPCR ports are specific to a PPCR and always start on an PPCR at zero, run up to the last port of that PPCR and start at zero for the next PPCR.

For example: A 40 x 1Gbps interface Module has four PPCRs. Each PPCR supports 10 interface Module ports. Port 38 of the interface Module is associated with the fourth PPCR. Consequently, port 31 of the interface Module is associated with PPCR port 0 and interface Port 38 is PPCR port 7 of the fourth packet processor.

Trunk_Mask - This variable depends on the number of ports you intend to use from a single PPCR. Table 12 describes the number of PPCRs on each of the interface Modules supported by the 9408sl and the interface module ports supported by each PPCR.

0x1 for two ports

0x3 for four ports

0x7 for eight ports

EXAMPLE:

Interface Module 4 in an 9408sl chassis is a 40 x 1 Gbps module. Trunk ports 4/11 to 4/14 and ports 4/25 to 4/28 are able to pass the Trunk Mask Test as shown in the following.

Interface ports 4/11, 4/12, 4/13, and 4/14 are associated with PPCR ports 0, 1, 2, and 3. The Trunk_Mask for four ports is 0x3. The mask is applied to each of the ports as shown in the following:

<PPCR_Port_Number> AND NOT<trunk_mask> == First PPCR port in range

0 AND NOT (0x3) == 0 (This is the 1st PPCR port number in the range)

1 AND NOT (0x3) == 0

2 AND NOT (0x3) == 0

3 AND NOT (0x3) == 0

Since each PPCR port belongs to a port set that begins with the same PPCR port (0). Therefore, this is a valid set of four ports to use in a trunk.

Interface ports 4/25, 4/26, 4/27, and 4/28 are associated with PPCR ports 4, 5, 6, and 7. The Trunk_Mask for four ports is 0x3. The mask is applied to each of the ports as shown in the following:

<PPCR_Port_Number> AND NOT<trunk_mask> == First PPCR port in range

4 AND NOT (0x3) == 4 (This is the 1st PPCR port number in the range)

5 AND NOT (0x3) == 4

6 AND NOT (0x3) == 4

7 AND NOT (0x3) == 4

Since each PPCR port belongs to a port set that begins with the same PPCR port (4) this is a valid set of four ports to use in a trunk.

Using these two sets of four ports, you can create a valid 8 port trunk.

Determining valid ports using Valid Port Tables

As described in “Determining Valid Ports for Trunking”, you must use particular PPCR ports when forming a trunk. Valid trunking ports can be determined by applying the Trunk Mask Test as described in “Determining Valid Ports Using the Trunk Mask Test” or by using the following tables. The valid port-sets for each of the current interface Modules are in one the following tables. Port-sets can be used alone to create a trunk of the desired size, or they can be mixed between interface Modules in a chassis and PPCRs on an individual module, as long as the trunk meets all of the other rules as described in “Other Rules for Forming a 9408sl Trunk”.

Table 13: 40 x 1 Gbps Interface Module Port-sets for trunking

Module type	Packet Processor #	Valid Port Sets 2-port	Valid Port Sets 4-port	Valid Port Sets 8-port
40 x 1G	PPRC 1	1 - 2 3 - 4 5 - 6 7 - 8 9 - 10	1 - 4 5 - 8	1-8
	PPRC 2	11 - 12 13 - 14 15 - 16 17 - 18 19 - 20	11 - 14 15 - 18	11-18

Module type	Packet Processor #	Valid Port Sets 2-port	Valid Port Sets 4-port	Valid Port Sets 8-port
	PPRC 3	21 - 22 23 - 24 25 - 26 27 - 28 29 - 30	21 - 24 25 - 28	21-28
	PPRC 4	31 - 32 33 - 34 35 - 36 37 - 38 39 - 40	31 - 34 35 - 38	31-38

Table 14: 60 X 1 Gbps Interface Module Port-sets for trunking

module type	Packet Processor #	Valid Port Sets 2-port	Valid Port Sets 4-port	Valid Port Sets 8-port
60 x 1G	PPRC 1	1 - 2 3 - 4 5 - 6 7 - 8 9 - 10 11 - 12 13 - 14 15 - 16 17 - 18 19 - 20	1 - 4 5 - 8 9 - 12 13 - 16 17 - 20	1 - 8 9 - 16
	PPRC 2	21 - 22 23 - 24 25 - 26 27 - 28 29 - 30 31 - 32 33 - 34 35 - 36 37 - 38 39 - 40	21 - 24 25 - 28 29 - 32 33 - 36 37 - 40	21 - 28 29 - 36
	PPRC 3	41 - 42 43 - 44 45 - 26 47 - 48 49 - 50 51 - 52 53 - 54 55 -56 57 - 58 59 - 60	41 - 44 45 - 48 49 - 52 53 - 56 57 - 60	41 - 48 49 - 56

Other Rules for Forming a 9408sl Trunk

Once you have determined the ports you intend to use for your trunk, you must make sure that they meet the requirements defined in the following list.

1. Physical port requirements

   All trunk ports must have the same physical port attributes; otherwise, the trunk is rejected.

2. Rate Limiting and PBR requirements

   Primary port policy will apply to all secondary ports. No trunk is rejected.

3. Mirroring/Monitoring requirements

   The trunk is rejected if any trunk port has mirroring or monitoring configured.

4. VLAN and inner-VLAN translation

   The trunk is rejected if any trunk port has vlan or inner-vlan translation configured.

5. Layer 2 requirements

   The trunk is rejected if the trunk ports:

   - do not have the same untagged VLAN component.
   - do not share the same superspan customer id (or cid).
   - do not share the same vlan membership
   - do not share the same uplink vlan membership
   - do not share the same protocol-vlan configuration
   - are configured as mrp primary and secondary interfaces

6. Layer 3 requirements

   The trunk is rejected if any of the secondary trunk port has any layer 3 configurations, such as Ipv4 or Ipv6 address, ospf, rip, ripng, etc.

7. Layer 4 (ACL) requirements

   All trunk ports must have the same ACL configurations; otherwise, the trunk is rejected.

Enhancements and Configuration Notes in 02.1.00

This section provides details about the enhancements and configuration differences in release 02.1.00 for the ProCurve 9408sl.

Layer 2 Access Control Lists

Layer 2 Access Control Lists (ACLs) filter incoming traffic based on Layer 2 MAC header fields in the Ethernet/IEEE 802.3 frame. Specifically, in release 02.1.00 you can configure Layer 2 ACLs to use the **etype** argument to filter on the following etypes (Ethertype):

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

Configuration Rules and Notes

- You cannot bind Layer 2 ACLs and IP ACLs to the same port. However, you can configure one port on the device to use Layer 2 ACLs and another port on the same device to use IP ACLs.
- You cannot bind a Layer 2 ACL to a virtual interface.
- By default, when Layer 2 ACLs are enabled on a port, the device filters traffic in hardware.

Configuring Layer 2 ACLs

Configuring a Layer 2 ACL is similar to configuring IPV4 standard and extended ACLs. Layer 2 ACL table IDs range from 400 to 499, for a maximum of 100 configurable Layer 2 ACL tables. Within each Layer 2 ACL table,

you can configure from 64 (default) to 256 clauses. Each clause or entry can define a set of Layer 2 parameters for filtering. Once you completely define a Layer 2 ACL table, you must bind it to the interface for filtering to take effect.

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

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

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

Creating a Layer 2 ACL Table

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

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

```
9408sl(config)# access-list 400 deny any etype arp
9408sl(config)# access-list 400 permit any any 100
```

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

For more examples of valid Layer 2 ACL clauses, see “Example Layer 2 ACL Clause” on page 49.

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

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

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

The **<src-mac> <mask> | any** parameter specifies the source MAC address. You can enter a specific address and a comparison mask or the keyword **any** to filter on all MAC addresses. Specify the mask using F’s and zeros. For example, to match on the first two bytes of the address aabb.ccdd.eeff, use the mask ffff.0000.0000. In this case, the clause matches all source MAC addresses that contain “aabb” as the first two bytes and any values in the remaining bytes of the MAC address. If you specify any, you don’t need to specify a mask and the clause matches on all MAC addresses.

The **<dest-mac> <mask> | any** parameters specify the destination MAC address. The syntax rules are the same as those for the **<src-mac> <mask> | any** parameter.

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

The **etype** <etype-str> argument specifies the value for the Ethernet type field of the incoming packet in-order for a match to occur. The <etype-str> can be one of the following keywords:

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

The **log-enable** parameter is optional and applies to clauses specified with a ‘deny’ action. If specified with a ‘permit’ action, the log-enable keyword is ignored and the user is warned that he cannot log permit traffic.

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

Example Layer 2 ACL Clause

The following shows an example of a valid Layer 2 ACL clause:

```
9408sl(config)# access-list 400 permit any any 100 etype ipv4
```

Binding a Layer 2 ACL Table to an Interface

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

```
9408sl(config)# int e 4/12
9408sl(config-int-e100-4/12)# mac access-group 400 in
```

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

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

Viewing Layer 2 ACLs

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

```
9408sl(config)# show access-list 400

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

Syntax: show access-list <num>

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

Example of Layer 2 ACL Deny by MAC Address

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

```
9408sl(config)# access-list 401 deny 0012.3456.7890 ffff.ffff.ffff 0011.2233.4455
ffff.ffff.ffff
9408sl(config)# access-list 401 permit any any
```

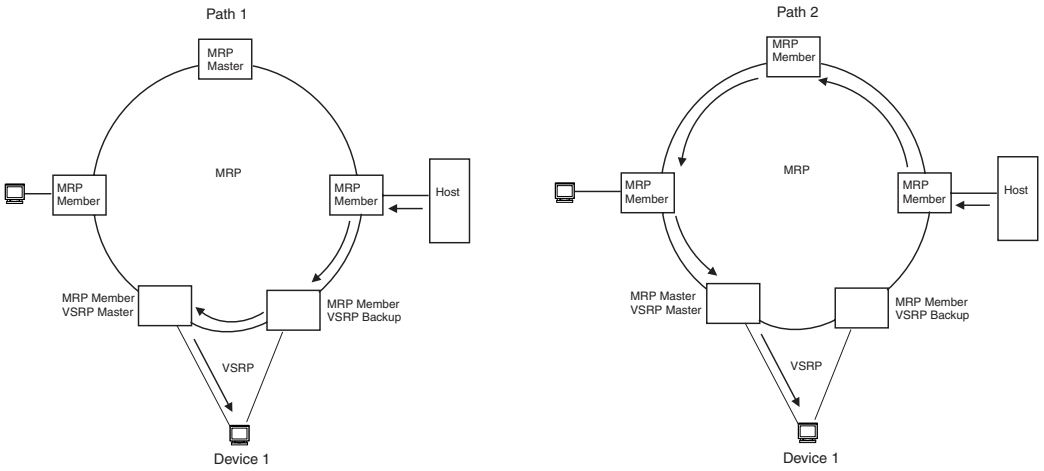
Using the mask, you can make the access list apply to a range of addresses. For instance if you changed the mask in the previous example for 0012.3456.7890 to ffff.ffff.fff0, all hosts with addresses from 0012.3456.7890 to 0012.3456.789f would be blocked. This configuration for this example is shown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

```
9408sl(config)# access-list 401 deny 0012.3456.7890 ffff.ffff.fff0 0011.2233.4455
ffff.ffff.ffff
9408sl(config)# access-list 401 permit any any
```

VSRP and MRP Signaling

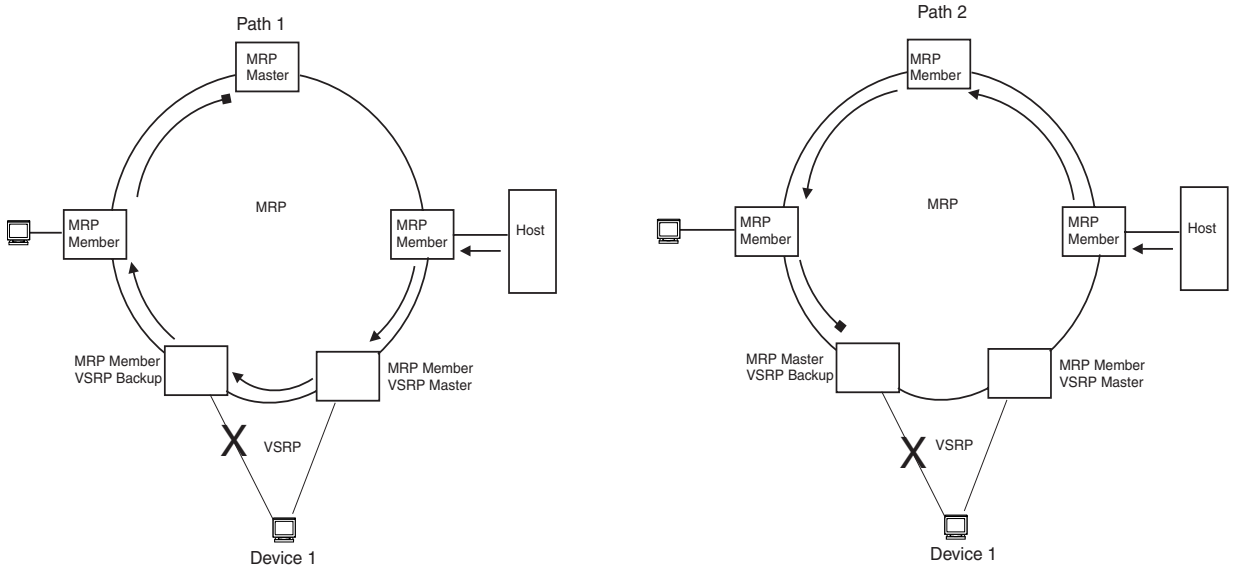
A device may connect to an MRP ring via VSRP to provide a redundant path between the device and the MRP ring. VSRP and MRP signaling ensures rapid failover by flushing MAC addresses appropriately. The host on the MRP ring learns the MAC addresses of all devices on the MRP ring and VSRP link. From these MAC addresses, the host creates a MAC database (table), which is used to establish a data path from the host to a VSRP-linked device. Figure 7 below shows two possible data paths from the host to Device 1.

Figure 7 Two data paths from host on an MRP ring to a VSRP-linked device



If a VSRP failover from master to backup occurs, VSRP needs to inform MRP of the topology change; otherwise, data from the host continues along the obsolete learned path and never reach the VSRP-linked device, as shown in Figure 8.

Figure 8 VSRP on MRP rings that failed over



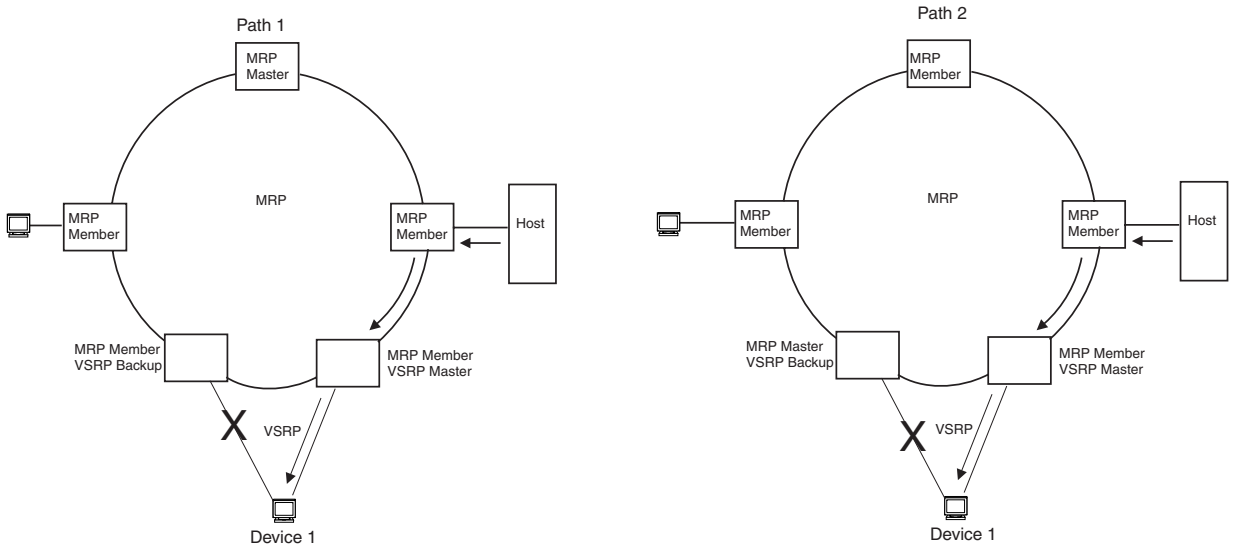
To ensure that MRP is informed of the topology change and to achieve convergence rapidly, software release 02.1.02 provides a new signaling process for the interaction between VSRP and MRP. When a VSRP node fails, a new VSRP master is selected. The new VSRP master finds all MRP instances impacted by the failover. Then each MRP instance does the following:

- The MRP node sends out an MRP PDU with the mac-flush flag set three times on the MRP ring.
- The MRP node that receives this MRP PDU empties all the MAC entries from its interfaces that participate on the MRP ring.
- The MRP node then forwards the MRP PDU with the mac-flush flag set to the next MRP node that is in

forwarding state.

The process continues until the Master MRP node's secondary (blocking) interface blocks the packet. Once the MAC address entries have been flushed, the MAC table can be rebuilt for the new path from the host to the VSRP-linked device (Figure 9).

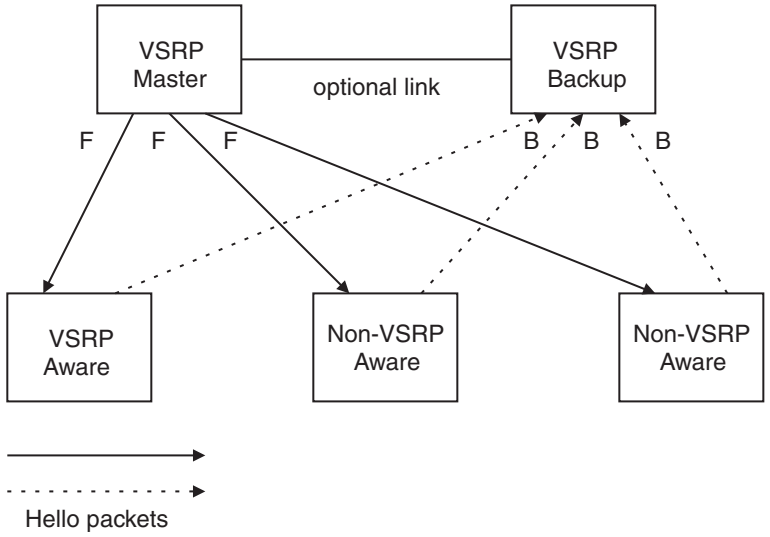
Figure 9 New path established



VSRP Fast Start

VSRP provides redundancy and sub-second failover in Layer 2 and Layer 3 mesh topologies. Two VSRP configured ProCurve devices provide the redundancy. One is the Master for the Virtual Router ID (VRID). The Master sets the state of all its VLAN ports to Forwarding. The other device is a Backup; it sets all its ports in its VRID VLAN to Blocking.

Figure 10 VSRP mesh – redundant paths for Layer 2 and Layer 3 traffic



If a failover occurs, the Backup becomes the new Master and changes all its VRID ports to the Forwarding state. The previous Master becomes the Backup. VSRP-aware devices quickly switch to the new Master to reconverge their connectivity to the network; however, reconvergence for non-VSRP aware devices occurs slowly.

VSRP-aware devices are ProCurve devices that do not have VSRP configured, but are connected to a ProCurve device that is the VSRP Master. Previously only ProCurve devices could be VSRP-aware. Software release 02.1.00 introduces the VSRP fast start feature, a way for non-ProCurve or non-VSRP aware devices to quickly switchover to the new Master when a VSRP failover occurs.

The VSRP fast start feature causes the port on a VSRP Master to restart when a VSRP failover occurs. When the port shuts down at the start of the restart, ports on the non-VSRP aware devices that are connected to the VSRP Master flush the MAC address they have learned for the VSRP master. After a specified time, the port on the previous VSRP Master (which now becomes the Backup) returns back online. Ports on the non-VSRP aware devices switch over to the new Master and learn its MAC address.

Special Considerations when Configuring VSRP Fast Start

- VSRP is sensitive to port status. When a port goes down, the VSRP instance lowers its priority based on the port up fraction. (see "VSRP Priority Calculation" in the *Installation and Basic Configuration Guide for the ProCurve 9408sl Routing Switch* for more information on how priority is changed by port status). Since the VSRP fast start feature toggles port status by bringing ports down and up it can affect VSRP instances because their priorities get reduced when a port goes down. To avoid this, the VSRP fast start implementation keeps track of ports that it brings down and suppresses port down events for these ports (as concerns VSRP).
- Once a VSRP restart port is brought up by a VSRP instance, other VSRP instances (in Master state) that have this port as a member do not go to forwarding immediately. This is a safety measure that is required to prevent transitory loops. This could happen if a peer VSRP node gets completely cut off from this node and assumed Master state. In this case, where there are 2 VSRP instances that are in Master state and forwarding, the port comes up and starts forwarding immediately. This would cause a forwarding loop. To avoid this, the VSRP instance delays forwarding.

Recommendations for Configuring VSRP Fast Start

The following recommendations apply to configurations where multiple VSRP instances are running between peer devices sharing the same set of ports.

- Multiple VSRP instances configured on the same ports can cause VSRP instances to be completely cut off from peer VSRP instances. This can cause VSRP instances to toggle back and forth between master and backup mode. For this reason, we recommend that you configure VSRP fast start on a per port basis rather than for the entire VLAN.
- We recommend that VSRP peers have a directly connected port without VSRP fast start enabled on it. This allows protocol control packets to be received and sent even if other ports between the master and standby are down.
- The VSRP restart time should be configured based on the type of connecting device since some devices can take a long time to bring a port up or down (as long as several seconds). In order to ensure that the port restart is registered by neighboring device, the restart time may need to be changed to a value higher than the default value of 1 second.

Configuring VSRP Fast Start

The VSRP fast start feature can be enabled on a VSRP-configured ProCurve device, either on the VLAN to which the VRID of the VSRP-configured device belongs (globally) or on a port that belongs to the VRID.

To globally configure a VSRP-configured device to shut down its ports when a failover occurs, then restart after five seconds, enter the following command:

```
9408sl(configure)# vlan 100
9408sl(configure-vlan-100)# vsrp vrid 1
9408sl(configure-vlan-100-vrid-1)# restart-ports 5
```

Syntax: restart-ports <seconds>

This command shuts down all the ports that belong to the VLAN when a failover occurs. All the ports will have the specified VRID.

To configure a single port on a VSRP-configured device to shut down when a failover occurs, then restart after a period of time, enter the following command:

```
9408sl(configure)# interface ethernet 1/1
9408sl(configure-if-1/1)# vsrp restart-port 5
```

Syntax: vsrp restart-port <seconds>

In both commands, the <seconds> parameter instructs the VSRP Master to shut down its port for the specified number of seconds before it starts back up. Enter a value between 1 – 120 seconds. The default is 1 second.

Displaying Ports that Have VSRP Fast Start Feature Enabled

The show vsrp vrid command shows the ports on which the VSRP fast start feature is enabled.

```
9408sl(config-vlan-100-vrid-100)#show vsrp vrid 100

VLAN 100
  auth-type no authentication
  VRID 100
  ========
   State      Administrative-status Advertise-backup Preempt-mode save-current
   master     enabled               disabled         true         false
   Parameter        Configured Current    Unit/Formula
   priority         100        50         (100-0)*(2.0/4.0)
   hello-interval   1          1          sec/1
   dead-interval    3          3          sec/1
   hold-interval    3          3          sec/1
   initial-ttl      2          2          hops

   next hello sent in 00:00:00.3
   Member ports:      ethe 2/5 to 2/8
   Operational ports: ethe 2/5 ethe 2/8
   Forwarding ports:  ethe 2/5 ethe 2/8
   Restart ports:     2/5(1) 2/6(1) 2/7(1) 2/8(1)
```

The “Restart ports:” line lists the ports that have the VSRP fast start enabled, and the downtime for each port.

Secure Shell (SSH) Version 2 Support

Secure Shell (SSH) is a mechanism for allowing secure remote access to management functions on a ProCurve device. SSH provides a function similar to Telnet, but with a secure, encrypted connection to the device.

Starting with release 02.1.00, the 9408sl supports SSH version 2 (SSHv2) and SSHv1 is not supported.

NOTE: This release supports SSH v2 only. Other versions of SSH are not supported. This will ordinarily not present a problem because most SSH clients in the market support SSHv1 and SSHv2 and they automatically determine which version to use depending on the server, which in this case is the 9408sl.

SSHv2 is a substantial revision of Secure Shell, comprising the following hybrid protocols and definitions:

- SSH Transport Layer Protocol
- SSH Authentication Protocol
- SSH Connection Protocol
- GSSAPI Authentication and Key Exchange for the Secure Shell Protocol
- Generic Message Exchange Authentication For SSH

- SECSH Public Key File Format
- SSH Fingerprint Format
- SSH Protocol Assigned Numbers
- SSH Transport Layer Encryption Modes
- Session Channel Break Extension
- SCP protocol

In this release, the CLI commands for setting up and configuring SSHv2 on a ProCurve device are similar to SSHv1 with the following exceptions:

The following CLI commands are removed, as they are not applicable to an SSHv2 implementation:

```
ip ssh key-size
```

```
ip ssh pub-key-file
```

```
crypto random-number-seed generate
```

The following CLI command for generating a crypto key has been changed:

Syntax: crypto key generate/zeroize rsa

in SSHv1 is changed to:

Syntax: crypto key generate/zeroize

in SSHv2.

The **rsa** option has been removed. There is no backward compatibility problem, as the command is a runtime command and the key is stored in the EEPROM.

While the SSH listener exists at all times, sessions can't be started from clients until a key is generated. Once a key is generated, clients can start sessions. The keys are also not displayed in the configuration file by default. If you would like them to be displayed, use the **ssh show-host-keys** command in Privileged EXEC mode as shown in the following:

```
9408sl#ssh show-host-keys
```

Syntax: [no] ssh show-host-keys

This command causes the keys to be displayed when the show running-config command is used as shown. The default is for the keys to not be displayed.

For further information on configuring SSH on ProCurve devices, see the *Security Guide for ProCurve 9300/9400 Series Routing Switches*.

ProCurve's SSHv2 implementation is compatible with all versions of the SSHv2 protocol (2.1, 2.2, and so on). At the beginning of an SSH session, the ProCurve device negotiates the version of SSHv2 to be used. The highest version of SSHv2 supported by both the ProCurve device and the client is the version that is used for the session. Once the SSHv2 version is negotiated, the encryption algorithm with the highest security ranking is selected to be used for the session.

Tested SSHv2 Clients

The following SSH clients have been tested with SSHv2:

- SSH Secure Shell 3.2.3
- Van Dyke SecureCRT 4.0
- F-Secure SSH Client 5.3
- Tera Term Pro 3.1.3
- PuTTY 0.54
- OpenSSH 3.5_p1

Supported Encryption Algorithms for SSHv2

The following encryption algorithms are supported with the ProCurve implementation of SSHv2:

- 3DES
- None selected

Supported MAC (Message Authentication Code) Algorithms

The following MAC algorithms are supported with the ProCurve implementation of SSHv2:

- SHA
- None selected

Enabling Support for More ACL Entries

Software release 02.1.00 provides support for up to 4K (4096) ACL statements on a ProCurve 9408sl.

Enabling ACL Duplication Check

For the 9408sl, the software does not check for duplicate ACL entries. This is so the device can support the increased maximum number of ACLs. In a system with several thousand ACL entries, checking for duplicate ACL entries may consume a significant amount of time.

If desired, you can enable software checking for duplicate ACL entries. To do so, enter the following command at the Global CONFIG level of the CLI:

```
9408sl(config)# acl-duplication-check
```

Syntax: [no]acl-duplication-check

Maximum Frame Size Support

In earlier releases, the ProCurve 9408sl had a default maximum frame size of 1518 bytes. With software release 02.1.00, the maximum frame size supported on a port is modified to dynamically change based upon the port's tagging characteristics as described:

Untagged Ports – The maximum frame size supported on an untagged port is 1518 bytes. This includes 1500 bytes for payload, 14 bytes for the MAC header, and 4 bytes for the CRC. This limit is defined for untagged ports in the IEEE 802.1 specification.

Tagged Ports – The maximum size supported on tagged ports is 1522 bytes. The additional 4 bytes over the untagged port maximum are allowed to support the additional bytes needed to include a VLAN tag.

Super-aggregated VLAN Support – A maximum of 1526 bytes are supported on ports where super-aggregated VLANs are configured. This allows for an additional 8 bytes over the untagged port maximum to allow for support of two VLAN tags.

Configuring the Management Port for an IPv6 Automatic Address Configuration

With software release 02.1.00, the ProCurve 9408sl can have its management port configured to automatically obtain an IPv6 address. This process is the same for any other port and is described in detail in the "Configuring a Global or Site-Local IPv6 Address with an Automatically Computed EUI-64 Interface ID" and "Configuring a Link-Local IPv6 Address" sections of the *IPv6 Configuration Guide for the ProCurve 9408sl Routing Switch*.

Enhancements to Rate Limiting on ProCurve Devices

ProCurve devices provide line-rate rate limiting in hardware on inbound and outbound ports.

Software release 01.1.00 for ProCurve devices introduced the following rate limiting types for inbound ports:

- Port-based for inbound ports – Limits the rate of inbound traffic on an individual physical port to a specified rate. Only one port-based inbound rate limiting policy can be applied to a port. (Refer to "Configuring Port-Based Rate Limiting For Inbound and Outbound Ports" on page 58.)

- Port-and-priority-based – Limits the rate on an individual hardware forwarding queue on an individual physical port. Only one port-and-priority-based rate limiting policy can be specified per priority queue for a port. This means that a maximum of four port-and-priority-based policies can be configured on a port. (Refer to “Configuring a Port-and-Priority-Based Rate Limiting Policy” on page 58.)
- Port-and-VLAN-based – Limits the rate of packets tagged with a specific VLAN on an individual physical port. Only one rate can be specified for each VLAN. Up to 10 VLAN-based policies can be configured for a port. (Refer to “Configuring a Port-and-VLAN-Based Rate Limiting Policy” on page 59.)
- Port-and-ACL-based – Limits the rate of IP traffic on an individual physical port that matches the permit conditions in IP Access Control Lists (ACLs). You can use standard or extended IP ACLs. Standard IP ACLs match traffic based on source IP address information. Extended ACLs match traffic based on source and destination IP address and IP protocol information. Extended ACLs for TCP and UDP also match on source and destination TCP or UDP addresses. and protocol information. (Refer to “Configuring a Port-and-ACL-Based Rate Limiting Policy” on page 61.)

Software Release 02.1.00 adds the following enhancements to the rate limiting feature:

- Port-based for outbound ports – Limits the rate of outbound traffic on an individual physical port to a specified rate. Only one port-based outbound rate limiting policy can be applied to a port. (Refer to “Configuring Port-Based Rate Limiting For Inbound and Outbound Ports” on page 58.)
- Port-and-Layer 2 ACL-based – Limits the rate of traffic on an individual physical port that matches the permit conditions a Layer 2 ACL. (Refer to “Configuring Port-and-Layer 2 ACL-based rate limiting” on page 62.)
- VLAN-and-priority based – Limits traffic on a physical port that is a member of a specified VLAN and has been assigned to specified forwarding queues. (Refer to “Configuring VLAN-and-priority based rate limiting” on page 59.)
- VLAN group based – Limits the traffic for a group of VLANs. Members of a VLAN group share the specified bandwidth defined in the rate limiting policy that has been applied to that group. (Refer to “Configuring VLAN Group Based Rate Limiting” on page 60.)
- Port-and-IPV6 ACL-based – Limits the rate of traffic on an individual physical port that matches the permit conditions of IPV6 ACL. These policies can be applied to inbound traffic only. (Refer to “Configuring Port-and-IPv6 ACL-based rate limiting” on page 62.)
- Filtering traffic denied by a rate limiting ACL – Drops traffic that matched an ACL deny filter in a port-and-ACL based rate limiting policy. (Refer to “Filtering Traffic Denied by a Rate Limiting ACL” on page 63.)
- New command to display rate limiting policies – Displays rate limiting policies that have been configured for a device, an interface, or a VLAN group. (Refer to “Display Rate Limiting Policies” on page 63 and “Displaying Rate Limit VLAN Groups” on page 64.)

This section presents all the rate limiting policies available on ProCurve devices. Except for port-based rate limiting policies, all rate limiting policy types can be applied only to inbound ports.

Rate Limiting Parameters and Algorithm

All rate limiting policies specify two parameters: average rate and maximum burst. These parameters are used to configure credits and credit totals.

Average Rate

The *Average Rate* is the maximum number of bits a port is allowed to receive during a one-second interval. The rate of the traffic that matches the rate limiting policy will not exceed the average rate.

The Average Rate represents a percentage of an interface's line rate (bandwidth), expressed in bits per second (bps). It cannot be smaller than 515,624 bits per second (bps) and it cannot be larger than the port’s line rate.

Average Rate must be entered in multiples of 515,624 bps. If you enter a number that is not a multiple of 515,624, the software adjusts the rate down to the lowest multiple of the number so that the calculation of credits does not result in a remainder of a partial Credit. For example, if you enter 600,000 bps, the value will be adjusted to 515,624 bps. The adjusted rate is sometimes called the *adjusted average rate*.

Maximum Burst

When the traffic on the port is less than the specified average rate, the rate limiting policy can accumulate credits up to a maximum of the ***maximum burst*** value. The accumulated credit allows traffic to pass through the port at a rate higher than the average rate for a short period of time. The time period is determined by the amount of credit accumulated and the rate of traffic passing through the port.

The maximum burst rate cannot be smaller than 65536 bits.

Credits and Credit Total

Each rate limiting policy is assigned a class. A *class* uses the average rate and maximum allowed burst in the rate limit policy to calculate credits and credit totals.

Credit size is measured in bytes. A credit is a forwarding allowance for a rate-limited port, and is the smallest number of bytes that can be allowed during a rate limiting interval. Minimum credit size can be 1 byte.

During a rate limiting interval, a port can send or receive only as many bytes as the port has Credits for. For example, if an inbound rate limiting policy results in a port receiving two credits per rate limiting interval, the port can send or receive a maximum of 2 bytes of data during that interval.

The credit size is calculated using the following algorithm:

Credit = (Average rate in bits per second)/(8*64453)

One second is divided into 64,453 intervals. In each interval, the number of bytes equal to the credit size is added to the running total of the class. The running total of a class represents the number of bytes that can be allowed to pass through without being subject to rate limiting.

The second calculation is the maximum *credit total*, which is also measured in bytes. The maximum credit total is calculated using the following algorithm.

Maximum credit total = (Maximum burst in bits)/8

The running total can never exceed the maximum credit total. When packets arrive at the port, a class is assigned to the packet based on the rate limiting policies. If the running total of the class is less than the size of the packet, then the packet is dropped. Otherwise, the size of the packet is subtracted from the running total and the packet is forwarded. If there is no traffic that matches the rate limiting criteria, then the running total can grow up to the maximum credit total.

Configuration Considerations

- Except for port-based rate limiting policies, all rate limiting policy types can be applied only to inbound ports of 9408sl devices.
- Only one type of inbound rate limiting policy can be applied on a physical port. For example, you cannot apply inbound port-and-ACL-based and inbound port-based rate limiting policies on the same port.
- Outbound port-based rate limiting policy can be combined with any type of inbound rate limiting policy.
- When a port-and-VLAN-based rate limiting policy is applied to a port, all the ports controlled by the same packet processor are rate limited for that VLAN. You cannot apply a port-and-VLAN-based rate limiting policy on another port of the same packet processor for the same VLAN ID.
- Any VLAN-based rate limiting can limit only tagged packets that match the VLAN ID specified in the policy. Untagged packets are not subject to rate limiting.
- The average rate in a rate limiting policy cannot be less than 515,624 bits per second, must be in multiples of 515,624, and cannot be more than the port's line rate.
- The maximum burst in a rate limit policy can be less than the average rate, but cannot be less than 65536 bits and cannot be more than the port's line rate.
- Control packets are not subject to rate limiting.
- You cannot apply Layer 4 ACL-based rate limiting policy on a physical port that is a member of a virtual routing interface.
- You cannot create a trunk if any of the physical ports that are members of the trunk has a rate limiting policy.

- You cannot apply a Layer 2 ACL-based rate limit policy and a Layer 4 ACL-based rate limit policy on a port at the same time.
- A Layer 4 ACL-based rate limiting policy applies only to Layer 3 traffic.
- The total number of source MAC-and-VLAN based, any ACL-based, and any VLAN-based rate limiting policies on ports controlled by the same packet processor cannot exceed:
  - 126 on a 4 x 10G interface module
  - 117 on a 40 x 1G interface module
  - 107 on a 60 x 1G interface module
- For any type of priority based rate limiting policy on a port: If the rates of the policies are the same, then the priorities are combined into one group. For example:

  ```
  9408sl(config-if-1/1)#rate-limit in priority q1 500000000 750000000
  9408sl(config-if-1/1)#rate-limit in priority q2 500000000 750000000
  ```

  These two policies will be combined and displayed as one policy:

  ```
  9408sl(config-if-1/1)#rate-limit in priority q1 q2 500000000 750000000
  ```

  All the traffic for hardware forwarding queues q1 and q2 will be rate limited individually to an average rate of 500Mbps with a maximum burst size of 750Mbits, even if the queues are combined into one policy.
- Certain features such as FDP, CDP, UDLD and LACP that make the port run in dual mode can cause traffic to be rate limited to less than the expected average rate. When the port is in dual mode, all incoming or outgoing packets are treated as tagged. An extra 4 bytes is added to the length of the packet to account for the tag, thus causing the average rate to be less than the expected average rate. Ports in dual mode are assumed to be tagged ports for rate limiting purpose.

Configuring Port-Based Rate Limiting For Inbound and Outbound Ports

ProCurve 9408sl software release 01.1.00 introduced rate limiting features for inbound ports. Software release 02.1.00 adds port-based rate limiting to outbound ports.

Port-based rate limiting limits the rate on an individual physical port to a specified rate.

To configure port-based rate limiting policy for outbound ports, enter commands such as the following at the interface level:

```
9408sl(config)# interface ethernet 1/1
9408sl(config-if-1/1)# rate-limit out 500000000 750000000
Average rate is adjusted to 499639656 bits per second
```

The commands configure a rate limiting policy for outbound traffic on port 1/1. The policy limits the average rate of all outbound traffic to 500 Mbps with a maximum burst size of 750 Mbps.

The complete syntax for configuring a port-based rate limiting policy is:

Syntax: [no] rate-limit in | out <average-rate> <maximum-burst>

The **in** parameter applies the policy to traffic on inbound ports, while **out** applies to outbound ports.

Only one inbound and one outbound port-based rate limiting policy can be applied to a port.

The <average-rate> parameter specifies the maximum rate allowed on a port during a one-second interval. The software automatically adjusts the number you enter to the lower multiple of 515,624 bps. Refer to the section “Average Rate” on page 56 for more details.

The <maximum-burst> parameter specifies the extra bits above the average-rate that traffic can have. Refer to the section “Maximum Burst” on page 57 for more details.

Configuring a Port-and-Priority-Based Rate Limiting Policy

To configure port-and-priority based rate limiting policy:

```
9408sl(config)# interface ethernet 1/1
9408sl(config-if-1/1)# rate-limit in priority q0 q2 500000000 750000000
Average rate is adjusted to 499639656 bits per second
```

These commands configure an rate limiting policy for an inbound port 1/1 that limits the average rate of all inbound traffic for hardware forwarding queues q0 and q2. Traffic on each hardware forwarding queue is limited to an average rate of 500 Mbps with a maximum burst size of 750 Mbits.

Syntax: [no] rate-limit in priority q0 | q1 | q2 | q3 <average-rate> <maximum-burst>

The **in** parameter applies the policy to traffic on inbound ports.

The **priority q0 | q1 | q2 | q3** parameter specifies the hardware forwarding queue to which the policy applies. The device prioritizes the queues from **q0** (normal priority) to **q3** (highest priority). Only one rate can be specified per priority queue for a port.

The <average-rate> parameter specifies the maximum rate allowed on a port during a one-second interval. The software automatically adjusts the number you enter to the lower multiple of 515,624 bps. Refer to the section “Average Rate” on page 56 for more details.

The <maximum-burst> parameter specifies the extra bits above the average-rate that traffic can have. Refer to the section “Maximum Burst” on page 57 for more details.

Configuring a Port-and-VLAN-Based Rate Limiting Policy

To configure a port-and-VLAN based rate limiting policy, enter commands such as the following:

```
9408sl(config)# interface ethernet 1/1
9408sl(config-if-1/1)# rate-limit in vlan 10 500000000 750000000
Average rate is adjusted to 499639656 bits per second
9408sl(config-if-1/1)# rate-limit in vlan 20 100000000 200000000
Average rate is adjusted to 99515432 bits per second
```

These commands configure two rate limiting policies that limit the average rate of all inbound traffic on port 1/1 with VLAN tag 10 and 20. The first policy limits packets with VLAN tag 10 to an average rate of 500 Mbps with a maximum burst size of 750 Mbits. The second policy limits packets with VLAN tag 20 to an average rate of 100 Mbps with a maximum burst size of 200 Mbits. Tagged packets belonging to VLANs other than 10 and 20 and untagged packets are not subject to rate limiting.

Syntax: [no] rate-limit in vlan <vlan-number> <average-rate> <maximum-burst>

The **in** parameter applies the policy to traffic on inbound ports.

The vlan <vlan-number> parameter specifies the VLAN ID to which the policy applies. Refer to “Configuration Considerations” on page 57 to determine the number of rate limiting policies that can be configured on a device.

The <average-rate> parameter specifies the maximum rate allowed on a port during a one-second interval. The software automatically adjusts the number you enter to the lower multiple of 515,624 bps. Refer to the section “Average Rate” on page 56 for more details.

The <maximum-burst> parameter specifies the extra bits above the average-rate that traffic can have. Refer to the section “Maximum Burst” on page 57 for more details.

Configuring VLAN-and-priority based rate limiting

VLAN-and-priority based rate limiting limits traffic on a physical port that is a member of a specified VLAN and has been assigned to specified forwarding queues. For example, you can configure a rate limiting policy for inbound traffic on port 1/1. The policy limits the average rate of all inbound packets with VLAN tag 10 destined for hardware forwarding queues q0 and q2 to an average rate of 500 Mbps for each queue with a maximum burst size of 750 Mbits for each queue. Enter commands such as the following:

```
9408sl(config)# interface ethernet 1/1
9408sl(config-if-1/1)# rate-limit in vlan 10 pri q0 q2 500000000 750000000
```

Syntax: [no] rate-limit in vlan <number> priority <queue> <average-rate> <maximum-burst>

The **in** parameter applies the policy to traffic on inbound ports.

Enter the VLAN ID for the **vlan** <number> parameter.

The **priority q0 | q1 | q2 | q3** parameter specifies the hardware forwarding queue to which the policy applies. The device prioritizes the queues from **q0** (normal priority) to **q3** (highest priority).

The <average-rate> parameter specifies the maximum rate allowed on a port during a one-second interval. The software automatically adjusts the number you enter to the lower multiple of 515,624 bps. Refer to the section "Average Rate" on page 56 for more details.

The <maximum-burst> parameter specifies the extra bits above the average-rate that traffic can have. Refer to the section "Maximum Burst" on page 57 for more details.

Configuring VLAN Group Based Rate Limiting

A rate limiting policy can be applied to a VLAN group. VLANs that are members of a VLAN group share the specified bandwidth defined in the rate limiting policy applied to that group.

To configure a rate limiting policy for a VLAN group, do the following:

1. Define the VLANs that you want to place in a rate limiting VLAN group.
2. Define a rate limiting VLAN group. This VLAN group is specific to the rate limiting feature. Enter commands such as the following:

   ```
   9408sl(config)# rl-vlan-group 10
   9408sl(config-vlan-rate-group)# vlan 3 5 to 7 10
   9408sl(config-vlan-rate-group)# exit
   ```

   The commands assign VLANs 3, 5,6, 7, and 10 to rate limiting VLAN group 10.

   Syntax: [no] rl-vlan-group <vlan-group-number>

   Syntax: [no] vlan <vlan-number> [to <vlan-number]

   The **rl-vlan-group** command takes you to the VLAN group rate limiting level. Enter the ID of the VLAN group that you want to create or update by entering a value for <vlan-group-number>.

   Use the vlan command to assign or remove VLANs to the rate limiting VLAN group. You can enter the individual VLAN IDs or a range of VLAN IDs.
3. Create a policy for the VLAN group and apply it to the interface you want. Enter commands such as the following:

   ```
   9408sl(config)# int e 1/1
   9408sl(config-if-1/1)# rate limit in group 10 500000000 750000000
   ```

   The command applies the rate limiting policy for rate limiting VLAN group 10 on port 1/1. This policy limits all traffic tagged with VLANs 3, 5, 6, 7, or 10 to an average rate of 500 Mbps with a maximum burst size of 750 Mbits.

   Syntax: rate limit in group <group-number> average-rate maximum-burst

   The **in** parameter indicates that the policy is for incoming traffic.

   Enter the rate limiting VLAN group ID for the **group** <group-number> parameter.

   The <average-rate> parameter specifies the maximum rate allowed on a port during a one-second interval. The software automatically adjusts the number you enter to the lower multiple of 515,624 bps. Refer to the section "Average Rate" on page 56 for more details.

   The <maximum-burst> parameter specifies the extra bits above the average-rate that traffic can have. Refer to the section "Maximum Burst" on page 57 for more details.
4. If you want to apply a rate limiting policy to a VLAN group whose traffic are prioritized by hardware forwarding queues, enter commands such as the following:

   ```
   9408sl(config)# int e 1/1
   9408sl(config-if-1/1)# rate limit in group 10 priority q1 q2 500000000 750000000
   ```

   The command applies the rate limiting policy for rate limiting VLAN group 10 on port 1/1. This policy limits all

traffic tagged with VLANs 3, 5, 6, 7, or 10 on each hardware forwarding queue. Rate for q1 is rate limited to an average rate of 500 Mbps with a maximum burst size of 750 Mbits. Rate for q2 is also rate limited to an average rate of 500 Mbps with a maximum burst size of 750 Mbits.

Configuration Considerations

When configuring VLAN group based rate limiting policies, consider the following rules:

- A rate limit VLAN group must have at least one VLAN member before it can be used in a rate limit policy. The list cannot be empty if it is being used in a rate limiting policy.
- A rate limit VLAN group cannot be deleted if it is being used in a rate limiting policy.
- If a rate limit policy for a VLAN group is applied to a port, the group cannot be used in any other rate limiting policies applied to other ports that are controlled by the same packet processor.
- A VLAN can be member of multiple rate limit VLAN groups, but two groups with common members cannot be applied on ports controlled by the same packet processor.
- VLAN-based rate limiting and VLAN groups based rate limiting policies can be applied on the same ports or ports controlled by the same packet processor as long as there are no common VLANs in the policies.

Configuring a Port-and-ACL-Based Rate Limiting Policy

You can use standard or extended IP ACLs for port-and-ACL-based rate limiting.

- Standard IP ACLs match traffic based on source IP address information.
- Extended ACLs match traffic based on source and destination IP addresses and IP protocol information. Extended ACLs for TCP and UDP protocol must also match on source and destination IP addresses and TCP or UDP protocol information.
- You can apply an ACL ID to a port-and-ACL-based rate limiting policy even before you define the ACL. The rate limiting policy does not take effect until the ACL is defined.
- It is not necessary to remove an ACL from a port-and-ACL-based rate limiting policy before deleting the ACL.

NOTE: Port-and-ACL-based rate limiting is supported for traffic on inbound ports only.

To configure port-and-ACL-based rate limiting policies, enter commands such as the following:

```
9408sl(config)#access-list 50 permit host 1.1.1.2
9408sl(config)#access-list 50 deny host 1.1.1.3
9408sl(config)#access-list 60 permit host 2.2.2.3
9408sl(config-if-1/1)# rate-limit in access-group 50 500000000 750000000
Average rate is adjusted to 499639656 bits per second
9408sl(config-if-1/1)# rate-limit in access-group 60 100000000 200000000
Average rate is adjusted to 99515432 bits per second
```

These commands first configure access-list groups that contain the ACLs that will be used in the rate limiting policy. Use the **permit** condition for traffic that will be rate limited. Traffic that match the **deny** condition are not subject to rate limiting and allowed to pass through. Refer to “Filtering Traffic Denied by a Rate Limiting ACL” on page 63 for information on how to drop traffic that matches deny conditions.

Next, the commands configure two rate limiting policies on port 1/1. The policies limit the average rate of all inbound IP traffic that match the permit rules of ACLs 50 and 60. The first policy limits the rate of all permitted IP traffic from host 1.1.1.2 to an average rate of 500 Mbps with a maximum burst size of 750 Mbits. Rate of all traffic from host 1.1.1.3 is not subject to rate limiting since it is denied by ACL 50; it is merely forwarded on the port.

The second policy limits the rate of all IP traffic from host 2.2.2.3 to an average rate of 100 Mbps with a maximum burst size of 200 Mbits.

All IP traffic that does not match ACLs 50 and 60 are not subject to rate limiting.

Syntax: [no] rate-limit- in vlan <vlan-number> <average-rate> <maximum-burst>

The **in** parameter applies the policy to traffic on inbound ports.

The **access-group**, group-number> parameter specifies the group number to which the ACLs used in the policy belong.

NOTE: An ACL must exist in the configuration before it can take effect in a rate limiting policy.

Refer to the “Configuration Considerations” on page 57 regarding the number of ACL-based rate limiting policies that can be configured.

The <average-rate> parameter specifies the maximum rate allowed on a port during a one-second interval. The software automatically adjusts the number you enter to the lower multiple of 515,624 bps. Refer to the section “Average Rate” on page 56 for more details.

The <maximum-burst> parameter specifies the extra bits above the average-rate that traffic can have. Refer to the section “Maximum Burst” on page 57 for more details.

Configuring Port-and-Layer 2 ACL-based rate limiting

The port-and-Layer 2 ACL-based rate limiting limits the rate of traffic on individual physical ports that match the permit conditions a Layer 2 ACL. For example,

```
9408sl(config)# access-list 400 deny any any any etype arp
9408sl(config)# access-list 400 deny any any any etype ipv4
9408sl(config)# access-list 400 permit any any 100

9408sl(config)# interface ethernet 1/1
9408sl(config-if-1/1)# rate-limit in access-group 400 100000000 200000000
Average rate is adjusted to 99515432 bits per second
```

These commands first configure access-list group 400. This group contains the ACLs that will be used in the rate limiting policy. Use the **permit** condition for traffic that will be rate limited. Traffic that match the **deny** condition are not subject to rate limiting.

The next set of commands configures a rate limiting policies on port 1/1. The policies limit the average rate of all inbound IP traffic that match the permit rules of ACL 400 to an average rate of 100 Mbps with a maximum burst size of 200 Mbits. Traffic denied by ACL 400 is merely forwarded on the port.

Syntax: [no] rate-limit in access-group <number> <average-rate> <maximum-burst>

The **in** parameter applies the policy to traffic on inbound ports.

The **access-group** <number> parameter identifies the Layer 2 ACL used to permit or deny traffic on a port. Permitted traffic is subject to rate limiting.

NOTE: Port-and Layer 2 ACL-based rate limiting and Port-and-Layer 4 ACL-based rate limiting cannot be applied on a port at the same time.

The <average-rate> parameter specifies the maximum rate allowed on a port during a one-second interval. The software automatically adjusts the number you enter to the lower multiple of 515,624 bps. Refer to the section “Average Rate” on page 56 for more details.

The <maximum-burst> parameter specifies the extra bits above the average-rate that traffic can have. Refer to the section “Maximum Burst” on page 57 for more details.

Configuring Port-and-IPv6 ACL-based rate limiting

Software release 02.1.00 supports port-and-IPv6 ACL-based rate limiting. The port-and-IPV6 ACL-based rate limiting limits the rate of traffic on individual physical ports that match the permit conditions of an IPV6 ACL. Traffic that matches the deny condition is not subject to rate limiting.

For example, the following commands in the Global Config mode configure the IPv6 access-list: “ipv6-acl” to permit any traffic from the 10:10::0:0/64 network and deny all other traffic.

```
9408sl(config)# ipv6 access-list ipv6-acl
9408sl(config-ipv6-access-list ipv6-acl)# permit ipv6 10:10::0:0/64 any
```

```
9408sl(config-ipv6-access-list ipv6-acl)# deny ipv6 any any
```

The following configuration creates a rate limiting policy on port 1/1. The policy limits the average rate of all inbound IP traffic that matches the permit rules of ACL “ipv6-rl” to an average rate of 100 Mbps with a maximum burst size of 200 Mbits. Traffic denied by ACL “ipv6-rl” is forwarded on the port.

```
9408sl(config)# interface ethernet 1/1
9408sl(config-if-1/1)# rate-limit in ipv6-named-access-group ipv6-rl 100000000
200000000

Average rate is adjusted to 99515432 bits per second
```

Syntax: [no] rate-limit in ipv6-named-access-group <name> <average-rate> <maximum-burst>

The **in** parameter applies the policy to traffic on inbound ports.

The **ipv6-named-access-group** <name> parameter identifies the IPv6 ACL used to permit or deny traffic on a port. Permitted traffic is subject to rate limiting. Denied traffic is forwarded on the port.

The <average-rate> parameter specifies the maximum rate allowed on a port during a one-second interval.

The <maximum-burst> parameter specifies the extra Mbits above the average-rate that traffic can have.

Filtering Traffic Denied by a Rate Limiting ACL

When you use a Layer 2 ACL-based or Layer 4 ACL-based rate limiting policy, traffic permitted by the ACL is subject to rate limiting; however, traffic denied by the ACL is simply forwarded on the port. With the strict ACL feature, you can configure a port to drop traffic that is denied by the rate limiting ACL instead of forwarding it.

NOTE: Once you configure a Layer 2 ACL-based or Layer 4 ACL-based rate limiting policy on a port, you cannot configure a regular (traffic filtering) ACL on the same port. To filter this type of traffic, you must enable the strict ACL feature.

To enable the device to drop traffic that is denied by a rate limiting ACL, enter the following command at the configuration level for the port:

```
9408sl(config-if-1/1)# rate-limit strict-acl
```

Syntax: [no] rate-limit strict-acl

Display Rate Limiting Policies

The **show rate-limit** command has been added to display the rate limiting policies that have been configured on an interface.

For example, to display rate limiting policy on a device, enter the following command:

```
9408sl(config)# show rate-limit
interface e 1/1
 rate-limit input group 3 8765608 9000000
 rate-limit input group 10 priority q1 515624 1000000
 rate-limit input group 10 priority q0 q2 2578120 3000000
interface e 1/2
 rate-limit input 8765608 9000000
interface e 1/3
 rate-limit input vlan-id 5 515624 1000000
```

To display rate limiting policy on a device with counters, enter the following command:

```
9408sl(config)# show rate-limit counters
interface e 1/1
 rate-limit input group 3 8765608 9000000
  Pkts fwd: 20 Pkts drop: 10 Total: 30
 rate-limit input group 10 priority q1 515624 1000000
  Pkts fwd: 90 Pkts drop: 15 Total: 105
 rate-limit input group 10 priority q0 q2 2578120 3000000
  Pkts fwd: 221 Pkts drop: 11 Total: 232
 rate-limit input group 20 priority q1 q2 q3 515624 1000000
  Pkts fwd: 0 Pkts drop: 0 Total: 0
interface e 1/2
 rate-limit input 8765608 9000000
  Pkts fwd: 440 Pkts drop: 20 Total: 460
interface e 1/3
 rate-limit input vlan-id 5 515624 1000000
  Pkts fwd: 0 Pkts drop: 0 Total: 0
```

To display the rate limiting policies on interface 1/3, enter the following command:

```
9408sl(config)# show rate-limit interface 1/3
interface e 1/3
 rate-limit input vlan-id 5 515624 1000000
```

You can also display rate limiting policies for an interface that includes counters by entering the following command:

```
9408sl(config)# show rate-limit counters interface 1/4
interface e 1/4
 rate-limit input priority q1 8765608 9000000
  Pkts fwd: 200 Pkts drop: 150 Total: 350
```

Syntax: show rate-limit [counters] [interface <slot-number/port-number>]

For inbound rate limiting policies, specify the **counters** parameter if you want counters to be included in the display. Counters show the estimated number of packets that matched a rate limiting policy and were either forwarded or dropped, based on the availability of credit. If you do not use this parameter, the counters are not included in the display.

Outbound port rate limiting policies have no counters.

Use the **interface** <slot-number/port-number> to display rate limiting policies for a specific interface.

Displaying Rate Limit VLAN Groups

To display the rate limit VLAN groups and their members, enter the following command:

```
9408sl#show rate-limit group
rl-vlan-group 3
 vlan 2 to 3
rl-vlan-group 10
 vlan 25 29 to 40 42 100 to 2000
```

To display VLAN members of a specific rate limit VLAN group, enter a command such as the following:

```
9408sl#show rate-limit group 3
rl-vlan-group 3
 vlan 2 to 3
```

Syntax: show rate-limit group <group-number>

Specify the rate limit group number for the **group** <group-number> parameter.

Enabling Support for Network-based ECMP Load Sharing for IPv6

In previous releases of ProCurve 9408sl software, only ECMP Load sharing by host was supported for IPv6. In that configuration, a simple round-robin mechanism is employed to distribute traffic across equal-cost paths based on the destination host IP address. Routes to each destination host are stored in CAM and accessed when a path to a host is required.

Beginning with software release 02.1.00, network-based ECMP load sharing is also supported. If this configuration is selected, traffic is distributed across equal-cost paths based on the destination network address. Routes to each network are stored in CAM and accessed when a path to a network is required. Because multiple hosts are likely to reside on a network, this method uses fewer CAM entries than load sharing by host. When you select network-based ECMP load sharing, you can choose either of the following two CAM modes:

Dynamic Mode – In the dynamic mode, routes are entered into the CAM dynamically using a flow-based scheme. In this mode routes are only added to the CAM as they are required. Once routes are added to the CAM, they are subject to being aged-out when they are not in use. Because this mode conserves CAM, it is useful for situations where CAM resources are stressed or limited.

Static Mode – In the static mode, routes are entered into the CAM whenever they are discovered. Routes aren't aged once routes are added to the CAM and they are subject to being aged-out when they are not in use.

Configuring the CAM Mode to Support Network-based ECMP Load Sharing for IPv6

To configure the CAM mode to support network-based ECMP load sharing for IPv6, use a command such as the following at the Global Configuration level:

```
9408sl(config)# #cam-mode ipv6 dynamic
```

Syntax: [no] cam-mode ipv6 [dynamic | static | host]

The **dynamic** parameter configures the 9408sl for network-based ECMP load sharing using the dynamic CAM mode.

The **static** parameter configures the 9408sl for network-based ECMP load sharing using the static CAM mode.

The **host** parameter configures the 9408sl for host-based ECMP load sharing using the dynamic CAM mode.

You must reload the router for this command to take effect.

Fast Direct Routing

Fast Direct Routing (FDR), also known as IP static cam mode, enables very large routing/forwarding tables (up to twice the published Internet routes) to be maintained at the interface module level so that all packet forwarding is done at wire speed without the need to learn the best routes in real-time. FDR can significantly reduce network convergence time to minimize customer impact in the case of a network topology change. To enable FDR on a ProCurve 9408sl you must perform the following procedures:

- "Configuring CAM Partitions for FDR"
- "Setting the CAM Mode to Enable FDR"

Configuring CAM Partitions for FDR

CAM partitioning is performed to allow you to dedicate CAM for specific purposes. Configuring FDR requires you to partition CAM by block and to then partition the blocks in more detail. This section describes what CAM partitioning is required to achieve high-performance from FDR.

CAM partitioning by block allows you to dedicate CAM blocks to the following applications: session-mac, ip-mac, out-session, ipv6, and ipv6-session. This feature is described in “CAM Partitioning by Block” on page 32. Because FDR maintains a large number of IP routes within CAM, we suggest that you assign a greater number of blocks to the IP MAC partition when configuring your router for FDR. The specific recommendation is described in “Configure CAM Partitioning by Block” on page 66.

Once you’ve configured a sufficient number of blocks of CAM for IP routes, the ip-mac partition can be more finely partitioned to assign routes to IP supernet levels based upon their prefix height. In this scheme, routes assigned to IP supernet level 1 are those with the maximum prefix length and the best routes and routes with a smaller prefix length are assigned to IP supernet levels greater than 1. Depending on the number of routes, there can be up to 32 IP supernet levels assigned. If there are only two routes, then the route with the shorter prefix length of the two routes will be assigned to the IP supernet level 2. Additional IP supernet levels are assigned as required.

For example, if the router needs to find a route to a host with the IP address 10.10.10.4, routes with the following two destinations would be considered qualified routes: 10.10.10.0/24 and 10.0.0.0/8. The route to the 10.10.10.0/24 network is much more specific than 10.0.0.0/8. Consequently, it is judged to be the more efficient route. If these were the only two routes, the route with the 10.10.10.0/24 destination would be assigned as the IP supernet level 1 route and the route with the 10.0.0.0/8 would be assigned as the IP supernet level 2 route. If a route is later discovered with the destination 10.10.0.0/16, it will be assigned as the IP supernet level 2 route and the route to 10.0.0.0/8 will be reassigned to become the IP supernet level 3 route. There are 32 IP supernet levels possible to reflect the 32 bits of an IP address. Directly connected hosts are a special case and are classified as IP supernet level 0 routes.

Different amounts of CAM are assigned to each of the IP supernet levels as described in “Configure CAM Partitioning by IP Supernet” on page 66.

Configure CAM Partitioning by Block

CAM partitioning by block allows you to dedicate CAM blocks to the following applications: session-mac, ip-mac, out-session, ipv6, and ipv6-session. This feature is described in “CAM Partitioning by Block” on page 32. The default CAM block allocations are listed in Table 8. To optimize your system for FDR, we recommend that you set these blocks to the levels specified in Table 15.

Table 15: CAM partition allocation for FDR

Number of Blocks	Allocation Parameter
1 block	session-mac
4 blocks	ip-mac
1 block	out-session
1 blocks	ipv6
1 block	ipv6-session

To configure the CAM partition blocks to the levels recommended for FDR, perform the following command:

```
9408sl(config)#cam-partition block session-mac 1 ip-mac 4 out-session 1 ipv6 1 ipv6-
session 1
```

Configure CAM Partitioning by IP Supernet

You can assign different amounts of CAM to each of the first 5 IP supernetting levels (Levels 0 -4). This can be done by assigning a specific number of routes that each IP supernet level can contain or by assigning percentages of available CAM to each level. Observations of routers on the internet suggest that greater than 90% of the routes can be classified as IP supernet level 1, between 6% and 7% as IP supernet 2, about 1% as IP supernet 3 and less than 1% as IP supernet levels of 4 or greater. Levels 5 and above are set to default values on the 9408sl and are not configurable.

To optimize your system for FDR, we recommend that you set the IP supernet levels 0 to 4 as described in Table 16.

Table 16: Recommended IP Supernet CAM allocation for FDR

Supernet Level	Allocation for Specified Level (# of routes)
Level 0	1024
Level 1	192935
Level 2	151158n
Level 3	2087
Level 4	1024

To configure the CAM for IP supernet levels 0 to 4 as described in Table 16, perform the following command:

```
9408sl(config)#cam-partition ip supernet 0 1024 1 192935 2 151158 3 2087 4 1024
```

Syntax: [no] cam-partition ip supernet <supernet-level> <cam-allocation>

The <supernet> variable specifies the IP supernet level that you are assigning CAM to. Levels 0 to 4 can be configured.

The <cam-allocation> variable specifies the amount of CAM that is allocated to the specified IP supernet level. This variable can be expressed as a number of routes or as a percentage of available CAM.

While these assignments will work in most cases, you can use the CAM partition show commands to monitor the actual CAM usage of your router. From this information, you can determine whether you need to change the settings. For information on how to use these commands, see “Using the Display Commands to Evaluate CAM Partition Assignment” on page 67.

Setting the CAM Mode to Enable FDR

The default IP CAM mode in this software release is dynamic CAM mode. To enable Fast Direct Routing (FDR), you can set the CAM mode to static IP CAM mode (FDR) using the following command:

```
9408sl(config)# cam-mode ip static
```

You must reload the router for this command to take effect.

Syntax: [no] cam-mode ip [dynamic | static]]

The **dynamic** parameter configures the ProCurve 9408sl for dynamic CAM mode. This is the default mode.

The **static** parameter configures the 9408sl for static CAM mode also known as FDR.

Using the Display Commands to Evaluate CAM Partition Assignment

While the recommended CAM assignments for IP supernetting levels will work in most cases, you can use the following display commands to determine your current settings and to examine if the settings are adequate to your application:

- “Using the Show Cam-partition Command” on page 67
- “Using the Show ip cam-failure Command” on page 68

Using the Show Cam-partition Command

The **show cam-partition** command allows you to see the number of routes that are configured to be available per IP supernet level on each interface module. In addition, you can also find out how much of the capacity is currently available for new routes. The output display from this command is extensive and would take up several pages to present here. Consequently, we only show the sections that are relevant to the IP subnet level settings and current usage.

To display CAM partition information, use the following command:

```
9408sl#show cam-partition slot 3
Slot 3 XPP/XTM 0:
# of CAM device                = 1
CAM device size                = 131072 entries (9Mbits)
Total CAM Size                 = 131072 entries (9Mbits)

...

IP  Size   = 24576
   0 Subpartition Size = 1024
   1 Subpartition Size = 43220
   2 Subpartition Size = 3500
   3 Subpartition Size = 512
   4 Subpartition Size = 256
...
```

The part of the output from the command shown, displays each of the configurable IP supernet levels and the number of routes that are configured to be available at that level. If you have used the **cam-partition ip supernet** command, these numbers should reflect the amounts that you have configured. Otherwise, they will reflect the default values.

In another section of the output for this command, the amount of free CAM is shown for each IP supernet level as shown below. The bolded sections show the IP supernet level on one side, and the number of free routes on the other for the levels that are user-configurable. As described earlier, IP supernet levels 5 and above are not user-configurable.

```
...

IP Section:      73728 (012000) -  98303 (017fff)
  IP Supernet    0:    64512 (00fc00) -  65535 (00ffff), free   1010
  IP Supernet    1:    21292 (00532c) -  64511 (00fbff), free  43220
  IP Supernet    2:    17792 (004580) -  21291 (00532b), free   3500
  IP Supernet    3:    17280 (004380) -  17791 (00457f), free    512
  IP Supernet    4:    17024 (004280) -  17279 (00437f), free    256
  IP Supernet    5:    16896 (004200) -  17023 (00427f), free    128
  IP Supernet    6:    16832 (0041c0) -  16895 (0041ff), free     64
 ...
```

If the number of free routes starts to get too small, this could be an indication that you need to increase the amount for that IP supernet level.

Using the Show ip cam-failure Command

Another way to determine if the number of entries assigned per IP supernet level are adequate to your application is to examine if there are any IP CAM failures. You can do this by using rconsole to log into an interface module and executing the **show ip cam-failure** command as shown in the following:

```
rconsole-4/1@LP#show ip cam-failure

   RecoveryRequired : 1 RecoveryInProgress 0
   Total invalid route count 0
   Number of CAM required count
   1      Total 100000
   2      Total  4000
   Number of CAM failure count
   1      Total  5
   Number of routes not in CAM
   1      Total  5
```

In this example, you can see that there was one failure that required a recovery. The Number of CAM required count specifies 100000 for IP supernet level 1 and 4000 for IP supernet level 2. These numbers represent the actual number of routes that are being held in CAM at each of these levels.

The **Number of CAM failure count** value is set at a total of 5 for IP supernet level 1. This and the next statistic, **Number of routes not in CAM** set equal to 5, indicates that there is not enough CAM available for supernet level 1 routes.

Using the Show ip prefix-height Command

Another way to determine the number of entries that the routing table has for each IP supernet level is to examine the number of routes that are contained in each IP supernet level. You can do this by using rconsole to log into an interface module and executing the **show ip prefix-height** command as shown in the following:

```
rconsole-4/1@LP#sh ip prefix-height

>From Trie
1     Total 612
2     Total 42014
3     Total 4803
Total number of routes = 47429

Calculated
1     Total 612
2     Total 42014
3     Total 4803
Total number of routes = 47429
```

The number at the left (shown bolded) is the IP supernet level and the total to the right of it is the number of routes that are currently contained at that level. If these numbers exceed or are close to the capacity set, that would indicate that the capacity should be increased.

Configuring SSL Security for the Web Management Interface

Starting with software release 02.1.00, the ProCurve 9408sl supports Secure Sockets Layer (SSL) for configuring the device using the Web Management interface. When enabled, the SSL protocol uses digital certificates and public-private key pairs to establish a secure connection to the ProCurve device. Digital certificates serve to prove the identity of a connecting client, and public-private key pairs provide a means to encrypt data sent between the device and the client.

Configuring SSL for the Web Management interface consists of the following tasks:

- Enabling the SSL server on the ProCurve device
- Importing an RSA certificate and private key file from a client (optional)
- Generating a certificate

Enabling the SSL Server on the ProCurve Device

To enable the SSL server, enter the following command:

```
9408sl(config)# web-management https
```

Syntax: [no] web-management http | https

You can enable either the HTTP or HTTPs servers with this command.

Importing Digital Certificates and RSA Private Key Files

To allow a client to communicate with the ProCurve 9408sl using an SSL connection, you configure a set of digital certificates and RSA public-private key pairs on the device. A digital certificate is used for identifying the connecting client to the server. It contains information about the issuing Certificate Authority, as well as a public key. You can either import digital certificates and private keys from a server, or you can allow the ProCurve device to create them.

If you want to allow the ProCurve device to create the digital certificates, see the next section, “Generating an SSL Certificate”. If you choose to import an RSA certificate and private key file from a client, you can use TFTP to transfer the files.

For example, to import a digital certificate using TFTP, enter a command such as the following:

```
9408sl(config)# ip ssl certificate-data-file tftp 192.168.9.210 certfile
```

Syntax: [no] ip ssl certificate-data-file tftp <ip-addr> <certificate-filename>

To import an RSA private key from a client using TFTP, enter a command such as the following:

```
9408sl(config)# ip ssl private-key-file tftp 192.168.9.210 keyfile
```

Syntax: [no] ip ssl private-key-file tftp <ip-addr> <key-filename>

The <ip-addr> is the IP address of a TFTP server that contains the digital certificate or private key.

Generating an SSL Certificate

After you have imported the digital certificate, generate the SSL certificate by entering the following command:

```
9408sl(config)# crypto-ssl certificate generate
```

Syntax: [no] crypto-ssl certificate generate

If you did not already import a digital certificate from a client, the device can create a default certificate. To do this, enter the following command:

```
9408sl(config)# crypto-ssl certificate generate default
```

Syntax: [no] crypto-ssl certificate generate default

Deleting the SSL Certificate

To delete the SSL certificate, enter the following command:

```
9408sl(config)# crypto-ssl certificate zeroize
```

Syntax: [no] crypto-ssl certificate zeroize

Setting Maximum Frame Size Per PPCR

Beginning with software release 02.1.00, when you set a maximum frame size, that maximum applies to all ports that are associated with the same packet processor (PPCR). Table 17 shows the ports of each interface module.

Table 17: Ports available per PPCR

Module type	Number of Packet Processors (PPCR)	Module Port Range Belonging to each PPCR PPCR 1	PPCR 2	PPCR 3	PPCR 4	PPCR 5	PPCR 6	PPCR 6	PPCR 8
4 x 10G	4	1	2	3	4	N/A	N/A	N/A	N/A
40 x 1G	4	1 - 10	11 - 20	21 -30	31 - 40	N/A	N/A	N/A	N/A
60 x 1G	3	1 - 20	21 - 40	41 - 60	N/A	N/A	N/A	N/A	N/A

To set a maximum frame size for all the ports attached to a PPCR, enter a command such as the following at the interface Configuration level:

```
9408sl(config)#interface ethernet 6/4
9408sl(config-if-e1000-6/4)#max-frame-size 1500 bytes.
```

In this example the maximum frame size is applied to port 4 of a 40 x 1G Ethernet interface module. That means that this maximum will apply to ports 1 to 10 on the interface module.

Syntax: max-frame-size <frame-size>

The <frame-size> variable specifies the maximum frame size for each port that is connected the same PPCR as described in Table 17. Values can be from 64 to 9212 bytes.

New Command for Setting Fan Speed

Previously the following two commands were used for setting fan speed:

In the Privileged EXEC mode:

`set-fan-speed`

In Global CONFIG mode:

`fan init/read-temperature/read-speed/set-speed`

Both of these commands have been eliminated and replaced with the following command:

Syntax: set-fan-speed [low | med | med-hi | high | auto]

The **low** parameter sets the fan speed to 50% of full speed

The **med** parameter sets the fan speed to 75% of full speed

The **med-hi** parameter sets the fan speed to 90% of full speed

The **high** parameter sets the fan speed to 100% of full speed.

The **auto** parameter set the fan speed to be adjusted by the monitoring service. This is the default setting. Since the “temperature monitoring service” sets both fans to the same speed, the new command also affects both fans.

If set the fan speed to anything other than “auto”, the fan mode becomes manual. In manual mode, the “temperature monitoring service” is stopped, and the fan speed will not change regardless of temperature changes to the chassis.

This command can be saved like other configuration commands.

Using the **show chassis** command you can determine if the chassis is in "auto mode" or "manual mode.

Downloading a New Image Using a Script

Beginning with software release 02.1.00, you can create a script to download new software images to your ProCurve 9408sl. Use this command to download an image using a script:

Syntax: copy tftp system <ip_addr> <download_script>

The **<ip_addr>** variable is used to identify the IP address of the tftp server that holds the script.

The **<download_script>** variable is the name of the script containing download specifications.

The CLI command first copies the download script specified to the system's memory. It then parses the script to perform the software download specified in the script.

The following section describes the download script syntax.

```
# download script syntax:
# <spec_line>
# ...
# <spec_line>
# where <spec_line> == KEYWORD:<val>;
#
# 1) Supported KEYWORD
#    SRC                    // specify source of the images, optional
#    DIR                    // image source directory, optional
#    MP_MON                 // MP monitor image
#    MP_APP                 // MP application image
#    LP_MON                 // LP monitor image
#    LP_APP                 // LP application image
#    XPP                    // FPGA XPP
#    XTM                    // FPGA XTM
#    PBIF                   // FPGA PBIF
#    XBRIDGE                // FPGA XBRIDGE
```

```
# Note: If SRC is not specified, the images are taken from the server specified in
the CLI command line.
#
# 2) Syntax of <val>
#
# It depends on the KEYWORD preceding it:
#
# SRC:tftp:<ip_addr>;
#
# MP_MON:<image_name>:[boot];  // [boot] is the option to copy monitor to boot.
# MP_APP:pri:<image_name>;
# MP_APP:sec:<image_name>;
#
# LP_MON:all:<image_name>:[boot];
# LP_MON:<slot#>[[,-]<slot#>]:<image_name>:[boot];
#
# LP_APP:pri:all:<image_name>;
# LP_APP:pri:<slot#>[[,-]<slot#>]:<image_name>;
# LP_APP:sec:all:<image_name>;
# LP_APP:sec:<slot#>[[,-]<slot#>]:<image_name>;
#
# XPP:all:<image_name>;
# XPP:<slot#>[[,-]<slot#>]:<image_name>;
#
# XTM:all:<image_name>;
# XTM:<slot#>[[,-]<slot#>]:<image_name>;
#
# PBIF:all:<image_name>;
# PBIF:<slot#>[[,-]<slot#>]:<image_name>;
#
# XBRIDGE:all:<image_name>;
# XBRIDGE:<slot#>[[,-]<slot#>]:<image_name>;
#
# Note: If one <spec_line> fails to parse, or it fails to copy, the script is
aborted.
```

Sample Install Script

The following example script installs software files on a ProCurve 9408sl using files previously stored on a TFTP server.

The script must be stored in the same directory as the image files. Be sure to change the script to match your needs, as noted in the script comments.

The script installs all files from the source area as follows:

1. Install file mb02100c.bin to MP in both the monitor area, and also boot flash.
2. Install file mpr02100c.bin to MP primary flash.
3. Install file lb02100c.bin to all LPs in both the monitor area, and also boot flash.
4. Install file lp02100c.bin to all LPs in the primary flash area.
5. Install FPGA file pbif02100c.bin to all LPs.
6. Install FPGA file xtm02100c.bin to all LPs.
7. Install FPGA file xpp02100c.bin to all LPs.

8. Install FPGA file xbridge02100c.bin to all LPs (error messages will indicate the slots that do not have 60-port modules, and therefore do not need or accept the XBRIDGE image).

After the script completes, use the **show flash** command to verify successful transfer of boot image, monitor image, and primary software image. (There is no command to verify successful FPGA file transfer until after the interface modules have been power-cycled.)

Then use this command to reboot the management module (which will cause a power-cycle of the interface modules), specifying **primary** to correspond with where the script placed the new software images:

- boot system flash primary

```
# Filename: 9408sl-Script-02100c-TFTP.txt
#
# Sample 9408sl install script for 02.1.00c.
#
# This version uses TFTP to install these
# images: boot-and-monitor, pri flash, FPGAs.
#
# CHANGES NEEDED TO USE THIS:
# 1) change IP address to be your TFTP server
#
# NOTES:
# i. Script must be stored in
#    same directory as image files.
# ii. If any line fails, script aborts!
#     Users MUST verify results!
# iii. After files are installed, 9408sl must be
#      rebooted for the upgrade to take effect.
#
# SYNTAX:
#    copy tftp system <ip-addr> <script-filename>
#
SRC:tftp:10.10.10.56;
MP_MON:mb02100c.bin:boot;
MP_APP:pri:mpr02100c.bin;
LP_MON:all:lb02100c.bin:boot;
LP_APP:pri:all:lp02100c.bin;

PBIF:all:pbif02100c.bin;
XTM:all:xtm02100c.bin;
XPP:all:xpp02100c.bin;
XBRIDGE:all:xbridge02100c.bin;
```

Enhancements and Configuration Notes in 02.2.01

Multi-Device Port Authentication

Multi-device port authentication is a way to configure a 9408sl to forward or block traffic from a MAC address based on information received from a RADIUS server. The *Security Guide* describes how this feature works and the CLI commands used to configure the feature. The options and CLI described in the *Security Guide* are supported on the 9408sl beginning with software release 02.2.01, with the following exceptions:

- The **mac-authentication enable** command, which enables multi-device port authentication is configured only on the global level, not the interface level.
- The Denial of Service Attack Protection, which is enabled or disabled using the **[no] mac-authentication dos-protection** command is not supported.

- The **mac-authentication disable-aging** command has been enhanced as follows:

  ```
  9408sl(config)#mac-authentication disable-aging denied-only
  ```

Syntax: mac-authentication disable-aging denied-only | permitted-only

Enter the command at the global or interface configuration level.

The **denied-only** parameter prevents denied sessions from being aged out, but ages out permitted sessions.

The **permitted-only** parameter prevents permitted (authenticated and restricted) sessions from being aged out and ages denied sessions.

- Multi-device port authentication and 802.1x port security can be enabled on the same interface. However, only one of these features will be used to authenticate a MAC addresses and 802.1x client. If an 802.1x client responds, the software assumes that the MAC address should be authenticated using 802.1x protocols. Multi-device port authentication for that MAC address is aborted.

Enhancement to 802.1X Port Security

In software release 02.2.01, this feature allows you to enable multi-device port authentication and 802.1X port security on the same interface. However, only one of these features will be used to authenticate a MAC addresses and 802.1X client. If an 802.1X client responds, the software assumes that the MAC address should be authenticated using 802.1X protocols. Multi-device port authentication for that MAC address is aborted.

The *Security Guide* describes 802.1X port security and the CLI commands used to configure the feature. The “Configuring 802.1X Port Security” chapter applies to the 9408sl, with the following exceptions:

- Flow-based, not rule-based, ACLs can be applied to 802.1X ports.
- You do not need to enter the **multi-user policy enable** command to enable the configuration of dynamic ACL and MAC filter assignments on an 802.1X multiple host configuration. This feature is enabled by default on the 9408sl.
- Instead of using the **mac filter-group** command to define MAC filters for EAP frames, the 9408sl uses the **mac access-group** command.

VLAN Byte Accounting

With software release 02.2.01, you can enable your 9408sl to perform accounting of the number of bytes received by all the member ports of a VLAN. This includes the preamble and the minimum inter-frame gap in Ethernet. The byte counts can then be viewed using the **show vlan** command. VLAN byte accounting is disabled by default.

Considerations When Configuring VLAN Byte Accounting

- VLAN byte accounting cannot be enabled for the default and control VLANs.
- The number of VLANs on which byte accounting can be enabled system-wide is restricted by the number of VLANs with byte accounting enabled on a given packet processor and the number of rate limiting policies enabled on the same packet processor ports.
- On a given packet processor, the total number of VLANs with byte accounting enabled and number of rate limiting policies based on ACLs and VLANs is dependent on the interface module. See Table 19 for details.
- If a port's VLAN has byte accounting enabled, you cannot enable rate limiting on that port. Similarly, if a port has rate limiting enabled, you cannot enable VLAN byte accounting on that port's VLAN.
- Clearing the rate limiting counters using **clear rate-limit counters** will also clear VLAN byte-accounting counters. It is recommended that when using rate limiting along with VLAN byte accounting, use individual port rate limiting counter clear command.

Configuring VLAN Byte Accounting

To enable VLAN accounting on a specified VLAN, use the following commands:

```
9408sl(config)# vlan 10
9408sl(config-vlan-10)# byte-accounting
```

Syntax: [no] byte-accounting

Displaying VLAN Byte Accounting Information

To display VLAN accounting information for all VLANs configured on a router, use the **show vlan** command as shown:

```
9408sl# show vlan

Configured PORT-VLAN entries: 2
Maximum PORT-VLAN entries: 512
Default PORT-VLAN id: 1

PORT-VLAN 1, Name DEFAULT-VLAN, Priority Level0
L2 protocols   : NONE
Untagged Ports : ethe 1/1 to 1/40 ethe 2/1 to 2/4

PORT-VLAN 10, Name [None], Priority Level0
L2 protocols   : NONE
Tagged Ports   : ethe 1/2 to 1/5
Bytes received : 18527
```

To display VLAN accounting information for a specific VLAN, use the **show vlan <vlan>** command as shown:

```
9408sl# show vlan 10

PORT-VLAN 10, Name [None], Priority Level0
L2 protocols   : NONE
Tagged Ports   : ethe 1/2 to 1/5
Bytes received : 5626
```

The byte accounting statistics are displayed above in **bold**. Table 18 describes the Byte Accounting statistics displayed when using the **show vlan** command.

Syntax: show vlan [<vlan-id>]

Table 18: VLAN Byte Accounting in Show VLAN

This Field...	Displays...
Bytes received	This field displays the number of bytes received by all member ports of all VLANs configured on the router if the command **show vlan** is used or if the **<VLAN>** variable is used with the **show vlan** command

Maximum Number of Rate Limiting Policies and VLANs with Byte Accounting

In software release 02.2.01, the maximum number of ACL-based, and VLAN-based rate limiting policies that can be configured on ports controlled by the same packet processor also depends on the number of VLANs with byte accounting enabled on the same packet processor.

On a given packet processor (PPCR), the total of:

Number of VLANs with byte accounting enabled
+
Number of rate limiting policies based on ACLs and VLANs

cannot exceed the maximum number per-PPCR as specified in Table 19.

Table 19: Maximum # of rate limiting policies and VLANs w/ byte accounting permitted per-PPCR

Module Type	PPCR Number	Ports supported by PPCR	Max # of rate limiting policies based on ACLs and VLANs + number of VLANs w/ byte accounting enabled
4 x 10G	PPCR 1	1	126
	PPCR 2	2	126
	PPCR 3	3	126
	PPCR 4	4	126
40 x 1G	PPCR 1	1 - 10	117
	PPCR 2	11 - 20	117
	PPCR 3	21 - 30	117
	PPCR 4	31 - 40	117
60 x 1G	PPCR 1	1 - 20	107
	PPCR 2	21 - 40	107
	PPCR 3	41 - 60	107

Clearing Counters

To clear the byte counter for a VLAN, enter a command such as the following:

```
9408sl(config)# clear vlan byte-accounting 10
```

You can also enter the following command to clear the byte counter for all VLANs

```
9408sl(config)# clear vlan byte-accounting all-vlans
```

Syntax: clear vlan byte-accounting <vlan-id> | all-vlans

Enter a VLAN ID if you want to clear the byte counters for a specific VLAN. Enter **all-vlans** to clear the byte counters for all VLANs.

Changes to Rate Limiting Counters in ProCurve 9408sl Release 02.2.00

In ProCurve 9408sl software release 02.2.00 and later, rate limiting counters have been changed to count bytes instead of packets. The CLI remains the same. The display will show bytes forwarded and dropped instead of packets forwarded and dropped.

```
9408sl(config)# show rate-limit counters

interface e 1/2 rate-limit
  input 8765608 9000000
  Bytes fwd: 440 Bytes drop: 20 Total: 460
```

The byte accounting statistics are displayed above in **bold**. The byte count includes the preamble and the minimum inter-frame gap in Ethernet.

Graceful Restart

The Graceful Restart feature provides support for high-availability routing. With this feature enabled, disruptions in forwarding are minimized and route flapping diminished to provide continuous service during times when a router experiences a restart. Graceful restart is effective during management module failover, not during a router reboot. Also, graceful restart does not function if the active management is hot-swapped (pulled from the chassis during operation). Graceful restart is only useful when the management module fails over to the standby management because of hardware failure, or because of a manually-initiated failover with the CLI switchover command. Beginning with software release 02.2.01, the graceful restart feature is supported for the following protocols:

- BGP (draft-ietf-idr-restart-10.txt)
- OSPF (RFC 3623)

Graceful Restart in BGP

Under normal operation, restarting a BGP router causes the network to be reconfigured. In this situation, routes available through the restarting router are first deleted when the router goes down and are then rediscovered and re-added to the routing tables when the router is back up and running. In a network where routers are restarted regularly, this can degrade performance significantly and limit availability of network resources. BGP graceful restart dampens the network topology changes and limits route flapping by allowing routes to remain available between routers during a restart. BGP Graceful restart operates between a router and its peers and must be configured on both.

A BGP router with graceful restart enabled advertises its graceful restart capability and restart timer to establish peering relationships with other routers. Once the restarting router is restarted, it begins to reestablish BGP connections and receive routing updates from its peers. When the restarting router receives all end-of-RIB markers from its helper neighbors that indicates that it has received all of the BGP route updates, all of the routes are recomputed, and newly computed routes replace the stale routes in the routing table.

During the restarting process, the helper neighbors will continue to use all of the routes learned from the restarting router and mark them as stale for the length of learned restart timer. If the restarting router doesn't come back up within the restart timer, the routes marked stale will be removed.

Configuring BGP Graceful Restart

To configure BGP Graceful Restart, you must enable it on all BGP peers where you want it to operate and set the following timers:

- Restart Timer
- Stale Routes Timer
- Purge Timer

NOTE: After configuring BGP Graceful Restart, you need to reset neighbor session using the **clear ip bgp neighbor** command whether or not the neighbor session is up. This command clears and re-establishes neighbor sessions.

Configuring BGP Graceful Restart on a Router

Use the following command to enable the BGP graceful restart feature on a 9408sl router:

```
9408sl(config)# router bgp
9408sl(config-bgp)# graceful-restart
```

Configuring BGP Graceful Restart Timer

Use the following command to specify the maximum amount of time a router will maintain routes from a restarting router and forward traffic to a restarting router.

```
9408sl(config)# router bgp
9408sl(config-bgp)# graceful-restart
9408sl(config-bgp)# restart-timer 60
```

Syntax: restart-timer <seconds>

The <seconds> variable sets the maximum number of seconds the restarting router will take to restart. Also, the peer routers waits this number of seconds to re-establish BGP connection and to keep using the learned routes from the restarting router. Enter 10 – 3600 seconds. The default value is 120 seconds.

Configuring BGP Graceful Restart Stale Routes Timer

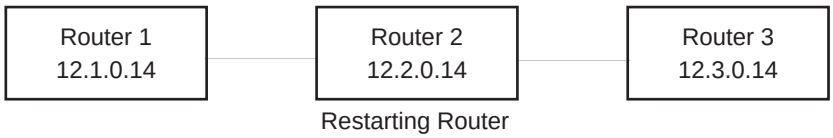
Use the following command to specify the maximum amount of time a helper router will wait for an end-of-RIB message from a restarting router before deleting stale routes learned from that restarting router:

```
9408sl(config-bgp)# stale-routes-time 30
```

Syntax: stale-routes-time <seconds>

The <seconds> variable sets the number of seconds that a helper router will wait for an end-of-RIB (restart complete) message from a restarting router. Enter 10 – 3600 seconds. The default value is 360 seconds.

EXAMPLE:



Router 1

```
9408sl(config)#router bgp
9408sl(config-bgp)#local-as 100
9408sl(config-bgp)#graceful-restart
9408sl(config-bgp)#neighbor 12.2.0.14 remote-as 250
9408sl(config-bgp)#write memory
```

Router 2

```
9408sl(config)#router bgp
9408sl(config-bgp)#local-as 100
9408sl(config-bgp)#graceful-restart
9408sl(config-bgp)#neighbor 12.1.0.14 remote-as 250
9408sl(config-bgp)#neighbor 12.3.0.14 remote-as 250
9408sl(config-bgp)#write memory
```

Router 3

```
9408sl(config)#router bgp
9408sl(config-bgp)#local-as 100
9408sl(config-bgp)#graceful-restart
9408sl(config-bgp)#neighbor 12.2.0.14 remote-as 250
9408sl(config-bgp)#write memory
```

Displaying BGP Graceful Restart information

You can display the BGP Graceful Restart configuration by entering the following command:

```
9408sl#show ip bgp neighbor 11.11.11.2
1 IP Address: 11.11.11.2, Remote AS: 101 (EBGP), RouterID: 101.101.101.1
Local AS: 200
State: ESTABLISHED, Time: 0h18m15s, KeepAliveTime: 60, HoldTime: 180
KeepAliveTimer Expire in 44 seconds, HoldTimer Expire in 167 seconds
RefreshCapability: Received
GracefulRestartCapability: Received
Restart Time 120 sec, Restart bit 0
afi/safi 1/1, Forwarding bit 0
GracefulRestartCapability: Sent
Restart Time 30 sec, Restart bit 0
afi/safi 1/1, Forwarding bit 0
Messages: Open Update KeepAlive Notification Refresh-Req
Sent : 1 5 15 0 0
Received: 1 1 15 0 0
Last Update Time: NLRI Withdraw NLRI Withdraw
Tx: --- --- Rx: --- ---
Last Connection Reset Reason:Unknown
Notification Sent: Unspecified
Notification Received: Unspecified
Neighbor NLRI Negotiation:
Peer Negotiated IPV4 unicast capability
Peer configured for IPV4 unicast Routes
TCP Connection state: ESTABLISHED
TTL check: 0, value: 0, rcvd: 64
Byte Sent: 628, Received: 363
Local host: 11.11.11.1, Local Port: 8190
Remote host: 11.11.11.2, Remote Port: 179
ISentSeq: 2123652 SendNext: 2124281 TotUnAck: 0
TotSent: 629 ReTrans: 1 UnAckSeq: 2124281
IRcvSeq: 2300094 RcvNext: 2300458 SendWnd: 65000
TotalRcv: 364 DupliRcv: 0 RcvWnd: 65000
SendQue: 0 RcvQue: 0 CngstWnd: 1460
```

Syntax: show ip bgp neighbor <address>

Graceful Restart in OSPF (RFC 3623)

With OSPF graceful restart enabled, a restarting router sends special LSAs to its neighbors called **grace-lsas**. These LSAs are sent to neighbors either before a planned OSPF restart or immediately after an unplanned restart. A grace LSA contains a grace period value that the requesting routers asks its neighbor routers to use for the existing routes, to and through the router after a restart. The restarting router comes up, it continues to use its existing OSPF routes to forward packets. In the background, it re-establishes OSPF adjacencies with its neighboring router, relearns all OPSF LSAs, recalculates its OSPF routes, and replaces them with new routes as necessary. Once the restarting router relearns all OSPF routes, it flushes the grace LSAs from the network, informing the helper routers of the completion of the restart process. If the restarting router does not re-establish adjacencies with the helper router within the restart time, the helper router stops the helping function and flushes the stale OSPF routes.

Configuring OSPF Graceful Restart

To configure OSPF Graceful Restart on a router, the restarting router and its directly connected OSPF peers must be configured with Graceful Restart.

```
9408sl(config)#router ospf
9408sl(config-ospf-router)#area 0
9408sl(config-ospf-router)#graceful-restart
```

Syntax: graceful-restart

Enabling and Disabling OSPF Helper

When OSPF is enabled, the helper mode is enabled by default. OSPF routers that do not have graceful restart enabled will act as if the graceful restart helper is enabled. To prevent the graceful restart from performing its function, disable it by entering the following command:

```
9408sl(config-ospf-router)#graceful-restart helper-disable
```

Syntax: [no] graceful-restart helper-disable

Use the **no** form of the command to re-enable the graceful restart helper.

Configuring OSPF Graceful Restart Timer

The OSPF graceful restart timer specifies the maximum amount of time an OSPF restarting router will take to re-establish OSPF adjacencies and relearn OSPF routes. This value will be sent to the neighboring routers in the grace LSA packets. Configure the timer by entering a command such as the following:

```
9408sl(config-ospf-router)#graceful-restart restart-time 120
```

Syntax: graceful-restart restart-time <seconds>

Enter 10 – 1200 for seconds. The default is 120.

Displaying OSPF Graceful Restart Information

Displaying if OSPF Graceful Restart is Enabled

Use the **show ip ospf data grace-link-state** and the **show ip ospf neighbor** commands to display information about OSPF graceful restart.

The following is an example of what the **show ip ospf data grace-link-state** command that is displayed during a restart event. The output is blank if the report is requested while the OSPF router is in normal operation.

```
9408sl#show ip ospf data grace-link-state
    Area  Interface  Router ID    Type  Age      Restart-Time  Seq
     0    3/27       12.1.0.14    9     27        120          0x80000001
     0     v31       12.1.0.14    9     27        120          0x80000001
     0     v32       12.1.0.14    9     27        120          0x80000001
     0     v33       12.1.0.14    9     27        120          0x80000001
     0     v34       12.1.0.14    9     27        120          0x80000001
```

The **show ip ospf neighbor** command displays the following information during normal operation:

```
9408sl#show ip ospf neighbor
Port  Address         Pri State      Neigh Address   Neigh ID        Ev Opt Cnt
3/1   30.1.0.5        0   FULL/OTHER 30.1.0.13       30.0.0.13       5  2   0
3/27  25.27.0.8       1   FULL/DR    25.27.0.14      12.1.0.14       20 2   0
v31   21.23.0.5       1   FULL/DR    21.23.0.14      12.1.0.14       15 2   0
v32   22.24.0.5       1   FULL/DR    22.24.0.14      12.1.0.14       15 2   0
v33   23.25.0.5       1   FULL/DR    23.25.0.14      12.1.0.14       15 2   0
v34   24.26.0.5       1   FULL/DR    24.26.0.14      12.1.0.14       15 2   0
```

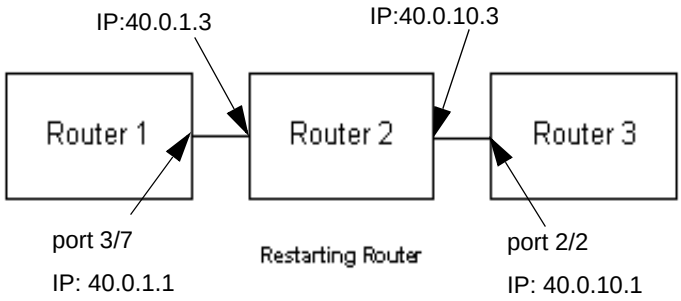
The **show ip ospf neighbor** command displays the following information during a restart event on a helper router. Note the “<in graceful restart state...>” entry appears only during restart. It does not appear once restart is complete.

```
9408sl#sh ip ospf neigh
Port  Address         Pri State      Neigh Address   Neigh ID        Ev Opt Cnt
3/1   30.1.0.5        0   FULL/OTHER 30.1.0.13       30.0.0.13       5  2   0
3/27  25.27.0.8       1   FULL/DR    25.27.0.14      12.1.0.14       20 2   0
  < in graceful restart state,  helping 1, timer 104 sec >
v31   21.23.0.5       1   FULL/DR    21.23.0.14      12.1.0.14       15 2   0
  < in graceful restart state,  helping 1, timer 104 sec >
v32   22.24.0.5       1   FULL/DR    22.24.0.14      12.1.0.14       15 2   0
  < in graceful restart state,  helping 1, timer 104 sec >
v33   23.25.0.5       1   FULL/DR    23.25.0.14      12.1.0.14       15 2   0
  < in graceful restart state,  helping 1, timer 104 sec >
v34   24.26.0.5       1   FULL/DR    24.26.0.14      12.1.0.14       15 2   0
  < in graceful restart state,  helping 1, timer 104 sec >
```

EXAMPLE:

OSPF Graceful Restart requires at least three routers as shown in Figure 1.

Figure 1 Restarting Router Topology



Before configuring graceful restart, use the **show ip ospf neighbor** command to determine the state of the OSPF neighbors. For example,

```
9408sl Router 1# show ip ospf neighbor
Port  Address         Pri State      Neigh Address   Neigh ID        Ev Opt Cnt
3/7   40.0.1.1        1   FULL/DR    40.0.1.3        9.0.1.24        23 2   0
```

Enable graceful restart on each OSPF router in Figure 1. For example,

Router 1

```
9408sl(config)#router ospf
9408sl(config-ospf-router)#graceful-restart
9408sl(config-ospf-router)#area 0
```

Router 2

```
9408sl(config)#router ospf
9408sl(config-ospf-router)#graceful-restart
9408sl(config-ospf-router)#area 0
```

Router 3

```
9408sl(config)#router ospf
9408sl(config-ospf-router)#graceful-restart
9408sl(config-ospf-router)#area 0
```

Use the **show ip ospf neighbor** command to display the state of the OSPF neighbors after enabling graceful restart. For example,

```
9408sl Router 1# show ip ospf neigh

Port  Address         Pri State      Neigh Address   Neigh ID        Ev Opt Cnt
3/7   40.0.1.1        1   EXST/DR    40.0.1.3        9.0.1.24        24 2   0
  < in graceful restart state,  helping 1, timer 112 sec >

9408sl Router 1# sh ip ospf neighbor
Port  Address         Pri State      Neigh Address   Neigh ID        Ev Opt Cnt
2/2   40.0.10.1       1   EXST/DR    40.0.10.3       8.0.0.23        23 2   0
< in graceful restart state,  helping 1, timer 111 sec >
```

Note the “<in graceful restart state...>” entry appears only during restart. It does not appear once restart is complete. The restarting router should resync LSDB with its peers when the restart has completed.

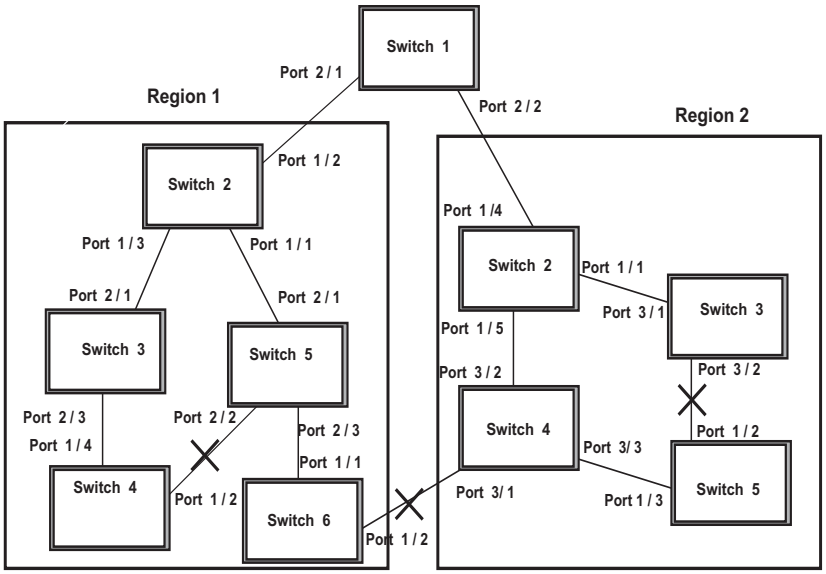
802.1s Multiple Spanning Tree Protocol

Multiple Spanning Tree Protocol (MSTP) as defined in IEEE 802.1s-2002 allows you to configure multiple STP instances. This ensures loop-free topology for 1 or more VLANs. Using MSTP, the entire network runs a common instance of RSTP. Within that common instance, one or more VLANs can be individually configured into distinct regions. The entire network runs the common spanning tree instance (CST) and the regions run a local instance. The local instance is known as Internal Spanning Tree (IST). The CST treats each instance of IST as a single bridge. Consequently, ports are blocked to prevent loops that might occur within an IST and also throughout the CST. In addition, MSTP can coexist with individual devices running STP or RSTP in the Common and Internal Spanning Trees instance (CIST). With the exception of the provisions for multiple instances, MSTP operates exactly like RSTP.

For example, in Figure 2 a network is configured with two regions: Region 1 and Region 2. The entire network is running an instance of CST. Each of the regions is running an instance of IST. In addition, this network contains Switch 1 running RSTP that isn’t configured in a region and consequently is running in the CIST instance. In this configuration, the regions are each regarded as a single bridge to the rest of the network, as is Switch 1. The CST prevents loops from occurring across the network. Consequently, a port is blocked at either port 1/2 of region 1 switch 6, or port 3/1 of region 2 switch 4.

Additionally, loops must be prevented in each of the IST instances. Within the IST Region 1, a port is blocked at port 1/2 of switch 4 to prevent a loop in that region. Within Region 2, a port is blocked at port 3/2 of switch 3 to prevent a loop in that region.

Figure 2 MSTP Configured Network



The following definitions describe the STP instances that define an MSTP configuration:

Common Spanning (CST) – MSTP runs a single instance of spanning tree, called the Common Spanning Tree (CST), across all the bridges in a network. This instance treats each region as a single bridge. It all other ways, it operates exactly like Rapid Spanning Tree (RSTP).

Internal Spanning Tree (IST) – Instances of spanning tree that operate within a defined region are called ISTs (Internal Spanning Tree).

Common and Internal Spanning Trees (CIST) – This is the default MSTP instance 0. It contains all of the ISTs and all bridges that are not formally configured into a region. This instance interoperates with bridges running legacy STP and RSTP implementations.

Multiple Spanning Tree Instance (MSTI) – The MSTI is identified by an MST identifier (MSTid) value between 1 and 4094. This defines an individual instance of an IST. One or more VLANs can be assigned to an MSTI. A VLAN cannot be assigned to multiple MSTIs.

MSTP Region – These are clusters of bridges that run multiple instances of the MSTP protocol. Multiple bridges detect that they are in the same region by exchanging their configuration (instance to VLAN mapping), name, and revision-level. Therefore, if you need to have two bridges in the same region, the two bridges must have identical configurations, names, and revision-levels.

Configuring MSTP

To configure a switch for MSTP, you could configure the name and the revision on each switch that is being configured for MSTP. This name is unique to each switch. You must then create an MSTP Instance and assign an ID. VLANs are then assigned to MSTP instances. These instances must be configured on all switches that interoperate with the same VLAN assignments. Port cost, priority and global parameters can then be configured for individual ports and instances. In addition, operational edge ports and point-to-point links can be created and MSTP can be disabled on individual ports.

Each of the commands used to configure and operate MSTP are described in the following:

- “Setting the MSTP Name”
- “Setting the MSTP Revision Number”
- “Configuring an MSTP Instance”
- “Configuring Port Priority and Port Path Cost”

- “Configuring Bridge Priority for an MSTP Instance”
- “Setting the MSTP Global Parameters”
- “Setting Ports To Be Operational Edge Ports”
- “Setting Point-to-Point Link”
- “Disabling MSTP on a Port”
- “Forcing Ports to Transmit an MSTP BPDU”
- “Enabling MSTP on a Switch”

Setting the MSTP Name

Each switch that is running MSTP is configured with a name. It applies to the switch which can have many different VLANs that can belong to many different MSTP regions. By default, the name is the MAC address of the device.

To configure an MSTP name, use a command such as the following at the Global Configuration level:

```
9408sl(config)# mstp name procurve
```

Syntax: [no] mstp name <name>

The **name** parameter defines an ASCII name for the MSTP configuration. The default name is the MAC address of the switch expressed as a string.

Setting the MSTP Revision Number

Each switch that is running MSTP is configured with a revision number. It applies to the switch which can have many different VLANs that can belong to many different MSTP regions.

To configure an MSTP revision number, use a command such as the following at the Global Configuration level:

```
9408sl(config)# mstp revision 4
```

Syntax: [no] mstp revision <revision-number>

The **revision** parameter specifies the revision level for MSTP that you are configuring on the switch. It can be a number from 0 and 65535.

Configuring an MSTP Instance

An MSTP instance is configured with an MSTP ID for each region. Each region can contain one or more VLANs. To configure an MSTP instance and assign a range of VLANs, use a command such as the following at the Global Configuration level:

```
9408sl(config)# mstp instance 7 vlan 4 to 7
```

Syntax: [no] mstp instance <instance-number> [vlan <vlan-id> | vlan-group <group-id>]

The **instance** parameter defines the number for the instance of MSTP that you are configuring.

The **vlan** parameter assigns one or more VLANs or a range of VLANs to the instance defined in this command.

The **vlan-group** parameter assigns one or more VLAN groups to the instance defined in this command.

Configuring Port Priority and Port Path Cost

Priority and path cost can be configured for a specified instance. To configure an MSTP instance, use a command such as the following at the Global Configuration level:

```
9408sl(config)# mstp instance 7 ethernet 3/1 priority 32 path-cost 200
```

Syntax: [no] mstp instance <instance-number> ethernet <slot/port> priority <port-priority> path-cost <cost>

The <instance-number> variable is the number of the instance of MSTP that you are configuring priority and path cost for.

The **ethernet** <slot/port> parameter specifies a port within a VLAN. The priority and path cost configured with this command will apply to the VLAN that the port is contained within.

You can set a **priority** to the port that gives it forwarding preference over lower priority instances within a VLAN or on the switch. A higher number for the priority variable means a lower forwarding priority. Valid values are 0 - 240 in increments of 16. The default value is 128.

A **path-cost** can be assigned to a port to bias traffic towards or away from a path during periods of rerouting. Possible values are 1 - 200000000.

Configuring Bridge Priority for an MSTP Instance

Priority can be configured for a specified instance. To configure priority for an MSTP instance, use a command such as the following at the Global Configuration level:

```
9408sl(config)# mstp instance 1 priority 8192
```

Syntax: [no] mstp instance <instance-number> priority <priority-value>

The <instance-number> variable is the number for the instance of MSTP that you are configuring.

You can set a **priority** to the instance that gives it forwarding preference over lower priority instances within a VLAN or on the switch. A higher number for the priority variable means a lower forwarding priority. Acceptable values are 0 - 61440 in increments of 4096. The default value is 32768.

Setting the MSTP Global Parameters

MSTP has many of the options available in RSTP as well as some unique options. To configure MSTP Global parameters for all instances on a switch:

```
9408sl(config)# mstp force-version 0 forward-delay 10 hello-time 4 max-age 8 max-
hops 9
```

Syntax: [no] mstp force-version <mode-number> forward-delay <value> hello-time <value> max-age <value> max-hops <value>

The **force-version** parameter forces the bridge to send BPDUs in a specific format. You can specify one of the following <mode-number> values:

- 0 – The STP compatibility mode. Only STP BPDUs will be sent. This is equivalent to single STP.
- 2 – The RSTP compatibility mode. Only RSTP BPDUS will be sent. This is equivalent to single STP.
- 3 – MSTP mode. In this default mode, only MSTP BPDUS will be sent.

The **forward-delay** <value> specifies how long a port waits before it forwards an RST BPDU after a topology change. This can be a value from 4 – 30 seconds. The default is 15 seconds.

The **hello-time** <value> parameter specifies the interval between two hello packets. The parameter can have a value from 1 – 10 seconds. The default is 2 seconds.

The **max-age** <value> parameter specifies the amount of time the device waits to receive a hello packet before it initiates a topology change. You can specify a value from 6 – 40 seconds. The default value is 20 seconds.

The **max-hops** <value> parameter specifies the maximum hop count. You can specify a value from 1 – 40 hops. The default value is 20 hops.

Setting Ports To Be Operational Edge Ports

You can define specific ports as edge ports for the region in which they are configured to connect to devices (such as a host) that are not running STP, RSTP, or MSTP. If a port is connected to an end device such as a PC, the port can be configured as an edge port. To configure ports as operational edge ports enter a command such as the following:

```
9408sl(config)# mstp admin-edge-port ethernet 3/1
```

Syntax: [no] mstp admin-edge-port ethernet <slot/port>

The <slot/port> parameter specifies a port or range of ports as edge ports in the instance they are configured in.

Setting Point-to-Point Link

You can set a point-to-point link between ports to increase the speed of convergence. To create a point-to-point link between ports, use a command such as the following at the Global Configuration level:

```
9408sl(config)# mstp admin-pt2pt-mac ethernet 2/5 ethernet 4/5
```

Syntax: [no] mstp admin-pt2pt-mac ethernet <slot/port>

The <slot/port> parameter specifies a port or range of ports to be configured for point-to-point links to increase the speed of convergence.

Disabling MSTP on a Port

To disable MSTP on a specific port, use a command such as the following at the Global Configuration level:

```
9408sl(config)# mstp disable 2/1
```

Syntax: [no] mstp disable <slot/port>

The <slot/port> variable specifies the location of the port that you want to disable MSTP for.

Forcing Ports to Transmit an MSTP BPDU

To force a port to transmit an MSTP BPDU, use a command such as the following at the Global Configuration level:

```
9408sl(config)# mstp force-migration-check ethernet 3/1
```

Syntax: [no] mstp force-migration-check ethernet <slot/port>

The <slot/port> variable specifies the port or ports that you want to transmit an MSTP BPDU from.

Enabling MSTP on a Switch

To enable MSTP on your switch, use a command such as the following at the Global Configuration level:

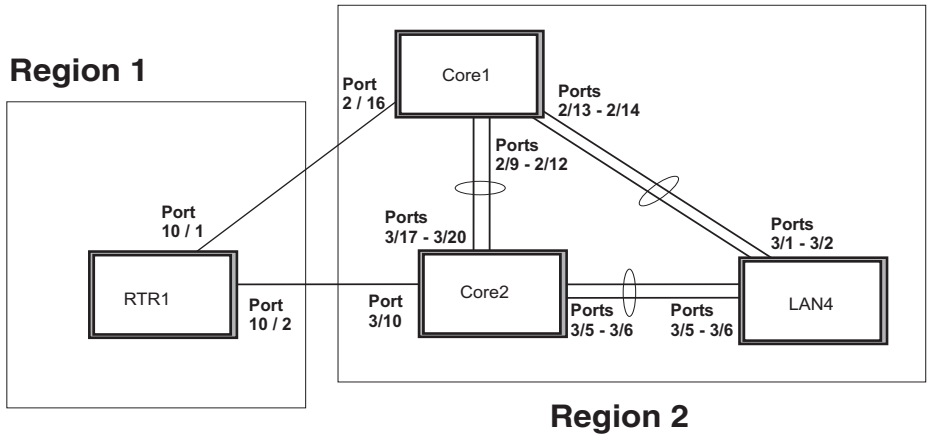
```
9408sl(config)# mstp start
```

Syntax: [no] start

Configuration Example

In Figure 3 four 9408sl switches are configured in two regions. There are four VLANs in four instances in Region 2. Region 1 is in the CIST.

Figure 3 SAMPLE MSTP Configuration



RTR1 Configuration

```
9408sl(config-vlan-4093)tagged ethe 10/1 to 10/2
9408sl(config-vlan-4093)exit
9408sl(config)mstp name Reg1
9408sl(config)mstp revision 1
9408sl(config)mstp instance 0 vlan 4093
9408sl(config)mstp admin-pt2pt-mac ethernet 10/1 to 10/2
9408sl(config)mstp start
```

```
9408sl(config)hostname RTR1
```

Core 1 Configuration

```
9408sl(config)trunk switch ethernet 2/9 to 2/12 ethernet 2/13 to 2/14
9408sl(config)vlan 1 name DEFAULT-VLAN by port
9408sl(config-vlan-1)no spanning-tree
9408sl(config-vlan-1) exit
9408sl(config)vlan 20 by port
9408sl(config-vlan-20)tagged ethernet 2/9 to 2/14 ethernet 2/16
9408sl(config-vlan-20)no spanning-tree
9408sl(config-vlan-20) exit
9408sl(config)vlan 21 by port
9408sl(config-vlan-21)tagged ethernet 2/9 to 2/14 ethernet 2/16
9408sl(config-vlan-21)no spanning-tree
9408sl(config-vlan-21)exit
9408sl(config)vlan 22 by port
9408sl(config-vlan-22)tagged ethernet 2/9 to 2/14 ethernet 2/16
9408sl(config-vlan-22)no spanning-tree
9408sl(config-vlan-22)exit
9408sl(config) mstp name HR
9408sl(config) mstp revision 2
9408sl(config) mstp instance 20 vlan 20
9408sl(config) mstp instance 21 vlan 21
9408sl(config) mstp instance 22 vlan 22
9408sl(config) mstp instance 0 priority 8192
9408sl(config) mstp admin-pt2pt-mac ethernet 2/9 to 2/14
9408sl(config) mstp admin-pt2pt-mac ethernet 2/16
9408sl(config) mstp start
9408sl(config) hostname CORE1
```

Core2 Configuration

```
9408sl(config)trunk switch ethernet 3/5 to 3/6 ethernet 3/17 to 3/20
9408sl(config) vlan 1 name DEFAULT-VLAN by port
9408sl(config-vlan-1)no spanning-tree
9408sl(config-vlan-1) exit
9408sl(config)vlan 20 by port
9408sl(config-vlan-20)tagged ethernet 3/5 to 3/6 ethernet 3/17 to 3/20
9408sl(config-vlan-20)no spanning-tree
9408sl(config-vlan-20)exit
9408sl(config)vlan 21 by port
9408sl(config-vlan-21)tagged ethernet 3/5 to 3/6 ethernet 3/17 to 3/20
9408sl(config-vlan-21)no spanning-tree
9408sl(config-vlan-21)exit
9408sl(config) vlan 22 by port
9408sl(config-vlan-22)tagged ethe 3/5 to 3/6 ethe 3/17 to 3/20
9408sl(config-vlan-22)no spanning-tree
9408sl(config-vlan-22)exit
9408sl(config)mstp name HR
9408sl(config)mstp revision 2
9408sl(config)mstp instance 20 vlan 20
9408sl(config)mstp instance 21 vlan 21
9408sl(config)mstp instance 22 vlan 22
9408sl(config)mstp admin-pt2pt-mac ethernet 3/17 to 3/20 ethernet 3/5 to 3/6
9408sl(config)mstp admin-pt2pt-mac ethernet 3/10
9408sl(config)mstp disable ethe 3/7 ethernet 3/24
9408sl(config)mstp start
9408sl(config)hostname CORE2
```

LAN 4 Configuration

```
9408sl(config) trunk switch ethernet 3/5 to 3/6 ethernet 3/1 to 3/2
9408sl(config)vlan 1 name DEFAULT-VLAN by port
9408sl(config-vlan-1)no spanning-tree
9408sl(config-vlan-1)exit
9408sl(config)vlan 20 by port
9408sl(config-vlan-20)tagged ethernet 3/1 to 3/2 ethernet 3/5 to 3/6
9408sl(config-vlan-20)no spanning-tree
9408sl(config-vlan-20)exit
9408sl(config)vlan 21 by port
9408sl(config-vlan-21)tagged ethernet 3/1 to 3/2 ethe 3/5 to 3/6
9408sl(config-vlan-21)no spanning-tree
9408sl(config-vlan-21)exit
9408sl(config)vlan 22 by port
9408sl(config-vlan-22)tagged ethernet 3/1 to 3/2 ethe 3/5 to 3/6
9408sl(config-vlan-22)no spanning-tree
9408sl(config-vlan-22)exit
9408sl(config)mstp config name HR
9408sl(config)mstp revision 2
9408sl(config)mstp instance 20 vlan 20
9408sl(config)mstp instance 21 vlan 21
9408sl(config)mstp instance 22 vlan 22
9408sl(config)mstp admin-pt2pt-mac ethernet 3/5 to 3/6 ethernet 3/1 to 3/2
9408sl(config)mstp start
9408sl(config)hostname LAN4
```

Displaying MSTP Statistics

MSTP statistics can be displayed using the commands shown below.

To display all general MSTP information, enter the following command:

```
9408sl(config)#show mstp

MSTP Instance 0 (CIST) - VLANs: 1
--------------------------------------------------------------------------------
Bridge           Bridge Bridge Bridge Bridge Root   Root  Root   Root
Identifier       MaxAge Hello  FwdDly Hop    MaxAge Hello FwdDly Hop
hex              sec    sec    sec    cnt    sec    sec   sec    cnt
8000000cdb80af01 20     2      15     20     20     2     15     19

Root             ExtPath   RegionalRoot     IntPath   Designated       Root
Bridge           Cost      Bridge           Cost      Bridge           Port
hex                        hex                        hex
8000000480bb9876 2000      8000000cdb80af01 0         8000000480bb9876 3/1

Port  Pri PortPath  P2P Edge Role       State      Designa-  Designated
Num       Cost      Mac Port                      ted cost  bridge
3/1   128 2000      T   F    ROOT       FORWARDING 0         8000000480bb9876

MSTP Instance 1 - VLANs: 2
--------------------------------------------------------------------------------
Bridge           Max RegionalRoot     IntPath   Designated       Root  Root
Identifier       Hop Bridge           Cost      Bridge           Port  Hop
hex              cnt hex                        hex                    cnt
8001000cdb80af01 20  8001000cdb80af01 0         8001000cdb80af01 Root  20

Port  Pri PortPath  Role       State      Designa-  Designated
Num       Cost                          ted cost  bridge
3/1   128 2000      MASTER     FORWARDING 0         8001000cdb80af01
```

Syntax: show mstp <instance-number>

The <instance-number> variable specifies the MSTP instance that you want to display information for.

Table 20: Output from Show MSTP

This Field...	Displays...
MSTP Instance	The ID of the MSTP instance whose statistics are being displayed. For the CIST, this number is 0.
VLANs:	The number of VLANs that are included in this instance of MSTP. For the CIST this number will always be 1.
Bridge Identifier	The MAC address of the bridge.
Bridge MaxAge sec	Displays configured Max Age.
Bridge Hello sec	Displays configured Hello variable.
Bridge FwdDly sec	Displays configured FwdDly variable.
Bridge Hop cnt	Displays configured Max Hop count variable.
Root MaxAge sec	Max Age configured on the root bridge.
Root Hello sec	Hello interval configured on the root bridge.

Table 20: Output from Show MSTP (Continued)

This Field...	Displays...
Root FwdDly sec	FwdDly interval configured on the root bridge.
Root Hop Cnt	Current hop count from the root bridge.
Root Bridge	Bridge identifier of the root bridge.
ExtPath Cost	The configured path cost on a link connected to this port to an external MSTP region.
Regional Root Bridge	The Regional Root Bridge is the MAC address of the Root Bridge for the local region.
IntPath Cost	The configured path cost on a link connected to this port within the internal MSTP region.
Designated Bridge	The MAC address of the bridge that sent the best BPDU that was received on this port.
Root Port	Port indicating shortest path to root. Set to “Root” if this bridge is the root bridge.
Port Num	The port number of the interface.
Pri	The configured priority of the port. The default is 128.
PortPath Cost	Configured or auto detected path cost for port.
P2P Mac	Indicates if the port is configured with a point-to-point link: • **T** – The port is configured in a point-to-point link • **F** – The port is not configured in a point-to-point link
Edge	Indicates if the port is configured as an operational edge port: • **T** – indicates that the port is defined as an edge port. • **F** – indicates that the port is not defined as an edge port
Role	The current role of the port: • Master • Root • Designated • Alternate • Backup • Disabled
State	The port’s current 802.1w state. A port can have one of the following states: • Forwarding • Discarding • Learning • Disabled
Designated Cost	Port path cost to the root bridge.

Table 20: Output from Show MSTP (Continued)

This Field...	Displays...
Max Hop cnt	The maximum hop count configured for this instance.
Root Hop cnt	Hop count from the root bridge.

Displaying MSTP Information for a Specified Instance

The following example displays MSTP information specified for an MSTP instance.

```
9408sl(config)#show mstp 1

MSTP Instance 1 - VLANs: 2
--------------------------------------------------------------------------------
Bridge           Max RegionalRoot     IntPath   Designated       Root  Root
Identifier       Hop Bridge           Cost      Bridge           Port  Hop
hex              cnt hex                        hex                    cnt
8001000cdb80af01 20  8001000cdb80af01 0         8001000cdb80af01 Root  20

Port  Pri PortPath  Role       State      Designa-  Designated
Num       Cost                          ted cost  bridge
3/1   128 2000      MASTER     FORWARDING 0         8001000cdb80af01
```

See Table 20 for details about the display parameters.

Displaying MSTP Information for CIST Instance 0

Instance 0 is the Internal Spanning Tree Instance (IST). When you display information for this instance there are some differences with displaying other instances. The following example displays MSTP information for CIST Instance 0.

```
9408sl(config)#show mstp 0

MSTP Instance 0 (CIST) - VLANs: 1
--------------------------------------------------------------------------------
Bridge           Bridge Bridge Bridge Bridge Root   Root  Root   Root
Identifier       MaxAge Hello  FwdDly Hop    MaxAge Hello FwdDly Hop
hex              sec    sec    sec    cnt    sec    sec   sec    cnt
8000000cdb80af01 20     2      15     20     20     2     15     19

Root             ExtPath   RegionalRoot     IntPath   Designated       Root
Bridge           Cost      Bridge           Cost      Bridge           Port
hex                        hex                        hex
8000000480bb9876 2000      8000000cdb80af01 0         8000000480bb9876 3/1

Port  Pri PortPath  P2P Edge Role       State      Designa-  Designated
Num       Cost      Mac Port                      ted cost  bridge
3/1   128 2000      T   F    ROOT       FORWARDING 0         8000000480bb9876
```

To display details about the MSTP configuration, enter the following command:

```
9408sl(config)#show mstp conf

MSTP CONFIGURATION
------------------
Name     : Reg1
Revision : 1
Version  : 3 (MSTP mode)
Status   : Started

Instance VLANs
-------- ----------------------------------------------------
0        4093
```

To display details about the MSTP that is configured on the device, enter the following command:

```
9408sl(config)#show mstp detail
MSTP Instance 0 (CIST) - VLANs: 4093
--------------------------------------------------------------------------------
Bridge: 800000b000c00000 [Priority 32768, SysId 0, Mac 00b000c00000]
FwdDelay 15, HelloTime 2, MaxHops 20, TxHoldCount 6

Port 6/54 - Role: DESIGNATED - State: FORWARDING
PathCost 20000, Priority 128, OperEdge T, OperPt2PtMac F, Boundary T
Designated - Root 800000b000c00000, RegionalRoot 800000b000c00000,
Bridge 800000b000c00000, ExtCost 0, IntCost 0
ActiveTimers - helloWhen 1
MachineState - PRX-DISCARD, PTX-IDLE, PPM-SENDING_RSTP, PIM-CURRENT
PRT-ACTIVE_PORT, PST-FORWARDING, TCM-INACTIVE
BPDUs - Rcvd MST 0, RST 0, Config 0, TCN 0
Sent MST 6, RST 0, Config 0, TCN 0
```

See Table 20 for explanation about the parameters in the output.

Syntax: show mstp [<mstp-id> | configuration | detail] [| begin <string> | exclude <string> | include <string>]

Enter an MSTP ID for <mstp-id>.

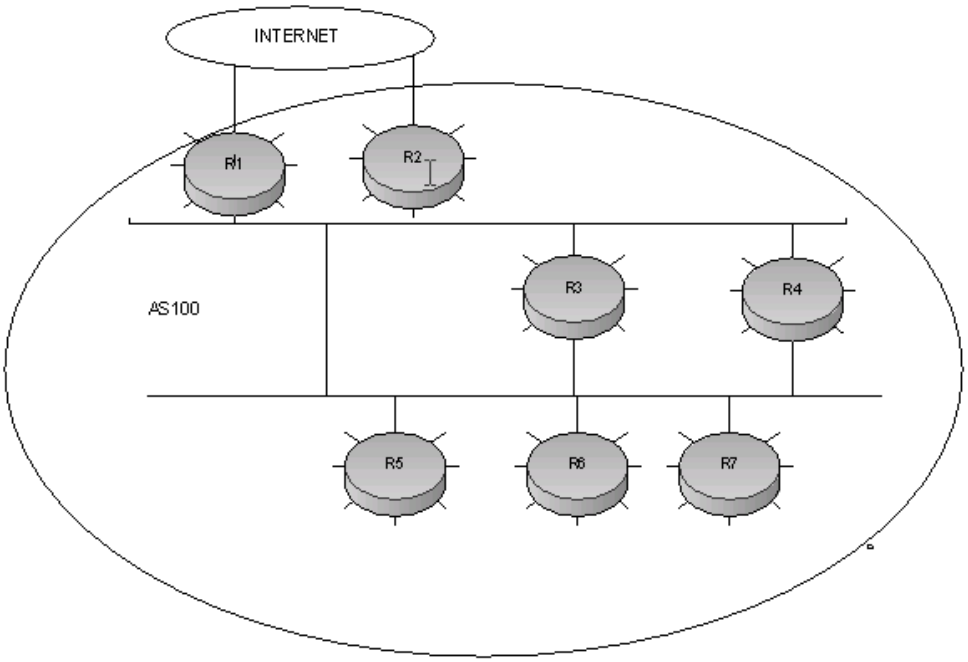
BGP Null0 Routing

In previous releases, null0 routes were treated as invalid routes for BGP next hop resolution. With software release 02.2.01, BGP can use the null0 route to resolve its next hop. Thus, null0 route in the routing table (for example, static route) is considered as a valid route by BGP. If the next hop for BGP resolves into a null0 route, the BGP route is also installed as a null0 route in the routing table.

The null0 routing feature allows network administrators to block certain network prefixes, by using null0 routes and route-maps. The combined use of null0 routes and route maps blocks traffic from a particular network prefix, telling a remote router to drop all traffic for this network prefix by redistributing a null0 route into BGP.

Figure 4 shows a topology for a null0 routing application example.

Figure 4 SAMPLE Null0 Routing Application



The following steps configure a null0 routing application for stopping denial of service attacks from remote host(s) on the internet.

Configuration Steps

1. Select one router, Router 6, to distribute null0 routes throughout the BGP network.
2. Configure a route-map to match a particular tag (50) and set the next-hop address to an unused network address (199.199.1.1).
3. Set the local-preference to a value higher than any possible internal/external local-preference (50).
4. Complete the route map by setting origin to IGP.
5. On Router 6, redistribute the static routes into BGP, using route-map <route-map-name> (redistribute static route-map block user).
6. On Router 1, the router facing the internet, configure a null0 route matching the next-hop address in the route-map (ip route 199.199.1.1/32 null0).
7. Repeat step 3 for all routers interfacing with the internet (edge corporate routers). In this case, Router 2 has the same null0 route as Router 1.
8. On Router 6, configure the network prefixes associated with the traffic you want to drop. The static route IP address references a destination address. You are required to point the static route to the egress port, for example, Ethernet 3/7, and specify the tag 50, matching the route-map configuration.

Configuration Examples

Router 6

The following configuration defines specific prefixes to filter:

```
9408sl(config)#ip route 110.0.0.40/29 ethernet 3/7 tag 50
9408sl(config)#ip route 115.0.0.192/27 ethernet 3/7 tag 50
9408sl(config)#ip route 120.0.14.0/23 ethernet 3/7 tag 50
```

The following configuration redistributes routes into BGP:

```
9408sl(config)#router bgp

9408sl(config-bgp-router)#local-as 100
9408sl(config-bgp-router)#neighbor <router1_int_ip address> remote-as 100
9408sl(config-bgp-router)#neighbor <router2_int_ip address> remote-as 100
9408sl(config-bgp-router)#neighbor <router3_int_ip address> remote-as 100
9408sl(config-bgp-router)#neighbor <router4_int_ip address> remote-as 100
9408sl(config-bgp-router)#neighbor <router5_int_ip address> remote-as 100
9408sl(config-bgp-router)#neighbor <router7_int_ip address> remote-as 100
9408sl(config-bgp-router)#redistribute static route-map blockuser
9408sl(config-bgp-router)#exit
```

The following configuration defines the specific next hop address and sets the local preference to preferred:

```
9408sl(config)#route-map blockuser permit 10
9408sl(config-routemap blockuser)#match tag 50
9408sl(config-routemap blockuser)#set ip next-hop 199.199.1.1
9408sl(config-routemap blockuser)#set local-preference 1000000
9408sl(config-routemap blockuser)#set origin igp
9408sl(config-routemap blockuser)#exit
```

Router 1

The following configuration defines the null0 route to the specific next hop address. The next hop address 199.199.1.1 points to 128.178.1.101, which gets blocked:

```
9408sl(config)# ip route 199.199.1.1/32 null0

9408sl(config)#router bgp
9408sl(config-bgp-router)#local-as 100
9408sl(config-bgp-router)#neighbor <router2_int_ip address> remote-as 100
9408sl(config-bgp-router)#neighbor <router3_int_ip address> remote-as 100
9408sl(config-bgp-router)#neighbor <router4_int_ip address> remote-as 100
9408sl(config-bgp-router)#neighbor <router5_int_ip address> remote-as 100
9408sl(config-bgp-router)#neighbor <router6_int_ip address> remote-as 100
9408sl(config-bgp-router)#neighbor <router7_int_ip address> remote-as 100
```

Router 2

The following configuration defines a null0 route to the specific next hop address. The next hop address 199.199.1.1 points to 128.178.1.101, which gets blocked:

```
9408sl(config)#ip route 199.199.1.1/32 null0
9408sl(config)#router bgp
9408sl(config-bgp-router)#local-as 100
9408sl(config-bgp-router)#neighbor <router1_int_ip address> remote-as 100
9408sl(config-bgp-router)#neighbor <router3_int_ip address> remote-as 100
9408sl(config-bgp-router)#neighbor <router4_int_ip address> remote-as 100
9408sl(config-bgp-router)#neighbor <router5_int_ip address> remote-as 100
9408sl(config-bgp-router)#neighbor <router6_int_ip address> remote-as 100
9408sl(config-bgp-router)#neighbor <router7_int_ip address> remote-as 100
```

Show Commands

After configuring the null0 application, you can display the output:

Router 6

Show ip route static output for Router 6:

```
9408sl# show ip route static

Type Codes - B:BGP  D:Connected  S:Static  R:RIP  O:OSPF;   Cost - Dist/Metric
          Destination          Gateway         Port        Cost       Type
1       110.0.0.40/29          DIRECT          eth 3/7     1/1        S
2       115.0.0.192/27         DIRECT          eth 3/7     1/1        S
3       120.0.14.0/23          DIRECT          eth 3/7     1/1        S
9408sl#
```

Router 1 and 2

Show ip route static output for Router 1 and Router 2:

```
9408sl# show ip route static

 Type Codes - B:BGP  D:Connected  S:Static  R:RIP  O:OSPF;   Cost - Dist/Metric
             Destination        Gateway         Port        Cost       Type
1            199.199.1.1/32     DIRECT          drop        1/1        S
9408sl#
```

Router 6

Show BGP routing table output for Router-6

```
Router-6#show ip bgp route

Total number of BGP Routes: 126
Status A:AGGREGATE B:BEST b:NOT-INSTALLED-BEST C:CONFED_EBGP D:DAMPED  E:EBGP
H:HISTORY I:IBGP L:LOCAL M:MULTIPATH S:SUPPRESSED s:STALE
       Prefix             Next Hop        Metric     LocPrf     Weight Status
1      30.0.1.0/24        40.0.1.3        0          100        0      BI
         AS_PATH:
.             ..                        .                      .          .      .
9      110.0.0.16/30      90.0.1.3                   100        0      I
         AS_PATH: 85
10     110.0.0.40/29      199.199.1.1     1        1000000 32768  BL
         AS_PATH:
11     110.0.0.80/28      90.0.1.3                   100        0      I
 .           ..                        .                      .          .       .
 .           ..                        .                      .          .       .
36     115.0.0.96/28      30.0.1.3                   100        0      I
         AS_PATH: 50
37     115.0.0.192/27     199.199.1.1     1       10000000 32768  BL
         AS_PATH:
 .             ..                      .                      .          .      .
64     120.0.7.0/24       70.0.1.3                   100        0      I
         AS_PATH: 10
65     120.0.14.0/23      199.199.1.1     1        1000000 32768  BL
         AS_PATH: ..                   .                      .          .      .
```

Router 1 and 2

The **show ip route** output for Router 1 and Router 2 shows “drop” under the Port column for the network prefixes you configured with null0 routing:

```
9408sl#show ip route

Total number of IP routes: 133
 Type Codes - B:BGP  D:Connected  S:Static  R:RIP  O:OSPF;   Cost - Dist/Metric
        Destination        Gateway         Port        Cost       Type
1       9.0.1.24/32        DIRECT          loopback 1  0/0    D
2       30.0.1.0/24        DIRECT          eth 2/7     0/0        D
3       40.0.1.0/24        DIRECT          eth 2/1     0/0        D
.
13      110.0.0.6/31       90.0.1.3        eth 2/2     20/1       B
14      110.0.0.16/30      90.0.1.3        eth 2/2     20/1       B
15      110.0.0.40/29      DIRECT          drop        200/0      B
.          ..                .              .          .          .
42      115.0.0.192/27     DIRECT          drop        200/0      B
43      115.0.1.128/26     30.0.1.3        eth 2/7     20/1       B
.          ..                .              .          .          .
69      120.0.7.0/24       70.0.1.3        eth 2/10    20/1       B
70      120.0.14.0/23      DIRECT          drop        200/0      B
.          ..                .              .          .          .
.          ..                .              .          .          .
131     130.144.0.0/12     80.0.1.3        eth 3/4     20/1       B
132    199.199.1.1/32     DIRECT          drop        1/1        S
9408sl#
```

Port Security MAC Deny

With software release 02.2.01, you can configure a new mode for port security, called the *violation deny mode*. The violation deny mode allows you to deny MAC addresses on a global level or on a per port level.

Global Deny Configuration

The following configuration example shows how to deny a MAC address on a VLAN.

```
9408sl(config)# port security
9408sl(config-port-security)# violation deny
9408sl(config-port-security)# deny-mac-address 0000.0000.0001 2
```

Syntax: [no] violation deny

Syntax: [no] deny-mac-address <MAC-address>

Global denied secure MAC addresses are denied system wide. These MAC entries are added to the MAC table as deny entries, when a flow is received. Only the configured MAC addresses are denied. All other MAC addresses are allowed.

A maximum of 512 deny MAC addresses can be configured on a global level.

Interface deny configuration

MAC addresses can be configured to be denied on an interface.

The following configuration example shows how to deny a MAC address on an interface.

```
9408sl(config)# int e 7/11
9408sl(config-if-e100-7/11)# port security
9408sl(config-port-security-e100-7/11)# violation deny
9408sl(config-port-security-e100-7/11)# deny-mac-addr 0000.1111.2222 4
```

Only the configured MAC addresses are denied. All other MAC addresses are allowed.

A maximum of 64 deny MAC addresses can be configured at an interface level.

Syntax: [no] violation deny

Syntax: [no] deny-mac-address <MAC-address>

Display Commands

The **show port security global-deny** command lists all the configured global deny MAC addresses:

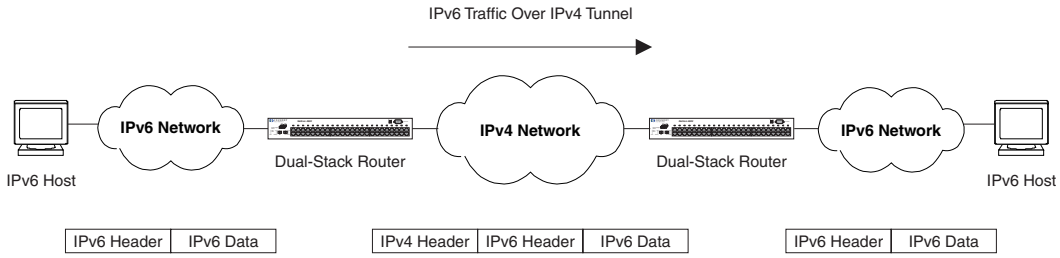
The **show port security denied mac** command lists all the denied MAC addresses in the system.

All other port security commands remain the same.

IPv6 Over IPv4 Tunnels

To enable communication between the isolated IPv6 domains using the IPv4 infrastructure, you can configure IPv6 over IPv4 tunnels. As shown in Figure 5, these tunnels encapsulate an IPv6 packet within an IPv4 packet.

Figure 5 IPv6 over an IPv4 tunnel



ProCurve supports the following IPv6 over IPv4 tunneling mechanisms:

- Manually configured tunnels
- Automatic 6to4 tunnels
- Automatic IPv4-compatible IPv6 tunnels

In general, a manually configured tunnel establishes a permanent link between routers in IPv6 domains, while the automatic tunnels establish a transient link that is created and taken down on an as-needed basis. (Although the feature name and description may imply otherwise, some configuration is necessary to set up an automatic tunnel.) Also, a manually configured tunnel has explicitly configured IPv4 addresses for the tunnel source and destination, while the automatic tunnels have an explicitly configured IPv4 address for the tunnel source and an automatically generated address for the tunnel destination.

NOTE: ProCurve's implementation of IPv6 supports automatic IPv4-compatible IPv6 tunnels. However, because of this tunneling mechanism's inherent dependence on IPv4 addresses, which diminishes the benefits of IPv6, ProCurve recommends using either manually configured tunnels or automatic 6to4 tunnels instead.

These tunneling mechanisms require that the router at each end of the tunnel run both IPv4 and IPv6 protocol stacks. (For information about configuring IPv4 and IPv6 protocol stacks on a router interface, see *IPv6 Configuration Guide for the ProCurve 9408sl Routing Switch*.) The routers running both protocol stacks, or ***dual-stack routers***, can interoperate directly with both IPv4 and IPv6 end systems and routers.

Configuring a Manual IPv6 Tunnel

You can use a manually configured tunnel to connect two isolated IPv6 domains. You should deploy this point-to-point tunnel mechanism if you need a permanent and stable connection.

To configure a manual IPv6 tunnel, enter commands such as the following on a Layer 3 Switch running both IPv4 and IPv6 protocol stacks on each end of the tunnel:

```
9408sl(config)# interface tunnel 1
```

```
9408sl(config-tnif-1)#tunnel source ethernet 3/1
9408sl(config-tnif-1)#tunnel destination 198.162.100.1
9408sl(config-tnif-1)#tunnel mode ipv6ip
9408sl(config-tnif-1)#ipv6 address 2001:b78:384d:34::/64 eui-64
```

This example creates tunnel interface 1 and assigns a global IPv6 address with an automatically computed EUI-64 interface ID to it. The IPv4 address assigned to Ethernet interface 3/1 is used as the tunnel source, while the IPv4 address 192.168.100.1 is configured as the tunnel destination. Finally, the tunnel mode is specified as a manual IPv6 tunnel.

Syntax: interface tunnel <number>

For the <number> parameter, specify a value from 1 – 32.

Syntax: ipv6 address <ipv6-prefix>/<prefix-length> [eui-64]

You must specify the <ipv6-prefix> parameter in hexadecimal using 16-bit values between colons as documented in RFC 2373.

You must specify the <prefix-length> parameter as a decimal value. A slash mark (/) must follow the <ipv6-prefix> parameter and keyword configures the global or site-local address with an EUI-64 interface ID in the low-order 64 bits. The interface ID is automatically constructed in IEEE EUI-64 format using the interface's MAC address.

Syntax: tunnel source <ipv4-address> | ethernet <port> | loopback <number> | ve <number>

You must specify the <ipv4-address> parameter using 8-bit values in dotted decimal notation.

NOTE: Starting with software release 02.2.01 for the 9408sl, the **tunnel** parameter for the **tunnel** <number> option is no longer available.

The **ethernet | loopback | ve** parameter specifies an interface as the tunnel source. If you specify an Ethernet interface, also specify the port number associated with the interface. If you specify a **loopback**, virtual routing interface (**ve**), or **interface**, you must also specify the loopback, ve, or interface number, respectively.

Syntax: tunnel destination <ipv4-address>

You must specify the <ipv4-address> parameter using 8-bit values in dotted decimal notation.

Syntax: tunnel mode ipv6ip

Clearing IPv6 Tunnel Statistics

You can clear all IPv6 tunnel statistics (reset all fields to zero) or statistics for a specified tunnel interface.

For example, to clear statistics for tunnel 1, enter the following command at the Privileged EXEC level or any of the Config levels of the CLI:

```
9408sl# clear ipv6 tunnel 1
```

Syntax: clear ipv6 tunnel <number>

The <number> parameter specifies the tunnel number.

Displaying IPv6 Tunnel Information

To display a summary of tunnel information, enter the following command at any level of the CLI:

```
9408sl# show ipv6 tunnel
IP6 Tunnels
  Tunnel  Mode         Packet Received  Packet Sent
  1       configured   0                0
  2       configured   0                22419
```

Syntax: show ipv6 tunnel

This display shows the following information.

Table 21: IPv6 tunnel information

This Field...	Displays...
Tunnel	The tunnel interface number.
Mode	The tunnel mode. Possible modes include the following: • configured – Indicates a manually configured tunnel.
Packet Received	The number of packets received by a tunnel interface.
Packet Sent	The number of packets sent by a tunnel interface.

Displaying Tunnel Interface Information

For example, to display status and configuration information for tunnel interface 1, enter the following command at any level of the CLI:

```
9408sl# show interfaces tunnel 1
Tunnel1 is up, line protocol is up
  Hardware is Tunnel
  Tunnel source ethernet 3/5 (10.10.10.93)
  Tunnel destination is 10.10.10.77
  Tunnel mode ipv6ip
  No port name
  MTU 1500 bytes
```

Syntax: show interfaces tunnel <number>

The <number> parameter indicates the tunnel interface number for which you want to display information.

The resulting display shows the following information.

Table 22: IPv6 tunnel interface information

This Field...	Displays...
Tunnel interface status	The status of the tunnel interface can be one of the following: • up – The tunnel interface is functioning properly. • down – The tunnel interface is not functioning and is down.
Line protocol status	The status of the line protocol can be one of the following: • up – The line protocol is functioning properly. • down – The line protocol is not functioning and is down.
Hardware is tunnel	The interface is a tunnel interface.
Tunnel source	The tunnel source can be one of the following: • An IPv4 address • The IPv4 address associated with an interface/port.
Tunnel destination	The tunnel destination can be an IPv4 address.

Table 22: IPv6 tunnel interface information (Continued)

This Field...	Displays...
Tunnel mode	The tunnel mode is: • ipvbip - Indicates a manual tunnel
Port name	The port name configured for the tunnel interface.
MTU	The setting of the IPv6 maximum transmission unit (MTU).

Displaying Interface Level IPv6 Settings

To display Interface level IPv6 settings for tunnel interface 1, enter the following command at any level of the CLI:

```
9408sl# show ipv6 inter tunnel 1
Interface Tunnel 1 is up, line protocol is up
  IPv6 is enabled, link-local address is fe80::3:4:2 [Preferred]
  Global unicast address(es):
    1001::1 [Preferred],  subnet is 1001::/64
    1011::1 [Preferred],  subnet is 1011::/64
  Joined group address(es):
    ff02::1:ff04:2
    ff02::5
    ff02::1:ff00:1
    ff02::2
    ff02::1
  MTU is 1480 bytes
  ICMP redirects are enabled
  No Inbound Access List Set
  No Outbound Access List Set
  OSPF enabled
```

The display command above reflects the following configuration:

```
9408sl# show running-config interface tunnel 1

!
interface tunnel 1
 port-name ManualTunnel1
 tunnel mode ipv6ip
 tunnel source loopback 1
 tunnel destination 2.1.1.1
 ipv6 address fe80::3:4:2 link-local
 ipv6 address 1011::1/64
 ipv6 address 1001::1/64
 ipv6 ospf area 0
```

Configuring Egress Priority Merging

Starting with release 02.2.01, egress priority merging is disabled by default; in earlier releases the feature is enabled by default.

In earlier releases, the ProCurve internal priority (calculated on ingress) is merged with the incoming VLAN tag priority at the egress on every port. This means that the highest value of these two priorities is used to set the outgoing packet priority.

The problem with this approach is that if an inbound ACL downgrades the priority (it does this by setting the internal priority to a lower value) it will not be honored for tagged packets, since the egress priority merge will override what the ACL attempted to do.

To turn on the egress priority merge on a per-port basis enter the following command:

```
9408sl(config)# interface ethernet 1/2
9408sl(config-if-e10000-1/2)# merge-egress-priority
```

This will enable egress priority merging on the interface.

Syntax: [no] merge-priority

IP Receive Access List

The *IP receive access list* feature uses IPv4 ACLs to filter the packets intended for the management process to protect the management module from being overloaded with heavy traffic that was sent to one of the Layer 3 Switch IP interfaces. The feature applies to IPv4 unicast and multicast packets.

Configuring IP Receive Access List

IP receive access list is a global configuration command. Once it is applied, the command will be effective on all the management modules on the device. To configure the feature, do the following:

1. Create a numbered ACL that will be used as the IP receive ACL. This ACL can be a standard (1– 99) or extended (100–199) ACL. Named ACLs are not supported.

   For example,

   ```
   9408sl(config)# access-list 10 deny host 209.157.22.26 log
   9408sl(config)# access-list 10 deny 209.157.29.12 log
   9408sl(config)# access-list 10 deny host IPHost1 log
   9408sl(config)# access-list 10 permit any
   9408sl(config)# write memory
   ```

2. Configure ACL 10 as the IP receive access list by entering the following command:

   ```
   9408sl(config)# ip receive access-list 10
   ```

   Syntax: [no] `ip receive access-list <num>`

   Specify an access list number for <num>.

   The IP receive ACL is applied globally to all interfaces on the device.

Displaying IP Receive Access List

To determine if IP receive access list has been configured on the device, enter the following command:

```
9408sl# show access-list bindings
L4 configuration:

ip receive access-list 101
```

New Rule for Creating Trunks

Release 02.0.02 introduced rules for creating trunks on the 9408sl. In that release, you can create trunks from multiple interface modules of either 2, 4, or 8 ports. Within that limit, not all combinations of ports can form a trunk. See the “Configuring Trunk Groups and Dynamic Link Aggregation” chapter in the *Installation and Basic Configuration Guide for ProCurve 9300 Series Routing Switches*.

In software release 02.2.01, trunk rules have been simplified. Trunks can now be formed from any number of ports, as long as they contain at least 2 ports and no more than 8 ports. Also, ports in a trunk must have the same speed and the same negotiation mode.

OSPF Point-to-Point Links

In an OSPF point-to-point network, where a direct Layer 3 connection exists between a single pair of OSPF routers, there is no need for Designated and Backup Designated Routers, as is the case in OSPF multi-access networks. Without the need for Designated and Backup Designated routers, a point-to-point network establishes adjacency and converges faster. The neighboring routers become adjacent whenever they can communicate directly. In contrast, in broadcast and non-broadcast multi-access (NBMA) networks, the Designated Router and Backup Designated Router become adjacent to all other routers attached to the network.

Configuration Notes and Limitations

- This feature is supported in 9408sl software releases 02.2.01 and later.
- This feature is supported on Gigabit Ethernet and 10-Gigabit Ethernet interfaces.
- This feature is supported on physical interfaces. It is not supported on virtual interfaces.
- ProCurve supports numbered point-to-point networks, meaning the OSPF router must have an IP interface address which uniquely identifies the router over the network. ProCurve does not support unnumbered point-to-point networks.

Configuring an OSPF Point-to-Point Link

To configure an OSPF point-to-point link, enter commands such as the following:

```
9408sl(config)# interface eth 1/5
9408sl(config-if-1/5)# ip ospf network point-to-point
```

This command configures an OSPF point-to-point link on Interface 5 in slot 1.

Syntax: [no] ip ospf network point-to-point

Viewing Configured OSPF Point-to-Point Links

You can use the show ip ospf interface command to display OSPF point-to-point information. Enter the following command at any CLI level:

:

```
9408sl# show ip ospf interface 192.168.1.1

Ethernet 2/1,OSPF enabled
     IP Address 192.168.1.1, Area 0
     OSPF state ptr2ptr, Pri 1, Cost 1, Options 2, Type pt-2-pt Events 1
     Timers(sec): Transit 1, Retrans 5, Hello 10, Dead 40
     DR:  Router ID 0.0.0.0           Interface Address 0.0.0.0
     BDR: Router ID 0.0.0.0           Interface Address 0.0.0.0
     Neighbor Count = 0, Adjacent Neighbor Count= 1
     Neighbor: 2.2.2.2
     Authentication-Key:None
     MD5 Authentication: Key None, Key-Id None, Auth-change-wait-time 300
```

Syntax: show ip ospf interface [<ip-addr>]

The <ip-addr> parameter displays the OSPF interface information for the specified IP address.

The following table defines the highlighted fields shown in the above example output of the **show ip ospf interface** command.

Table 23: Output of the show ip ospf interface command

This field	Displays
IP Address	The IP address of the interface.
OSPF state	ptr2ptr (point to point)
Pri	The link ID as defined in the router-LSA. This value can be one of the following: 1 = point-to-point link 3 = point-to-point link with an assigned subnet

Table 23: Output of the show ip ospf interface command

This field	Displays
Cost	The configured output cost for the interface.
Options	OSPF Options (Bit7 - Bit0): • unused:1 • opaque:1 • summary:1 • dont_propagate:1 • nssa:1 • multicast:1 • externals:1 • tos:1
Type	The area type, which can be one of the following: • Broadcast = 0x01 • NBMA = 0x02 • Point to Point = 0x03 • Virtual Link = 0x04 • Point to Multipoint = 0x05
Events	OSPF Interface Event: • Interface_Up = 0x00 • Wait_Timer = 0x01 • Backup_Seen = 0x02 • Neighbor_Change = 0x03 • Loop_Indication = 0x04 • Unloop_Indication = 0x05 • Interface_Down = 0x06 • Interface_Passive = 0x07
Adjacent Neighbor Count	The number of adjacent neighbor routers.
Neighbor:	The neighbor router's ID.

IP Fragmentation Protection

Beginning with software release 02.2.01, IP packet filters on the 9408sl will drop undersized fragments and overlapping packet fragments to prevent tiny fragment attacks as explained in RFC 1858. When packets are fragmented on the network, the first fragment of a packet must be large enough to contain all the necessary header information, and fragments, once reassembled, must meet certain criteria before they are allowed to pass through the network. There are no CLI commands for this new security feature.

IP Option Attack Protection

An attack on the network could be accomplished using the options field of an IP packet header. For example, the source routing option makes it possible for the sender to specify a route to follow.

To protect against attacks contained in the option field, 9408sl devices drop any IP packet that contains an option in its header, except for IGMP packets. IGMP packets are processed even if they contain IP options. If you want other packets that contain options in their headers to be processed, enter a command such as the following:

```
9408sl(config)#ip ip-option-process
```

Syntax: ip ip-option-process

Static Route Tagging

Static routes can be configured with a tag value, which can be used to color routes and filter routes during a redistribution process. When tagged static routes are redistributed to OSPF or to a protocol that can carry tag information, they are redistributed with their tag values.

To add a tag value to a static route, enter commands such as the following:

```
9408sl(config)#ip route 192.122.12.1 255.255.255.0 192.122.1.1 tag 20
```

Syntax: ip route <dest-ip-addr> <dest-mask> | <dest-ip-addr>/<dest-mask> <next-hop-ip-address> tag <value>

The <dest-ip-addr> is the route's destination. The <dest-mask> is the network mask for the route's destination IP address. Alternatively, you can specify the network mask information by entering a forward slash followed by the number of bits in the network mask. For example, you can enter 192.0.0.0 255.255.255.0 as 192.0.0.0/24. You can enter multiple static routes for the same destination for load balancing or redundancy.

The <next-hop-ip-address> is the IP address of the next-hop router (gateway) for the route. In addition, the <next-hop-ip-address> can also be a virtual routing interface (for example. ve 100), or a physical port (for example, ethernet 1/1) that is connected to the next hop router.

Enter 0 – 4294967295 for **tag** <value>. The default is 0, meaning no tag.

MTU for IPv4 and IPv6

In previous releases, the maximum value of MTU was 1500 bytes. Beginning with software release 02.2.01, you can configure IP MTU for IPv4 and IPv6 to be greater than 1500 bytes, although the default remains at 1500 bytes.

The value of the MTU you can define depends on the following:

- For a physical port, the maximum value of the MTU is the equal to the maximum frame size of the port minus 18 (Layer 2 MAC header + CRC).
- For a virtual routing interface, the maximum value of the MTU is the maximum frame size configured for the VLAN to which it is associated, minus 18 (Layer 2 MAC header + CRC). If a maximum frame size for a VLAN is not configured, then configure the MTU based on the smallest maximum frame size of all the ports of the VLAN that corresponds to the virtual routing interface, minus 18 (Layer 2 MAC header + CRC).

You can define MTU globally and per interface.

To define IPv4 MTU globally, enter a command such as the following:

```
9408sl(config)#ip mtu 1000
```

To define IPv4 MTU on an interface, enter the following command:

```
9408sl(config)#interface ethernet 1/3
9408sl(config-if-e10000-1/3)ip mtu 1000
```

Syntax: ip mtu <value>

To define IPv6 MTU globally, enter:

```
9408sl(config)#ipv6 mtu 1300
```

To define IPv6 MTU on an interface, enter:

```
9408sl(config-if-e1000-2/1)#ipv6 mtu
```

Syntax: ipv6 mtu <value>

NOTE: if a jumbo packet is received on a port whose maximum frame size - 18 (Layer 2 MAC header + CRC) is greater than the outgoing port's IPv4/IPv6 MTU, then it will be forwarded in the CPU.

Enhancement to the LACP system-priority Command

In previous releases, the LACP **system-priority** command was available at the interface level. For example:

```
9408sl(config-if-e1000-4/3)#link-aggregate configure
  key
  port-priority
  system-priority
```

Starting with software release 02.2.01, this command was moved to global configuration level:

```
9408sl(config)#lacp system-priority
```

Enhancement to the ip ssh rsa-authentication Command

The syntax for the **ip ssh rsa-authentication no** command has been changed to make the command more generic. The syntax for the enhanced command is **ip ssh key-authentication no**.

Neighbor Local-AS Feature

Software release 02.2.01 introduces the Neighbor Local Autonomous System (AS) feature. This feature allows a router that is a member of one AS to appear to also be a member of another AS. This feature is useful, for example, if Company A purchases Company B, but Company B does not want to modify it peering configurations.

This feature can only be used for true EBGP peers. When establishing a BGP connection, the router will use the configured neighbor local AS, instead of the system AS number.

For example, if you want a router to use AS 200, instead of 100 when peering with neighbor 11.11.11.2, enter commands such as the following:

```
9408sl(config)#router bgp
9408sl(config-bgp-router)#local-as 100
9408sl(config-bgp-router)#graceful-restart restart-time 30
9408sl(config-bgp-router)#graceful-restart
9408sl(config-bgp-router)#neighbor 11.11.11.2 remote-as 101
9408sl(config-bgp-router)#neighbor 11.11.11.2 local-as 200
```

Syntax: [no] neighbor <ip-address> local-as <local-as-number>

Enter the IP address of the neighbor with which the device will be peering for <ip-address>.

Enhancements and Configuration Notes in 02.3.00

ACL Logging of Denied Packets

In previous releases, if logging is enabled on an interface and the **log** option is included in an ACL entry, packets that match the conditions in a **deny** statement are sent to the CPU for processing.

Starting in release 02.3.00, a new logging method is implemented to prevent the Syslog buffer from being overloaded with entries that are logged each time a packet is denied access by an ACL entry. With the new method, the first packet, which matches a **deny** statement with the **log** option configured, is sent to the CPU for logging. After a certain "waiting' period, the next packets that match the **deny** condition are dropped in hardware; no additional Syslog message is written for packets that are denied access during this time. When the waiting period expires, a Syslog message is written if the routing switch receives another packet that matches the **deny** condition and a new waiting period begins.

Enabling the New ACL Logging Method

No new CLI commands are introduced with the new method for logging packets denied by individual ACL entries. However, to activate the new ACL logging method, you must configure the following settings:

- Enable Syslog logging as follows:

  ```
  9408sl(config)# logging on
  ```

- Add the **log** option to individual ACL statements as in the following examples:

  ```
  9408sl(config)# access-list 400 permit any any log-enabled
  ```

  Or

  ```
  9408sl(config)# ip access-list standard hello
  9408sl(config-std-nacl)# permit any log
  ```

- Enter the **ip access-group enable-deny-logging** command on an interface. If this command is not entered, packets denied by ACLs are not logged.

  ```
  9408sl(config)# interface ethernet 5/1
  9408sl(config-if-e1000-5/1)# ip access-group enable-deny-logging
  ```

Syntax: ip access-group enable-deny-logging

Specifying the Wait Time

To specify how long the system waits before it writes a new message for denied packets in the Syslog, enter the following command:

```
9408sl(config)# ip access-list logging-age 2
```

Syntax: ip access-list logging-age <minutes>

The valid range for the <minutes> parameter is from 1 to 10. The default is 5 minutes.

Reordering ACL Entries

When you configure an ACL, the software places the ACL entries in the order in which they are entered. For example, if you enter the following ACL statements in the order shown below, the software always applies the entries to traffic in the same order.

```
9408sl(config)# access-list 1 deny 209.157.22.0/24
9408sl(config)# access-list 1 permit 209.157.22.26
```

If a packet matches the first ACL entry and is denied, the software does not compare the packet to the remaining ACL entries. In this example, packets from host 209.157.22.26 will always be dropped, even though packets from this host match the second entry.

You can reorder entries within an ACL by individually removing the ACL entries using the **no** form of the command, followed by the ACL entry. Repeat the **no** form of the command for each ACL entry in the ACL that you want to move. After removing entries from the ACL, you then re-enter each statement in the ACL in the order in which you want them to filter traffic.

Although this method of reordering ACL entries works well for small ACLs, it can be impractical for large ACLs that contain many entries. An alternative method for reordering ACL entries is introduced that allows you to edit a large ACL on a TFTP file server and later download the ACL to the running configuration on a routing switch. The different procedures for editing and downloading ACLs are described in the following sections:

- “Editing and Downloading an ACL from a TFTP Server” on page 107
- “Editing ACL Entries” on page 108
- “Displaying the Number of Configured ACLs” on page 108
- “Inserting an ACL Entry” on page 109
- “Deleting an ACL Entry” on page 110

Editing and Downloading an ACL from a TFTP Server

The alternative method for re-ordering entries in an ACL allows you to edit an ACL on a file server and then download the edited ACL from a TFTP server to the routing switch, replacing the ACL in the running-config file with the downloaded list. You can later save the modified ACL entries to the startup-config file on the routing switch.

Because an ACL list contains only ACL entries, not the assignment of the ACL to one or more interfaces, you must separately apply an ACL to individual interfaces on a routing switch.

NOTE: The only commands that are valid in the ACL list are the **access-list** and **end** commands. The routing switch ignores other commands in the file.

To modify an ACL on a file server and download the edited file to a routing switch, follow these steps:

1. Use a text editor to create a new text file. When you name the file, use 8.3 format (up to eight characters in the name and up to three characters in the extension).

NOTE: Make sure the routing switch has network access to the TFTP server.

2. Optionally, clear the ACL entries from the ACLs you are changing by placing commands such as the following at the top of the file:

   ```
   no access-list 1
   no access-list 101
   ```

   When you load the ACL list into the routing switch, the software adds the ACL entries in the file after any entries that already exist in the same ACLs. Therefore, if you intend to replace all entries in an ACL, you must use the **no access-list** <*num*> command to clear the entries from the ACL before you add new ones.

3. In the text file, enter each command that creates an ACL entry. The order in which you apply separate ACLs does not matter, but the order of the entries in each ACL is important. The software applies the entries in an ACL in the order in which they are listed in the ACL. Here is an example of some ACL entries:

   ```
   access-list 1 deny host 209.157.22.26 log
   access-list 1 deny 209.157.22.0 0.0.0.255 log
   access-list 1 permit any
   access-list 101 deny tcp any any eq http log
   ```

   The software applies the entries in ACL 1 in the order shown and stops at the first match. If a packet is denied by one of the first two entries, the packet will not be permitted by the third entry, even if the packet matches the comparison values in this entry.

4. Enter the **end** command on a separate line at the end of the file. This command indicates to the software that the entire ACL list has been read from the file.

5. Save the text file.

6. On the routing switch, enter the following command at the Privileged EXEC level of the CLI:

   copy tftp running-config <tftp-ip-addr> <filename>

   NOTE: If you enter any commands besides the **access-list** and **end** (as the last entry in the file) commands in the text file, the **copy tftp running-config** command does not function correctly. Only the **access-list** and **end** commands are supported in a file that you load using the **copy tftp running-config** command.

7. To save the changes to the startup-config file of the routing switch, enter the following command at the Privileged EXEC level of the CLI:

write memory

The following example displays a complete ACL configuration file that can be downloaded from a TFTP server:

```
no access-list 1
no access-list 101
access-list 1 deny host 209.157.22.26 log
access-list 1 deny 209.157.22.0 0.0.0.255 log
access-list 1 permit any
access-list 101 deny tcp any any eq http log
end
```

NOTE: Do not enter other commands in the file. The routing switch reads only the ACL information in the file and ignores other commands, including **ip access-group** commands. To assign ACLs to interfaces, use the CLI.

Editing ACL Entries

By default, newly created ACL entries are appended to the end of an ACL list. Since ACL entries are applied to data packets in the order in which they are read in a list, you also need to create ACLs in the order you want them to be applied.

In a routing switch running release 02.2.01 and later, the CLI command for standard and extended ACLs has been enhanced with the following options:

- Insert a new ACL entry within an existing ACL.
- Delete an entry from an ACL
- Replace an existing ACL entry.

The **show access-list** and **show ip access-list** commands display the entries in an ACL with line numbers.

Displaying the Number of Configured ACLs

You can quickly determine the number of ACLs configured on a routing switch by entering the **show access-list count** command; for example:

```
9408sl(config)# show access-list count

ACL filter counts:
Total 0 IPV4 L3/L4 ACLs exist.
Total 0 L2 MAC ACLs exist.
```

Syntax: show access-list count

The command output displays the number of IPv4, L2, and IPv6 ACLs (if enabled) configured on the routing switch.

Numbered ACLs

To display the contents of numbered ACLs, enter the **show access-list** command:

```
9408sl(config)# show access-list 99

Standard IP access list 99
1. deny host 1.2.4.5
2. deny host 5.6.7.8
3. permit any
```

Syntax: show access-list <acl-num> | all

Named ACLs

To display the contents of named ACLs, enter the **show ip access-list** command:

```
Standard IP access list melon
1. deny host 1.2.4.5
2. deny host 5.6.7.8
3. permit any
```

Syntax: show ip access-list <acl-num> | <acl-name>

Inserting an ACL Entry

To insert an entry in an ACL, enter the **show access-list** or **show ip access-list** command to display the contents of the ACL and determine where you want to insert the new entry. Then enter the appropriate commands as described in the following sections.

Numbered ACLs

The following example inserts a new ACL entry in line 2 of ACL 99.

```
9408sl(config)# access-list 99 insert 2 deny 5.6.7.8
```

Syntax: access-list <acl-num> insert <line-num>

The <acl-num> parameter specifies the numbered ACL in which the new entry will be inserted. This number can be from 1 to 99 for standard ACLs or from 100 to 199 for extended ACLs.

The **insert** <line-num> parameter specifies the lien number at which a new ACL entry is inserted. After the new entry is inserted, the ACL list is renumbered.

Named ACLs

To insert an ACL entry that denies host 10.1.1.1 as the second entry in an ACL named “melon”, first enter the **show ip access-list** command to display the current entries in “melon”:

```
9408sl(config)# show ip access-list standard melon

Standard IP access-list melon
1. deny host 1.2.4.5
2. deny host 5.6.7.8
3. permit any
```

To add an entry that denies IP host 10.1.1.1 from any source at line 2, enter the **ip access-list** and **insert deny** commands:

```
9408sl(config)# ip access-list standard melon
9408sl(config-std-nacl)# insert 2 deny host 10.1.1.1
```

To display the updated ACL, enter the **show ip access-list** command.

```
9408sl(config)# show ip access-list standard melon

Standard IP access-list melon
1. deny host 1.2.4.5
2. deny host 10.1.1.1
3. deny host 5.6.7.8
4. permit any
```

The updated list has been renumbered.

Syntax: ip access-list standard | extended <acl-name> | <acl-number>

Syntax: insert <line-num> deny <options>

Syntax: insert <line-number> permit <options>

Syntax: insert <line-number> remark <comment-text>

In the **ip access-list** command:

- The **standard | extended** parameter specifies the ACL type.
- The <acl-name> parameter is the ACL name. The valid values are a string of up to 256 alphanumeric characters. You can use blanks in the ACL name if you enclose the name in quotation marks (for example, “ACL for Net1”).
- The <acl-num> parameter specifies an ACL number if you prefer. You can specify a number from 1 to 99 for standard ACLs or from 100 to 199 for extended ACLs.

Enter the number of the ACL for <acl-num>. You can configure optional settings for ACL entries by using the **ip access-list** command at the standard or extended ACL configuration level.

In the **insert** commands:

- The **insert** <line-num> parameter specifies the line in the list where the ACL entry is to be inserted.
- Enter **deny** if you are creating an entry that denies access to the routing switch.
- Enter **permit** to create an entry that allows access to a routing switch.
- Enter **remark** to add a remark to an ACL entry, followed by a text string for <comment-text>.
- Enter a value for <options> in a standard or extended ACL entry to specify additional ACL criteria. For information about the optional settings that you can configure for standard or extended named ACLs, refer to the ACL chapter in the "*Advanced Configuration and Management Guide for ProCurve 9300/9400 Series Routing Switches*".

Deleting an ACL Entry

If you want to delete an ACL entry in a list, first enter the **show access-list** or **show ip access-list** command to determine the line number of the entry you want to delete. Then enter the appropriate commands, as described in the following sections, to delete the entry.

Numbered ACLs

If you want to delete the second entry from a numbered ACL (such as ACL 99), first display the contents of the list using the **show access-list** command:

```
9408sl(config)# show access-list 99

Standard IP access-list 99
1. deny host 1.2.4.5
2. deny host 5.6.7.8
3. permit any
```

Then enter the following command:

```
9408sl(config)# access-list 99 delete 2
```

To verify the deletion, redisplay the contents of the updated list:

```
9408sl(config)# show ip access-list 99

Standard IP access-list 99
1. deny host 1.2.4.5
2. permit any
```

The updated list has been renumbered.

Syntax: access-list <acl-number> delete <line-number>

The <line-number> parameter specifies the ACL entry to be deleted. The <acl-num> parameter allows you to specify an ACL number if you prefer. If you specify a number, enter a number from 1 to 99 for standard ACLs or from 100 to 199 for extended ACLs.

The delete <line-num> parameter specifies the ACL entry to be deleted.

Named ACLs

To delete an ACL entry from a named ACL, first enter the **show ip access-list** command to list the contents of the ACL; for example:

```
9408sl#show access-list melon

Standard IP access list melon
1. deny host 1.2.4.5
2. deny host 10.1.1.1
3. deny host 5.6.7.8
4. permit any
```

To delete the second ACL entry from the list, enter the following commands:

```
9408sl(config)# ip access-list standard melon
9408sl(config-std-nacl)# delete 2
```

Enter the **show access-list** command to display the updated list.

```
9408sl(config)# show access melon all
```

```
Standard IP access list melon
1. deny host1.2.4.5
2. deny host 5.6.7.8
3. permit any
```

The updated list has been renumbered.

Syntax: ip access-list standard | extended <acl-name> | <acl-number>

Syntax: delete <line-number>

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

The <acl-name> parameter is the ACL name. The valid values are a string of up to 256 alphanumeric characters. You can use blanks in the ACL name if you enclose the name in quotation marks (for example, “ACL for Net1”).

The <acl-num> parameter specifies an ACL number if you prefer. Enter a number from 1 to 99 for standard ACLs or from 100 to 199 for extended ACLs.

Enter the **delete** parameter at the standard or extended IP access-list level. The **delete** <line-number> parameter specifies the first ACL entry to be deleted.

Replacing an ACL Entry

To replace the definition of an ACL entry, enter the **show access-list** or **show ip access-list** command to determine the line number of the entry you want to update. Then enter the appropriate command as described in the following sections.

Numbered ACLs

To replace an entry in a numbered ACL, first display the ACL list; for example:

```
9408sl(config)# show access-list 99
```

```
Standard IP access-list 99
1. deny host 1.2.4.5
2. deny host 10.8.9.1
3. permit any
```

To modify the second entry in ACL 99, enter the following command:

```
9408sl(config)# access-list 99 replace 2 permit 5.6.7.8
```

To verify the replacement, redisplay the updated list:

```
9408sl(config)# show access-list 99
```

```
Standard IP access-list 99
1. deny host 1.2.4.5
2. permit host 5.6.7.8
3. permit any
```

Syntax: access-list <acl-num> replace <line-num>

The <acl-num> parameter specifies the ACL list in which the new entry is inserted. This number can be from 1 to 99 for standard ACLs or from 100 to 199 for extended ACLs.

The **replace** <line-num> parameter specifies the ACL entry to be replaced in the ACL list. After the new entry is inserted, the ACL list is renumbered.

Complete the configuration by specifying <options> for a **deny** or **permit** entry. For information about the optional settings that you can configure for standard or extended numbered ACLs, refer to the ACL chapter in the "*Advanced Configuration and Management Guide for ProCurve 9300/9400 Series Routing Switches*".

Named ACLs

To replace an entry in a named ACL, display the list of ACL entries; for example:

```
9408sl(config)# show ip access-list melon

Standard IP access-list melon
1. deny ip host 1.2.4.5
2. deny 10.1.1.1
3. permit any
```

To modify the second entry in the ACL, enter the following commands:

```
9408sl(config)# ip access-list standard melon
9408sl(config-std-nacl)# replace 2 permit 5.6.7.8
```

Enter the **show ip access-list** command to display the updated list:

```
9408sl(config)# show ip access-list melon

Standard IP access-list melon
1. ip host 1.2.4.5
2. permit ip host 5.6.7.8
3. permit any
```

Syntax: ip access-list standard | extended <acl-name> | <acl-num> replace <line-num>

Syntax: replace <line-number> deny <options>

Syntax: replace <line-number> permit <options>

Syntax: replace <line-number> remark <comment-text>

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

The <acl-name> parameter specifies the ACL name. You can specify a string of up to 256 alphanumeric characters. You can also include blank spaces if you enclose the name in quotation marks (for example, "ACL for Net1").

Optionally, the <acl-num> parameter specifies an ACL number if you prefer. Enter a number from 1 to 99 for standard ACLs or from 100 to 199 for extended ACLs.

The **replace** <line-num> parameter specifies line number of the ACL entry to be replaced.

You specify the contents of the new line entry at the standard or extended ACL level:

- Enter **deny** if you are creating an entry that denies access to the routing switch.
- Enter **permit** to create an entry that allows access to the routing switch.
- Enter **remark** to add a text comment to an ACL entry.

Complete the configuration by specifying <options> for a standard or extended ACL **deny** or **permit** entry. For information about the optional settings that you can configure for standard or extended named ACLs, refer to the ACL chapter in the "*Advanced Configuration and Management Guide for ProCurve 9300/9400 Series Routing Switches*".

Adding a Comment to an ACL Entry

You can add, insert, replace, or delete comments in an ACL entry. First enter the **show access-list** or **show ip access-list** command to determine the line number of the entry you want to update or where you want to insert a new ACL entry. Then enter the appropriate command as described in the following sections.

Numbered ACL: Adding or Replacing a Comment

To add a comment to an entry in a numbered ACL, follow these steps:

1. Enter the **show access-list** command to display the entries in an ACL; for example:

```
9408sl(config)# show access-list 99

Standard IP access-list 99
1. deny host 1.2.4.5
2. permit host 5.6.7.8
3. permit any
```

2. Add a comment (“Permit all users”) to the second entry in the list by entering the **access-list insert remark** command; for example:

```
9408sl(config)# access-list 99 insert 2 remark Permit all users
```

3. Enter the **show access-list** command to display the updated ACL list:

```
9408sl(config)# show access-list 99

Standard IP access-list 99
1. deny host 1.2.4.5
Permit all users
2. permit host 5.6.7.8
3. permit any
```

Syntax: access-list <acl-num> insert <line-number> | replace <line-number> remark <comment-text>

To add a remark to the next ACL entry that you create, enter the **access-list** <acl-num> **remark** <comment-text> command (without the **insert** or **replace** parameters).

The **insert** <line-number> parameter specifies the line number in the ACL at which a new comment is added.

The **replace** <line-number> parameter specifies the line number in the ACL at which a new comment replaces an existing comment.

The **remark** <comment-text> adds a comment to an ACL entry. The remark can be up to 128 characters in length. The comment must be entered separately from the actual ACL entry; that is, you cannot enter the ACL entry and the ACL comment with the same command. Also, in order for the remark to be displayed correctly in **show** command output, you must enter the comment immediately before the ACL entry it describes.

Numbered ACLs: Deleting a Comment

To delete a remark from a numbered ACL entry, enter the following command:

9408sl(config)# access-list 99 delete 2 remark

Syntax: delete <line-number> remark

Named ACLs: Adding a Comment to a New ACL

To add a comment to an entry in a named ACL, follow these steps:

1. Enter the **show access-list** command to display the contents of the ACL. For example, the **show access-list** command output for an ACL named “melon” shows that it has only one entry:

```
9408sl(config)# show access-list melon

Standard IP access-list 99
1. deny host 1.2.4.5
```

2. Add a new entry with a remark to a named ACL by entering the **remark** and **deny** or **permit** commands at the IP access-list level; for example:

```
9408sl(config)# ip access-list standard melon
9408sl(config-std-nacl)# remark Deny traffic from Marketing
9408sl(config-std-nacl)# deny 5.6.7.8
```

3. Enter the **show access-list** command to display the new ACL entry with its remark:

```
9408sl(config)# show access-list melon

Standard IP access-list melon
1. deny host 1.2.4.5
Deny traffic from Marketing
```

```
   2. permit host 5.6.7.8
```

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

Syntax: remark <comment-text>

Syntax: deny <options> | permit <options>

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

The <acl-name> parameter is the ACL name. You can specify a string of up to 256 alphanumeric characters. You can use blanks in the ACL name if you enclose the name in quotation marks (for example, “ACL for Net1”). The <acl-num> parameter allows you to specify an ACL number if you prefer. If you specify a number, enter a number from 1 to 99 for standard ACLs or from 100 to 199 for extended ACLs.

The **remark** <comment-text> adds a comment to the ACL entry that you are about to create. The comment can have up to 128 characters in length. The comment must be entered separately from the actual ACL entry; that is, you cannot enter the ACL entry and the ACL comment with the same command. Also, in order for the remark to be displayed correctly in **show** command output, you must enter the comment immediately before the ACL entry it describes.

Enter the **deny** parameter to deny specified traffic or the **permit** parameter to allow the traffic.

Complete the configuration by specifying **deny** or **permit** <options> for the standard or extended ACL entry. For information about the optional settings that you can configure for standard or extended named ACLs, refer to the ACL chapter in the “*Advanced Configuration and Management Guide for ProCurve 9300/9400 Series Routing Switches*”.

Named ACLs: Inserting or Replacing Comments to Existing ACL Entries

To insert a comment in an existing ACL entry or to replace a comment for an ACL entry, first display the list of entries in the ACL by entering the **show access-list** command; for example:

```
9408sl(config)# show access-list melon

Standard IP access-list melon
1. deny host 1.2.4.5
2. permit host 5.6.7.8
3. permit any
```

To add a comment (“Permit all users”) to the second entry in the list, enter the **ip access-list** and **insert remark** commands; for example:

```
9408sl(config)# ip access-list standard melon
9408sl(config-std-nacl)# insert 3 remark Permit all users
```

Enter the **show access-list** command to display the updated ACL:

```
9408sl(config)# show access-list melon

Standard IP access-list melon
1. deny host 1.2.4.5
2. permit host 5.6.7.8
Permit all users
3. permit ip any any
```

To replace the comment for the third entry, enter the **replace remark** command at the IP access-list level; for example:

```
9408sl(config)# ip access-list standard melon
9408sl(config-std-nacl)# replace 3 remark All users allowed
```

Enter the **show access-list** command to display the contents of the updated ACL:

```
9408sl(config)# show access-list melon

Standard IP access-list melon
1. deny host 1.2.4.5
2. permit host 5.6.7.8
```

```
All users allowed
3. permit ip any any
```

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

Syntax: insert <line-number> | replace <line-number> remark <comment-text>

The **standard** | **extended** parameter specifies the ACL type. The <acl-name> parameter is the ACL name. You can enter a string of up to 256 alphanumeric characters. You can use blanks in the ACL name if you enclose the name in quotation marks (for example, “ACL for Net1”). The <acl-num> parameter allows you to enter an ACL number if you prefer. Valid values for the <acl-num> parameter are from 1 to 99 for standard ACLs or from 100 to 199 for extended ACLs.

The **insert** <line-number> parameter specifies the line number at which the comment is added.

The **replace** <line-number> command specifies the line number at which a new comment replaces an existing comment.

The **remark** <comment-text> adds a comment to an ACL entry. The remark is a text string with up to 128 characters in length. You must enter the comment separately from the actual ACL entry; that is, you cannot enter the ACL entry and the ACL comment with the same **access-list** command. Also, in order for the remark to be displayed correctly in the **show** command output, you must enter the comment immediately before the ACL entry it describes.

Deleting a Comment from a Named ACL.

To delete a comment from a named ACL, enter the **delete remark** command at the IP access-list level; for example:

```
9408sl(config)# ip access-list standard melon
9408sl(config-std-nacl)# delete 3 remark
```

Syntax: delete <line-number> remark

Specifying a Host Name in an ACL Entry

When you configure <options> values in a **deny or permit** entry, you can also specify a host name. For example, the following are valid ACL entries:

```
access-list 101 deny ip host www.google.com host www.ibm.com
access-list 102 permit tcp host www.sbc.com any eq telnet log
```

When you enter an ACL statement that contains a host name in the CLI, the routing switch tries to resolve the host name using the DNS resolver. The switch checks its DNS cache for an entry that corresponds to the host name; if an entry is not found, the switch sends a DNS query to the configured DNS server. When the host name is resolved to an IP address, a Layer 4 CAM entry is created for the ACL.

If the host name cannot be resolved, the ACL statement is not activated. When you enter the **show run** command, the “not in effect” comment is displayed at the end of each ACL statement that specifies an unresolved host name; for example:

```
permit udp host 3.3.3.3 host www.yahoo.com (not in effect)
```

When the host name in an ACL statement cannot be resolved, the routing switch still tries to periodically resolve it. In addition, the routing switch periodically tries to resolve host names that have been resolved previously, but are no longer resolved because their time-to-live (TTL) value expired. If an attempted host-name resolution is successful and the IP address has changed, the ACL’s Layer 4 CAM entry is updated; otherwise, it remains unchanged.

Usage Notes

- In order for the ACL host name feature to function correctly, the following prerequisites exist:
  - DNS must be properly configured on the routing switch.
  - The configured DNS server must be reachable from the routing switch.
- If a host name times out, the routing switch tries to resolve the host name again. If the resolution is successful, one of the following actions is taken:

  - When the host-name-to-IP-address mapping is unchanged, the Layer 4 CAM entry remains the same.
  - When the host-name-to-IP-address mapping changes, the Layer 4 CAM entry is flushed with the new IP address for the host.

- If a host-name resolution is unsuccessful, the Layer 4 CAM entry for the ACL is removed. The output of the **show run** command displays the ACL entry as “not in effect”.

  For example, if the host name “www.foundrynet.com” is not resolved, the output of the **show run** command displays the following line:

  ```
  permit udp host 3.3.3.3 host www.foundrynet.com (not in effect)
  ```

  The routing switch then makes multiple attempts to resolve the host name. If and when the host name is resolved, a new Layer 4 CAM entry is created for the newly resolved IP address.

- When the routing switch boots up, it may take a few seconds for an ACL statement containing a host name to take effect. This is because the switch must first resolve the host name and create a Layer 4 CAM entry. During this brief period, the output of the **show run** command displays the ACL entry as “not in effect”.

OSPF Non-Broadcast Multi-Access Networks

OSPF routers generally use broadcast packets to establish neighbor relationships and broadcast route updates on Ethernet and virtual routing (VE) interfaces. Starting with software releases 02.3.00, you can configure an interface to send OSPF unicast packets rather than broadcast packets to its neighbor by configuring non-broadcast multi-access (NBMA) networks.

NBMA networks are similar to broadcast networks except the packets are sent as unicast. An NBMA network may be required when using multicast traffic is not possible (for example, when a firewall does not allow multicast packets).

You configure NBMAs on an interface. The routers at the other end of that interface must have a non-broadcast neighbor configured. There is no restriction on the number of routers sharing a non-broadcast interface (for example, through a hub or switch).

Configuring OSPF NBMA

To configure NBMA on an interface, do the following:

1. Create an OSPF area on an interface and then enable NBMA on the interface.

   For example, the following commands configure VE 20 as a non-broadcast interface in OSPF area 0.

   ```
   9408sl(config)# interface ve 20
   9408sl(config-vif-20)# ip ospf area 0
   9408sl(config-vif-20)# ip ospf network non-broadcast
   9408sl(config-vif-20)# exit
   ```

   Syntax: [no] ip ospf network non-broadcast

2. At the router OSPF level, specify the IP address of the neighbor in the OSPF network.

   For example, the following commands configure 1.1.20.1 as an OSPF neighbor address. The specified IP address must be in the same subnet as the non-broadcast interface.

   ```
   9408sl(config)# router ospf
   9408sl(config-ospf-router)# neighbor 1.1.20.1
   ```

   Syntax: neighbor <ip-address>

3. Repeat Step 1 on each peer OSPF router in the NBMA subnet to configure the non-broadcast interface.

4. Repeat Step 2 on each peer OSPF router in the NBMA subnet to configure the IP address of all neighbors that share the non-broadcast interface.

For example, to configure the OSPF Non-Broadcast Multi-Access feature in a network with three routers connected by a hub or switch:

- Each router must have the linking interface configured as a non-broadcast interface.

- Each router must have the IP addresses of the other NBMA routers in the subnet configured as OSPF neighbor addresses.

The output of the **show ip ospf interface** command has been enhanced to display information about non-broadcast interfaces and neighbors that are configured in the same subnet.

```
9408sl# show ip ospf interface
v20,OSPF enabled

IP Address 1.1.20.4, Area 0
OSPF state BD, Pri 1, Cost 1, Options 2, Type non-broadcast Events 6
Timers(sec): Transit 1, Retrans 5, Hello 10, Dead 40
DR: Router ID 1.1.13.1        Interface Address 1.1.20.5
BDR: Router ID 2.2.2.1        Interface Address 1.1.20.4
Neighbor Count = 1, Adjacent Neighbor Count= 2
Non-broadcast neighbor config: 1.1.20.1, 1.1.20.2, 1.1.20.3, 1.1.20.5
Neighbor: 1.1.20.5
Authentication-Key:None
MD5 Authentication: Key None, Key-Id None, Auth-change-wait-time 300
```

In the Type field, “non-broadcast” indicates that this is a non-broadcast interface.

When the interface type is non-broadcast, the “Non-broadcast neighbor config” field displays the IP addresses of the neighbors that are configured in the same subnet. If no neighbors are configured, a message such as the following is displayed:

```
***Warning! no non-broadcast neighbor config in 1.1.100.1 255.255.255.0
```

DHCP Relay Agent for IPv6

A client locates a DHCP server using a reserved, link-scoped multicast address. For this reason, it is a requirement for direct communication between the client and the server that they be attached by the same link. However, in some situations in which ease of management, economy, and scalability is a concern, it is useful to allow a DHCP client to send a message to a DHCP server by using a DHCP relay agent.

A DHCP relay agent, which resides on a client’s link to a DHCP server, is used to relay messages between the client and the server. The DHCP relay agent is transparent to the client. When the relay agent receives a message to be relayed from a client to another relay agent or a DHCP server, it creates a new Relay-forward message, sets the original DHCP message to relay forward option, and includes its own address and the address it received in the same option.

Configuring DHCP for IPv6 Relay Agent

You can enable the DHCP for IPv6 relay agent function on an interface and specify a relay destination address by entering the following command at the interface level:

```
9408sl(config)# interface ethernet 2/3
9408sl(config-if-e10000-2/3)# ipv6 dhcp-relay-dest FE80::250:A2FF:FEBF:A056
ethernet 1/3
```

Syntax: ipv6 dhcp-relay < ipv6-address >

The <ipv6-address> parameter specifies a destination address to which the client messages are forwarded and enables DHCP for IPv6 relay services on the interface.

Organization of Product Documentation

NOTE: ProCurve periodically updates the ProCurve 9300/9400 Series Routing Switch documentation. For the latest version of any of these publications, visit the ProCurve website at:

http://www.procurve.com

Click on **Technical Support**, then **Product manuals**.

NOTE: All manuals listed below are available on the ProCurve website, and also on the Documentation CD shipped with your ProCurve product.

Installation and Basic Configuration Guide for ProCurve 9300 Series Routing Switches

This is an electronic (PDF) guide containing product safety and EMC regulatory statements as well as installation and basic configuration information, and software and hardware specifications.

Topics Specific to the 9300 Series Routing Switches

- Product mounting instructions
- Module installation
- Basic access and connectivity configuration (passwords, IP addresses)
- Redundant management module commands and file systems
- Cooling system commands and information
- Basic software feature configuration (SNMP, clock, mirror/monitor ports)
- Configuring for these features:
  - Uni-Directional Link Detection (UDLD)
  - Metro Ring Protocol (MRP)
  - Virtual Switch Redundancy Protocol (VSRP)
  - GVRP (dynamic VLANs)
- Software update instructions
- Hardware specs
- Software specs (e.g. RFC support, IEEE compliance)

Information on Configuring Features for 9300 Series and 9408sl Routing Switches

- Port settings
- VLANS
- Trunks
- Spanning Tree Protocol
- Syslog

Quick Start Guide for ProCurve 9300 Series Routing Switches

This is a printed guide you can use as an easy reference to the installation and product safety information needed for out-of-box setup, plus the general product safety and EMC regulatory statements of which you should be aware when installing and using a Routing Switch.

Installation and Basic Configuration Guide for the ProCurve 9408sl Routing Switch

This is a printed guide that describes the ProCurve 9408sl and provides procedures for installing modules and AC power supplies into the ProCurve 9408sl, cabling the 10-Gigabit Ethernet interface ports, and performing a basic configuration of the software.

Topics Specific to the 9408sl Routing Switch

- Product overview and architecture

- Product mounting instructions
- Module installation
- Basic access and connectivity configuration (passwords, IP addresses)
- Management Module redundancy and file systems
- Interacting with the cooling system, switch fabric module, and interface modules
- Basic software feature configuration (SNMP, clock, mirror/monitor ports)
- Hardware maintenance instructions
- Software update instructions
- Hardware specs
- Safety and regulatory statements
- Software specs (e.g. RFC support, IEEE compliance)

Advanced Configuration and Management Guide for ProCurve 9300/9400 Series Routing Switches

This is an electronic (PDF) guide that contains advanced configuration information for routing protocols and Quality of Service (QoS). In addition, appendixes in this guide contain reference information for network monitoring, policies, and filters.

Information on Configuring Features

- Quality of Service (QoS)
- Access Control Lists (ACLs)
- Rate limiting
- IPv4 routing
- RIP
- IP Multicast
- OSPF
- BGP4
- Multi-protocol BGP (MBGP)
- Network Address Translation (NAT)
- VRRP and VRRPE (VRRP extended)
- IPX routing
- AppleTalk routing
- Route health injection
- RMON, NetFlow, and sFlow monitoring

IPv6 Configuration Guide for the ProCurve 9408sl Routing Switch

This is an electronic (PDF) guide that describes the IPv6 software and features. It provides conceptual information about IPv6 addressing and explains how to configure basic IPv6 connectivity and the IPv6 routing protocols. The software procedures explain how to perform tasks using the CLI.

Command Line Interface Reference for ProCurve 9300/9400 Series Routing Switches

This is an electronic (PDF) guide that provides a dictionary of CLI commands and syntax.

Security Guide for ProCurve 9300/9400 Series Routing Switches

This is an electronic (PDF) guide that provides procedures for securing management access to ProCurve devices and for protecting against Denial of Service (DoS) attacks.

Diagnostic Guide for ProCurve 9300/9400 Series Routing Switches

This is an electronic (PDF) guide that describes the diagnostic commands available on ProCurve devices. The software procedures show how to perform tasks using the Command Line Interface (CLI).

Removing and Installing XENPAK Optics

This is a printed instruction sheet describing the correct preparation and procedure for removing and installing XENPAK optics on the 10-Gigabit Ethernet modules.

Read Me First

The "Read Me First" document, printed on bright yellow paper, is included with every chassis and module. It contains an overview of software release information, a brief "Getting Started" section, an included parts list, troubleshooting tips, operating notes, and other information that is not included elsewhere in the product documentation. It also includes:

- software update instructions
- operating notes for this release

Release Notes

These documents describe features and other information that becomes available between revisions of the main product guides. New releases of such documents will be available on the ProCurve website. To register to receive email notice from ProCurve when a new software release is available, visit:

http://www.procurve.com

In the "**My Procurve**" box on the right, click on "**Register**".

Product Documentation CD: A Tool for Finding Specific Information and/or Printing Selected Pages

This CD is shipped with your ProCurve Routing Switch product and provides the following:

- A **README** file describing the CD contents and use, including easy instructions on how to search the book files for specific information
- A **Contents** file to give you easy access to the documentation on the CD
- Separate PDF files of the individual chapters and appendixes in the major guides, enabling you to easily print individual chapters, appendixes, and selected pages
- Single PDF files for each of the major guides, enabling you to use the Adobe® Acrobat® Reader to easily search for detailed information
- Additional files. These may include such items as additional Read Me files and release notes.

Software Fixes

The following table lists the software issues that were fixed in each release.

Table 24: Software Issues Fixed in each Release

Bug ID	Bug Description	Protocol/ Feature	Version Found	Version Fixed
37478	Some SNMP MIB OIDs would not return data when a get was performed	SNMP	02.0.04	02.1.00a
43519	Router fails to send TTL expired message to the traceroute originator when target is not down.	Traceroute	02.0.04	02.1.00a
43749	Multiple LP Crashes with EXCEPTION 0300, Data Storage Current task :timer	LP Crash	02.0.04	02.1.00a
44220	Router crashed at flash_get_free_block_space(pc) after issuing “write mem” and “show config” multiple times; “show config” also returns no config occasionally.	MP Crash	02.0.04	02.1.00a
44399	Error Message: “Error:send_timeout_ind: itc_send_request () to app_id 0x0000000d" seen on console followed by a crash.	MP Crash	02.0.04	02.1.00a
44477	Policy based routing does not work with one arm routing topology.	PBR, OAR	02.1.00	02.1.00a
44986	The “power-off lp xx” command will cause many error messages to post to the screen such as "power_off_lp: HAL_TURN_OFF_PORT for port 128 failed".	CLI	02.1.00	02.1.00b
44869	CLI hangs when pasting large configs when using SSHv2	CLI, SSHv2	02.1.00	02.1.00c
44986	Powering off a line card with the "power-off lp xxx" command will send errors to the CLI.	CLI	02.1.00	02.1.00c
45222	SSHv2 sessions are not getting cleared properly and SSH sessions are taking too many CPU cycles.	SSv2	02.1.00	02.1.00c
45273	Dynamic disabling / enabling IPv6 RIP does not work correctly.	IPv6 RIP	02.1.00	02.1.00c
45421	IPv6 packets are not switched over a L2 vlan when another vlan containing the same ports is configured with an IPv6 address.	IPv6	02.1.00	02.1.00c
45472	The initial couple of routed packets are dropped in IPv6.	IPv6, IPv4	02.1.00	02.1.00c
45062	VRRP-E interfaces are added to the route table as host-routes when they shouldn't.	VRRP-e	02.1.00	02.1.00d
48161	MP and LP Multicast PIM dense crash in pim_search_routes(pc), if a route's lookup Next-Hop points to a null route.	PIM Dense	02.1.00	02.1.00d

Table 24: Software Issues Fixed in each Release

Bug ID	Bug Description	Protocol/ Feature	Version Found	Version Fixed
47034	LP crash seen after routes were being flushed and or updated with task: Machine Check Crash @ fpip_delete_network_route.	IP Stack	02.1.00	02.1.00e
41397	Router MP crashed with Soft check Timeout: task: "ospf_r_calc" in "ospf_construct_routing_table" while executing the command 'no area 0'.	OSPF	02.1.00	02.1.00f
44402	Multicast and Broadcast packet replication on many ports in the same tower may cause unicast traffic to be dropped on the 60-port 10/100/1000-T module.	60-Port 10/100/1000-T Module, Bcast/ Multicast replication	02.1.00	02.1.00f
49050	While enabling 'debug ip bgp damp' and sending output to SSH session caused MP Crash in Current Task : bgp_io, Data Storage Exception in "ssh_event_handler(pc)".	BGP, SSH	02.1.00	02.1.00f
50271	Applying an ACL on an interface with a network with a scope greater than a class C will cause packets to be dropped.	ACL	02.1.00	02.1.00g
51405	9408sl continues to route to VRRP-E address even when the VRRP-E configuration is deleted.	VRRP-e	02.1.00	02.1.00g
51426	When inserting the second mgmt module, FID error messages are printed to the console. Error: "Error:MpSync: component_id 3 can't be here Warn:sys_fid_update: Sync to standby MP failed for FID 49857 (c2c1) (err = Busy)".	Hot-Swap	02.1.00	02.1.00g
51449	AAA authorization error occurs when running additional script on the TACACS+ server with usernames including the @ sign.	AAA	02.1.00	02.1.00g
49745	Error messages "OSPF:RCV bad check sum" in log.	OSPF	02.1.00	02.1.00h
37478	Some SNMP MIB OIDs would not return data when a get was performed.	SNMP	N/A	02.2.00
44477	Policy-based routing did not work in a one arm routing topology.	Policy-based Routing	N/A	02.2.00
44699	SSH session fails under certain circumstances.	SSHv2	N/A	02.2.00
44736	Telnet authentication hung at the login name prompt.	RADIUS	N/A	02.2.00
44783	A traceroute issued to a host that did not respond resulted in inconsistent IP addresses responding with !H.	Traceroute	N/A	02.2.00
44810	Some interface module ports were disabled after reload and power cycling even though they are configured as enabled.	Interface Module	N/A	02.2.00

Table 24: Software Issues Fixed in each Release

Bug ID	Bug Description	Protocol/ Feature	Version Found	Version Fixed
44813	The counter for the number of routes programmed is not accurate as displayed by the show ip network command.	Counters	N/A	02.2.00
44869	The 9408sl hangs when pasting large configurations such as ACLs when using SSHv2 and takes a long time to output the show task results.	SSHv2	N/A	02.2.00
45222	Sometimes SSH sessions do not get cleared.	SSHv2	N/A	02.2.00
45273	Dynamically disabling or enabling IPv6 RIP did not work correctly.	RIPng	N/A	02.2.00
45421	On the 4-port 10-GbE module, IPv6 packets were not switched over a Layer-2 VLAN when another VLAN containing the same ports was configured with an IPv6 address.	IPv6, 4-Port 10GbE Module	N/A	02.2.00
45472	The first few IPv6 routed packets were dropped for a new flow.	IPv6	N/A	02.2.00
33408	The SNMP MIB object "snVsrpVirRtrState" showed the VSRP state of a device to be in "backup state" when it was in "initialization state".	VSRP, SNMP	N/A	02.2.01
33619	Changing the reference bandwidth to "auto-cost reference-bandwidth xxx" did not change the cost on the OSPF physical interfaces.	OSPF	N/A	02.2.01
41397	A software reload occurred when the device was executing a no area 0 command in the unconfigured part of the OSPF stub area.	OSPF	N/A	02.2.01
44608	If you used a script to update software images, the script failed if you did not specify a location or a PCMCIA slot. You should be able to load software images from where the script was executed if you do not specify a location.	Software Update Script	N/A	02.2.01
44072	The value for the "specific_number" information in an SNMP Syslog message was incorrect when a module's temperature returns to normal.	SNMP	N/A	02.2.01
44074	Layer-3 packets were not forwarded if the ingress port MTU (max-frame-size) was greater than egress port MTU in a routed VE configuration. For example, if the ingress port MTU was set to 4092 bytes and the egress port MTU was set to 1518 bytes, packets should get fragmented and sent out the egress port. This did not occur.	Jumbo Packets	N/A	02.2.01
44399	The following message was continuously sent to the console immediately proceeding a BGP failure: Error:send_timeout_ind: itc_send_request() to app_id 0x0000000d, type 1310722 failed (ret = 8).	BGP	N/A	02.2.01

Table 24: Software Issues Fixed in each Release

Bug ID	Bug Description	Protocol/ Feature	Version Found	Version Fixed
44402	Multicast and broadcast packet replication on many ports on the same device could cause unicast traffic to be dropped.	Multicast, Broadcast	N/A	02.2.01
44467	The management module performed a software reload when an interface module was inserted into the chassis.	Hot Swap	N/A	02.2.01
44477	Policy based routing did not work with one-arm routing topology.	IP Policy	N/A	02.2.01
44700	An error was displayed when you used public key authentication with SSH, and then returned to password authentication.	SSH	N/A	02.2.01
44725	You could not configure IPv4 MTU to be greater than 1500 bytes for jumbo packets.	Jumbo Packets	N/A	02.2.01
44726	OSPF DBD MTU setting was not compliant with RFC1583. It should be set to "0" when "no rfc1583-compatibility" was configured.	OSPF	N/A	02.2.01
44763	The 9408sl allows a user to configure link fault signaling on Gigabit Ethernet interfaces, even though this feature applies only to 10-Gigabit Ethernet interfaces.	CLI	N/A	02.2.01
44769	Every sixty seconds, the 9408sl restored all static IGMP entries. As a result, mcache entries that were removed because their ports went down were restored, even though these ports were still down. This problem may have caused packet loss when there was a significant amount of traffic flowing across the backplane toward ports that were actually down.	IGMP	N/A	02.2.01
44844	While doing an snmpset on the Qbridge MIB values, the 9408sl reloaded the software.	SNMP	N/A	02.2.01
44862	The 9408sl allowed a user to globally configure IPv6 traffic filters, even though this feature is not supported.	IPv6 Extended ACL	N/A	02.2.01
44869	The device would hang when you cut and pasted a large configuration in the CLI during an SSHv2 session.	SSHv2, CLI	N/A	02.2.01
44986	With the 40-port mini-GBIC module, many error messages were displayed when the interface module was being powered off using the "power-off lp" command.	40-Port Mini-GBIC Module	N/A	02.2.01

Table 24: Software Issues Fixed in each Release

Bug ID	Bug Description	Protocol/ Feature	Version Found	Version Fixed
45021	If the same IPv6 link-local address was configured on an arbitrary interface and on the remote end of a different IPv6 interface on the same router, that remote link-local address was not reachable because it is considered a local IPv6 address. Consequently, OSPF and BGP did not work correctly.	IPv6	N/A	02.2.01
45062	VRRP-E interfaces are added to the route table as host-routes when they shouldn't.	VRRP-E	N/A	02.2.01
45222	SSHv2 sessions were not being cleared properly. SSH was acting as if the maximum number of session had been opened, while only one active session existed. Also, SSH sessions required a large amount of CPU cycles.	SSHv2	N/A	02.2.01
45273	You could not enable or disable IPv6 RIP dynamically.	RIPng	N/A	02.2.01
45388	Using SSH to connect to the 9408sl from a Solaris server did not work. The following error message displays on the console: The authenticity of host xxx.xxx.xxx.x can't be established.	SSHv2, Solaris	N/A	02.2.01
45439	The 9408sl did not allow the user to create a VLAN using the dot1qVlanStaticTable SNMP MIB object.	SNMP	N/A	02.2.01
45472	The first few routed packets were being dropped when IPv6 was enabled on the device.	IPv6	N/A	02.2.01
45597	Replacing a 4-port 10-GbE module (removing the configuration then re-adding it), caused the 9408sl to incorrectly configure RSTP parameters on VLAN ports. Note that this error did not occur when a module was simply added to the configuration.	CLI	N/A	02.2.01
45660	Copying images to flash during an SSH session using a script did not work. The console immediately displayed the following message, but did not copy the images: TFTP to flash done.	SSHv2, Software Upgrade	N/A	02.2.01
47193	Deleting a VLAN or an interface with an IP then re-creating it, causes the router to transmit the traffic as untagged frames, even though the frames egress along a tagged port.	VLAN Tagging	N/A	02.2.01
47383	The CLI command snmp-server community did not provide an option to restrict access via an IPv6 ACL.	IPv6, ACL, CLI	N/A	02.2.01

Table 24: Software Issues Fixed in each Release

Bug ID	Bug Description	Protocol/ Feature	Version Found	Version Fixed
47606	The Management Module's InDiscard counter, which showed the number of inbound packets that will be discarded, increased when reconnecting a cable to the device. The counter also increased when there was no traffic on the device. Neither of these incidents should have caused the counter to increase.	Counters	N/A	02.2.01
47729	There was a difference between the port utilization value displayed by the CLI and the one displayed by the SNMP MIB.	SNMP	N/A	02.2.01
48031	With the 4-port 10-GbE module, when the number of active member ports dropped below a specified threshold value, all member ports were disabled. However, once the link came back up, all member ports were still disabled and failed to return to the active state.	Trunking	N/A	02.2.01
48161	A software reload occurred because the next hop for a PIM Dense route pointed to a null route.	PIM Dense	N/A	02.2.01
48554	When you configured Inner VLANs, then reloaded the device, the configuration did not appear in a show run display.	CLI	N/A	02.2.01
48694	ARP packets were being sent to the network address.	ARP	N/A	02.2.01
48919	The <cr> option was missing from the list of show RSTP options.	RSTP, CLI	N/A	02.2.01
49050	While enabling 'debug ip bgp damp' and sending output to SSH session caused MP Crash in Current Task : bgp_io, Data Storage Exception in "ssh_event_handler(pc)".	BGP,SSH	02.1.00	02.1.00f
49194	When using OSPFv3 in IPv6 jumbo mode, the OSPF DBD MTU never increased above 1500 bytes.	OSPFv3, Jumbo packets	NA	02.2.01
49745	The error message "OSPF:RCV bad check sum" appeared in the Syslog.	BGP and SSH	N/A	02.2.01
50089	When you removed a user, then added that user again with the old password, the device displayed an error, telling you to use another password. However, the user was able to login using the old password.	AAA	N/A	02.2.01
50154	The ip ospf mtu-ignore command did not work after a software reload.	OSPF	N/A	02.2.01
50177	Entering a no telnet server command caused the outbound Telnet session to hang when you closed the session.	Telnet	N/A	02.2.01

Table 24: Software Issues Fixed in each Release

Bug ID	Bug Description	Protocol/ Feature	Version Found	Version Fixed
50271	In a one-armed routing scenario, packets were dropped by the CPU when you applied an ACL to an interface on a network that has a scope greater than a CLASS C. The packets should have been permitted by the ACL.	ACLs	N/A	02.2.01
50498	With a 4-port 10-GbE module, entering a show mac command for an interface showed the total number of learned MAC addresses; however, the MAC addresses were not listed in the output. Also, the port could not learn new MAC addresses.	Port Security	N/A	02.2.01
50499	With a 4-port 10-GbE module, when a MAC address caused a security violation on a port, the MAC address could not be learned on another port.	Port Security	N/A	02.2.01
50500	With a 4-port 10-GbE module, the information displayed by a show port sec denied and a show port sec denied ethernet <port-number> was inconsistent. The show port sec denied command displayed the denied mac address in the "Secure-SRC-Addr" field while the show port sec denied ethernet <port-number> command displayed the allowed secure MAC address learned on the port in the "Secure-SRC-Addr" field.	Port Security	N/A	02.2.01
50506	When port security was configured, an error message should be displayed if a MAC Address that has been received on one port was also received on another port.	Port Security	N/A	02.2.01
50507	When a port was brought down by port security, the port state showed "down/down" instead of "disabled". Also to re-enable the port, you needed to enter a disable command first, then an enable command. You should be able to enter just enable.	Port Security	N/A	02.2.01
50527	On devices where port security was enabled, secure MAC addresses were learned on the correct ports of an interface module; however, when you powered off an interface module and entered a show run command, the secure MAC addresses were shown on a different port.	MAC Security	N/A	02.2.01
50577	With a 60-port 10/100/1000-T module, if you wanted to re-enable an interface that had been shutdown (for example, after a port security violation and the interface becomes disabled), you entered a disable command then an enable command. You could not enter just an enable command.	Port Security	N/A	02.2.01

Table 24: Software Issues Fixed in each Release

Bug ID	Bug Description	Protocol/ Feature	Version Found	Version Fixed
50578	With a 60-port 10/100/1000-T module, an error message was displayed when you tried to clear a secure MAC address after a port security violation occurred.	Port Security	N/A	02.2.01
50718	You could not configure LACP on a port if port monitoring was configured that port.	LACP, Port Security	N/A	02.2.01
50795	With a 40-port 10/100/1000-T module, an ACL that was applied to a virtual routing interface of a tagged VLAN was also being applied to a VLAN on the same physical interface, with no virtual routing interfaces.	ACLs	N/A	02.2.01
50804	An exception message was displayed when LACP was configured on interfaces that have port security enabled.	LACP, Port Security	N/A	02.2.01
51435	When you run traceroute using SSH session, the results were not being displayed line by line. The results were displayed after the traceroute was completed.	SSH, Traceroute	N/A	02.2.01
51449	AAA authorization error occurred when additional scripts were running on the TACACS+ server.	AAA	N/A	02.2.01
51637	Entering a show ip vrrp-e command during a Telnet session caused excessive CPU utilization, which can cause in a VSRP failover.	VSRP, CLI	N/A	02.2.01
51789	You could not issue an ipv6 ospf mtu-ignore command after a software reload occurred.	OSPFv3	N/A	02.2.01
52290	Cross-module trunking does not preserve jumbo frames on the secondary ports of a trunk after reload	trunks, jumbo frames	02.2.01	02.2.01b
51626	Syslog messages are not sync'd with Standby Mgmt after reload.	Syslog	02.2.01	02.2.01c
52346	Allow configuration of LACP trunk ports on VLANs if trunk is formed.	LACP	02.1.00	02.2.01c
32773	Configuring the command 'no area range' under Router OSPF causes Exception Type 300 crash in Current Task : ospf_r_calc with "ospf_rtm_add_entry_in_routing_table(pc)" in the stack trace.	OSPF	02.1.00	02.2.01d
44068	Using the scripted upgrade and not specifying a location will cause the script to fail. Error: "Error:perform_lp_file_type_check: sys_open() failed for file lb02004d.bin".	Upgrade	02.1.00	02.2.01d

Table 24: Software Issues Fixed in each Release

Bug ID	Bug Description	Protocol/ Feature	Version Found	Version Fixed
44726	OSPF DBD MTU must be set to zero to be compliant with RFC1583, section A.3.3. On the 9408sl, even when "no rfc1583-compatibility" is not present, the OSPF DBD MTU is always set to the value you can see with "show ip interface".	OSPF	02.1.00	02.2.01d
48187	Error: "OSPF intf rcvd bad pkt: Bad Checksum" msgs occasionally received [NOTE: this is the complete fix for bug 49745, originally fixed in 02.1.00h].	OSPF	02.1.00	02.2.01d
48669	Powering off one module of a multi-slot trunk causes traffic to stop.	Trunk Group	02.1.00	02.2.01d
50498	Port security stats show 6553 number of learned MACs but no MAC address in seen in the MAC table.	Port Security	02.1.00	02.2.01d
51636	Incorrect (100%) port utilization seen on 10Gig ports when using the “sh stat e x/y” command.	CLI	02.1.00	02.2.01d
51924	Module failure on a cross module LACP Trunk will cause L2 traffic not to flood to remaining trunk ports.	LACP	02.1.00	02.2.01d
52632	ARP ref count issue found, replacement DA's are incorrect.	ARP, IPv4	02.1.00	02.2.01d
52676	IGMP PDUs are not L2-flooded within the VLAN even though multicast is disabled on box.	IGMP	02.1.00	02.2.01d
52973	IGMPv3 "joins" are accepted, and clients are added to flows, even though IGMPv3 is not supported in 02.2.01d software. (Fix is that IGMPv3 packets are treated as unknown protocols.)	IGMPv3	02.1.00	02.2.01d
53077	MP crash current task: "mcast in pim_sm_check_route_change" when MBGP peer went down.	MBGP	02.1.00	02.2.01d
53453	PIM-Sparse sends registration to RP if the input ip source address matches one of its own subnets.	PIM-Sparse	02.1.00	02.2.01d
52571	Metric for default route sent to neighbor is not sent when it is configured via a route map.	BGP	02.2.00	02.2.01e
52572	Route-map policy change is not dynamically updating routes when a change is made.	BGP	02.2.00	02.2.01e
52689	Exception Type 0300 (Data Storage Interrupt) crash in task: OSPF, while doing an snmpwalk.	SNMP, OSPF	02.2.00	02.2.01e
53068	LP reset in Current Task: timer, when issuing the “show campartition brief” command on the LP.	CLI, System	02.2.00	02.2.01e
53109	Port mirroring causes multicast packets to be duplicated on the mirror port.	Mirror/ Monitoring	02.2.00	02.2.01e

Table 24: Software Issues Fixed in each Release

Bug ID	Bug Description	Protocol/ Feature	Version Found	Version Fixed
53671	The command 'show ip multicast statistics VLANID' caused the router to crash with "l2mcast_process_show_stats(pc)".	CLI, Multicast	02.2.00	02.2.01e
53730	LP CPU levels spiking or sustained at around 30% due to route changes requiring background	CPU, ARP	02.2.00	02.2.01e
53771	Static IGMP entries, when removed and then learned dynamically, still show up as static.	IGMP	02.2.00	02.2.01e
53867	When a MAC address is locked on a port, it gets removed when seeing this MAC on another port.	Port Security	02.2.00	02.2.01e
54482	LP Crash in - EXCEPTION 0000, Soft Check (Timeout) with task: timer at "sw_l4_construct_source_mask_for_rule_based_acl(pc)" when certain ACL configuration is used.	ACL	02.2.00	02.2.01e
54636	MP Crash EXCEPTION 0000, Soft Check (Timeout) in task: BGP at "ip_find_exact_entry_in_routing_table_trie(pc)" when redistributing static routes.	BGP	02.2.00	02.2.01e
49469	The command 'show ip igmp traffic' does not have the correct interface information (e.g. slot 1 does not exist but slot 1 ports are displayed).	Multicast, IGMP	01.0.00	02.2.01f
52053	PIM enabled VLAN will not L2 forward multicast packets with a TTL=1.	Multicast, PIM	02.2.01	02.2.01f
53601	OSPF crash with task: ospf_r_calc at "ospf_get_next_hop_router_path_type(pc)" when external routes are seen on multiple ASBRs in rfc2328 mode when 'no rfc1583-compatibility' is configured.	OSPF	02.2.01	02.2.01f
53885	When receiving around 30-60% line rate on a 10-Gig interface, occasionally the interface will not change the destination MAC.	routing, 10-Gig	02.1.00	02.2.01f
54156	MP routing table changes are not reflected in the LP routing table for up to 60 seconds with static routes.	static routes	02.2.01	02.2.01f
54206	LSAs are generated for a static route whenever there is a link flap on a non-OSPF interface.	OSPF	02.2.01	02.2.01f
54495	The 'no module' command deletes cam partition commands in running config.	CLI, CAM partitions	02.2.01	02.2.01f
55261	Non-RP sends many register packets for one second to the RP Router in spite of received register stop packet from the RP.	PIM Dense	02.2.01	02.2.01f
55302	Unable to delete below "prefix-list ... in" from config.	CLI	02.2.01	02.2.01f

Table 24: Software Issues Fixed in each Release

Bug ID	Bug Description	Protocol/ Feature	Version Found	Version Fixed
55349	RIP V1 is not advertising routes correctly. It also appears that when the 9408sl receives rip routes (10.1.1.x/24), the 9408sl indicates the netmask as 10.1.1.x/32 in routing table.	RIP	02.2.01	02.2.01f
22002	The command 'show ip cache' shows incorrect id for LP module.	CLI	01.1.00	02.2.01g
53013	Snmpwalk on 'snChas' is not returning installed power supplies.	SNMP	02.2.01	02.2.01g
53308	"TFTP: image download failed - Flash write failed, either flash is full or access violation" - flash download error.	Flash	02.2.01	02.2.01g
53866	MSDP originator-id is incorrectly added to the configuration file.	MSDP	02.2.01	02.2.01g
54927	RSTP does not recover within a second for all failure scenarios.	RSTP	02.2.00	02.2.01g
55006	Flash corruption - Router configuration removed when system reload if both console and SSH are used and “write memory” from console is issued.	SSH, Flash	02.2.01	02.2.01g
55281	The SNMP OID snAgGblDynMemTotal and snAgGblDynMemFree snmp objects report incorrect values.	SNMP	02.2.01	02.2.01g
55286	The LP boot monitor download to all I/O cards fails - flash download error from telnet session "lb02201b138.bin from tftp (MP) -> monitor (LP 2) failed (RECEIVER ERROR)."	Flash	02.2.01	02.2.01g
56253	The command 'power-on/power-off lp X' for an empty slot affects slot state and power budget.	Power	02.1.00	02.2.01g
55233	Occasionally, Optillion SR Xenpak is not recognized by the chassis.	XENPAK	02.1.00	02.2.01h
56296	Show Flash shows inconsistent time stamps for the startup configuration on Active and Standby module.	Flash	N/A	02.2.01h
56322	Losing up to 4 pings between PCs connected to systems in a RSTP ring topology when link is enabled.	RSTP	N/A	02.2.01h
56462	Static trunk is created when link-agg active is there on the secondary port, causing interface to show up as LACP blocking.	LACP	N/A	02.2.01h
56489	"Error triggered port disable" capability added: "link-error-disable <number of flaps> <time period of polling> <amount of time to disable>".	Ports	N/A	02.2.01h

Table 24: Software Issues Fixed in each Release

Bug ID	Bug Description	Protocol/ Feature	Version Found	Version Fixed
56907	error-disable command cli parser adds the help menu to the cli when <tab> is pressed.	Ports	N/A	02.2.01h
56908	error-disable on the trunk interfaces works only on pri interface, show error-disable returns empty.	Ports	N/A	02.2.01h
56925	9408sl with 60-port 10/100/1000-T module receives packet with correct dest/src mac address, but does not learn source mac address correctly on the receive port. The 40-port 10/100/1000-T module does not exhibit this problem.	60-port 10/100/1000-T Module	N/A	02.2.01h
57011	10-Gig PHY doesn't come up properly causing rx packets corruption, "Interpacket junk" errors.	10-Gig	N/A	02.2.01h
57053	Crash in sFlow subsampling interval while setting sFlow configuration on Mgmt port via SNMP.	sFlow	N/A	02.2.01h
57137	Reprogram PBIF during boot up to prevent module not initializing on insertion.	FPGA	N/A	02.2.01h
57401	Excessive logging of stripe-sync failure messages.	Log	N/A	02.2.01h
57402	Command to perform "dm status" commands for line cards enabled on management cli.	CLI	N/A	02.2.01h
57484	When all Gigabit copper ports on a module are enabled, and then disabled, the 9408sl crashes.	Crash	N/A	02.2.01h
57498	'invalid input-> telnet authentication, enable telnet authentication' after reload, parser rejects telnet authentication configuration.	Telnet	N/A	02.2.01h
57682	Cannot update software from PCMCIA card, using a 256-Meg or larger PCMCIA card. Also, the attempt to load software from a 256-Meg or larger PCMCIA card could cause the management module to no longer come up (status: "init state").	PCMCIA	N/A	02.2.01h
57813	Removing link-error-disable does not remove the threshold from the interface.	Ports	N/A	02.2.01h
50709	When both management modules boot up, the lp-primary-0 file in the standby Management Processor (MP) software is deleted.	Dual Management	N/A	02.2.01i
58071	Starting in release 02.2.01i, the **dm sample** command is supported to sample CPU usage and display the active tasks and amount of CPU capacity that each task uses.	Syslog	N/A	02.2.01i
58281	Starting in release 02.2.01i, the logging of LP and standby MP messages on the console is supported. You can view the logged messages on the active MP or by copying the messages to a PCMCIA card with the **console-logging** command.	Syslog	N/A	02.2.01i

Table 24: Software Issues Fixed in each Release

Bug ID	Bug Description	Protocol/ Feature	Version Found	Version Fixed
44284	When you copy a large running configuration file to a file on a local PCMCIA flash card, the destination file becomes corrupted.	System	N/A	02.2.01j
58208	Poor packet forwarding performance occurs after you upgrade a 4-Port 10 Gigabit Ethernet module with Xenpak optics to release 2.2.01f.	LP	N/A	02.2.01j
58283	The syslog message, “access-list fails to function”, is logged if you configure an access-list on a physical or virtual interface that allows traffic.	ACL	N/A	02.2.01j
58691	A timestamp value has been added to crash dumps and additional **debug** commands have been added for buffer management.	Debug	N/A	02.2.01j
52487	In the Web Management interface, on the Ethernet Port Configuration panel, “Full Duplex” is always displayed as the port mode even if a different port mode value is configured.	Web Management	N/A	02.2.01k
54575	In an SSH session command output may stop in the middle of a line and require that you press any key to continue receiving the output.	SSH	N/A	02.2.01k
55240	A BGP prefix list for outbound traffic does not take effect when the routing switch boots up.	BGP	N/A	02.2.01k
55750	A 40-Port 1 Gigabit Ethernet module sends an ARP broadcast every three minutes without taking into account the configured ARP age value.	ARP	N/A	02.2.01k
56810	A port’s maximum transmission unit (MTU) that is displayed in the Web management interface does not always match the configured port MTU value.	Web Management	N/A	02.2.01k
57569	After you exit from a VLAN configuration, CLI commands may be at rejected at system startup.	CLI	N/A	02.2.01k
58068	Some OSPF static routes are not correctly entered.	OSPF	N/A	02.2.01k
58734	A system reload occurs when processing an invalid value in the **ip-option** field.	IP Stack	N/A	02.2.01k
58852	A BGP peer reset occurs when a community attribute list that exceeds 64 entries (or 256 bytes) is received.	BGP	N/A	02.2.01k
58913	A RIP route-map applied on inbound traffic may be displayed as “not defined” in **show ip rip interface ethernet** <*slot*/*port*> command output.	RIP V2/V1	N/A	02.2.01k
59309	When a LACP trunk is activated, the configured **port-name** value is removed from the running-config file.	LACP	N/A	02.2.01k

Table 24: Software Issues Fixed in each Release

Bug ID	Bug Description	Protocol/ Feature	Version Found	Version Fixed
59370	After several weeks of operation, leakage occurs in the TCP send buffer.	TCP	N/A	02.2.01k
59395	The IP Stack is increased when an ITC message for queue depth is received.	System	N/A	02.2.01k
59556	The **enable telnet authentication** command may be rejected by the system parser when the startup configuration is loaded.	CLI	N/A	02.2.01k
44964	The LP may cause a system reload due to a “0200 Machine check”.	System	N/A	02.2.01l
52558	When you enter the **aaa authentication snmp-server** command, the SNMP client cannot set the snAgGblPassword parameter due to an SNMP timeout.	SNMP	N/A	02.2.01l
57611	From the web management interface, you cannot modify the configuration of a virtual routing interface (VE).	Web Management	N/A	02.2.01l
58537	When configuring an extended ACL with the web management interface, if you select the **named acl** option, an error occurs resulting in an unsuccessful configuration.	Web Management	N/A	02.2.01l
60017	An MP module installed in the lower slot fails to adjust the fan speed correctly in auto mode when the temperature falls below the configured threshold values.	System	N/A	02.2.01l
60186	On an IPv6-enabled Gigabit Ethernet module, CRC errors may occur due to a low-level signaling problem.	FPGA	N/A	02.2.01l

Known Issues and Feature Limitations

This section lists the known issues and feature limitations in this release.

The **P** column indicates the priority of the issue, as follows:

- 0 = Critical
- 1 = Major
- 2 = Medium
- 3 = Minor

Table 25: Known Issues and Feature Limitations in Release 02.2.01h

Bug ID #	P	Description	Protocol Feature
41970	2	Module: 9408sl Management Module SSH and Telnet trap and Syslog messages for a logout event are not sent if you enter the **exit** command at the Privileged EXEC level of the CLI to close a session. If you enter the **exit** command at the User EXEC level to end an SSH and/or Telnet session, logout trap and syslog messages are sent. The messages are also sent if you end the session by entering the **logout** command. Workaround: Because the **exit** command entered at the Privileged EXEC level does not disconnect you from a Telnet session, but only returns you to the User EXEC mode, a logout event is not sent. From the User EXEC level, you can the **exit** command to disconnect from the Telnet session. Ending the session generates a logout event that triggers the sending of a logout trap and Syslog message.	System
43839	2	Module: 9408sl Management Module Layer 3 packets are CPU-switched if the ingress port MTU (**max-frame-size** parameter) is smaller than the egress port MTU in a physical-port IP configuration. For example, if the ingress port MTU is set to 2048 bytes and the egress port is set to 9212 bytes, packets are sent to the CPU.	Jumbo Packets
43850	1	Module: 9408sl Management Module The packet payload is cut for packets whose size is either greater than the configured MTU (**max-frame-size** parameter) minus 4 bytes or equal to the MTU limit. This behavior occurs because an internal tag is needed for CRC calculation. • MTU Boundary = 2048 - 4 = 2044 (MP Setting) Results: Four bytes are removed from the payload of packets whose size is 2045, 2046, 2047, and 2048 bytes (FCS included). The packets are then received as 2044-byte packets. • MTU Boundary = 4096 - 4 = 4092 (MP Setting) Results: Four bytes are removed from the payload of packets whose size is 4093, 4094, 4095, and 4096 bytes (FCS included). The packets are received as 4092-byte packets. • MTU Boundary = 8192 - 4 = 8188 (MP Setting) Results: Four bytes are removed from the payload of packets whose size is 8193, 8194, 8195, and 8196 bytes (FCS included). The packets are received as 8188-byte packets. • MTU Boundary = 9216 - 4 = 9212 (MP Setting) Results: Four bytes are removed from the payload of packets whose size is 9213, 9214, 9215, and 9216 bytes (FCS included). The packets are received as 9212-byte packets.	Jumbo Packets

Table 25: Known Issues and Feature Limitations in Release 02.2.01h

Bug ID #	P	Description	Protocol Feature
43993	1	Module: 9408sl Management Module The system forwards Layer 2 packets whose size is greater than the configured MTU (**max-frame-size** parameter) on the local interface. This behavior occurs under the following conditions: • The local and remote systems have an ingress port MTU larger than the egress port MTU. • The link between the two systems is configured with the same MTU size. The traffic is directed from the local system to the remote system. 4000-byte packets were sent to the ingress port of the local system. The following behavior was discovered with these module types: • 9408sl-1Gx40: 4000-byte packet gets cut down to egress MTU size and is forwarded to the remote system egress port. • 9408sl-10Gx4: 4000-byte packet gets forwarded to the ingress port of the remote system.	Jumbo Packets
44284	1	Module: 9408sl Management Module If the running configuration is copied to the PCMCIA card on a management module, the running-config file is corrupted and cannot be used. Workaround: Save the running configuration to the start-up config file by entering the **write memory** command and then copy the startup-config file to the PCMCIA card.	PCMCIA
44637	1	Modules: 40-Port Mini-GBIC (J8684A) and 40-Port 10/100/1000-T (J8685A) On the 9408sl routing switch, the following restrictions apply to MAC authorization: • The **mac-authenticate auth-fail-dot1x-override** global command is not supported. • The *simultaneous* operation of 802.1X and MAC authentication clients on the same port is not supported. • Dynamic ACL configuration on ports that are members of a virtual routing interface is not supported. • 802.1X or multi-device port security cannot be enabled on tagged ports. • If an ACL assigned by 802.1X or multi-device port security could not be bound to an interface, the user is still authenticated. If the ACL was also created by 802.1X or multi-device port security, the ACL definition is shown in the running configuration. Check the Syslog for any messages regarding the failure of the ACL bind operation • The **mac-authentication enable** command at the interface level is not supported.	MAC Authorization

Table 25: Known Issues and Feature Limitations in Release 02.2.01h

Bug ID #	**P**	Description	Protocol Feature
51917	1	Module: 40-Port 10/100/1000-T (J8685A) Layer 3 jumbo packets are forwarded by the CPU on the egress port if the IP MTU on the egress port is *not* equal to the maximum frame size of the ingress port minus 18 bytes. For example, if the ingress port is set to a maximum frame size of 5000, the IP MTU on the egress port must be set to 4982 in order for the jumbo packets to be forwarded in hardware on the egress port. If no value is assigned on the egress port, then the egress port takes the default value of 1500 or a value of the global IP MTU. Layer 3 jumbo packets that are sent out on this port will then be fragmented and CPU forwarded.	Jumbo packets
56935	1	Module: 9408sl Management Module After booting up with software version 02.2.01h, the redundant (standby) management module does not have a spare copy of the LP Primary Application image. Workaround: You must enter the **sync-standby** command after every bootup with release 02.2.01h, including both soft and hard bootups.	Redundant Management

5991-4721
June 2007