Aruba 2930F / 2930M Advanced Traffic Management Guide for ArubaOS-Switch 16.08

a Hewlett Packard
Enterprise company

Part Number: 5200-5484b
Published: June 2019
Edition: 3

Notices

Acknowledgments

This guide provides information on how to configure traffic management features.

Applicable products

This guide applies to these products:

Aruba 2930F Switch Series (JL253A, JL254A, JL255A, JL256A, JL258A, JL259A, JL260A, JL261A, JL262A, JL263A, JL264A, JL557A, JL558A, JL559A)

Aruba 2930M Switch Series (JL319A, JL320A, JL321A, JL322A, JL323A, JL324A, R0M67A, R0M68A)

Switch prompts used in this guide

Examples in this guide are representative and may not match your particular switch/environment. Examples use simplified prompts as follows:

Prompt	Explanation
`switch#`	`#` indicates manager context (authority).
`switch>`	`>` indicates operator context (authority).
`switch(config)#`	`(config)` indicates the config context.
`switch(vlan-`*`x`*`)#`	`(vlan-`*`x`*`)` indicates the vlan context of config, where *x* represents the VLAN ID. For example: `switch(vlan-128)#`.
`switch(eth-`*`x`*`)#`	`(eth-`*`x`*`)` indicates the interface context of config, where `x` represents the interface. For example: `switch(eth-48)#`.
`switch-Stack#`	`Stack` indicates that stacking is enabled.
`switch-Stack(config)#`	`Stack(config)` indicates the config context while stacking is enabled.
`switch-Stack(stacking)#`	`Stack(stacking)` indicates the stacking context of config while stacking is enabled.
`switch-Stack(vlan-`*`x`*`)#`	`Stack(vlan-`*`x`*`)` indicates the vlan context of config while stacking is enabled, where *x* represents the VLAN ID. For example: `switch-Stack(vlan-128)#`.
`switch-Stack(eth-`*`x/y`*`)#`	`Stack(eth-`*`x/y`*`)` indicates the interface context of config, in the form `(eth-`*`<member-in-stack>/<interface>`*`)`. For example: `switch(eth-1/48)#`

Understanding VLANs

Aruba-OS wired switches are 802.1Q VLAN-enabled. In the factory default state, the switch is enabled for up to 256 VLANs. You can reconfigure the switch to support more VLANs. The maximum VLANs allowed varies according to the switch series.

A group of networked ports assigned to a VLAN form a broadcast domain configured on the switch. On a given switch, packets are bridged between source and destination ports that belong to the same VLAN.

VLANs enable grouping users by logical function not physical location. They manage bandwidth usage in networks by:

- Enabling grouping high-bandwidth users on low-traffic segments.
- Organizing users from different LAN segments according to their need for common resources and individual protocols.
- Improving traffic control at the edge of networks by separating traffic of different protocol types.
- Enhancing network security by creating subnets to control in-band access to specific network resources.
- Cross-domain broadcast traffic in the switch is eliminated and bandwidth saved by not allowing packets to flood out all ports.

When configuring VLANs, you will need to plan your VLAN strategy as follows:

Procedure

1. Configure static VLANs with:
   - a name
   - VLAN ID number (VID)
   - port members
2. Include port configuration planning to use dynamic VLANs.
3. Create a map of the logical topology.
4. Create a map of the physical topology.
5. Consider the interaction between VLANs and other features:
   - Spanning Tree Protocol
   - port trunking
   - IGMP
6. Configure at least one VLAN in addition to the default VLAN.
7. Configure all ports that pass traffic for a particular subnet address on the same VLAN.

8. Assign the desired switch ports to the new VLANs.
9. Ensure that the VLAN through which you manage the switch has an IP address, if you are managing VLANs with SNMP in an IP network.

For information on the restrictions when you configure an IP address on a VLAN interface, see the "Comparing port based and protocol based VLAN" table in **Static VLAN operation**.

Static VLAN operation

Static VLANs are configured with a name, VLAN ID number (VID) and port members. For dynamic VLANs, see **GVRP**. 802.1Q compatibility enables you to assign each switch port to multiple VLANs.

Table 1: *Port based and protocol based VLAN*

Function	Port-Based VLANs	Protocol-Based VLANs
IP Addressing	Usually configured with at least one unique IP address. A port-based VLAN can have no IP address. However, this limits switch features available to ports on that VLAN. See "How IP addressing affects switch operation" in the chapter "Configuring IP Addressing" in the *Basic Operation Guide* for the switch. Multiple IP addresses allow multiple subnets within the same VLAN. See the chapter on "Configuring IP Addressing" in the *ArubaOS-Switch Basic Operation Guide* for the switch.	You can configure IP addresses on all protocol VLANs, but IP addressing is used only on IPv4 and IPv6 VLANs. **Restrictions**: Loopback interfaces share the same IP address space with VLAN configurations. The maximum number of IP addresses supported on a switch is 2048; this includes all IP addresses configured for both VLANs and loopback interfaces (except for the default loopback IP address 127.0.0.1). Each IP address configured on a VLAN interface must be unique in the switch; it cannot be used by a VLAN interface or another loopback interface. For more information, see the chapter on "Configuring IP Addressing" in the *ArubaOS-Switch Basic Operation Guide*.
Untagged VLAN Membership	A port can be a member of one untagged, port-based VLAN. All other port-based VLAN assignments for that port must be tagged.	A port can be an untagged member of one protocol VLAN of a specific protocol type, such as IPX or IPv6. If the same protocol type is configured in multiple protocol VLANs, then a port can be an untagged member of only one of those. For example, if you have two protocol VLANs, 100 and 200 and both include IPX, then a port can be an untagged member of either VLAN 100 or VLAN 200, but not both. A port's untagged VLAN memberships can include up to four different protocol types. It can be an untagged member of one of the following: • Four single-protocol VLANs • Two protocol VLANs where one VLAN includes a single protocol and the other includes up to three protocols • One protocol VLAN where the VLAN includes four protocols.
Tagged VLAN Membership	A port can be a tagged member of any port-based VLAN.	A port can be a tagged member of any protocol-based VLAN.

Table Continued

Function	Port-Based VLANs	Protocol-Based VLANs
Routing	If the switch configuration enables IP routing, the switch can internally route IP (IPv4) traffic between port-based VLANs and between port-based and IPv4 protocol-based VLANs. If the switch is not configured to route traffic internally between port-based VLANs, then an external router must be used to move traffic between VLANs.	If the switch configuration enables IP routing, the switch can internally route IPv4 traffic as follows: • Between multiple IPv4 protocol-based VLANs • Between IPv4 protocol-based VLANs and port-based VLANs. Other protocol-based VLANs require an external router for moving traffic between VLANs. **NOTE:** NETbeui and SNA are non-routable protocols. End stations intended to receive traffic in these protocols must be attached to the same physical network.

Commands for Configuring Static VLANs	`vlan <vid> {tagged | untagged <port-list>}`	`vlan <vid> protocol {ipx | ipv4 | ipv6 | arp | appletalk | sna | netbeui}`
		`vlan <vid> {tagged | untagged <port-list>}`

VLAN environments

You can configure different VLAN types in any combination. The default VLAN will always be present. For more on the default VLAN, see **VLAN support and the default VLAN**.

VLAN environment	Elements
The default VLAN (port-based; VID of 1) only	In the default VLAN configuration, all ports belong to VLAN 1 as untagged members. VLAN 1 is a port-based VLAN.
Multiple VLAN environment	In addition to the default VLAN, the configuration can include one or more other port-based VLANs and one or more protocol VLANs. The maximum VLANs allowed on a switch vary according to the switch. For details on the maximum VLANs allowed for your switch, see **Changing the number of VLANs allowed on the switch (CLI)** on page 46. UsingVLAN tagging, ports can belong to multiple VLANs of all types.Enabling routing on the switch enables it to route IPv4 and IPv6 traffic between port-based VLANs and between port-based VLANs and IPv4 protocol VLANs. Routing other types of traffic between VLANs requires an external router capable of processing the appropriate protocols.

VLAN operation

General VLAN operation

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

Types of static VLANs available in the switch

Port-based VLANs

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

Protocol-based VLANs

This type of static VLAN creates a layer-3 broadcast domain for traffic of a particular protocol and is composed of member ports that bridge traffic of the specified protocol type among themselves. Some protocol types are routable on the switches covered in this guide.

Designated VLANs

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

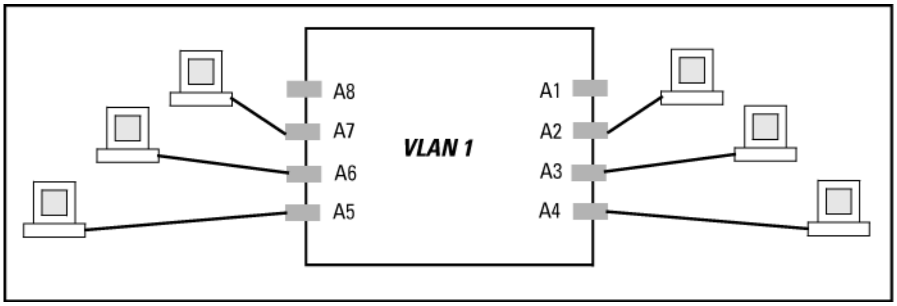
Default VLAN:

This port-based VLAN is always present in the switch and, in the default configuration, includes all ports as members. See **VLAN support and the default VLAN** on page 55.

Except for an IP address and subnet, no configuration steps are needed.

A switch in the default VLAN configuration

In this example, devices connected to these ports are in the same broadcast domain.



Primary VLAN:

The switch uses this port-based VLAN to run certain features and management functions, including DHCP/Bootp responses for switch management. In the default configuration, the Default VLAN is also the Primary VLAN. However, any port-based, non-default VLAN can be designated the Primary VLAN. See **The primary VLAN** on page 55.

Secure Management VLAN:

This optional, port-based VLAN establishes an isolated network for managing switches that support this feature. Access to this VLAN and to the switch's management functions are available only through ports configured as members. See **The primary VLAN** on page 55.

Voice VLANs:

This optional, port-based VLAN type enables separating, prioritizing, and authenticating voice traffic moving through your network, avoiding the possibility of broadcast storms affecting VoIP Voice-over-IP) operation. See **Using voice VLANs** on page 53.

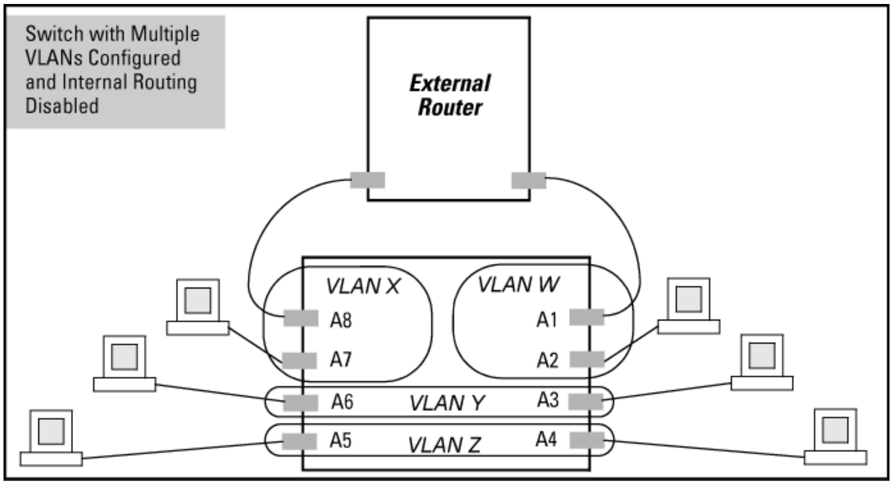
NOTE: In a multiple-VLAN environment that includes older switch models there may be problems related to the same MAC address appearing on different ports and VLANs on the same switch. In such cases, the solution is to impose cabling and VLAN restrictions. For more on this topic, see **Multiple VLAN considerations** on page 30.

Multiple port-based VLANs

In the following example, routing within the switch is disabled (the default). Thus, communication between any routable VLANs on the switch must go through the external router. In this case, VLANs W and X can exchange traffic through the external router, but traffic in VLANs Y and Z is restricted to the respective VLANs.

VLAN 1(the default) is present but not shown. The default VLAN cannot be deleted from the switch, but ports assigned to other VLANs can be removed from the default VLAN. If internal (IP) routing is enabled on the switch, then the external router is not needed for traffic to move between port-based VLANs.

A switch with multiple VLANs configured and internal routing disabled


Switch with Multiple VLANs Configured and Internal Routing Disabled
External Router
VLAN X
VLAN W
A8
A7
A1
A2
A6
VLAN Y
A3
A5
VLAN Z
A4

Protocol VLAN environment

The figure in **Multiple port-based VLANs** illustrates a protocol VLAN environment also. In this case, VLANs W and X represent routable protocol VLANs. VLANs Y and Z can be any protocol VLAN.

As noted for the discussion of multiple port-based VLANs, VLAN 1 is not shown. Enabling internal (IP) routing on the switch allows IP traffic to move between VLANs on the switch, but routable, non-IP traffic always requires an external router.

Routing options for VLANs

Table 2: *Options for routing between VLAN types in the switch*

Note that SNA and NETbeui are not routable protocol types. End stations intended to receive traffic in these protocols must be attached to the same physical network.

		Port-Based	IPX	IPv4	IPv6	ARP	AppleTalk	SNA	NETbeui
Port-Based		Yes	—	Yes	—	—	—	—	—
Protocol	IPX	—	Yes	—	—	—	—	—	—
	IPX4	Yes	—	Yes	—	—	—	—	—
	IPV6	—	—	—	Yes[1]	—	—	—	—
	ARP	—	—	—	—	Yes[1]	—	—	—
	AppleTalk	—	—	—	—	—	Yes[1]	—	—
	SNA	—	—	—	—	—	—	—	—
	NETbeui	—	—	—	—	—	—	—	—

802.1Q VLAN tagging

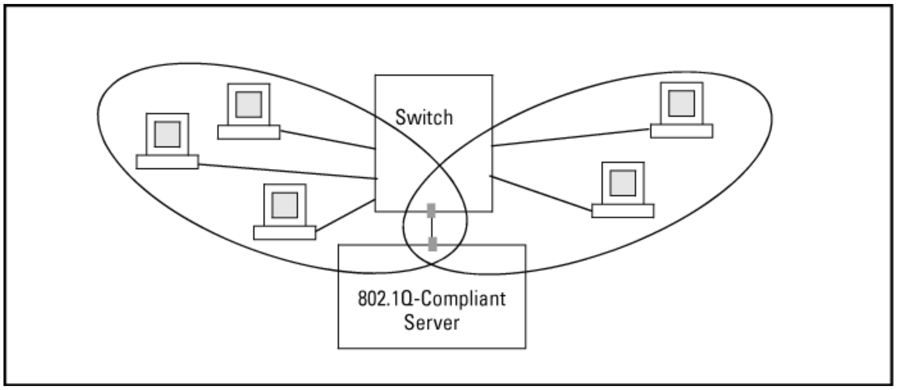
A port can be a member of more than one VLAN of the same type if the device to which the port connects complies with the 802.1Q VLAN standard.

For example, a port connected to a central server using a network interface card (NIC) that complies with the 802.1Q standard can be a member of multiple VLANs, allowing members of multiple VLANs to use the server.

- Although these VLANs cannot communicate with each other through the server, they can all access the server over the same connection from the switch.
- Where VLANs overlap in this way, VLAN "tags" are used in the individual packets to distinguish between traffic from different VLANs.
- A VLAN tag includes the particular VLAN ID. (VID) of the VLAN on which the packet was generated.

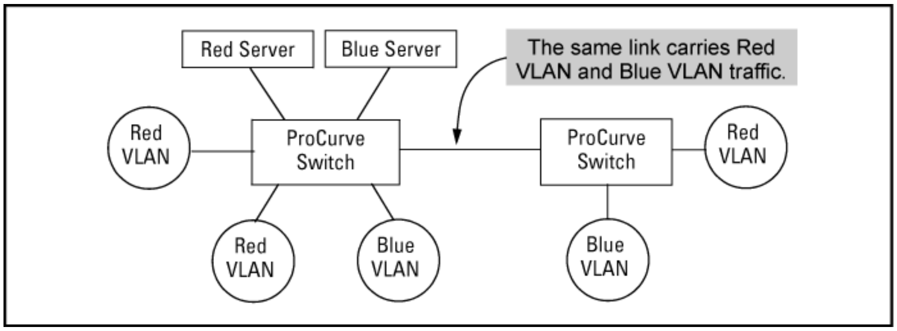
For more on this topic, see **Configuring or changing static VLAN per-port settings (CLI)** on page 37.

Overlapping VLANs using the same server



Similarly, using 802.1Q-compliant switches, you can connect multiple VLANs through a single switch-to-switch link.

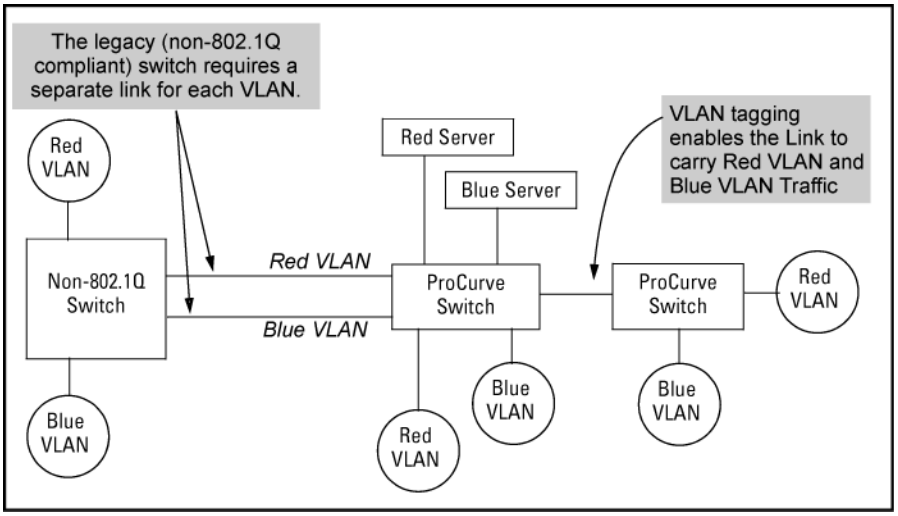
Connecting multiple VLANs through the same link



Introducing tagged VLANs into legacy networks running only untagged VLANs

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

Tagged and untagged VLAN technology in the same network


The legacy (non-802.1Q compliant) switch requires a separate link for each VLAN.
Red VLAN
Red Server
Blue Server
VLAN tagging enables the Link to carry Red VLAN and Blue VLAN Traffic
Non-802.1Q Switch
Red VLAN
Blue VLAN
ProCurve Switch
ProCurve Switch
Red VLAN
Blue VLAN
Blue VLAN
Red VLAN
Blue VLAN

VLAN tagging rules

When tagging is needed

When a port belongs to two or more VLANs of the same type, they remain as separate broadcast domains and cannot receive traffic from each other without routing.

NOTE:

If multiple, non-routable VLANs exist in the switch—such as NETbeui protocol VLANs—they cannot receive traffic from each other.

Inbound tagged packets

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

If a tagged packet arrives on a port that is not a tagged member of the VLAN indicated by the packet's VID, the switch drops the packet.

Similarly, the switch drops an inbound, tagged packet if the receiving port is an untagged member of the VLAN indicated by the packet's VID.

Untagged packet forwarding

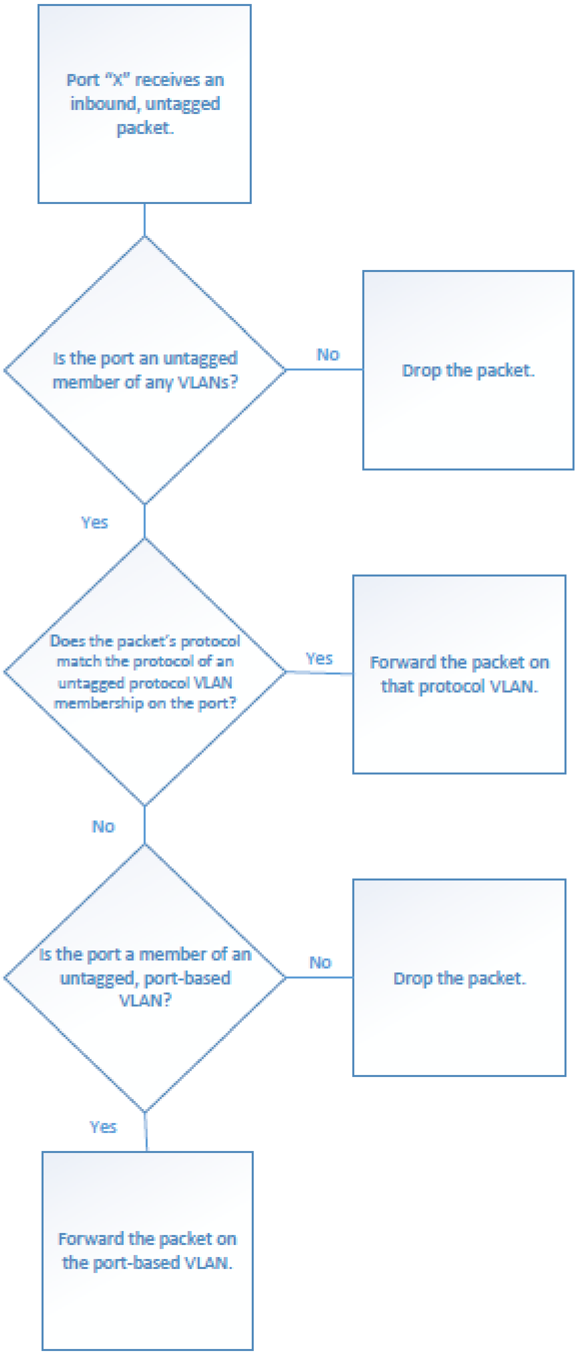
If the only authorized, inbound VLAN traffic on a port arrives untagged, then the port must be an untagged member of that VLAN. This is the case where the port is connected to a non-802.1Q compliant device or is assigned to only one VLAN.

To enable an inbound port to forward an untagged packet, the port must be an untagged member of either a protocol VLAN matching the packet's protocol, or an untagged member of a port-based VLAN.

That is, when a port receives an incoming, untagged packet, it processes the packet according to the following ordered criteria:

1. If the port has no untagged VLAN memberships, the switch drops the packet.
2. If the port has an untagged VLAN membership in a protocol VLAN that matches the protocol type of the incoming packet, then the switch forwards the packet on that VLAN.
3. If the port is a member of an untagged, port-based VLAN, the switch forwards the packet to that VLAN. Otherwise, the switch drops the packet.

Figure 1: *Untagged VLAN operation*

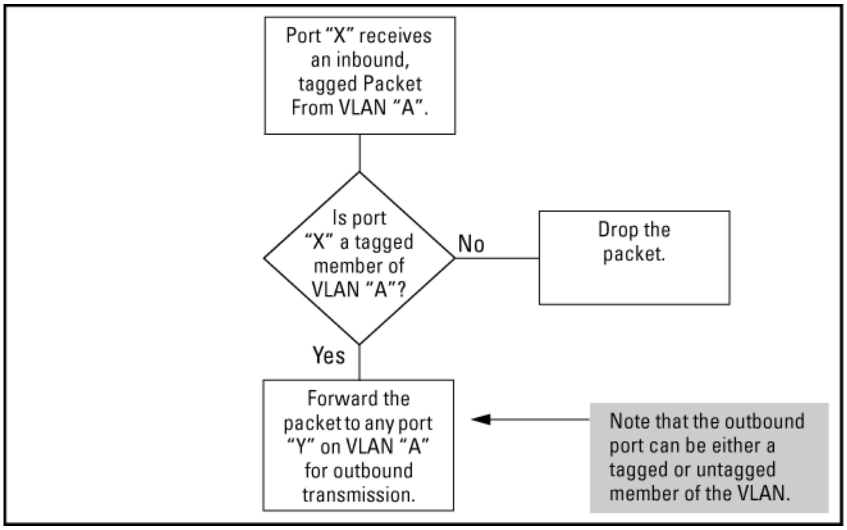

Port "X" receives an inbound, untagged packet.
Is the port an untagged member of any VLANs?
No
Drop the packet.
Yes
Does the packet's protocol match the protocol of an untagged protocol VLAN membership on the port?
Yes
Forward the packet on that protocol VLAN.
No
Is the port a member of an untagged, port-based VLAN?
No
Drop the packet.
Yes
Forward the packet on the port-based VLAN.

Tagged packet forwarding

If a port is a tagged member of the same VLAN as an inbound, tagged packet received on that port, then the switch forwards the packet to an outbound port on that VLAN.

To enable the forwarding of tagged packets, any VLAN to which the port belongs as a tagged member must have the same VID as that carried by the inbound, tagged packets generated on that VLAN.

Figure 2: *Tagged VLAN operation*


Port "X" receives an inbound, tagged Packet From VLAN "A".
Is port "X" a tagged member of VLAN "A"?
No
Drop the packet.
Yes
Forward the packet to any port "Y" on VLAN "A" for outbound transmission.
Note that the outbound port can be either a tagged or untagged member of the VLAN.

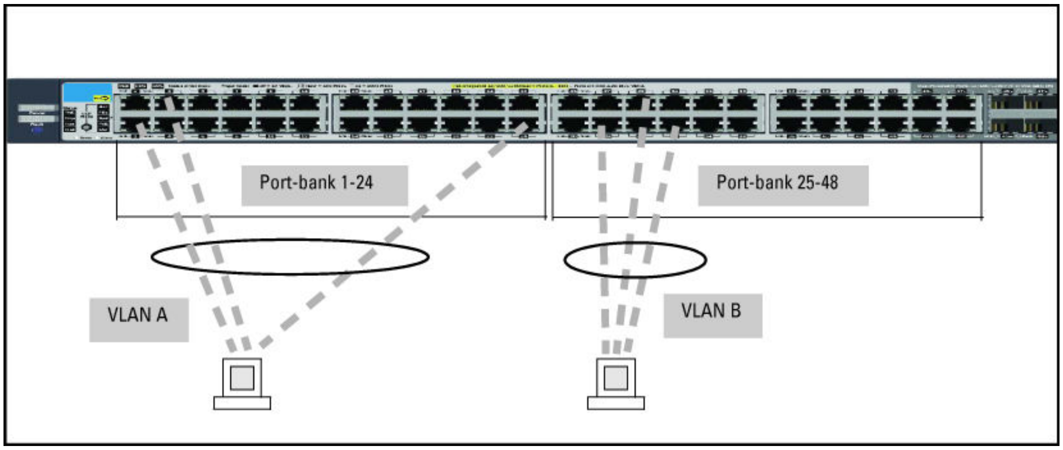
See also **Multiple VLAN considerations** on page 30.

CAUTION: Rate limiting may behave unpredictably on a VLAN if the VLAN spans multiple modules or port-banks.

This also applies if a port on a different module or port-bank is added to an existing VLAN. Hewlett Packard Enterprise does not recommend configuring rate limiting on VLANs that include ports spanning modules or port-banks.

In the following example, ports 2, 3 and 24 form one VLAN, with ports 1 through 24 in the same port-bank. Ports 28, 29 and 32 form a second VLAN. These ports are also in the same port-bank, which includes ports 25 through 48. Rate limiting will operate as expected for these VLANs.

Figure 3: *Example of VLANs using ports from the same port-bank for each VLAN*

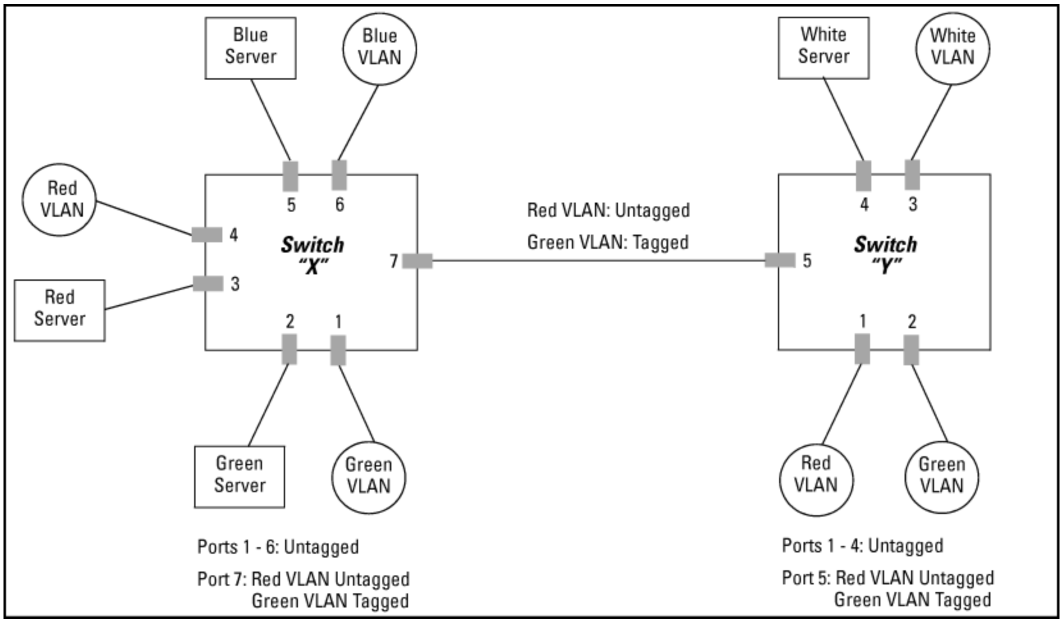


Applying VLAN tagging

Example of tagged and untagged VLAN port assignments

If port 7 on an 802.1Q-compliant switch is assigned to only the Red VLAN, the assignment can remain "untagged" because the port will forward traffic only for the Red VLAN. However, if both the Red and Green VLANs are assigned to port 7, then at least one of those VLAN assignments must be "tagged" so that Red VLAN traffic can be distinguished from Green VLAN traffic.

Figure 4: *Tagged and untagged VLAN port assignments*



In switch X:

- VLANs assigned to ports X1 - X6 can be untagged because there is only one VLAN assignment per port. Red VLAN traffic will go out only the Red ports, Green VLAN traffic will go out only the Green ports, and so on. Devices connected to these ports do not have to be 802.1Q-compliant.
- However, because both the Red VLAN and the Green VLAN are assigned to port X7, at least one of the VLANs must be tagged for this port.

In switch Y:

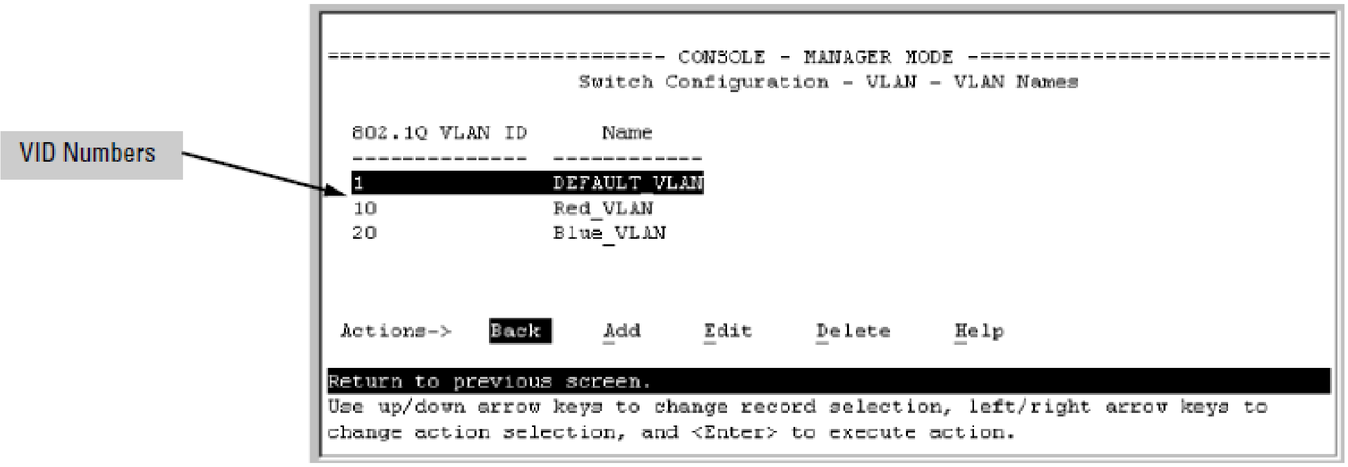
- VLANs assigned to ports Y1 - Y4 can be untagged because there is only one VLAN assignment per port. Devices connected to these ports do not have to be 802.1Q-compliant.
- Because both the Red VLAN and the Green VLAN are assigned to port Y5, at least one of the VLANs must be tagged for this port.

In both switches:

The ports on the link between the two switches must be configured the same. As shown in the following figure, the Red VLAN must be untagged on port X7 and Y5 and the Green VLAN must be tagged on port X7 and Y5, or the opposite way.

NOTE: Each 802.1Q-compliant VLAN must have its own unique VID number and that VLAN must be given the same VID in every device where configured. That is, if the Red VLAN has a VID of 10 in switch X, then 10 must also be the Red VID in switch Y.

Figure 5: *Example of VLAN ID numbers assigned in the VLAN names screen*



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

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



 Actions->  Back     Add     Edit     Delete     Help

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

Additional VLAN tagging considerations

- Since the purpose of VLAN tagging is to allow multiple VLANs on the same port, any port that has only one VLAN assigned to it can be configured as "Untagged" (the default) if the authorized inbound traffic for that port arrives untagged.
- Any port with two or more VLANs of the same type can have one such VLAN assigned as "Untagged." All other VLANs of the same type must be configured as "Tagged," that is:

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

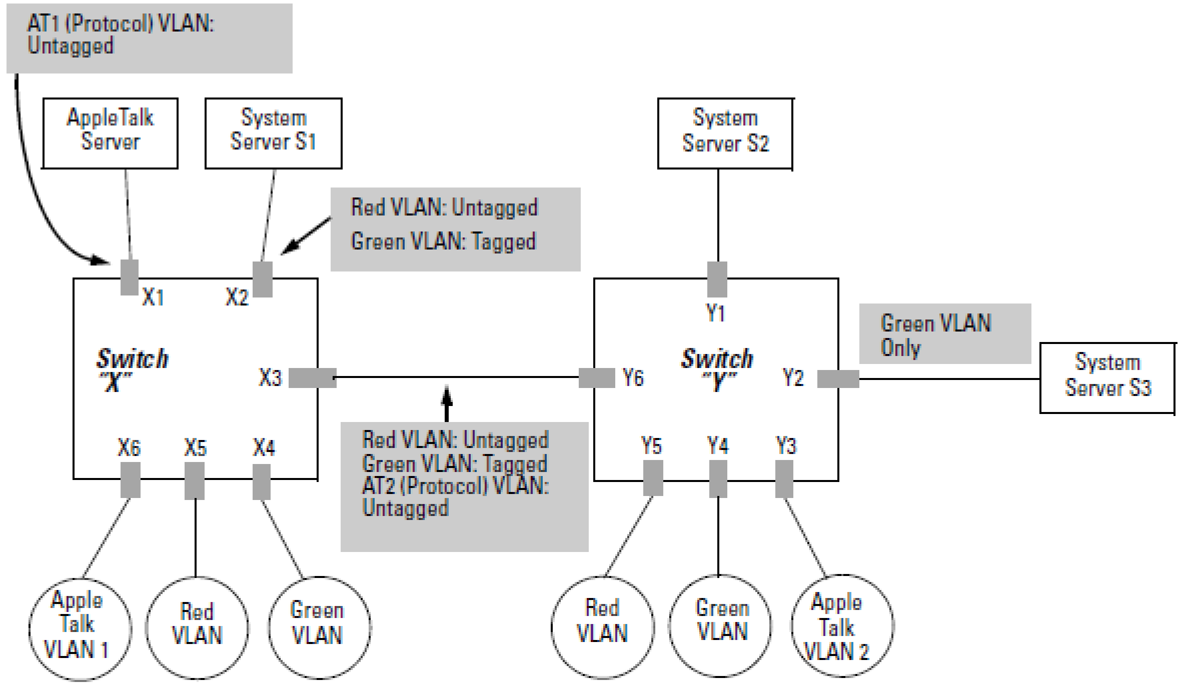
- If all end nodes on a port comply with the 802.1Q standard and are configured to use the correct VID, you can configure all VLAN assignments on a port as "Tagged" if doing so either makes it easier to manage your VLAN assignments, or if the authorized, inbound traffic for all VLANs on the port will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

For a summary and flowcharts of untagged and tagged VLAN operation on inbound traffic, see the following under **VLAN tagging rules** on page 24:

- **"Inbound Tagged Packets"**
- "Untagged Packet Forwarding" and **Figure 1: Untagged VLAN operation** on page 25
- "Tagged Packet Forwarding" and **Figure 2: Tagged VLAN operation** on page 26

Example of Networked 802.1Q-compliant devices with multiple VLANs on some ports

In the following network, switches X and Y and servers S1, S2, and the AppleTalk server are 802.1Q-compliant. (Server S3 could also be 802.1Q-compliant, but it makes no difference for this example.) This network includes both protocol-based (AppleTalk) VLANs and port-based VLANs.



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

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

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

Table Continued

Switch X					Switch Y				
Port	AT-1 VLAN	AT-2 VLAN	Red VLAN	Green VLAN	Port	AT-1 VLAN	AT-2 VLAN	Red VLAN	Green VLAN
X5	No	No	Untagged	No	Y5	No	No	Untagged	No
X6	Untagged	No	No	No	Y6	No	Untagged	Untagged	Tagged

NOTE: VLAN configurations on ports connected by the same link must match. Because ports X2 and Y5 are opposite ends of the same point-to-point connection, both ports must have the same VLAN configuration, configuring the Red VLAN as "Untagged" and the Green VLAN as "Tagged."

Multiple VLAN considerations

Switches use a forwarding database to maintain awareness of which external devices are located on which VLANs. Some switches, such as the switches covered in this guide, have a multiple forwarding database, which means the switch allows multiple database entries of the same MAC address, with each entry showing the (different) source VLAN and source port. Other switch models have a single forwarding database, which allows only one database entry of a unique MAC address, along with the source VLAN and source port on which it is found. All VLANs on a switch use the same MAC address. Thus, connecting a multiple forwarding database switch to a single forwarding database switch where multiple VLANs exist imposes some cabling and port VLAN assignment restrictions. The following table illustrates the functional difference between the two database types.

Table 3: *Forwarding database content*

Multiple forwarding database			Single forwarding database		
MAC address	Destination VLAN ID	Destination port	MAC address	Destination VLAN ID	Destination port
0004ea-84d9f4	1	A5	0004ea-84d9f4	100	A9
0004ea-84d9f4	22	A12	0060b0-880af9	105	A10
0004ea-84d9f4	44	A20	0060b0-880a81	107	A17
0060b0-880a81	33	A20			
This database allows multiple destinations for the same MAC address. If the switch detects a new destination for an existing MAC entry, it just adds a new instance of that MAC to the table.			This database allows only one destination for a MAC address. If the switch detects a new destination for an existing MAC entry, it replaces the existing MAC instance with a new instance showing the new destination.		

All switches covered in this guide use a multiple forwarding database.

Single forwarding database operation

When a packet arrives with a destination MAC address that matches a MAC address in the switch's forwarding table, the switch tries to send the packet to the port listed for that MAC address. But if the destination port is in a different VLAN than the VLAN on which the packet was received, the switch drops the packet. This is not a problem for a switch with a multiple forwarding database because the switch allows multiple instances of a given MAC address, one for each valid destination. However, a switch with a single forwarding database allows only one instance of a given MAC address.

TIP: If you connect both switch types through multiple ports or trunks belonging to different VLANs and enable routing on the switch with the multiple-forwarding database, then the port and VLAN record maintained on the switch with the single-forwarding database for the multiple-forwarding database can change frequently. This may cause poor performance and the appearance of an intermittent or broken connection.

Switch performance is unreliable

The following example provides a method to identify and correct an unsupported configuration.

Symptom

Poor switch performance, unreliable switch performance, dropped packets, discarded packets, appearance of intermittent or broken links.

Cause

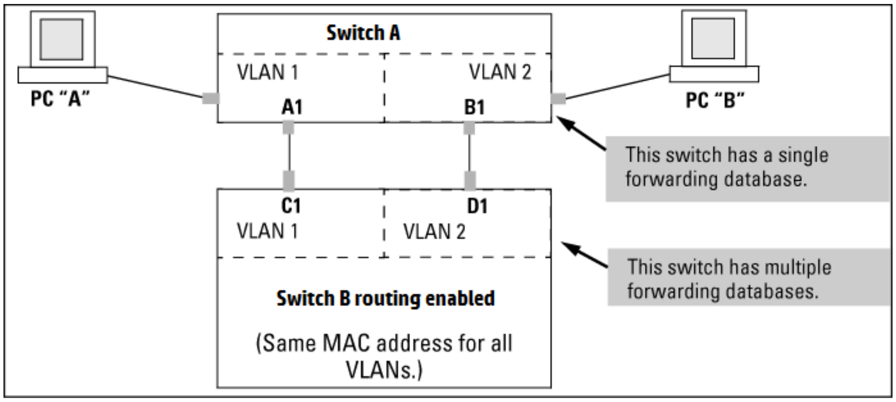
Incorrect switch configuration.

As shown in the following figure, two switches are connected using two ports on each, and the MAC address table for Switch A will sometimes record the switch as accessed on port A1 (VLAN 1) and at other times as accessed on port B1 (VLAN 2).

Procedure

1. **PC A** sends an IP packet to **PC B**.
2. The packet enters VLAN 1 in the switch with the MAC address of the switch in the destination field. Because the switch has not yet learned this MAC address, it does not find the address in its address table and floods the packet out all ports, including the VLAN 1 link (port A1) to the switch. The switch then routes the packet through the VLAN 2 link to the switch, which forwards the packet on to PC B. Because the switch received the packet from the switch on VLAN 2 (port B1), the switch's single forwarding database records the switch as being on port B1 (VLAN 2).
3. **PC A** now sends a second packet to **PC B**. The packet again enters VLAN 1 in the switch with the MAC address of the switch in the destination field. However, this time the switch's single forwarding database indicates that the switch is on port B1 (VLAN 2) and the switch **drops** the packet instead of forwarding it.
4. Later, the switch transmits a packet to the switch through the VLAN 1 link and the switch updates its address table to show that the switch is on port A1 (VLAN 1) instead of port B1 (VLAN 2). Thus, the switch's information on the location of the switch **changes over time**, and the switch discards some packets directed through it for the switch. This causes poor performance and the appearance of an intermittent or broken link.

Figure 6: *Invalid forwarding configuration*


PC “A”
Switch A
VLAN 1
VLAN 2
A1
B1
PC “B”
This switch has a single forwarding database.
C1
D1
VLAN 1
VLAN 2
This switch has multiple forwarding databases.
Switch B routing enabled
(Same MAC address for all VLANs.)

Action/solution

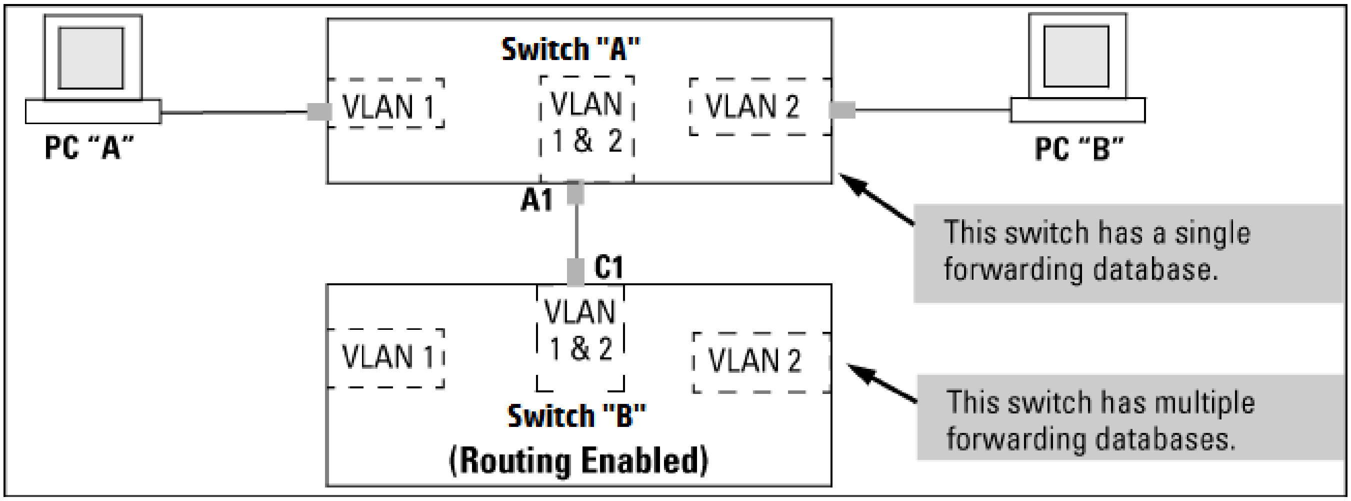
Reconfigure the switches in the configuration.

Procedure

1. Use only one cable or port trunk between single-forwarding and multiple-forwarding database devices.
2. Configure the link with multiple, tagged VLANs.
3. To increase network bandwidth of the connection between devices, use a trunk of multiple physical links.

Following these rules, the switch forwarding database always lists the switch MAC address on port A1 and the switch will send traffic to either VLAN on the switch.

Figure 7: *Solution for single-forwarding to multiple-forwarding database devices in a multiple VLAN environment*



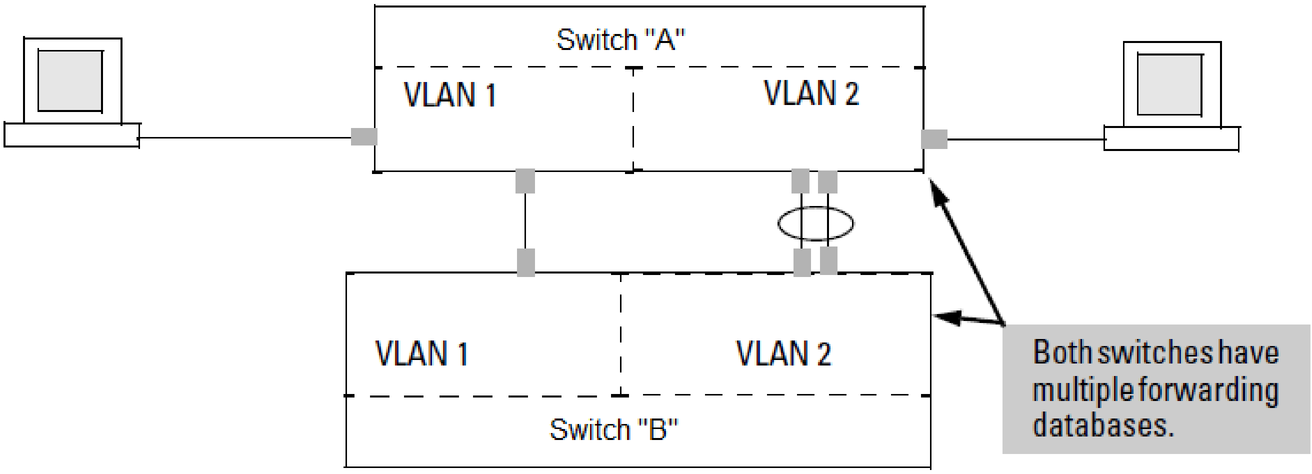
Connecting the Switch to another switch with a multiple forwarding database (Example)

Use one or both of the following connection options:

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

Allowing multiple entries of the same MAC address on different VLANs enables topologies such as the following:

Figure 8: *Topology for devices with multiple forwarding databases in a multiple VLAN environment*



Configuring VLANs

The CLI configures and displays port-based and protocol-based VLANs.

In the factory default state, the switch is enabled for up to 256 VLANs, all ports belong to the default primary VLAN and are in the same broadcast/multicast domain. You can reconfigure the switch to support more VLANs . The maximum VLANs allowed varies according to the switch series.

The number of VLANs allowed on a switch

The factory default number of VLANs is 256.

You can reconfigure the switch to support more VLANs using the `max-vlans` command or the GUI. The maximum VLANs allowed varies according to the switch series. The maximum VLAN values for the switch documented in this guide are as follows:

Attribute	MAX Number of VLANs
2930 Switch Series; WC code	
VLAN	2048
IP VLAN	512 total with up to: • 512 IPv4 • 512 IPv6
static routes	256 total

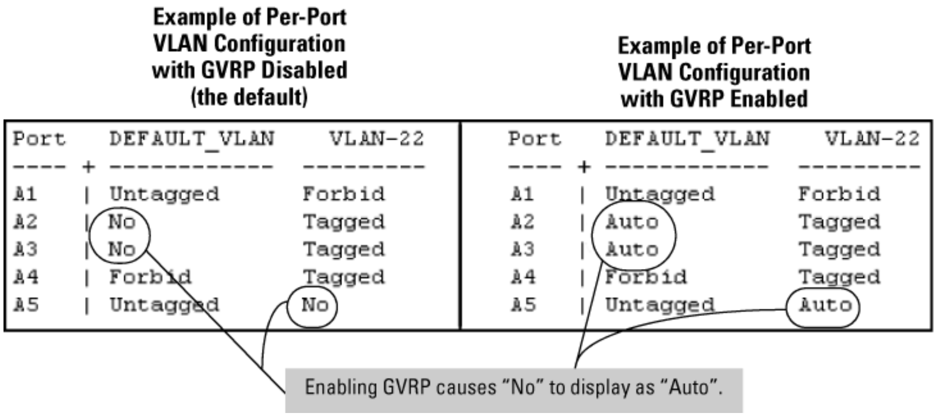
The maximum VIDs is 4094.

Per-port static VLAN configuration options example

This example shows the options available to assign individual ports to a static VLAN.

GVRP, if configured, affects these options and the VLAN behavior on the switch.

Figure 9: *Comparing per-port VLAN options with and without GVRP*



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

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

Example of Per-Port
VLAN Configuration
with GVRP Enabled

```
Port   DEFAULT_VLAN    VLAN-22
---- + ------------  ---------
A1   | Untagged      Forbid
A2   | Auto          Tagged
A3   | Auto          Tagged
A4   | Forbid        Tagged
A5   | Untagged      Auto
```

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

Table 4: *Per-port VLAN configuration options*

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

Configuring port-based VLAN parameters

NOTE: The CLI configures and displays both port-based and protocol-based VLANs (see **Using the CLI to configure port-based and protocol-based VLAN parameters** on page 35.

In the factory default state, the switch is enabled for up to 256 VLANs, all ports belong to the default primary VLAN and are in the same broadcast/multicast domain. The default VLAN is also the default Primary VLAN; see **The primary VLAN** on page 55. In addition to the default VLAN, you can configure additional static VLANs by adding new VLAN names and VIDs, and then assigning one or more ports to each VLAN. (The maximum of VLANs includes the default VLAN, all additional static VLANs you configure, and any dynamic VLANs the switch creates if you enable GVRP; see **GVRP** on page 66.) Each port can be assigned to multiple VLANs by using VLAN tagging; see **VLAN tagging rules** on page 24.)

Using the CLI to configure port-based and protocol-based VLAN parameters

In the factory default state, all ports on the switch belong to the port-based default VLAN (DEFAULT_VLAN; VID=1) and are in the same broadcast/multicast domain.

The default VLAN is also the Primary VLAN.

You can configure additional static VLANs by adding new VLAN names and then assigning one or more ports to each VLAN.

The maximum VLANs accepted by the switch varies according to the switch series. VIDs numbered up to 4094 are allowed. This must include the default VLAN and any dynamic VLANs the switch creates if you enable GVRP (see **GVRP** on page 66).

NOTE: Each port can be assigned to multiple VLANs by using VLAN tagging. See **VLAN tagging rules** on page 24.

Creating a new static VLAN (port-based or protocol-based) (CLI)

The `vlan <vid>` command operates in the global configuration context to configure a static VLAN and/or take the CLI to a specified VLAN's context.

Syntax:

```
vlan <vid> | <ascii-name-string>
```

```
no vlan <vid>
```

If *<vid>* does not exist in the switch, this command creates a port-based VLAN with the specified *<vid>*

If the command does not include options, the CLI, moves to the newly created VLAN context.

If an optional name is not specified, the switch assigns a name in the default format `VLAN` **n**, where `n` is the *<vid>* assigned to the VLAN.

If the VLAN exists and you enter either the *<vid>* or the *<ascii-name-string>* ,the CLI moves to the specified VLAN's context.

The `no` form of the command deletes the VLAN as follows:

If one or more ports belong only to the VLAN to be deleted, the CLI notifies you that these ports will be moved to the default VLAN and prompts you to continue the deletion. For member ports that also belong to another VLAN, there is no **move** prompt.

```
protocol [ipx | ipv4 | ipv6 | arp | appletalk | sna | netbeui]
```

Configures a static, protocol VLAN of the specified type.

If multiple protocols are configured in the VLAN, the `no` form removes the specified protocol

If a protocol VLAN is configured with only one protocol type and you use the `no` form of this command to remove that protocol, the switch changes the protocol VLAN to a port-based VLAN (if the VLAN does not have an untagged member port).

If an untagged member port exists on the protocol VLAN, you must either convert the port to a tagged member or remove the port from the VLAN before removing the last protocol type from the VLAN.

NOTE: If you create an IPv4 protocol VLAN, you must assign the ARP protocol option to it to provide IP address resolution. Otherwise, IP packets are not deliverable. A Caution message appears in the CLI if you configure IPv4 in a protocol VLAN that does not already include the ARP protocol option. The same message appears if you add or delete another protocol in the same VLAN.

`name` *`<ascii-name-string>`*

When included in a `vlan` command to create a new static VLAN, this command specifies a non-default VLAN name. Also used to change the current name of an existing VLAN.

NOTE: Avoid spaces and the following characters in the *`<ascii-name-string>`* entry: `@`, `#:`, `$`, `^`, `&`, `*`, `(` and`)`. To include a blank space in a VLAN name, enclose the name in single or double quotes.

`voice`

Designates a VLAN for VoIP use. For more on this topic, see **Using voice VLANs** on page 53.

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

Creating a new port-based static VLAN

The following example shows how to create a new port-based, static VLAN with a VID of 100 using the following steps:

1. To create the new VLAN, type the `vlan 100` command.
2. To show the VLANs currently configured in the switch, type the `show vlans` command.

If the Management VLAN field (`Primary VLAN : DEFAULT_VLAN Management VLAN` shown in the display information below) is empty, a Secure Management VLAN is not configured in the switch. For more information on configuring a secure management VLAN, see **The secure Management VLAN** on page 56.

```
switch(config)# vlan 100
switch(config)# show vlans

 Status and Counters - VLAN Information
 Maximum VLANs to support : 16
 Primary VLAN : DEFAULT_VLAN
 Management VLAN :

 VLAN ID Name                 Status       Voice Jumbo
 ------- -------------------- ------------ ----- -----
 1       DEFAULT_VLAN         Port-based   No    No
 100     VLAN100              Port-based   No    No
```

Changing the VLAN context level

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

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

Configuring or changing static VLAN per-port settings (CLI)

Syntax:

```
vlan <vid>
```

```
no vlan <vid>
```

This command, used with the options listed below, changes the name of an existing static VLAN and the per-port VLAN membership settings.

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

```
tagged <port-list>
```

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

```
untagged <port-list>
```

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

```
forbid <port-list>
```

Used in port-based VLANs, configures `<port-list>` as forbidden to become a member of the specified VLAN, as well as other actions. Does not operate with option not allowed protocol VLANs. The `no` version sets the port to either `No` or (if GVRP is enabled) to `Auto`. See **GVRP** on page 66.

```
auto <port-list>
```

Available if GVRP is enabled on the switch. Returns the per-port settings for the specified VLAN to `Auto` operation. `Auto` is the default per-port setting for a static VLAN if GVRP is running on the switch. For information on dynamic VLAN and GVRP operation, see **GVRP** on page 66.

Changing the VLAN name and set ports to tagged

Suppose that there is a VLAN named VLAN100 with a VID of 100 and all ports are set to **No** for this VLAN. To change the VLAN name to `Blue_Team` and set ports A1 - A5 to Tagged, use the following commands:

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

Moving the context level

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

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

Changing tagged ports

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

At the global config level, use:

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

or

At the VLAN 100 context level, use:

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

NOTE: You cannot use these commands with dynamic VLANs. Attempting to do so displays the message `VLAN already exists` with no change.

Converting a dynamic VLAN to a static VLAN (CLI)

Syntax:

```
static-vlan <vlan-id>
```

Converts a dynamic, port-based VLAN membership to static, port-based VLAN membership (allows port-based VLANs only).

For this command, *<vlan-id>* refers to the VID of the dynamic VLAN membership. Use `show vlan` to help identify the VID.

This command requires that GVRP is running on the switch and a port is currently a dynamic member of the selected VLAN.

After you convert a dynamic VLAN to static, you must configure the switch's per-port participation in the VLAN in the same way that you would for any static VLAN. For GVRP and dynamic VLAN operation, see **GVRP** on page 66.

Converting a dynamic VLAN to a port-based static VLAN

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

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

Deleting a static VLAN (CLI)

Syntax:

```
vlan <vid>

no vlan <vid>
```

CAUTION: Before deleting a static VLAN, reassign all ports in the VLAN to another VLAN.

Deleting a static VLAN

If ports B1-B5 belong to both VLAN 2 and VLAN 3 and ports B6-B10 belong to VLAN 3, deleting VLAN 3 causes the CLI to prompt you to approve moving ports B6 - B10 to VLAN 1 (the default VLAN). (Ports B1-B5 are not moved because they still belong to another VLAN.)

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

Deleting multiple VLANs

The `interface` command enables you to add or delete interfaces from multiple tagged or untagged VLANs or SVLANs using a single command. Interfaces can be added or deleted for up to 256 VLANs at a time. If more than 256 VLANs are specified, an error is displayed. The `forbid` option prevents an interface from becoming a member of the specified VLANs or SVLANs when used with GVRP.

Syntax

```
interface <port-list> <tagged | untagged | forbid> <vlan | svlan <vlan-id-list>>
```

```
no interface <port-list> <tagged | untagged | forbid> <vlan | svlan <vlan-id-list>>
```

The specified interfaces are added to existing VLANs or SVLANs. If a VLAN or SVLAN does not exist, an error message displays.

The `no` form of the command removes the specified interfaces from the specified VLANs or SVLANs.

The `forbid` option prevents an interface from becoming a member of the specified VLANs or SVLANs. It is executed in interface context.

Removing an interface from several VLANs

The `vlan-id-list` includes a comma-separated list of VLAN IDs and/or VLAN ID ranges.

To remove interface 1 from VLANs 1, 3, 5, 6, 7, 8, 9, 10

```
switch(config)# no interface 1,6,7-10 tagged vlan 1,3,5-10
```

To specify that an interface cannot become a member of VLANs 4 and 5

```
switch(config)# interface 2 forbid vlan 4-5
```

Using IP enable/disable for all VLANs

You can administratively disable the IP address on specified VLANs with static IP addresses without removing the Layer 3 configuration. The switch can be pre-configured as a backup router, then quickly transition from backup to active by re-enabling Layer 3 routing on one or more VLANs. While the switch is in “backup” mode, it will still be performing Layer 2 switching.

A MIB object will be toggled to make Layer 3 routing active or inactive on a VLAN.

Interaction with other features

This feature affects management access to the switch as follows:

- IP—SNMP, Telnet, SSH, HTTP, TFTP, SCP, SFTP
- Routing—RIP, OSPF, PIM, VRRP

When the `disable layer3` command is configured on a VLAN, the behavior is as if no IP address were configured for that VLAN. There is no other change in behavior.

Syntax:

```
disable layer3 vlan <vid> <vid range>
```

```
no disable layer3 vlan <vid> <vid range>
```

In config context, turns off Layer 3 routing for the specified VLAN or VLANs. When executed in vlan context, turns off Layer 3 routing for that VLAN.

The `no` form turns on Layer 3 routing for the specified VLAN or VLANs.

The `show ip` command displays `disabled` in the IP Config column if Layer 3 has been disabled, or if the VLAN has no IP configuration. You can tell which is the case by viewing the remaining columns; if there is no IP configuration, the remaining columns are blank.

Displaying a VLAN disabled for Layer 3

```
switch(config)# show ip

 Internet (IP) Service

  IP Routing : Disabled

  Default Gateway : 172.22.16.1
  Default TTL     : 64
  Arp Age         : 20
  Domain Suffix   :
  DNS server      :

                                                                 Proxy ARP
  VLAN              | IP Config  IP Address      Subnet Mask     Std   Local
  ----------------- + ---------- --------------- --------------- ----  -----
  DEFAULT_VLAN      | DHCP/Bootp 172.22.18.100   255.255.248.0   No    No
  VLAN3             | Disabled   172.17.17.17    255.255.255.0   No    No
  VLAN6             | Disabled
  VLAN7             | Manual     10.7.7.1        255.255.255.0   No    No
```

For IPv6, the `Layer 3 Status` field displays the status of Layer 3 on that VLAN.

Displaying IPv6 Layer 3 status for a VLAN

```
switch(config)# show ipv6

 Internet (IPv6) Service

  IPv6 Routing     : Disabled
  Default Gateway :
  ND DAD           : Enabled
  DAD Attempts    : 3

  Vlan Name       : DEFAULT_VLAN
  IPv6 Status     : Disabled
  Layer 3 Status  : Enabled

  Vlan Name       : layer3_off_vlan
  IPv6 Status     : Disabled
  Layer 3 Status  : Disabled

  Address    |                                             Address
  Origin     | IPv6 Address/Prefix Length                  Status
---------- + ------------------------------------------- -----------
  manual     | abcd::1234/32                               tentative
  autoconfig | fe80::218:71ff:febd:ee00/64                 tentative
```

Interactions with DHCP

Disabling Layer 3 functionality and DHCP are mutually exclusive, with DHCP taking precedence over disable layer3 on a VLAN. The following interactions occur:

- If the `disable layer3` command is executed when DHCP is already configured, no disabling of the VLAN occurs. This error message displays: “Layer 3 cannot be disabled on a VLAN that has DHCP enabled.”
- From the CLI: If `disable layer3` is configured already and an attempt is made to configure DHCP, DHCP takes precedence and will be set. The warning message displays: “Layer 3 has also been enabled on this VLAN since it is required for DHCP.”
- From the CLI: When disabling a range of VLAN IDs, this warning message displays: “Layer 3 will not be disabled for any LANs that have DHCP enabled.”
- From SNMP: If the `disable layer3` command is executed when DHCP is already configured, no disabling of the VLAN occurs. An INCONSISTENT_VALUE error is returned.
- From SNMP: If `disable layer3` is configured already and an attempt is made to configure DHCP, DHCP takes precedence and will be set.

Changing the Primary VLAN (CLI)

For more information on Primary VLANs, see **The primary VLAN** on page 55.

To change the Primary VLAN (CLI), use the following command:

```
primary-vlan vid <ascii-name-string>
```

In the default VLAN configuration, the port-based default VLAN (`DEFAULT_VLAN`) is the Primary VLAN. This command reassigns the Primary VLAN function to an existing, port-based, static VLAN. The switch cannot reassign the Primary VLAN function to a protocol VLAN.

If you reassign the Primary VLAN to a non-default VLAN, to delete the Primary VLAN from the switch, you must assign the Primary VLAN to another port-based static VLAN.

To identify the current Primary VLAN and list the available VLANs and their respective VIDs, use `show vlans`.

Reassigning, renaming and displaying the VLAN command sequence

The following example shows how to reassign the Primary VLAN to VLAN 22 (first command line), rename the VLAN **`22-Primary`** (second command line) and then display the result (third command line):

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

Status and Counters - VLAN Information
Maximum VLANs to support : 8
Primary VLAN : 22-Primary
Management VLAN :

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

Configuring a secure Management VLAN (CLI)

Preparation

Procedure

1. Determine a VID and VLAN name suitable for your Management VLAN.
2. Plan your topology to use switches that support Management VLANs. See **The secure Management VLAN** on page 56.
3. Include only the following ports:
   a. Ports to which you will connect authorized management stations, such as Port A7 in the "Management VLAN control in a LAN" example in **The secure Management VLAN**.
   b. Ports on one switch that you will use to extend the Management VLAN to ports on other switches, such as ports A1 in the "Management VLAN control in a LAN" example in **The secure Management VLAN**.
4. Half-duplex repeaters dedicated to connecting management stations to the Management VLAN can also be included in this topology. Any device connected to a half-duplex repeater in the Management VLAN will also have Management VLAN access.
5. Configure the Management VLAN on the selected switch ports.
6. Test the Management VLAN from all of the management stations authorized to use it, including any SNMP-based network management stations. Also test any Management VLAN links between switches.

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

Configuring an existing VLAN as the Management VLAN (CLI)

Syntax:

`management-vlan <vlan-id> | <vlan-name>`

`no management-vlan <vlan-id> | <vlan-name>`

Configures an existing VLAN as the Management VLAN.

The `no` form disables the Management VLAN and returns the switch to its default management operation.

Default: Disabled. In this case, the VLAN returns to standard VLAN operation.

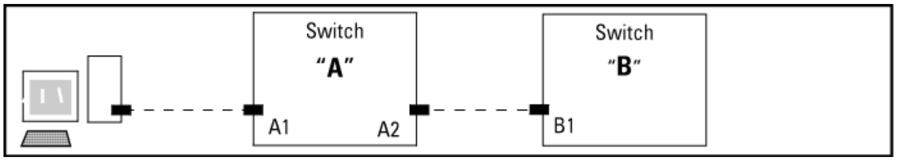
Switch configuration

You have configured a VLAN named My_VLAN with a VID of 100 and want to configure the switch to do the following:

- Use My_VLAN as a Management VLAN (tagged, in this case) to connect port A1 on switch "A" to a management station. The management station includes a network interface card with 802.1Q tagged VLAN capability.
- Use port A2 to extend the Management VLAN to port B1 which is already configured as a tagged member of My_VLAN, on an adjacent switch that supports the Management VLAN feature.

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

Configuration Example

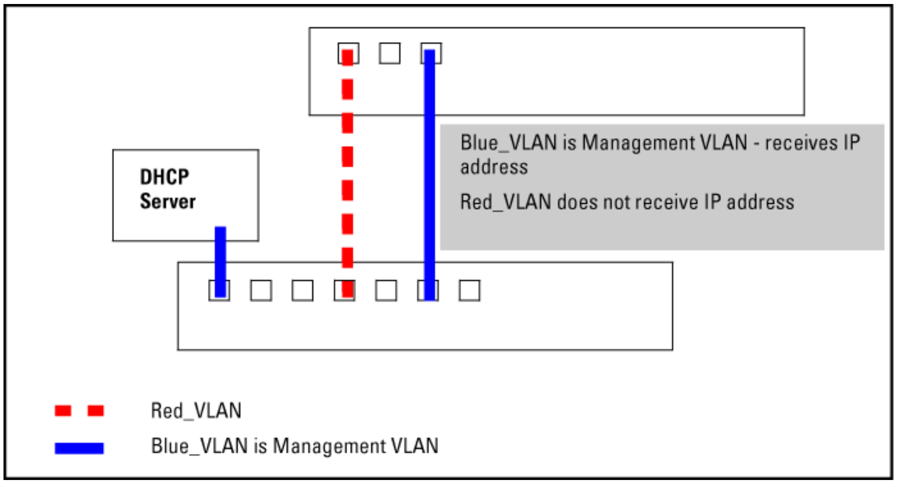


Obtaining an IP address using DHCP (CLI)

Use DHCP to obtain an IPv4 address for your Management VLAN or a client on that VLAN. The following examples illustrate when an IP address will be received from the DHCP server.

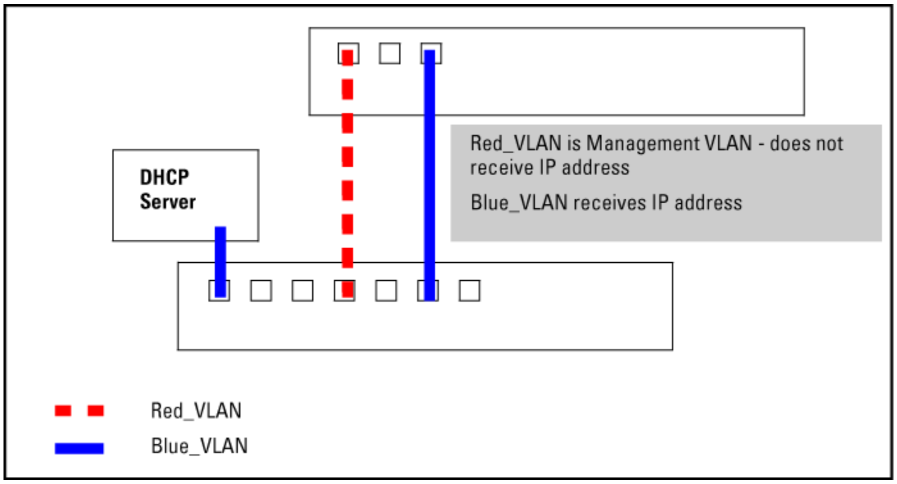
DHCP server on a Management VLAN

If Blue_VLAN is configured as the Management VLAN and the DHCP server is also on Blue_VLAN, Blue_VLAN receives an IP address. Because DHCP Relay does not forward onto or off the Management VLAN, devices on Red_VLAN cannot get an IP address from the DHCP server on Blue_VLAN (Management VLAN) and Red_VLAN does not receive an IP address.


DHCP
Server
Blue_VLAN is Management VLAN - receives IP address
Red_VLAN does not receive IP address
Red_VLAN
Blue_VLAN is Management VLAN

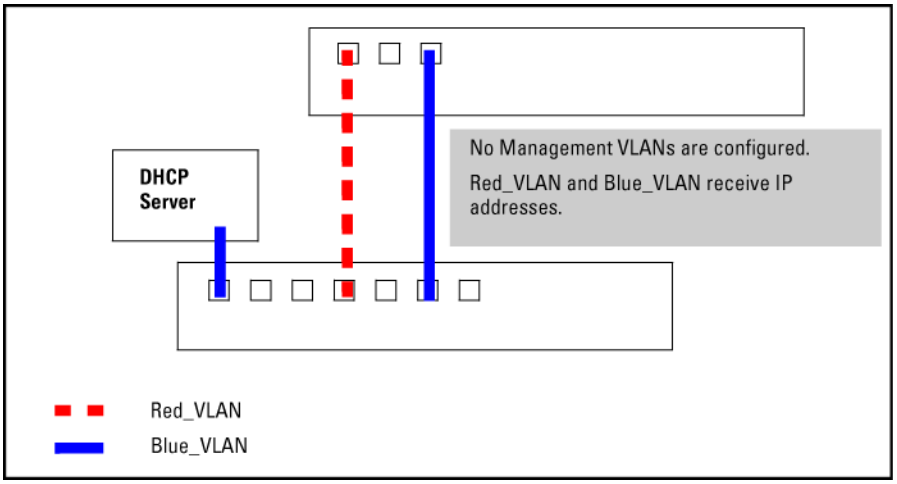
DHCP server on a different VLAN from the Management VLAN

If Red_VLAN is configured as the Management VLAN and the DHCP server is on Blue_VLAN, Blue_VLAN receives an IP address but Red_VLAN does not.


DHCP Server
Red_VLAN is Management VLAN - does not receive IP address
Blue_VLAN receives IP address
Red_VLAN
Blue_VLAN

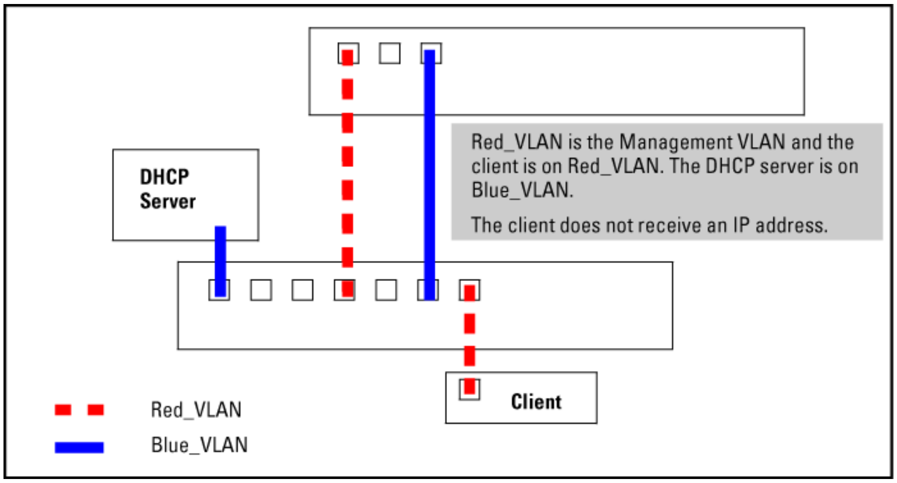
No Management VLANs configured

If no Management VLAN is configured, both Blue_VLAN and Red_VLAN receive IP addresses.


DHCP Server
No Management VLANs are configured.
Red_VLAN and Blue_VLAN receive IP addresses.
Red_VLAN
Blue_VLAN

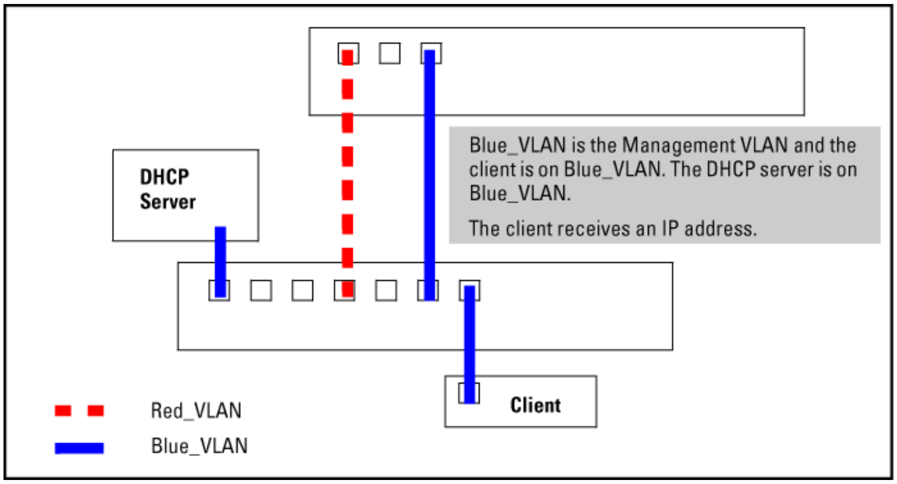
A client on a different Management VLAN from the DHCP server

If Red_VLAN is configured as the Management VLAN and the client is on Red_VLAN, but the DHCP server is on Blue_VLAN, the client will not receive an IP address.


DHCP Server
Red_VLAN is the Management VLAN and the client is on Red_VLAN. The DHCP server is on Blue_VLAN.
The client does not receive an IP address.
Client
Red_VLAN
Blue_VLAN

A DHCP server and client on the Management VLAN

If Blue_VLAN is configured as the Management VLAN, the client is on Blue_VLAN and the DHCP server is on Blue_VLAN, the client receives an IP address.



Blue_VLAN is the Management VLAN and the client is on Blue_VLAN. The DHCP server is on Blue_VLAN.

The client receives an IP address.

Obtaining the IP address for a host that is on a different VLAN than the DHCP server

In the following example, the host is on VLAN 20 and is connected on port number 2 of the switch. The DHCP server, however, is in VLAN 10 and is connected on port 10 of the switch.

Obtaining the IP address for a host that is on a different VLAN than the DHCP server

```
switch(config)# vlan 10
name "VLAN 10"
untagged 10
ip address 10.1.1.2 255.255.255.0
exit
vlan 20
name "VLAN 20"
untagged 2
ip address 100.99.1.1 255.255.255.0
ip helper-address 10.1.1.1
exit
```

Disabling the Management feature (CLI)

You can disable the Secure Management feature without deleting the VLAN.

Disabling the secure management feature

The following commands disable the Secure Management feature in the above example:

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

For more information, see **The secure Management VLAN** on page 56.

Changing the number of VLANs allowed on the switch (CLI)

Syntax:

```
max-vlans<max number of vlans>
```

Use this command to specify the maximum number of VLANs allowed on the switch. The minimum value is 16. The maximum value varies according to the switch series.

For the 2930 switch series you can enter a `max-vlans` value of between 16–2048.

The total number of allowed IP VLANs (IPv6 + IPv4) is 512.

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

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

The following example shows the command sequence for changing the number of VLANs allowed to 20. You can execute the commands to `write memory` and `boot` at another time.

Example of changing the number of allowed VLANs

```
switch(config)# max-vlans 20
This command will take effect after saving the configuration
and rebooting the system.
switch(config)# write memory
switch(config)# boot
This will reboot the system from the primary image, do you want to continue [y/n]? Y
```

Error Messages

An error message will be displayed, if you set the `max-vlans` value to a number that exceeds the allowable value for the switch series.

If you set the `max-vlans` and later try to downgrade to an earlier version of the switch software that does not allow that number of `max-vlans`, successful downgrade may be prevented.

Displaying a switch VLAN configuration

The `show vlans` command lists the VLANs currently running in the switch, with VID, VLAN name, and VLAN status. Dynamic VLANs appear only if the switch is running with GVRP enabled and one or more ports has dynamically joined an advertised VLAN. In the default configuration, GVRP is disabled.

Syntax:

```
show vlans
```

The following describes the fields displayed with this command (see the example output):

Maximum VLANs to support

Shows the number of VLANs the switch is currently configured to support.

Primary VLAN

See **The primary VLAN** on page 55.

Management VLAN

See **The secure Management VLAN** on page 56.

802.1Q VLAN ID

The VLAN identification number, or VID.

Name

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

Status

Port-Based

Port-Based, static VLAN

Protocol

Protocol-Based, static VLAN

Dynamic

Port-Based, temporary VLAN learned through GVRP

Voice

Indicates whether a port-based VLAN is configured as a voice VLAN. See **Using voice VLANs** on page 53.

Jumbo

Indicates whether a VLAN is configured for Jumbo packets. For more on jumbos, see "Port Traffic Controls" in the management and configuration guide for your switch.

This example shows the listing from the `show vlans` command. When GVRP is disabled (the default), Dynamic VLANs do not exist on the switch and do not appear in this listing. For more information, see **GVRP** on page 66.

Displaying VLAN listing with GVRP enabled

```
switch# show vlans

 Status and Counters - VLAN Information

  Maximum VLANs to support : 256
  Primary VLAN : DEFAULT_VLAN
  Management VLAN :

  VLAN ID Name                 | Status     Voice Jumbo
  ------- -------------------- + ---------- ----- -----
  1       DEFAULT_VLAN         | Port-based No    No
  10      VLAN_10              | Port-based Yes   Yes
  15      VLAN_15              | Port-based No    No
  20      VLAN_20              | Protocol   No    No
  33      VLAN_33              | Dynamic    No    No
```

Viewing the VLAN membership of one or more ports (CLI)

Syntax:

```
show vlan ports <port-list> [detail]
```

Displays VLAN information for an individual port or a group of ports, either cumulatively or on a detailed per-port basis.

`port-list`

Specifies a single port number or a range of ports (for example, `a1-a16`), or `all` for which to display information.

`detail`

Displays detailed VLAN membership information on a per-port basis.

The following describes the fields displayed by the command (see example output):

Port name

The user-specified port name, if one has been assigned.

VLAN ID

The VLAN identification number, or VID.

Name

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

Status

Port-Based

Port-Based, static VLAN.

Protocol

Protocol-Based, static VLAN.

Dynamic

Port-Based, temporary VLAN learned through GVRP.

Voice

Indicates whether a port-based VLAN is configured as a voice VLAN.

Jumbo

Indicates whether a VLAN is configured for jumbo packets. For more on jumbos, see "Port Traffic Controls" in the management and configuration guide for your switch.

Mode

Indicates whether a VLAN is tagged or untagged.

Displaying VLAN ports (cumulative listing)

```
switch(config)#show vlan ports a1-a24

Status and Counters - VLAN Information - for ports A1-A24

VLAN ID Name                 | Status     Voice Jumbo
------- -------------------- + ---------- ----- -----
1       DEFAULT_VLAN         | Port-based No    No
10      VLAN_10              | Port-based Yes   No
15      VLAN_15              | Protocol   No    No
```

Displaying VLAN ports (detailed listing)

```
switch(config)#show vlan ports a1-a3 detail
```

```
Status and Counters - VLAN Information - for ports A1

VLAN ID Name                 | Status     Voice Jumbo Mode
------- -------------------- + ---------- ----- ----- --------
1       DEFAULT_VLAN         | Port-based No    No    Untagged
10      VLAN_10              | Port-based Yes   No    Tagged

Status and Counters - VLAN Information - for ports A2

VLAN ID Name                 | Status     Voice Jumbo Mode
------- -------------------- + ---------- ----- ----- --------
1       DEFAULT_VLAN         | Port-based No    No    Untagged
20      VLAN_20              | Protocol   No    No    Untagged

Status and Counters - VLAN Information - for ports A3

VLAN ID Name                 | Status     Voice Jumbo Mode
------- -------------------- + ---------- ----- ----- --------
1       DEFAULT_VLAN         | Port-based No    No    Untagged
33      VLAN_33              | Port-based No    No    Tagged
```

Viewing the configuration for a particular VLAN (CLI)

Syntax:

```
show vlans <vlan-id>
```

Uses the VID to identify and display the data for a specific static or dynamic VLAN.

The following describes the fields displayed with this command (see example output):

802.1Q VLAN ID

The VLAN identification number, or VID.

Name

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

Status

Port-Based

Port-Based, static VLAN.

Protocol

Protocol-Based, static VLAN

Dynamic

Port-Based, temporary VLAN learned through GVRP. See **GVRP** on page 66.

Voice

Indicates whether a port-based VLAN is configured as a voice VLAN. See **Using voice VLANs** on page 53.

Jumbo

Indicates whether a VLAN is configured for Jumbo packets. For more on jumbos, see "Port Traffic Controls" in the management and configuration guide for your switch.

Port Information

Lists the ports configured as members of the VLAN.

DEFAULT

Shows whether a port is a tagged or untagged member of the listed VLAN.

Unknown VLAN

Shows whether the port can become a dynamic member of an unknown VLAN for which it receives an advertisement. GVRP must be enabled to allow dynamic joining to occur.

Status

Shows whether the port is participating in an active link.

Displaying information for a specific static VLAN

```
switch(config)#show vlans 22

Status and Counters - VLAN Information - VLAN 22

 VLAN ID : 22
 Name : VLAN22
 Status : Port-based
 Voice : Yes
 Jumbo : No

 Port Information Mode     Unknown VLAN Status
 ---------------- -------- ------------ ----------
 12               Untagged Learn        Up
 13               Untagged Learn        Up
 14               Untagged Learn        Up
 15               Untagged Learn        Down
 16               Untagged Learn        Up
 17               Untagged Learn        Up
 18               Untagged Learn        Up
```

Displaying information for a specific dynamic VLAN

The following example shows the information displayed for a specific dynamic VLAN. The `show vlans` command lists this data when GVRP is enabled and at least one port on the switch has dynamically joined the designated VLAN.

```
switch(config)# show vlans 22

 Status and Counters - VLAN Information - VLAN 22

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

  Port Information Mode     Unknown VLAN Status
  ---------------- -------- ------------ ----------
  6                Auto     Learn        Up
```

Customizing the show VLANs output (CLI)

Syntax

```
show vlans custom [port <port-list>] <column-list>
```

Specifies the order you want information to display for the `show vlans` command. Displays information for one port or a range of ports. If *<port-list>* is not specified, all ports display.

Fields that can be included in the customized display:

Field	Display	Example	Default width
id	VLAN id	5	6
name	VLAN name	Vlan55	32
status	Status	Port-based	10
voice	Voice enabled	No	5
jumbo	Jumbos enabled	No	5
ipconfig	How the IP address was configured	Manual Disabled DHCP/BootP	10
ipaddr (IPv4) ipaddr (IPv6)	The IP addresses	10.10.10.3 fe80::212:79ff:fe8d:8000	15 for IPv4 46 for IPv6
ipmask	The subnet masks	255.255.255.6/64 (prefix for IPv6 is in format "/XX")	15
proxyarp	Whether proxy ARP is configured	No	5
localproxyarp	Whether local proxy ARP is configured	No	9
state	"Up" if at least one port is up	Up	5

Customizing the VLAN display

The following example displays `id` at its default width and `name:20` allows up to 20 characters of the VLAN `name` to be displayed. The columns selected for display are separated by spaces.

If the width of the column requested is smaller than the header name of the column, the display of the header name is truncated.

```
switch(config)# show vlan custom A1-A3 id name:20 ipaddr state

Status and Counters - VLAN Information - Custom view
```

```
VLANID VLAN name            IP Addr                            State
------ -------------------- ---------------------------------  -----
1      DEFAULT_VLAN         15.255.134.74                      Up
33     Vlan33               10.10.10.01                        Up
44     Vlan44               15.255.164.13                      Up
55     Vlan55               15.255.178.2                       Down
                            15.255.178.3
                            15.255.178.4
60     Vlan60               fe80::212:79ff:fe8d:8000%vlan60    Up
```

Wrapping column headers

The total output wraps if it is longer than the terminal width; it is not truncated.

```
switch(config)# show vlan custom id
Status and Counters - VLAN Information - Custom view

 VLANID
 ------
 1
 33
 44

switch(config)# show vlan custom id:2
Status and Counters - VLAN Information - Custom view

 VL
 --
 1
 33
 44
```

Using pattern matching with the show VLANs custom command

If a pattern matching command is in a search for a field in the output of the `show vlan custom` command and it produces an error, the error message may not be visible. For example, if you enter a command with the pattern matching `include` option that contains an error (such as 'vlan' is misspelled) as in the following example, the output may be empty:

```
switch(config)# show vlans custom 1-3 name vlun include vlan1
```

Hewlett Packard Enterprise recommends that you try the `show vlans custom` command first to ensure that there is output and then enter the command again with the pattern matching option.

Creating an alias for show VLAN commands (CLI)

Create an alias for a frequently used `show vlans custom` command to avoid entering the selected columns each time you use the command.

Using a VLAN alias

```
switch(config)# alias showvlanstatus = “show vlan custom A1-A3 id name:20 status”

switch(config)# show vlan status
Status and Counters - VLAN Information - Custom view
```

```
VLANID VLAN name            Status
------ -------------------- ----------
1      DEFAULT_VLAN         Port-based
33     Vlan33               Port-based
```

Configuring a VLAN MAC address with heartbeat interval

When installing routing switches in the place of existing routers in a network configuration, you can achieve Layer 3 VLAN migration by using the `ip-recv-mac-address` command at the VLAN configuration level to:

- Configure the MAC address of the previously installed router on each VLAN interface of a routing switch.
- Optionally configure the time interval to use for sending heartbeat packets with the configured MAC address.

Syntax

```
ip-recv-mac-address <mac-address> interval <seconds>
```

```
no ip-recv-mac-address <mac-address> interval <seconds>
```

Configures a VLAN interface with the specified MAC address. Enter the `no` version of the command to remove the configured MAC address and return to the original MAC address of the switch.

Parameters

`interval` ***`<seconds>`***

(Optional) Configures the time interval in seconds used between transmissions of heartbeat packets to all network devices configured on the VLAN. Valid values are from one to 255 seconds.

Default: 60 seconds.

Displaying a VLAN MAC address configuration (CLI)

Syntax:

```
show ip-recv-mac-address
```

Displaying a VLAN MAC address

```
switch# show ip-recv-mac-address

VLAN L3-Mac-Address Table

VLAN                           L3-Mac-Address                Timeout
-------------                  ------------------------      -----------
DEFAULT_VLAN                  001635-024467                 60
VLAN2                         001635-437529                 100
```

Using voice VLANs

Configuring voice VLANs separates voice traffic from data traffic and shields your voice traffic from broadcast storms.

Operating rules for voice VLANs

- You must statically configure voice VLANs. GVRP and dynamic VLANs do not support voice VLAN operation.
- Configure all ports in a voice VLAN as tagged members of the VLAN. This ensures retention of the QoS (Quality of Service) priority included in voice VLAN traffic moving through your network.
- If a telephone connected to a voice VLAN includes a data port used for connecting other networked devices (such as PCs) to the network, then you must configure the port as a tagged member of the voice VLAN and a tagged or untagged member of the data VLAN you want the other networked device to use.

Components of voice VLAN operation

- Voice VLAN: Configure one or more voice VLANs on the switch. Some reasons for having multiple voice VLANs include:
  - Employing telephones with different VLAN requirements
  - Better control of bandwidth usage
  - Segregating telephone groups used for different, exclusive purposes

  Where multiple voice VLANs exist on the switch, you can use routing to communicate between telephones on different voice VLANs.
- Tagged/Untagged VLAN Membership: If the appliances using a voice VLAN transmit tagged VLAN packets, then configure the member ports as tagged members of the VLAN. Otherwise, configure the ports as untagged members.

Voice VLAN access security

You can use port security configured on an individual port or group of ports in a voice VLAN. That is, you can allow or deny access to a phone having a particular MAC address. See chapter titled "Configuring and Monitoring Port Security" in the *Access Security Guide* for your switch.

NOTE: MAC authentication is not recommended in voice VLAN applications.

Prioritizing voice VLAN QoS (Optional)

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

- If the ports in a voice VLAN are not tagged members, the switch forwards all traffic on that VLAN at "normal" priority.
- If the ports in a voice VLAN are tagged members, then the switch forwards all traffic on that VLAN at whatever priority the traffic has when received inbound on the switch.

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

Syntax:

```
vlan <vid> qos priority <0-7>
```

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

If you configure a voice VLAN with a VID of 10 and want the highest priority for all traffic on this VLAN, execute the following commands:

```
switch(config)# vlan 10 qos priority 4
switch(config)# write memory
```

You also have the option of resetting the DSCP (DiffServe Codepoint) on tagged voice VLAN traffic moving through the switch. For more information, see **Quality of Service (QoS): Managing bandwidth effectively** on page 178.

If all port memberships on the voice VLAN are tagged:

- The priority level set for voice VLAN traffic is carried to the next device.
- You can enforce a QoS priority policy moving through the switch and network.

For more information, see **Using voice VLANs** on page 53.

Special VLAN types

VLAN support and the default VLAN

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

- You can partition the switch into multiple virtual broadcast domains by configuring one or more additional VLANs and moving ports from the default VLAN to the new VLANs.
- The switch supports up to 2048 static and dynamic VLANs, with VIDs numbered up to 4094. You can change the name of the default VLAN, but not its VID, which is always 1.
- You can remove all ports from the default VLAN by placing them in another port-based VLAN, but this VLAN remains and cannot be deleted from the switch.

For details on port VLAN settings, see **Configuring or changing static VLAN per-port settings (CLI)** on page 37.

The primary VLAN

As certain features and management functions run on only one VLAN in the switch and because DHCP and Bootp can run per-VLAN, there is a need for a dedicated VLAN to manage these features and ensure that multiple instances of DHCP or Bootp on different VLANs do not result in conflicting configuration values for the switch.

The Primary VLAN is the VLAN the switch uses to run and manage these features and data. In the factory-default configuration, the switch designates the default VLAN (DEFAULT_VLAN; VID=1) as the Primary VLAN. However you can designate another static, port-based VLAN as primary.

To summarize, designating a non-default VLAN as primary means that:

- The switch reads DHCP responses on the Primary VLAN instead of on the default VLAN. This includes such DHCP-resolved parameters as the TimeP server address, Default TTL and IP addressing—including the Gateway IP address—when the switch configuration specifies DHCP as the source for these values.
- The default VLAN continues to operate as a standard VLAN you cannot delete it or change its VID.
- Any ports not specifically assigned to another VLAN will remain assigned to the Default VLAN, even if it is the Primary VLAN.

Candidates for Primary VLAN include any static, port-based VLAN currently configured on the switch.

Protocol-Based VLANs and dynamic (GVRP-learned) VLANs that have not been converted to a static VLAN cannot be the Primary VLAN. To display the current Primary VLAN, use the CLI `show vlan` command.

NOTE: If you configure a non-default VLAN as the Primary VLAN, you cannot delete that VLAN unless you first select a different VLAN to serve as primary.

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

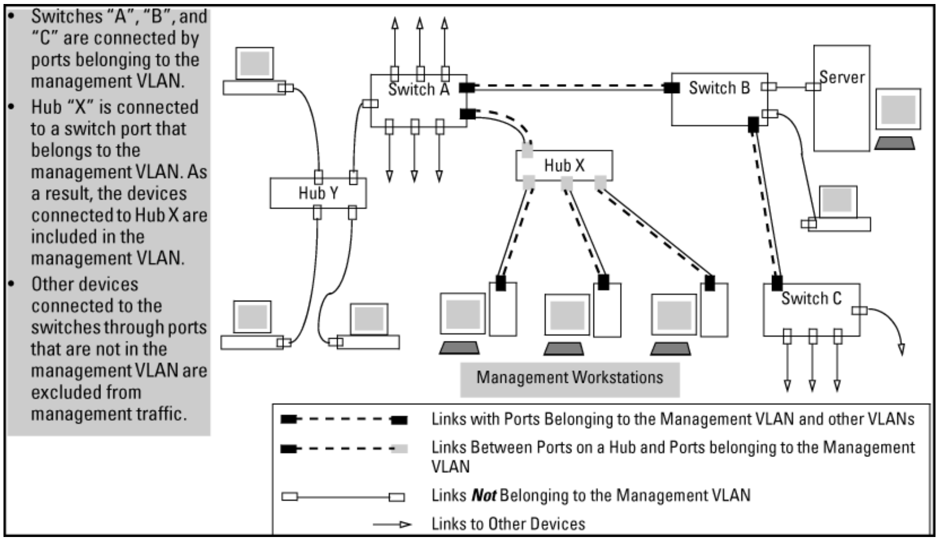
The secure Management VLAN

Configuring a secure Management VLAN creates an isolated network for managing the switches that support this feature. Access to a secure Management VLAN and the switch's management functions is available only through ports configured as members.

- Multiple ports on the switch can belong to the Management VLAN. This allows connections for multiple management stations to the Management VLAN, while allowing Management VLAN links between switches configured for the same Management VLAN.
- Only traffic from the Management VLAN can manage the switch, which means that only the workstations and PCs connected to ports belonging to the Management VLAN can manage and reconfigure the switch.

Potential security breaches in a network

This illustrates use of the Management VLAN feature to support management access by a group of management workstations.



- Switches “A”, “B”, and “C” are connected by ports belonging to the management VLAN.
- Hub “X” is connected to a switch port that belongs to the management VLAN. As a result, the devices connected to Hub X are included in the management VLAN.
- Other devices connected to the switches through ports that are not in the management VLAN are excluded from management traffic.

Switch A
Switch B
Server
Hub X
Hub Y
Switch C
Management Workstations
Links with Ports Belonging to the Management VLAN and other VLANs
Links Between Ports on a Hub and Ports belonging to the Management VLAN
Links *Not* Belonging to the Management VLAN
Links to Other Devices

Management VLAN control in a LAN

In this example, Workstation 1 has management access to all three switches through the Management VLAN, while the PCs do not. This is because configuring a switch to recognize a Management VLAN automatically excludes attempts to send management traffic from any other VLAN.

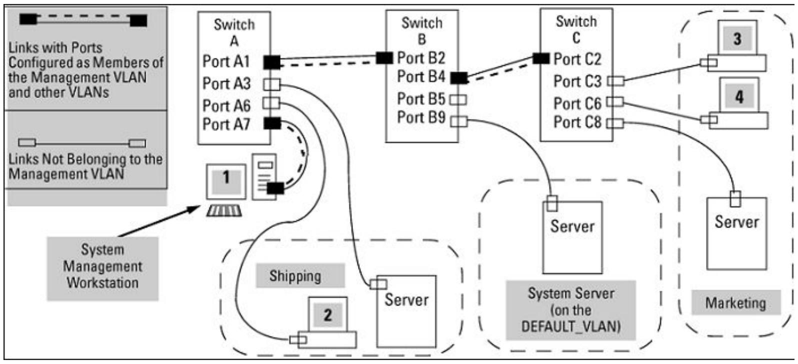

Links with Ports Configured as Members of the Management VLAN and other VLANs
Links Not Belonging to the Management VLAN
System Management Workstation
Switch A
Port A1
Port A3
Port A6
Port A7
1
Switch B
Port B2
Port B4
Port B5
Port B9
Switch C
Port C2
Port C3
Port C6
Port C8
3
4
Shipping
2
Server
Server
System Server (on the DEFAULT_VLAN)
Server
Marketing

Table 5: *VLAN membership in Management VLAN control in a LAN*

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

See **Configuring a secure Management VLAN (CLI)** on page 42 for configuration details.

Operating notes for Management VLANs

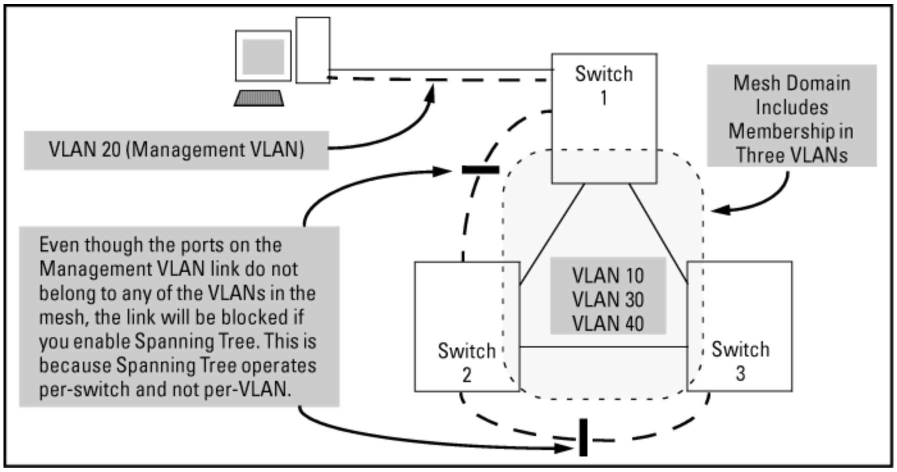
- Use only a static, port-based VLAN for the Management VLAN.
- The Management VLAN feature applies to both IPv4 and IPv6 traffic.
- The Management VLAN does not support IGMP operation.
- Routing between the Management VLAN and other VLANs is not allowed.
- If there are more than 25 VLANs configured on the switch, reboot the switch after configuring the Management VLAN.
- Only one Management VLAN can be active in the switch. If one Management VLAN VID is saved in the startup-config file and you configure a different VID in the running-config file, the switch uses the running-config version until you either use the `write-memory` command or reboot the switch.
- During a Telnet session to the switch, if you configure the Management VLAN to a VID that excludes the port through which you are connected to the switch, you will continue to have access only until you terminate the session by logging out or rebooting the switch.

NOTE: The Management VLAN feature does not control management access through a direct connection to the switch's serial port.

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

- Enabling Spanning Tree between a pair of switches where there are multiple links using separate VLANs, including the Management VLAN, will force the blocking of one or more links. This may include the link carrying the Management VLAN, which will cause loss of management access to some devices.
- Monitoring Shared Resources: The Management VLAN feature shares internal switch resources with several other features. The switch provides ample resources for all features. However, if the internal resources become fully subscribed, the Management VLAN feature cannot be configured until the necessary resources are released from other uses. For information on determining the current resource availability and usage, see the appendix titled "Monitoring Resources" in the *ArubaOS-Switch Management and Configuration Guide* for your switch.

Inadvertently blocking a Management VLAN link by implementing spanning tree


Switch 1
Mesh Domain Includes Membership in Three VLANs
VLAN 20 (Management VLAN)
Even though the ports on the Management VLAN link do not belong to any of the VLANs in the mesh, the link will be blocked if you enable Spanning Tree. This is because Spanning Tree operates per-switch and not per-VLAN.
VLAN 10
VLAN 30
VLAN 40
Switch 2
Switch 3

VLAN operating notes

DHCP/Bootp

If you are using DHCP/Bootp to acquire the switch's configuration, packet time-to-live and TimeP information, designates the VLAN on which DHCP is configured as the Primary VLAN.

NOTE:

In the factory-default configuration, the DEFAULT_VLAN is the Primary VLAN.

Per-VLAN features

IGMP and some other features operate on a per VLAN basis. This means you must configure such features separately for each VLAN in which you want them to operate.

Default VLAN

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

VLAN port assignments

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

Voice-Over-IP (VoIP)

VoIP operates only over static, port-based VLANs.

Multiple VLAN types configured on the same port

A port can simultaneously belong to both port-based and protocol-based VLANs.

Protocol Capacity

A protocol-based VLAN can include up to four protocol types. In protocol VLANs using the IPv4 protocol, to support normal IP network operation ARP must be one of these protocol types (to support normal IP network operation). Otherwise, IP traffic on the VLAN is disabled.

If you configure an IPv4 protocol VLAN that does not include the ARP VLAN protocol, the switch displays the following message which indicates a protocol VLAN configured with IPv4 but not ARP:

```
switch(config)# vlan 97 protocol ipv4

IPv4 assigned without ARP, this may result in undeliverable IP packets.
```

Deleting Static VLANs

A VLAN can be deleted even if there are currently ports belonging to it. The ports are moved to the default VLAN.

Adding or Deleting VLANs

Changing the number of VLANs supported on the switch, requires a reboot.

NOTE:

From the CLI, you must perform a `write memory` command before rebooting. Other VLAN configuration changes are dynamic.

Effects of VLANs on other switch features

Spanning Tree operation with VLANs

Depending on the spanning tree option configured on the switch, the spanning tree feature may operate as:

- A single instance across all ports on the switch regardless of VLAN assignments
- Multiple instances per-VLAN

For single-instance operation, if redundant physical links exist between the switch and another 802.1Q device, all but one link will be blocked, even if the redundant links are in separate VLANs. In this case, you can use port trunking to prevent Spanning Tree from unnecessarily blocking ports (and to improve overall network performance). For multiple-instance operation, physically redundant links belonging to different VLANs can remain open. For more information, see **Multiple instance spanning tree operation**.

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

Spanning Tree operates differently in different devices

IP interfaces

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

VLAN MAC address

The switches have one unique MAC address for all of their VLAN interfaces. You can send an 802.2 test packet to this MAC address to verify connectivity to the switch and you can assign an IP address to the VLAN interface. When you Ping that address, ARP will resolve the IP address to this single MAC address.

In a topology where a switch has multiple VLANs and must be connected to a device having a single forwarding database, some cabling restrictions apply. For more on this topic, see **Multiple VLAN considerations** on page 30.

Port trunks

When assigning a port trunk to a VLAN, all ports in the trunk are automatically assigned to the same VLAN. A port trunk is tagged, untagged, or excluded from a VLAN the same way as individual, untrunked ports.

Port monitoring

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

Jumbo packet support

Jumbo packet support is enabled per-VLAN and applies to all ports belonging to the VLAN. For more information, see the chapter titled "Port Traffic Controls" in the *ArubaOS-Switch Management and Configuration Guide* for your switch.

VLAN restrictions

- A port must be a member of at least one VLAN. In the factory default configuration, all ports are assigned to the default VLAN (DEFAULT_VLAN; VID=1).
- A port can be a member of one untagged, port-based VLAN. All other port-based VLAN assignments for that port must be tagged. The "Untagged" designation enables VLAN operation with non 802.1Q-compliant devices.
- A port can be an untagged member of one protocol-based VLAN of each protocol type. When assigning a port to multiple, protocol-based VLANs sharing of the same type, the port can be an untagged member of only one such VLAN.
- With routing enabled on the switch, the switch can route traffic between:
  - Multiple, port-based VLANs
  - A port-based VLAN and an IPv4 protocol-based VLAN
  - A port-based VLAN and an IPv6 protocol-based VLAN
  - An IPv4 protocol-based VLAN and an IPv6 protocol VLAN

  Other, routable, protocol-based VLANs must use an external router to move traffic between VLANs. With routing disabled, all routing between VLANs must be through an external router.
- Before deleting a static VLAN, t first reassign all ports in the VLAN to another VLAN. You can use the `no vlan <vid>` command to delete a static VLAN. For more information, see **Creating a new static VLAN (port-based or protocol-based) (CLI)** on page 35.
- Protocol-based VLANs, port-based VLANs and LLDP radio port VLANs cannot run concurrently with RPVST+.

Migrating Layer 3 VLANs using VLAN MAC configuration

Switches provide for maintaining Layer 3 VLAN configurations when migrating distribution routers in networks not centrally managed, by configuring the MAC address of the previous router on the VLAN interfaces of the routing switch.

VLAN MAC address reconfiguration

Switches use one unique MAC address for all VLAN interfaces. If you assign an IP address to a VLAN interface, ARP resolves the IP address to the MAC address of the routing switch for all incoming packets.

The Layer 3 VLAN MAC Configuration feature lets you reconfigure the MAC address used for VLAN interfaces, using the CLI. Packets addressed to the reconfigured Layer 3 MAC address, such as ARP and IP data packets, are received and processed by the routing switch.

Packets transmitted from the routing switch (packets originating from the router and forwarded packets) use the original Switch MAC address as the source MAC address in Ethernet headers.

ARP reply packets use the reconfigured MAC address in both the:

- ARP Sender MAC address field
- Source MAC address field in the Ethernet frame header

When reconfiguring the MAC address, you may specify a keepalive timeout to transmit heartbeat packets that advertise the new MAC address

By configuring the MAC address of the previously installed router as the MAC address of each VLAN interface on the Switch, you can swap the physical port of a router to the Switch after the switch has been properly configured in the network.

Handling incoming and outgoing VLAN Traffic

Incoming VLAN data packets and ARP requests

These are received and processed on the routing switch according to the MAC address of the previously installed router configured for each VLAN interface.

Outgoing VLAN traffic

This uses the MAC address of the switch as the source MAC address in packet headers. The MAC address configured on VLAN interfaces is not used on outbound VLAN traffic.

When the routing switch receives an ARP request for the IP address configured on a VLAN interface, the ARP reply uses the reconfigured MAC address in both the:

- ARP Sender MAC address field
- Source MAC address field in the Ethernet frame header

When proxy ARP is enabled on a VLAN interface, the ARP reply sent for an ARP request received from VLAN devices located outside the directly connected IP subnets also contains the reconfigured MAC address in both the:

- ARP Sender MAC address field
- Source MAC address field in the Ethernet frame header

NOTE: The Virtual Router Redundancy Protocol (VRRP) is not supported on VLAN interfaces on which the MAC address for incoming traffic has been reconfigured.

To hosts in the network, VLAN traffic continues to be routed (using the reconfigured MAC address as destination address), but outbound VLAN traffic appears to be sent from another router attached to the same subnet (using the Switch MAC address as source address) attached to the same subnet. Although it appears as an asymmetric path to network hosts, the MAC address configuration feature enables Layer 3 VLAN migration. (A successful VLAN migration is achieved because the hosts do not verify that the source MAC address and the destination MAC address are the same when communicating with the routing switch.)

Sending heartbeat packets with a configured MAC Address

On the VLAN interfaces of a routing switch, the user-defined MAC address only applies to inbound traffic. As a result, any connected switches need to learn the new address that is included in the Ethernet frames of outbound VLAN traffic transmitted from the routing switch.

If a connected switch does not have the newly configured MAC address of the routing switch as a destination in its MAC address table, it floods packets to all of its ports until a return packet allows the switch to learn the correct destination address. As a result, the performance of the switch is degraded as it tries to send Ethernet packets to an unknown destination address.

To allow connected switches to learn the user-configured MAC address of a VLAN interface, the routing switch can send periodic heartbeat-like Ethernet packets. The Ethernet packets contain the configured MAC address as the source address in the packet header. IP multicast packets or Ethernet service frames are preferred because they do not interrupt the normal operation of client devices connected on the segment.

Because the aging time of destination addresses in MAC address tables varies on network devices, you must also configure a time interval to use for sending heartbeat packets.

Heartbeat packets are sent at periodic intervals with a specific Switch unicast MAC address in the destination field. This MAC address is assigned to the Switch and is not used by other non- routers. Because the heartbeat packet contains a unicast MAC address, it does not interrupt host operation. Even if you have multiple 1-65 Static Virtual LANs (VLANs) Introducing tagged VLAN technology into networks running untagged VLANs switches connected to the network, there is no impact on network performance because each switch sends heartbeat packets with its configured MAC address as the destination address.

The format of a heartbeat packet is an extended Ethernet OUI frame with an extended OUI Ethertype (88B7) and a new protocol identifier in the 5-octet protocol identifier field.

Displaying a VLAN MAC address configuration (CLI)

Syntax:

```
show ip-recv-mac-address
```

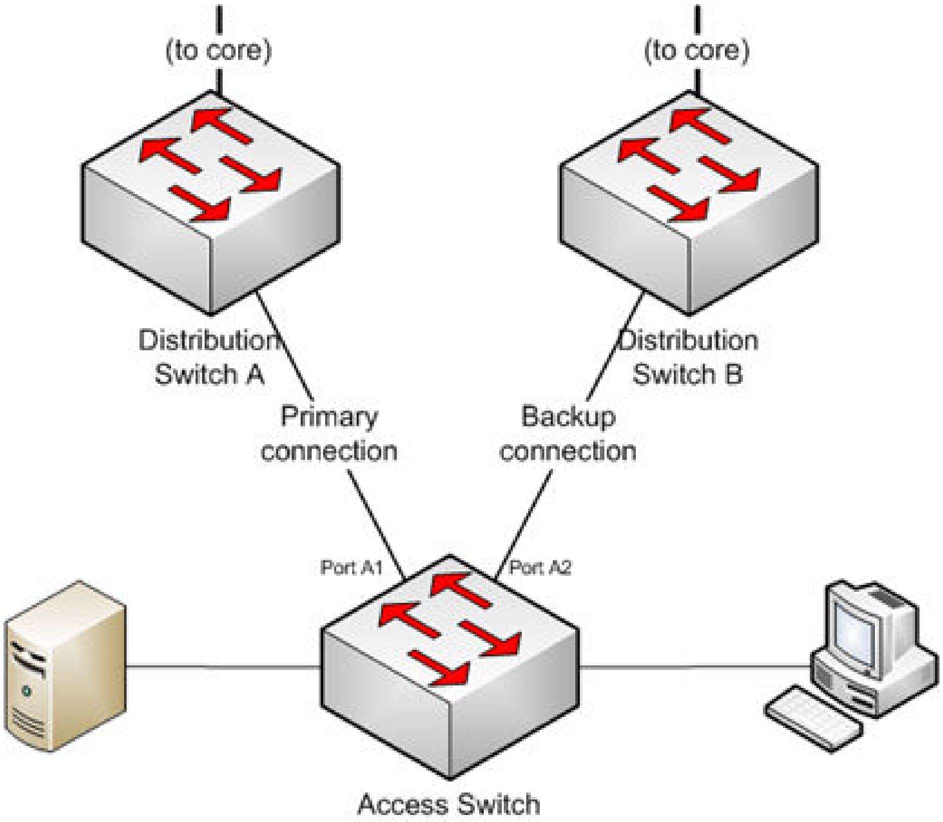
Displaying a VLAN MAC address

```
switch# show ip-recv-mac-address

VLAN L3-Mac-Address Table

VLAN                           L3-Mac-Address              Timeout
-------------                  ------------------------    -----------
DEFAULT_VLAN                   001635-024467               60
VLAN2                          001635-437529               100
```

Smart Link

Introduction to Smart Link

Smart link is a switch feature that provides effective, simple, and fast-converging link redundancy in network topology with dual uplink between different layers of the network. It requires an active (master) and a backup (slave) link. The active link carries the uplink traffic. Upon failure of the active link, a switchover is triggered and the traffic is directed to the backup link.



- In the previous figure, ports A1 and A2 are configured as part of a Smart link group. The connection from the access switch to Distribution Switch A is the master, and the connection from the access switch to Distribution Switch B is the slave.
- Only the master interface forwards traffic for a group of VLANs (referred to as protected VLAN group).
- The other interface is in standby mode for this protected group. If port A1 goes down, port A2 starts forwarding traffic for this protected VLAN group.
- If port A1 comes back up, it goes to standby mode and does not forward traffic. Port A2 continues forwarding traffic. This is the case if preemption-mode is configured as “role”. If preemption-mode is not configured as “role”, when the master (A1) comes back up, it becomes Active (forwarding) after the configured ‘preemption-delay’.
- Since a Smart link group has the information readily available via configuration as to which port should be forwarding for the protected VLAN group in the case of the active link failure, the failover is much quicker when compared with STP.

Configuring Smart Link

Use the following commands to configure Smart link:

- Create a smart link group, using optional parameters as necessary:

```
switch(config)# no smart-link group group-id
```

Parameters

```
master port
```

```
slave port
```

```
protected-vlans vid-list
```

```
send-control-vlan vid
```

```
preemption-mode {off | forced | bandwidth}
```

```
preemption-delay 10...max
```

```
trap {enable | disable}
```

NOTE: When executed without any parameters, this command enters into Smart link group context.

- Configure VLANs to receive flush messages. This is an interface level command. It must be executed for both the master and the slave port.

  ```
  switch(config)# smart-link recv-control-vlan vid-list
  ```

Enable debug messages for a Smart link group:

```
switch(config)#debug smart-link [group group-id | all] [flush-packets]
```

Configuration example

The following example illustrates Smart link configuration with VLAN load-balancing:

- vlans 1-10 mapped to smart-link group 1:

  ```
  switch(config)#smart-link group 1 master a1 slave a2
  switch(config)#smart-link group 1 protected-vlans 1-10
  switch(config)#smart-link group 1 send-control-vlan 1
  switch(config)#smart-link group 1 preemption-mode role
  switch(config)#smart-link group 1 preemption-delay 10
  ```

- vlans 11-20 mapped to smart-link group 2:

  ```
  switch(config)#smart-link group 2 master a2 slave a1
  switch(config)#smart-link group 2 protected-vlans 11-20
  switch(config)#smart-link group 2 send-control-vlan 10
  switch(config)#smart-link group 2 preemption-mode role
  switch(config)#smart-link group 2 preemption-delay 15
  ```

Viewing Smart Link information

Smart link supports the following show commands:

- Show the Smart link group information. Detailed output is displayed if group is specified, otherwise only basic information is displayed for all groups.

  ```
  show smart-link group {group-id | all}
  ```

The H3C-equivalent display command is:

```
display smart-link group [group-id | all] [begin | exclude | include] regular-expression
```

- Show statistics of received flush packets

```
show smart-link flush-statistics
```

The H3C-equivalent display command is:

```
display smart-link flush [begin | exclude | include] regular-expression
```

- Show receive control VLANs configured on per port basis:

```
show smart-link recv-control-vlans
```

Clearing statistics

Use the following command to clear group and flush statistics

```
clear smart-link [flush-statistics] [group group-id | all]
```

About GVRP

GVRP (GARP VLAN Registration Protocol) is an application of GARP (Generic Attribute Registration Protocol.) It enables a switch to dynamically create 802.1Q-compliant VLANs on links with other devices running GVRP and automatically create VLAN links between GVRP-aware devices. (A GVRP link can include intermediate devices that are not GVRP-aware.) This operation reduces the chance for errors in VLAN configurations by automatically providing VID (VLAN ID) consistency across the network. After the switch creates a dynamic VLAN, the CLI `static <vlan-id>` command can be used to convert it to a static VLAN. GVRP can also be used to dynamically enable port membership in static VLANs configured on a switch.

GVRP uses GVRP BPDUs (GVRP Bridge Protocol Data Units) to advertise static VLANs; this a GVRP BPDU is called an **advertisement**. On a switch, advertisements are sent outbound from ports to the devices directly connected to those ports.

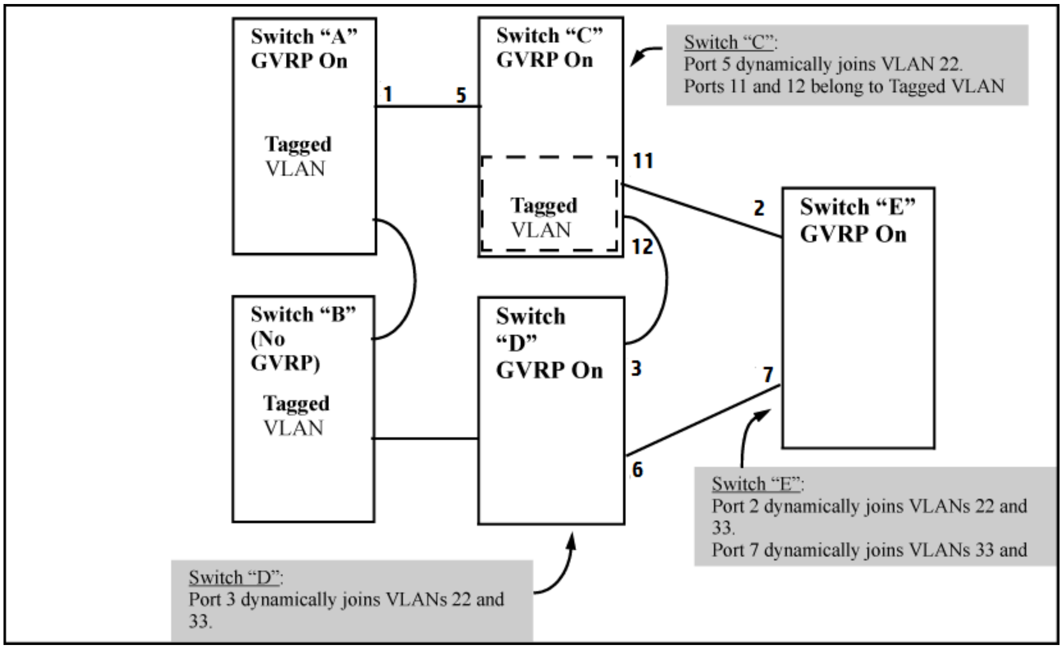
GVRP operational rules

- A dynamic VLAN must be converted to a static VLAN before it can have an IP address.
- For the switches covered in this guide, GVRP can be enabled only if `max-vlans` is set to no more than 512 VLANs.
- The total number of VLANs on the switch (static and dynamic combined) cannot exceed the current Maximum VLANs setting. For example, in the factory default state, the switch supports up to 256 VLANs. Any additional VLANs advertised to the switch will not be added unless you first increase the Maximum VLANs setting. In the global config level of the CLI, use **max-vlans**.
- Converting a dynamic VLAN to a static VLAN and then executing the `write memory` command saves the VLAN in the startup-config file and makes it a permanent part of the switch's VLAN configuration.
- Within the same broadcast domain, a dynamic VLAN can pass through a device that is notGVRP-aware. This is because a half-duplex repeater, a hub, or a switch that is not GVRP-aware will flood the GVRP (multicast) advertisement packets out all ports.
- GVRP assigns dynamic VLANs as tagged VLANs. To configure the VLAN as untagged, convert it to a static VLAN.
- Rebooting a switch on which a dynamic VLAN exists deletes that VLAN. However, the dynamic VLAN reappears after the reboot if GVRP is enabled and the switch again receives advertisements for that VLAN through a port configured to add dynamic VLANs.
- By receiving advertisements from other devices running GVRP, the switch learns of static VLANs on those other devices and dynamically (automatically) creates tagged VLANs on the links to the advertising devices. Similarly, the switch advertises its static VLANs to other GVRP-aware devices, as well as the dynamic VLANs the switch has learned.
- A GVRP-enabled switch does not advertise any GVRP-learned VLANs out of the ports on which it originally learned of those VLANs.
- While GVRP is enabled on the switch, you cannot apply any ACLs to VLANs configured on the same switch.
- A VLAN enabled for jumbo traffic cannot be used to create a dynamic VLAN; a port belonging to a statically configured, jumbo-enabled VLAN cannot join a dynamic VLAN.

Example of GVRP operation

In the following example, Tagged VLAN ports on switch A and switch C advertise VLANs 22 and 33 to ports on other GVRP-enabled switches that can dynamically join the VLANs.

GVRP operation



Switch "C":
Port 5 dynamically joins VLAN 22.
Ports 11 and 12 belong to Tagged VLAN

Switch "E":
Port 2 dynamically joins VLANs 22 and 33.
Port 7 dynamically joins VLANs 33 and

Switch "D":
Port 3 dynamically joins VLANs 22 and 33.

Options for a GVRP-aware port receiving advertisements

- If there is not already a static VLAN with the advertised VID on the receiving port, such a port can dynamically create the VLAN and become a member.
- If the switch already has a static VLAN assignment with the same VID as in the advertisement and the port is configured to `Auto` for that VLAN, then the port will dynamically join the VLAN and begin moving that VLAN's traffic. For more detail on `Auto`, see **Enabling a port for dynamic joins**.
- Ignore the advertisement for that VID.
- Not participate in that VLAN.

Options for a port belonging to a Tagged or Untagged static VLAN

- Send VLAN advertisements
- Receive advertisements for VLANs on other ports and dynamically join those VLANs.
- Send VLAN advertisements but ignore advertisements received from other ports.
- Avoid GVRP participation by not sending advertisements and dropping any advertisements received from other devices.

IP addressing

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

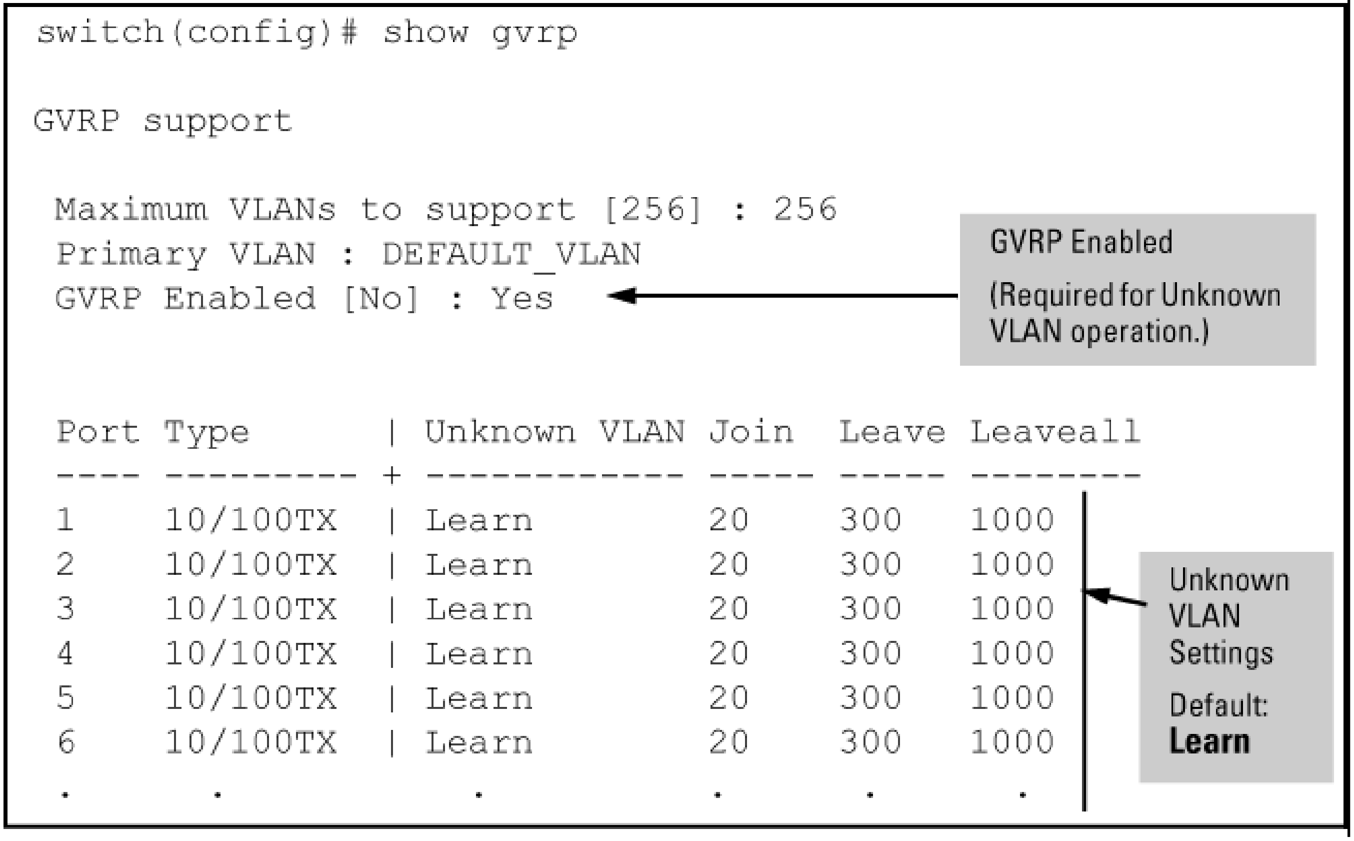
Per-port options for handling GVRP "unknown VLANs"

An "unknown VLAN" is a VLAN that the switch learns of by receiving an advertisement for that VLAN on a port that is not already a member of that VLAN. If the port is configured to learn unknown VLANs, then the VLAN is dynamically created and the port becomes a tagged member of the VLAN.

GVRP unknown VLAN settings

Suppose that in the **Example of GVRP operation**, port 1 on switch A is connected to port 5 on switch C. Because switch A has VLAN 22 statically configured, while switch C does not have this VLAN statically configured (and does not "Forbid" VLAN 22 on port 5), VLAN 22 is handled as an "Unknown VLAN" on port 5 in switch C. Conversely, if VLAN 22 was statically configured on switch C, but port 5 was not a member, port 5 would become a member when advertisements for VLAN 22 were received from switch A.

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



```
switch(config)# show gvrp

GVRP support

 Maximum VLANs to support [256] : 256
 Primary VLAN : DEFAULT_VLAN
 GVRP Enabled [No] : Yes


 Port Type      | Unknown VLAN Join  Leave Leaveall
 ---- --------- + ------------ ----- ----- --------
 1    10/100TX  | Learn        20    300   1000
 2    10/100TX  | Learn        20    300   1000
 3    10/100TX  | Learn        20    300   1000
 4    10/100TX  | Learn        20    300   1000
 5    10/100TX  | Learn        20    300   1000
 6    10/100TX  | Learn        20    300   1000
 .      .          .           .      .      .
```

Per-port options for dynamic VLAN advertising and joining

GVRP must be enabled and VLANs must be configured to one or more switches, depending on the topology.

Initiating advertisements

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

Enabling a port for dynamic joins

You can configure a port to dynamically join a static VLAN. The join will occur if that port subsequently receives an advertisement for the static VLAN. This is done by using the Auto and Learn options described in the table **Controlling VLAN behavior on ports with static VLANs**.

Parameters for controlling VLAN propagation behavior

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

Table 6: *Controlling VLAN behavior on ports with static VLANs*

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

Per-Port "Unknown VLAN" (GVRP) configuration	Static VLAN Options—Per VLAN Specified on Each Port[1]		
	Port Activity: Tagged or Untagged (Per VLAN)[2]	Port Activity: Auto[2] (Per VLAN)	Port Activity: Forbid (Per VLAN)[2]
	• Will not join any advertised VLANs. • Will not advertise VLANs.	• Will not become a member of the specified VLAN. • Will ignore GVRP PDUs. • Will not join any dynamic VLANs. • Will not advertise VLANs.	• Will not join any dynamic VLANs. • Will not advertise VLANs.

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

[2] To configure tagging, Auto, or Forbid, see **Configuring or changing static VLAN per-port settings (CLI)** on page 37.

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

NOTE: In the table above, the Unknown VLAN parameters are configured on a per-port basis using the CLI. The Tagged, Untagged, Auto, and Forbid options are configured per static VLAN on every port.

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

GVRP and VLAN access control

Enabling GVRP allows a port to advertise and join dynamic VLANs. If a port has not received an advertisement for an existing dynamic VLAN during the time-to-live (10 seconds), the port removes itself from that dynamic VLAN.

Advertisements and dynamic joins

When you enable GVRP on a switch, the default GVRP parameter settings allow all of the switch's ports to transmit and receive dynamic VLAN advertisements (GVRP advertisements) and to dynamically join VLANs.

Enabling GVRP:

- Allows a port to both advertise and join dynamic VLANs (Learn mode—the default).
- Allows a port to send VLAN advertisements, but not receive them from other devices; that is, the port cannot dynamically join a VLAN but other devices can dynamically join the VLANs it advertises (Block mode).
- Prevents a port from participating in GVRP operation (Disable mode).

Port-Leave from a dynamic VLAN

A dynamic VLAN continues to exist on a port for as long as the port receives its advertisements from another device connected to that port, or until:

- Converting the VLAN to a static VLAN
- Reconfiguring the port to `Block` or `Disable`
- Disabling GVRP
- Rebooting the switch.

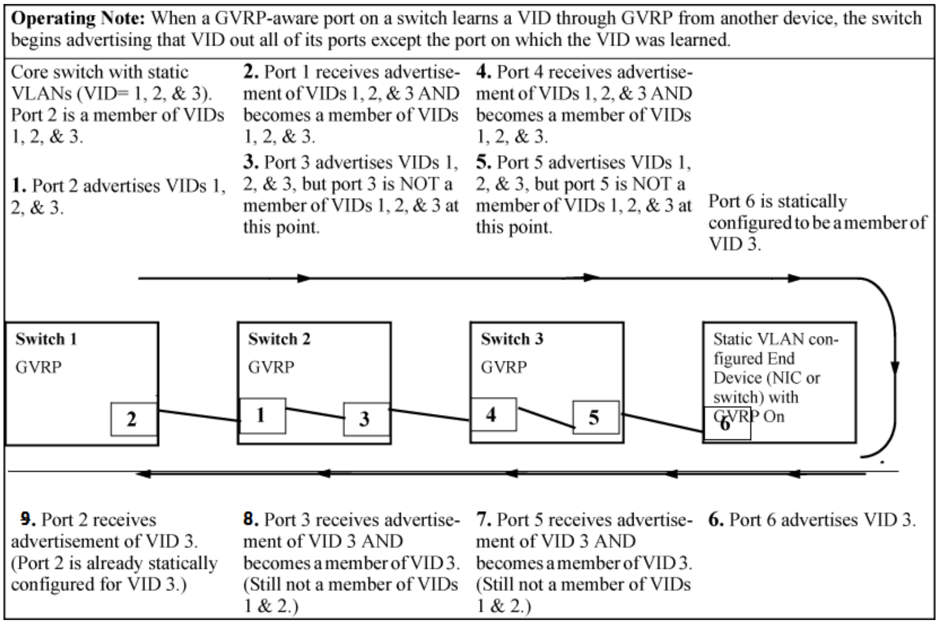
The time-to-live for dynamic VLANs is 10 seconds, if a port has not received an advertisement for an existing dynamic VLAN during that time, the port removes itself from that dynamic VLAN.

Using GVRP

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

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

Figure 10: *Forwarding advertisements and dynamic joining*



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

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

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

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

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

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

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

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

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

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

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

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

6. Port 6 advertises VID 3.

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

NOTE:

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

Planning for GVRP operation

To set up dynamic VLANs for a segment:

Procedure

1. Determine the VLAN topology required for each segment (broadcast domain) on the network.
2. Determine which VLANs must be static and which can be dynamically propagated.
3. Determine the devices on which static VLANs must be manually created to propagate VLANs throughout the segment.
4. Determine security boundaries and how individual ports in the segment are to handle dynamic VLAN advertisements (see **Options for handling unknown VLAN advertisements** and **Controlling VLAN behavior on ports with static VLANs**).
5. Enable GVRP on all devices to be used with dynamic VLANs and configure the appropriate "Unknown VLAN" parameter (Learn, Block, or Disable) for each port.
6. Configure static VLANs on the switches, where needed, with their per-VLAN parameters (Tagged, Untagged, Auto, and Forbid—see **Options for handling unknown VLAN advertisements** and **Controlling VLAN behavior on ports with static VLANs**) on each port.
7. Dynamic VLANs will then appear automatically, according to the chosen configuration options.
8. Convert dynamic VLANs to static VLANs, where dynamic VLANs are to become permanent.

Displaying switch current GVRP configuration (CLI)

Syntax:

```
show gvrp
```

Shows GVRP status (enabled or disabled), current maximum number of VLANs supported and the current Primary VLAN.

Displaying GVRP status with GVRP disabled

```
switch(config)# show gvrp

GVRP support

 Maximum VLANs to support [256] : 256
 Primary VLAN : DEFAULT_VLAN
 GVRP Enabled [No] : No
```

Displaying GVRP status with GVRP enabled

This example shows the output for the `show gvrp` command with GVRP enabled. It includes non-default settings for the Unknown VLAN field for some ports (see Port number 3, 4, 5 below).

```
switch(config)# show gvrp

 GVRP support

  Maximum VLANs to support [256] : 256
  Primary VLAN : DEFAULT_VLAN
  GVRP Enabled [No] : Yes
```

```
Port Type      | Unknown VLAN Join  Leave Leaveall
---- --------- + ------------ ----- ----- --------
1    10/100TX  | Learn       20     300   1000
2    10/100TX  | Learn       20     300   1000
3    10/100TX  | Block       20     300   1000
4    10/100TX  | Disable     20     300   1000
5    10/100TX  | Disable     20     300   1000
6    10/100TX  | Learn       20     300   1000
7    10/100TX  | Learn       20     300   1000
```

Displaying switch current GVRP configuration (CLI)

Syntax:

```
show gvrp
```

Shows GVRP status (enabled or disabled), current maximum number of VLANs supported and the current Primary VLAN.

Displaying GVRP status with GVRP disabled

```
switch(config)# show gvrp

GVRP support

 Maximum VLANs to support [256] : 256
 Primary VLAN : DEFAULT_VLAN
 GVRP Enabled [No] : No
```

Displaying GVRP status with GVRP enabled

This example shows the output for the `show gvrp` command with GVRP enabled. It includes non-default settings for the Unknown VLAN field for some ports (see Port number 3, 4, 5 below).

```
switch(config)# show gvrp

 GVRP support

  Maximum VLANs to support [256] : 256
  Primary VLAN : DEFAULT_VLAN
  GVRP Enabled [No] : Yes


Port Type      | Unknown VLAN Join  Leave Leaveall
---- --------- + ------------ ----- ----- --------
1    10/100TX  | Learn       20     300   1000
2    10/100TX  | Learn       20     300   1000
3    10/100TX  | Block       20     300   1000
4    10/100TX  | Disable     20     300   1000
5    10/100TX  | Disable     20     300   1000
6    10/100TX  | Learn       20     300   1000
7    10/100TX  | Learn       20     300   1000
```

Enabling and disabling GVRP on the switch (CLI)

Syntax:

```
gvrp
```

Enables GVRP on the switch.

```
no gvrp
```

Disables GVRP on the switch.

NOTE:

GVRP can be enabled only if `max-vlans` is set to no more than 256 VLANs. While GVRP is enabled on the switch, you cannot apply any ACLs to VLANs configured on the same switch. A GVRP link can include intermediate devices that are not GVRP-aware. To understand and use GVRP, you need a working knowledge of 802.1Q VLAN tagging. See **802.1Q VLAN tagging** on page 22.

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

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

Controlling how individual ports handle advertisements for new VLANs (CLI)

When GVRP is enabled on the switch, use the `unknown-vlans` command to change the Unknown VLAN field for one or more ports.

Syntax:

```
interface <port-list> unknown-vlans [learn | block | disable]
```

Changes the Unknown VLAN field to control how one or more ports handle advertisements. Use at either the Manager or interface context level for a port.

Changing the Unknown VLANs field

In the following example, the first command changes the configuration to Block, the second command displays the new configuration:

```
switch(config)# interface 1-2 unknown-vlans block

Switch(config)# show gvrp
 GVRP support
  Maximum VLANs to support [256] : 256
  Primary VLAN : DEFAULT_VLAN
  GVRP Enabled [No] : Yes

  Port Type      | Unknown VLAN Join  Leave Leaveall
  ---- --------- + ------------ ----- ----- --------
  1    10/100TX  | Block        20    300   1000
  2    10/100TX  | Block        20    300   1000
  3    10/100TX  | Learn        20    300   1000
  4    10/100TX  | Learn        20    300   1000
```

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

Table 7: *Options for handling unknown VLAN advertisements*

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

Listing static and dynamic VLANs on a GVRP-enabled switch (CLI)

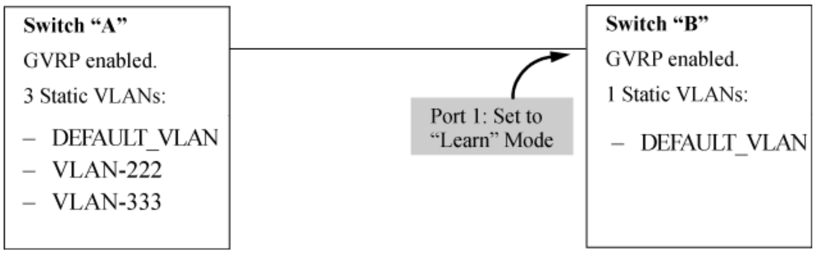
Syntax:

```
show vlans
```

Lists all VLANs present in the switch.

Using the `show vlans` command

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



The `show vlans` command lists the dynamic (and static) VLANs in switch B after it has learned and joined VLAN-222 and VLAN-333.

```
Switch-B> show vlans

 Status and Counters - VLAN Information

   VLAN support : Yes
   Maximum VLANs to support : 16
   Primary VLAN : DEFAULT_VLAN

   VLAN ID        NAME          Status
   -------------- ------------- ------
   1              DEFAULT_VLAN  Static
   222              GVRP_222       Dynamic
   333              GVRP_333       Dynamic
```

Converting a Dynamic VLAN to a Static VLAN (CLI)

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

```
static-vlan <dynamic-vlan-id>
```

Converting a dynamic VLAN 333 to a static VLAN

When converting a dynamic VLAN to a static VLAN as shown here, all ports on the switch are assigned to the VLAN in Auto mode.

```
switch(config)# static-vlan 333
```

Multiple VLAN Registration Protocol overview

Multiple VLAN Registration Protocol (MVRP) is a registration protocol defined by IEEE, which propagates VLAN information dynamically across devices. It also enables devices to learn and automatically synchronize VLAN configuration information, thereby reducing the configuration workload.

It is an enhanced version of GVRP and improves declaration efficiency. It allows a participant (port) to make or withdraw declaration of attributes (VLANs). These declarations (or withdraws) are resulted in registration (or removal of registrations) with other switches in the network.

Salient features

- Complaint as per IEEE 802.1Q-2011(Clause 11.2).
- Supports conversion of dynamic VLAN to static VLAN.
- Supports propagation of radius assigned dynamic VLANs.
- Supports immediate registration and propagation of VLAN attributes during spanning tree topology changes.
- Supports registrar’s administrative control values such as normal, fixed, and forbid.
- Supports MVRP objects on the following standard MIBs:
  - IEEE8021-Q-BRIDGE-MIB (version 200810150000Z)
  - IEEE8021-BRIDGE-MIB (version 200810150000Z)

NOTE: Supports other MVRP objects with the help of proprietary MIB, HPE-ICF-MVRP-MIB `(hpicfMvrp.mib)`.

- Supports on both physical and LAG ports, which include the manual (trunk), static lacp, and dynamic lacp trunks.
- Supports High Availability hitless.
- Supports configuring MVRP using CLI and SNMP commands.
- Supports configurable timers — Join, Leave, Leave-All, and Periodic.
- Supports fast logging for important MVRP events and error conditions.
- Supports debug logging for all MVRP enabled ports.
- MVRP can be used to manage VLANs on dynamic trunk.

MVRP operating notes

MVRP is an enhanced version of Generic Attribute Registration Protocol (GARP). It is a generic registration framework defined by the IEEE 802.1ak amendment to the IEEE 802.1Q standard. As GVRP, the same rules for dynamic propagation and registration of VLANs is also applicable for MVRP on Aruba switches.

- A dynamic VLAN must be converted to a static VLAN before it can have an IP address.
- On the switches covered in this guide, MVRP can be enabled only if `max-vlans` is not more than 512 VLANs.

- The total number of VLANs on the switch (static and dynamic combined) cannot exceed the current maximum VLANs setting. For example, in the factory default state, the switch supports up to 256 VLANs. Any additional VLANs advertised to the switch are not added unless you increase the maximum VLANs setting.
- Converting a dynamic VLAN to a static VLAN and then executing the `write memory` command saves the VLAN in the `startup-config` file and makes it a permanent part of the switch's VLAN configuration.
- When you enable MVRP globally, it is enabled by default on dynamic trunks. Based on your requirement, you can disable MVRP on dynamic trunks. You cannot modify any other MVRP port parameters.
- Within the same broadcast domain, a dynamic VLAN can pass through a device that is not MVRP-aware. This is because a half-duplex repeater or a switch that is not MVRP-aware floods the MVRP (multicast) advertisement packets out of all ports.
- Rebooting a switch on which a dynamic VLAN exists deletes the VLAN. However, the dynamic VLAN reappears after the reboot, if MVRP is enabled. The switch again receives advertisement for the particular VLAN through a port configured to add dynamic VLANs.
- By receiving advertisements from other devices running MVRP, the switch learns of static VLANs on those devices and dynamically (automatically) creates tagged VLANs on the links to the advertising devices. Similarly, the switch advertises its static VLANs and the dynamic VLANs to other MVRP-aware devices, which the switch has learnt.
- An MVRP enabled switch does not advertise any MVRP learned VLANs out of the ports (on which it originally learned of those VLANs), until it is dynamically learnt on at least two ports.
- While MVRP is enabled on the switch, you cannot apply any ACLs to VLANs configured on the same switch.

Listing static and dynamic VLANs on an MVRP-enabled switch

Syntax

```
show vlan
```

Description

Displays both static and dynamic VLANs in the switch.

Example output

```
switch(config)# show vlan

Status and Counters - VLAN Information

  Maximum VLANs to support : 256
  Primary VLAN : DEFAULT_VLAN
  Management VLAN :

  VLAN ID Name                                 | Status     Voice Jumbo
  ------- -------------------------------- + ---------- ----- -----
  1       DEFAULT_VLAN                         | Port-based No    No
  40      MVRP_40                              | Dynamic
```

Converting a dynamic VLAN to a static VLAN

Syntax

```
static-vlan <dynamic-vlan-id>
```

Description

If a port on the switch has joined a dynamic VLAN, use the command to convert dynamic VLAN to static VLANs in the switch.

Example output

```
switch(config)# static-vlan 40
switch(config)# show vlan

Status and Counters - VLAN Information

  Maximum VLANs to support : 256
  Primary VLAN : DEFAULT_VLAN
  Management VLAN :

  VLAN ID Name                             | Status     Voice Jumbo
  ------- -------------------------------- + ---------- ----- -----
  1       DEFAULT_VLAN                     | Port-based No    No
  40      VLAN40                           | Port-based No    No
```

Viewing the current MVRP configuration on a switch

show mvrp

Syntax

```
show mvrp [config|state|statistics]
```

Description

Displays the MVRP settings and status.

Example output

```
switch# show mvrp
 config                  Show the MVRP configuration for all ports.
 state                   Show the MVRP state.
 statistics              Show MVRP statistics.
```

show mvrp config

Syntax

```
show mvrp config
```

Description

Displays the MVRP configuration for all ports.

Example output

```
switch# show mvrp config

 Configuration and Status - MVRP

 Global MVRP status : Disabled

 Port     Status    Periodic Registration Join  Leave     LeaveAll Periodic
                    Timer    Type         Time  Timer     Timer    Timer
 -------  --------- -------- ------------ ----  --------- -------- --------
 1        Disabled Enabled  Normal       20    300       1000     100
 2        Disabled Enabled  Normal       20    300       1000     100
 3        Disabled Enabled  Normal       20    300       1000     100
```

show mvrp state

Syntax

`show mvrp state <VLAN-ID> [<PORT-NUM>]`

Description

Displays the MVRP state.

Parameters

<VLAN-ID>

Specify the MVRP state for VLAN ID.

<PORT-NUM>

Specify the port number to display the MVRP state.

Example output

```
switch(config)# show mvrp state
 VLAN-ID               Enter a VLAN identifier or the VLAN name if configured.
switch(config)# show mvrp state 1
 [ethernet] PORT-NUM
switch(config)# show mvrp state 1

 Configuration and Status - MVRP state for VLAN 1

 Port     VLAN  Registrar Applicant Forbid
                State     State     Mode
 -------- ----- --------- --------- ---------
 1        1     MT        QA        No
```

show mvrp statistics

Syntax

`show MVRP statistics [<PORT-LIST>]`

Description

Displays the MVRP statistics.

Parameter

`PORT-LIST`

Displays the MVRP statistics at the specified port.

Example output

```
switch(config)# show mvrp statistics

Status and Counters - MVRP

  MVRP statistics for port  : A1
  ----------------------------
  Failed registration    : 0
  Last PDU origin        : 40a8f0-9e11ff
  Total PDU Transmitted  : 53
  Total PDU Received     : 72
  Frames Discarded       : 0

  Message type  Transmitted     Received
-------------- ------------    ------------
  New           0               0
  Empty          16466          258
  In             4              0
  Join Empty     0              72
  Join In        53             55
  Leave          0              0
  Leaveall       4              2
```

clear mvrp statistics

Syntax

`clear mvrp statistics [<PORT-LIST>]`

Description

Clears the statistics for MVRP on a port or all ports.

Parameters

`PORT-LIST`

Specify a port number or list of ports or all ports.

Example output

```
switch# clear mvrp statistics
 [ethernet] PORT-LIST  Enter a port number, a list of ports or 'all' for all ports.
switch# clear mvrp statistics all
```

debug mvrp

Syntax

`debug mvrp {all | event| packet | state-machine | timer} [<PORT-LIST>]`

Description

Enables debug messages.

Parameters

`all`

Display all MVRP debug messages.

`event`

Display all MVRP event messages.

`packet`

Display all MVRP packet messages.

`state-machine`

Display all MVRP state-machine messages.

`timer`

Display all MVRP timer messages.

`PORT-LIST`

Display all MVRP debug messages for a port.

Example output

```
switch(config)# debug mvrp all
switch(config)# show debug
```

```
Debug Logging

  Source IP Selection: Outgoing Interface
  Origin identifier: Outgoing Interface IP
  Destination: None

  Enabled debug types:
   mvrp event include port A1-A24,F1-F24
   mvrp packet include port A1-A24,F1-F24
   mvrp state-machine include port A1-A24,F1-F24
   mvrp timer include port A1-A24,F1-F24
```

Configuring MVRP

Enabling MVRP globally

MVRP must be enabled globally to allow the device to participate in the protocol.

Syntax

`mvrp {enable | disable}`

`no mvrp`

Description

Enables MVRP globally on a switch. MVRP must be enabled globally and at least on one interface. The `no` form of the command disables MVRP.

Parameters

`enable`

Enable MVRP.

`disable`

Disable MVRP.

Example output

```
switch# show mvrp config

 Configuration and Status - MVRP

 Global MVRP status : Enabled

 Port     Status   Periodic Registration Join  Leave    LeaveAll Periodic
                   Timer    Type         Time  Timer    Timer    Timer
 -------  -------- -------- ------------ ----  -------- -------- --------
 1        Disabled Enabled  Normal       20    300      1000     100
 2        Disabled Enabled  Normal       20    300      1000     100
```

Enabling MVRP on an interface

By default, MVRP is disabled on all interfaces.

Syntax

`mvrp {enable | disable}`

`no mvrp`

Description

Enables MVRP on an interface. MVRP must be enabled globally and at least on one interface.

Use `no mvrp` to disable MVRP.

Parameters

`enable`

Enable `mvrp`

`disable`

Disable `mvrp`

Example output

```
switch(config)# mvrp
disable               Disable MVRP.
enable                Enable MVRP.
switch(config)# mvrp enable
switch(config)# interface 1
switch(eth-1)# mvrp enable
switch(eth-1)# show mvrp config

 Configuration and Status - MVRP

 Global MVRP status : Enabled

 Port     Status   Periodic Registration Join  Leave    LeaveAll Periodic
                   Timer    Type         Time  Timer    Timer    Timer
 -------  --------- --------- ------------- ----  --------- --------- ---------
 1        Enabled  Enabled  Normal       20    300      1000     100
 2        Disabled Enabled  Normal       20    300      1000     100
```

MVRP timers

MVRP supports four types of timers:

- Join Timer
- Leave Timer
- LeaveAll Timer
- Periodic Timer

Join Timer

The Join Timer controls the transmission of Join messages. To avoid a PDU storm, an MVRP participant waits for a duration of the Join Timer after sending a join message, and ensures that all participants transmit at different times. This is a per port timer and is applicable to all applicants for the port.

mvrp join-timer

Syntax

`mvrp join-timer <centiseconds>`

`no mvrp join-timer`

Description

Sets the Join Timer for the port. You can use the timer to space MVRP join messages. To ensure that join messages are transmitted to other participants, an MVRP participant waits for a specified time before sending a join message. The Join Timer must be less than half of the Leave Timer. The default value is 20 centiseconds.

Use `no mvrp join-timer` to set the interval to the default value.

Parameters

```
centiseconds
```

Set the Join Timer for the port.

Usage

```
mvrp join-timer <20-100>
```

The MVRP Join Timer ranges from 20 –100 in centiseconds.

Example output

```
switch(eth-1)# mvrp join-timer
<20-100>              Set the join timer for the port.
switch# mvrp join-timer 40
switch# show mvrp config

 Configuration and Status - MVRP

 Global MVRP status : Enabled

 Port     Status   Periodic Registration Join  Leave    LeaveAll Periodic
                   Timer    Type         Time  Timer    Timer    Timer
 -------  -------- -------- ------------ ----  -------- -------- --------
 1        Enabled  Enabled  Normal       40    300      1000     100
 2        Disabled Enabled  Normal       20    300      1000     100
 3        Disabled Enabled  Normal       20    300      1000     100
```

Leave Timer

The Leave Timer controls the time duration for which the Registrar state machine waits in the LV state before changing to the MT state. The Leave Timer is started only when a leave message is received by the applicant state. The attribute is deregistered, if there are requests to join before the expiry of the Leave Timer. This is a per port timer and is applicable to all registrars for the port.

mvrp leave-timer

Syntax

```
mvrp leave-timer <centiseconds>
```

```
no mvrp leave-timer
```

Description

The Leave Timer must be at least twice the Join Timer and must be less than the LeaveAll Timer. The default value is 300 centiseconds.

Use `no mvrp leave-timer` to set the interval to the default value.

Parameter

```
centiseconds
```

Set the Leave Timer for the port.

Usage

```
mvrp leave-timer <40-1000000>
```

The MVRP Leave Timer ranges from 40 –1000000 in centiseconds.

Example output

```
switch(eth-1)# mvrp leave-timer
<40-1000000>          Set the leave timer for the port.
```

```
switch(eth-1)# mvrp leave-timer 500
switch(eth-1)# show mvrp config

 Configuration and Status - MVRP

 Global MVRP status : Enabled

 Port     Status   Periodic Registration Join  Leave    LeaveAll Periodic
                   Timer    Type         Time  Timer    Timer    Timer
 -------  -------- -------- ------------ ----  -------- -------- --------
 1        Enabled  Enabled  Normal       40    500      1000     100
 2        Disabled Enabled  Normal       20    300      1000     100
 3        Disabled Enabled  Normal       20    300      1000     100
```

LeaveAll Timer

The LeaveAll Timer controls the frequency with which the LeaveAll state machine generates LeaveAll PDUs. When a LeaveAll Timer expires, the MVRP sends out LeaveAll messages and restarts the LeaveAll Timer. The LeaveAll Timer is set to a random value `T` which ranges from `LeaveAllTime < T < 1.5*LeaveAllTime`, where LeaveAll time is the configured LeaveAll time. The default value is 1000 centiseconds. This is a per port timer.

mvrp leaveall-timer

Syntax

`mvrp leaveall-timer <centiseconds>`

`no mvrp leaveall-timer`

Description

The LeaveAll Timer is the time duration between sending LeaveAll messages. The LeaveAll Timer must be greater than the Leave Timer.

Use `no mvrp leaveall-timer` to set the interval to the default value.

Parameter

`centiseconds`

Set the LeaveAll Timer for the port.

Usage

`mvrp leaveall-timer <500-1000000>`

The MVRP LeaveAll Timer ranges from 500 –1000000 in centiseconds.

Example output

```
switch# mvrp leaveall-timer
<500-1000000> Set the leaveall timer for the port.
switch# mvrp leaveall-timer 700
switch(eth-1)# show mvrp config

 Configuration and Status - MVRP

 Global MVRP status : Enabled

 Port     Status   Periodic Registration Join  Leave    LeaveAll Periodic
                   Timer    Type         Time  Timer    Timer    Timer
 -------  -------- -------- ------------ ----  -------- -------- --------
 1        Enabled  Enabled  Normal       40    500      700      100
 2        Disabled Enabled  Normal       20    300      1000     100
 3        Disabled Enabled  Normal       20    300      1000     100
```

Periodic Timer

The Periodic Timer controls the frequency with which the periodic transmission state machine generates periodic events. This is a per port timer. On start, the Periodic Timer is set to one second. You can enable or disable the Periodic Timer. By default, it is enabled. The default value is 100 centiseconds.

mvrp periodic timer

Syntax

`mvrp periodic-timer <centiseconds>`

`no mvrp periodic-timer`

Description

Set the Periodic Timer transmission interval for the port.

Use `no mvrp periodic-timer` to set the interval to the default value.

Parameters

`centiseconds`

Set the Periodic Timer transmission interval for the port.

Usage

`mvrp periodic-timer <100-1000000>`

The MVRP Periodic Timer ranges from 100 –1000000 in centiseconds.

Example output

```
switch(eth-1)# mvrp periodic-timer
<100-1000000>         Set the periodic timer transmission interval for the port.
switch(eth-1)# mvrp periodic-timer 300
switch(eth-1)# show mvrp config

 Configuration and Status - MVRP

 Global MVRP status : Enabled

 Port     Status   Periodic Registration Join  Leave    LeaveAll Periodic
                   Timer    Type         Time  Timer    Timer    Timer
 -------- -------- -------- ------------ ----  -------- -------- --------
 1        Enabled  Enabled  Normal       40    500      700      300
 2        Disabled Enabled  Normal       20    300      1000     100
 3        Disabled Enabled  Normal       20    300      1000     100
```

mvrp periodic-timer-enable

Syntax

`mvrp periodic-timer-enable`

`no mvrp periodic-timer-enable`

Description

Enable Periodic Timer transmission for the port. By default, it is enabled.

Use `no mvrp periodic-timer-enable` to disable the Periodic Timer on an interface.

MVRP registration modes

MVRP supports three registration modes:

- **Normal**

  In this mode, a port can register and deregister dynamic VLANs. By default, the registrar mode is normal.

- **Fixed**

  In this mode, a port cannot register or deregister dynamic VLANs. However, if a static VLAN exists in the system, the port changes to registered state on receipt of join message.

- **Forbidden**

  In this mode, a port does not register dynamic VLANs, ignores all MRP messages, and remains in MT state (unregistered).

mvrp registration

Syntax

`mvrp registration {normal |fixed}`

Description

Configures the port response to MRP messages.

Parameters

`normal`

Port response is normal for the incoming MRP messages.

`fixed`

Ignores the MRP messages and remains registered.

Example output

```
switch# mvrp registration
 fixed                 The port ignores all MRP messages and remains registered.
 normal                The port responds normally to incoming MRP messages.

switch(config)# interface A1 mvrp registration fixed
switch(config)# show mvrp config

Configuration and Status - MVRP

Global MVRP status : Enabled

Port     Status   Periodic Registration Join  Leave    LeaveAll Periodic
                   Timer    Type         Time  Timer    Timer    Timer
-------  -------- -------- ------------ ----  -------- -------- --------
A1       Enabled  Enabled  Fixed        20    300      1000     100
A2       Disabled Enabled  Normal       20    300      1000     100
A3       Disabled Enabled  Normal       20    300      1000     100
```

show tech mvrp

Syntax

`show tech mvrp`

Description

Displays statistics of all the MVRP enabled ports.

Example output

```
switch# show tech mvrp

show mvrp statistics

Status and Counters - MVRP

  MVRP statistics for port  : A1
  ----------------------------
  Failed registration    : 0
  Last PDU origin        : 40a8f0-9e11ff
  Total PDU Transmitted  : 620
  Total PDU Received     : 755
  Frames Discarded       : 0



  Message type  Transmitted     Received
-------------- ------------    ------------
  New           0               0
  Empty         117370          2506
  In            17              0
  Join Empty    1               519
  Join In       658             697
  Leave         0               0
  Leaveall      28              37

mvrpDumpGlobalData

MVRP global enabled status    : enabled
MVRP enabled ports            : A1
Total MVRP enabled ports      : 1
Dyn trunk auto disable count  : 0
Total Static VLANs in system  : 1
Total Dynamic VLANs in system : 1
Max VLANs supported           : 512

Display VLAN_GROUP to VLANs Mapping:

Group ID      Mapped VLANs
----------    ----------------
       0      1-4094

Display timer Ports:

Group ID     Timer Value
----------   -------------

Display Blocked Ports:

Group ID     Blocked Ports
----------   ---------------

mvrppconfig

Mvrp Port state info:

Port   MvrpState   LinkState   Registrar   Value
-----   ---------   ---------   ---------   -----
   A1      Enable          Up      Normal   0X05
   A2     Disable          Up      Normal   0X04
   A3      Disable        Down      Normal   0000
   A4      Disable        Down      Normal   0000
   A5      Disable        Down      Normal   0000
   A6      Disable        Down      Normal   0000
   A7      Disable        Down      Normal   0000
   A8      Disable        Down      Normal   0000
   A9      Disable        Down      Normal   0000
  A10      Disable        Down      Normal   0000
  A11      Disable        Down      Normal   0000
  A12      Disable        Down      Normal   0000
  A13      Disable        Down      Normal   0000
  A14      Disable        Down      Normal   0000
  A15      Disable        Down      Normal   0000
  A16      Disable        Down      Normal   0000
  A17      Disable        Down      Normal   0000
  A18      Disable        Down      Normal   0000
  A19      Disable        Down      Normal   0000
  A20      Disable        Down      Normal   0000
  A21      Disable        Down      Normal   0000
  A22      Disable        Down      Normal   0000
  A23      Disable        Down      Normal   0000
  A24      Disable        Down      Normal   0000
   F1      Disable        Down      Normal   0000
   F2      Disable        Down      Normal   0000
   F3      Disable        Down      Normal   0000
   F4      Disable        Down      Normal   0000
   F5      Disable        Down      Normal   0000
   F6      Disable        Down      Normal   0000
   F7      Disable        Down      Normal   0000
   F8      Disable        Down      Normal   0000
   F9      Disable        Down      Normal   0000
  F10      Disable        Down      Normal   0000
  F11      Disable        Down      Normal   0000
  F12      Disable        Down      Normal   0000
  F13      Disable        Down      Normal   0000
  F14      Disable        Down      Normal   0000
  F15      Disable        Down      Normal   0000
  F16      Disable        Down      Normal   0000
  F17      Disable        Down      Normal   0000
  F18      Disable        Down      Normal   0000
  F19      Disable        Down      Normal   0000
  F20      Disable        Down      Normal   0000
  F21      Disable          Up      Normal   0X04
  F22      Disable          Up      Normal   0X04
  F23      Disable        Down      Normal   0000
  F24      Disable        Down      Normal   0000

Mvrp Port timer values:

Port  join  leave  leaveall  periodic  periodic-enabled
-----  ----  -----  --------  --------  ----------------
   A1     20    300     1000       100          enabled
   A2     20    300     1000       100          enabled
   A3     20    300     1000       100          enabled
   A4     20    300     1000       100          enabled
   A5     20    300     1000       100          enabled
   A6     20    300     1000       100          enabled
   A7     20    300     1000       100          enabled
   A8     20    300     1000       100          enabled
   A9     20    300     1000       100          enabled
  A10     20    300     1000       100          enabled
  A11     20    300     1000       100          enabled
  A12     20    300     1000       100          enabled
  A13     20    300     1000       100          enabled
  A14    20    300      1000       100          enabled
  A15    20    300      1000       100          enabled
  A16    20    300      1000       100          enabled
  A17    20    300      1000       100          enabled
  A18    20    300      1000       100          enabled
  A19    20    300      1000       100          enabled
  A20    20    300      1000       100          enabled
  A21    20    300      1000       100          enabled
  A22    20    300      1000       100          enabled
  A23    20    300      1000       100          enabled
  A24    20    300      1000       100          enabled
   F1    20    300      1000       100          enabled
   F2    20    300      1000       100          enabled
   F3    20    300      1000       100          enabled
   F4    20    300      1000       100          enabled
   F5    20    300      1000       100          enabled
   F6    20    300      1000       100          enabled
   F7    20    300      1000       100          enabled
   F8    20    300      1000       100          enabled
   F9    20    300      1000       100          enabled
  F10    20    300      1000       100          enabled
  F11    20    300      1000       100          enabled
  F12    20    300      1000       100          enabled
  F13    20    300      1000       100          enabled
  F14    20    300      1000       100          enabled
  F15    20    300      1000       100          enabled
  F16    20    300      1000       100          enabled
  F17    20    300      1000       100          enabled
  F18    20    300      1000       100          enabled
  F19    20    300      1000       100          enabled
  F20    20    300      1000       100          enabled
  F21    20    300      1000       100          enabled
  F22    20    300      1000       100          enabled
  F23    20    300      1000       100          enabled
  F24    20    300      1000       100          enabled

mvrpmapringShow

Mvrp list info:
-------------------------
Port A1     : connected

Mvrp Map Count Info:

Vlan   Vid    Reg-Count
-----  -----  ----------
1      1      1
2      40     1

=== The command has completed successfully. ===
```

MVRP limitations

- MVRP and GVRP are mutually exclusive, and cannot coexist.
- MVRP and Smartlink are mutually exclusive. Smartlinks can be enabled on ports, which are not MVRP enabled and vice versa.
- MVRP and PVST are mutually exclusive. When MVRP is globally enabled, spanning tree mode cannot be set as PVST and vice versa.
- MVRP can be enabled on a provider bridge environment, but does not support SVLAN ports in mixed mode configuration.

- MVRP can be used to manage VLANs on dynamic trunk.
- Enable `aaa port-access gvrp-vlans` to support RADIUS-assigned VLANs. When you enable `aaa port-access gvrp-vlans`, dynamic VLANs created by MVRP or GVRP can be used for radius port assignment.
- An OpenFlow member VLAN cannot be a dynamic VLAN. As a result, a dynamic VLAN must be converted to static to be handled by the OpenFlow controller.
- For security purposes, MVRP is disabled by default. MVRP packets are blocked on MVRP disabled ports, but can be enabled on ports which are security enabled.
- MVRP and private VLAN cannot coexist.
- DIPLDv6 cannot be configured on MVRP enabled ports.
- MVRP support is limited to 512 VLANs and 128 logical ports due to CPU and memory resource availability.

Table 8: *MVRP supported ports*

Platforms	Maximum MVRP ports supported

`Aruba 2920`
`Aruba 2930`
	128

`Aruba 2530`	24
`Aruba 2540`	54

Table 9: *MVRP supported VLANs*

Platforms	Maximum VLANs	Maximum MSTP instance	Maximum ports

`Aruba 2920`
`Aruba 2930`
	512	16	128

`Aruba 2530`	512	16	24
`Aruba 2540`	512	16	54

MVRP statistics

The MVRP statistics generated using `show mvrp statistics`, records any registration failures, tracks MAC addresses to derive statistics.

- **Registration failure**

  Maintains the count of registration requests received but failed due to MVRP limitation.
- **Peer tracking**

Records the MAC address of the MVRP PDU that has caused the recent state change for the registrar machine. A maximum of one MAC address per port of the originator switch is stored.

- **PDU event statistics**

  Collects the data on numbers of events (join, leave, and so on) transmitted and received.

For more information, see **show mvrp statistics**.

Overview of MSTP

Without spanning tree, having more than one active path between a pair of nodes causes loops in the network, which can result in duplication of messages leading to a "broadcast storm" that can bring down the network.

NOTE: MSTP cannot protect against loops when there is an unmanaged device on the network that drops spanning tree packets, or may fail to detect loops where this is an edge port configured with client authentication (802.1X, Web and MAC authentication). To protect against the formation of loops in these cases, you can use the loop protection feature.

Multiple-Instance spanning tree operation (802.1s) ensures that only one active path exists between any two nodes in a spanning tree instance. A spanning tree instance comprises a unique set of VLANs, and belongs to a specific spanning tree region. A region can comprise multiple spanning tree instances (each with a different set of VLANs), and allows one active path among regions in a network. Applying VLAN tagging to the ports in a multiple-instance spanning tree network enables blocking of redundant links in one instance while allowing forwarding over the same links for non-redundant use by another instance.

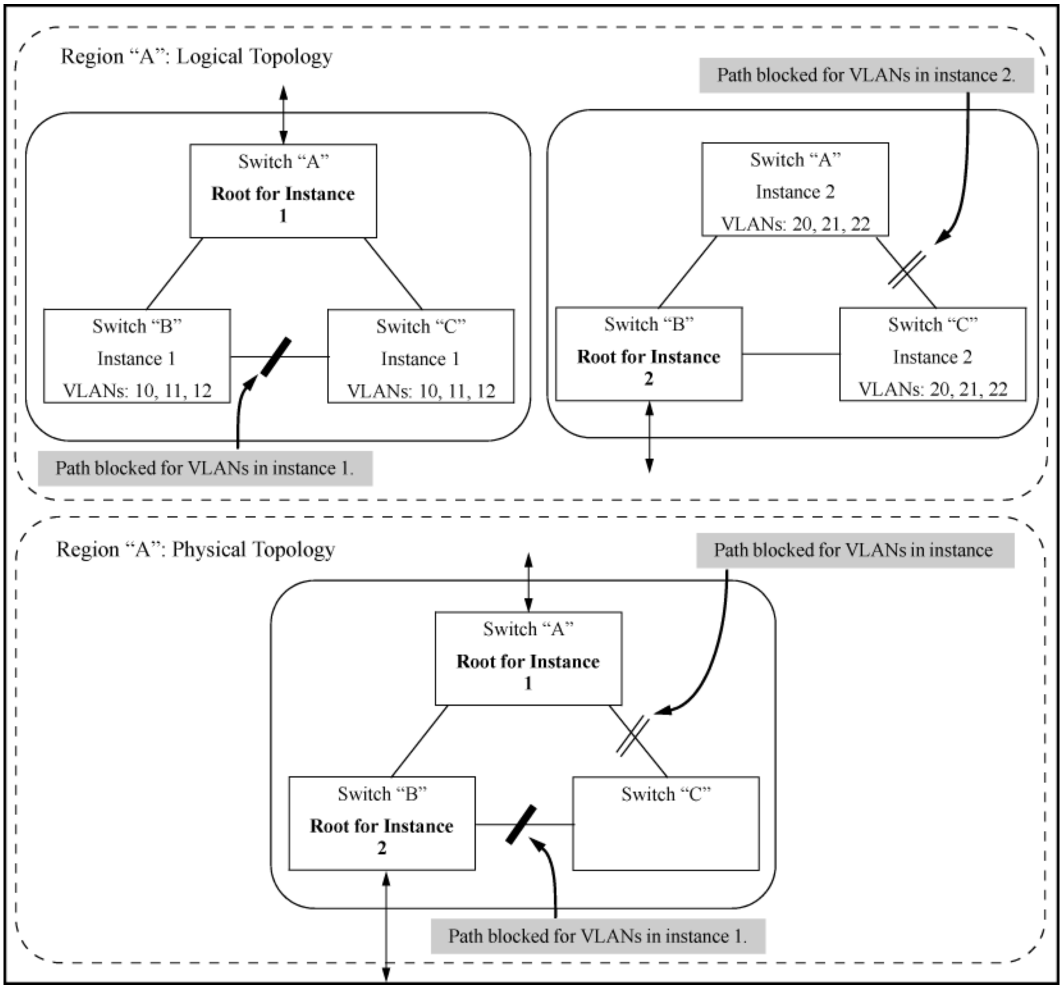
VLAN/Instance groupings

Suppose that there are three switches in a region configured with VLANs grouped into two instances, as follows:

VLANs	Instance 1	Instance 2
10, 11, 12	Yes	No
20, 21, 22	No	Yes

The logical and physical topologies resulting from these VLAN/Instance groupings result in blocking on different links for different VLANs:

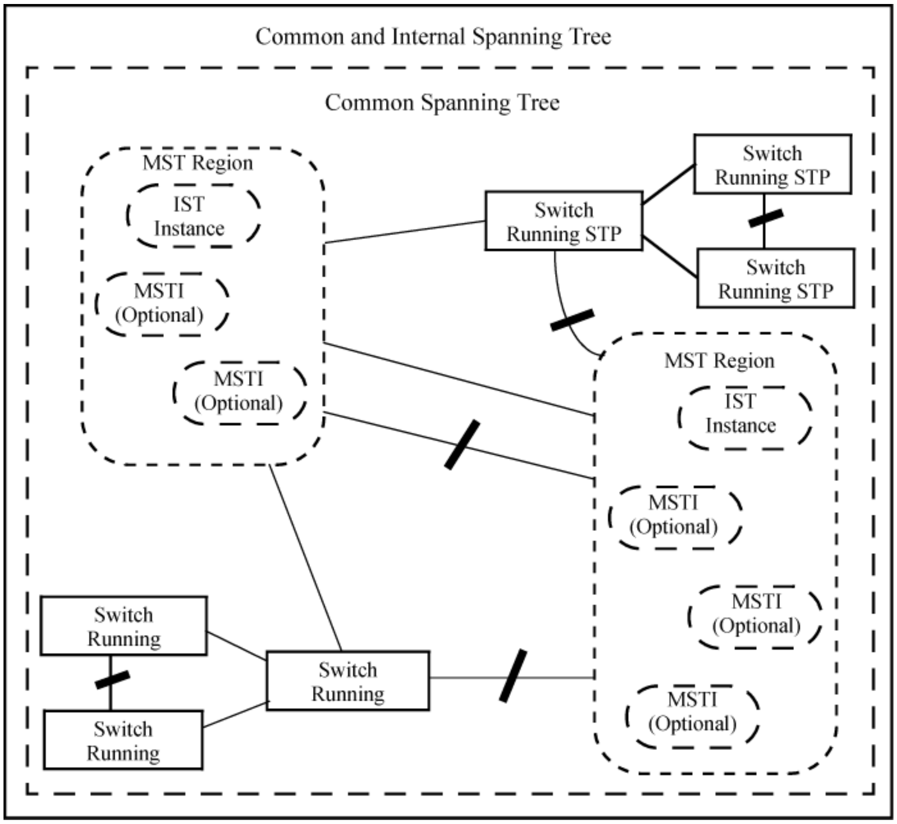
A multiple spanning tree application



MSTP structure

MSTP maps active, separate paths through separate spanning tree instances and between MST regions. Each MST region comprises one or more MSTP switches. Note that MSTP recognizes an STP or RSTP LAN as a distinct spanning tree region.

Figure 11: *An MSTP network with legacy STP and RSTP devices connected*



How MSTP operates

In the factory default configuration, spanning tree operation is off. Also, the switch retains its currently configured spanning tree parameter settings when disabled. Thus, if you disable spanning tree, then later re-enable it, the parameter settings will be the same as before spanning tree was disabled. The switch also includes a `pending` feature that enables you to exchange MSTP configurations with a single command.

NOTE:

The switch automatically senses port identity and type, and automatically defines spanning tree parameters for each type, and parameters that apply across the switch. Although these parameters can be adjusted, HPE strongly recommends leaving these settings in their default configurations unless the proposed changes have been supplied by an experienced network administrator who has a strong understanding of the IEEE 802.1D/w/s standards and operation.

802.1s Multiple Spanning Tree Protocol (MSTP)

The switches covered in this guide use the IEEE 802.1s Multiple Spanning Tree Protocol (MSTP) standard.

The 802.1D and 802.1w spanning tree protocols operate without regard to a network's VLAN configuration, and maintain one common spanning tree throughout a bridged network. Thus, these protocols map one loop-free, logical topology on a given physical topology. The 802.1s Multiple Spanning Tree protocol (MSTP) uses VLANs to create multiple spanning trees in a network, which significantly improves network resource utilization while maintaining a loop-free environment.

While the per-VLAN spanning tree approach adopted by some vendors overcomes the network utilization problems inherent in using STP or RSTP, using a per-VLAN technology with multiple VLANs can overload the switch's CPU. MSTP on the switches covered in this guide complies with the IEEE 802.1s standard, and extends STP and RSTP functionality to map multiple independent spanning tree instances onto a physical topology. With MSTP, each spanning tree instance can include one or more VLANs and applies a separate, per-instance forwarding topology. Thus, where a port belongs to multiple VLANs, it may be dynamically blocked in one spanning tree instance, but forwarding in another instance. This achieves load-balancing across the network while keeping the switch's CPU load at a moderate level (by aggregating multiple VLANs in a single spanning tree instance). MSTP provides fault tolerance through rapid, automatic reconfiguration if there is a failure in a network's physical topology.

With MSTP-capable switches, you can create a number of MST regions containing multiple spanning tree instances. This requires the configuration of a number of MSTP-capable switches. However, it is not necessary to do this. You can enable MSTP on an MSTP-capable switch and a spanning tree instance is created automatically. This instance always exists by default when spanning tree is enabled, and is the spanning tree instance that communicates with STP and RSTP environments. The MSTP configuration commands operate exactly like RSTP commands and MSTP is backward-compatible with the RSTP-enabled and STP-enabled switches in your network.

CAUTION:

Spanning tree interprets a switch mesh as a single link. Because the switch automatically gives faster links a higher priority, the default MSTP parameter settings are usually adequate for spanning tree operation. Because incorrect MSTP settings can adversely affect network performance, do not change the MSTP settings from their default values unless you have a strong understanding of how spanning tree operates.

In a mesh environment, the default MSTP timer settings (`Hello Time` and `Forward Delay`) are usually adequate for MSTP operation. Because a packet crossing a mesh may traverse several links within the mesh, using smaller-than-default settings for the MSTP `Hello Time` and `Forward Delay` timers can cause unnecessary topology changes and end-node connectivity problems.

MST regions

All MSTP switches in a given region must be configured with the same VLANs, and each MSTP switch within the same region must have the same VLAN-to-instance assignments. In addition, a VLAN can belong to only one instance within any region. Within a region:

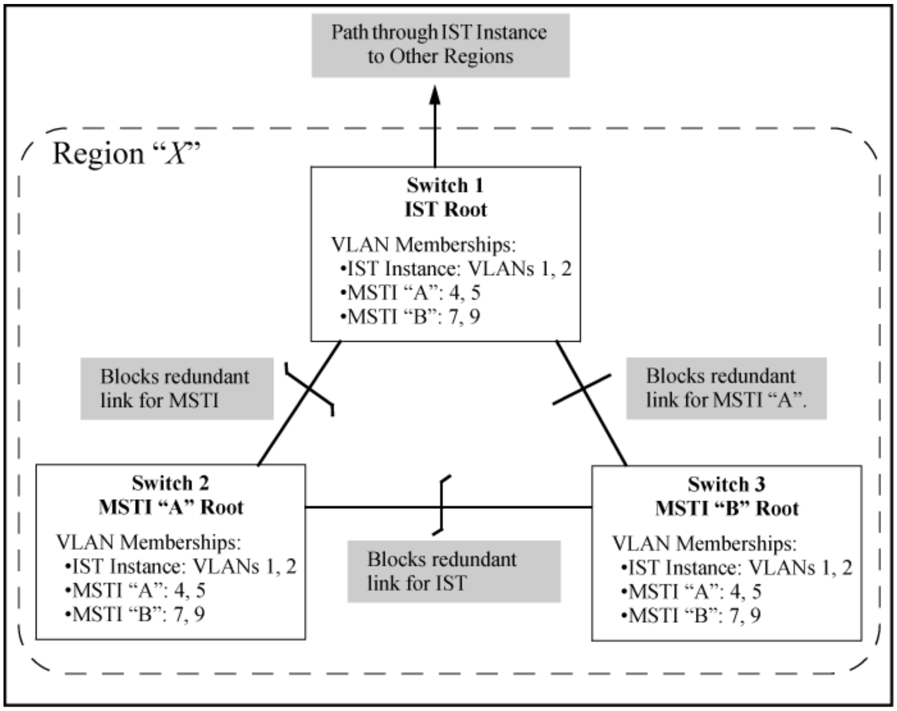
- All of the VLANs belonging to a given instance compose a single, active spanning tree topology for that instance.
- Each instance operates independently of other regions.

Between regions, there is a single, active spanning tree topology.

How separate instances affect MSTP

Assigning different groups of VLANs to different instances ensures that those VLAN groups use independent forwarding paths. For example, in the following figure, each instance has a different forwarding path.

Figure 12: *Active topologies built by three independent MST instances*


Path through IST Instance to Other Regions
Region "X"
Switch 1
IST Root
VLAN Memberships:
•IST Instance: VLANs 1, 2
•MSTI "A": 4, 5
•MSTI "B": 7, 9
Blocks redundant link for MSTI
Blocks redundant link for MSTI "A".
Switch 2
MSTI "A" Root
VLAN Memberships:
•IST Instance: VLANs 1, 2
•MSTI "A": 4, 5
•MSTI "B": 7, 9
Blocks redundant link for IST
Switch 3
MSTI "B" Root
VLAN Memberships:
•IST Instance: VLANs 1, 2
•MSTI "A": 4, 5
•MSTI "B": 7, 9

While allowing only one active path through a given instance, MSTP retains any redundant physical paths in the instance to serve as backups (blocked) paths in case the existing active path fails. Thus, if an active path in an instance fails, MSTP automatically activates (unblocks) an available backup to serve as the new active path through the instance for as long as the original active path is down. Note also that a given port may simultaneously operate in different states (forwarding or blocking) for different spanning tree instances within the same region. This depends on the VLAN memberships to which the port is assigned. For example, if a port belongs to VLAN 1 in the IST instance of a region and also belongs to VLAN 4 in MSTI "x" in the same region, the port may apply different states to traffic for these two different instances.

Within a region, traffic routed between VLANs in separate instances can take only one physical path. To ensure that traffic in all VLANs within a region can travel between regions, all of the boundary ports for each region should belong to all VLANs configured in the region. Otherwise, traffic from some areas within a region could be blocked from moving to other regions.

All MSTP switches (including STP and RSTP switches) in a network use BPDUs (Bridge Protocol Data Units) to exchange information from which to build multiple, active topologies in the individual instances within a region and between regions. From this information:

- The MSTP switches in each LAN segment determine a designated bridge and designated port or trunk for the segment.
- The MSTP switches belonging to a particular instance determine the root bridge and root port or trunk for the instance.
- For the IST instance within a region, the MSTP switches linking that region to other regions (or to STP or RSTP switches) determine the IST root bridge and IST root port or trunk for the region. (For any Multiple

spanning tree instance—MSTI—in a region, the regional root may be a different switch that is not necessarily connected to another region.)

- The MSTP switches block redundant links within each LAN segment, across all instances, and between regions, to prevent any traffic loops.

As a result, each individual instance (spanning tree) within a region determines its regional root bridge, designated bridges, and designated ports or trunks.

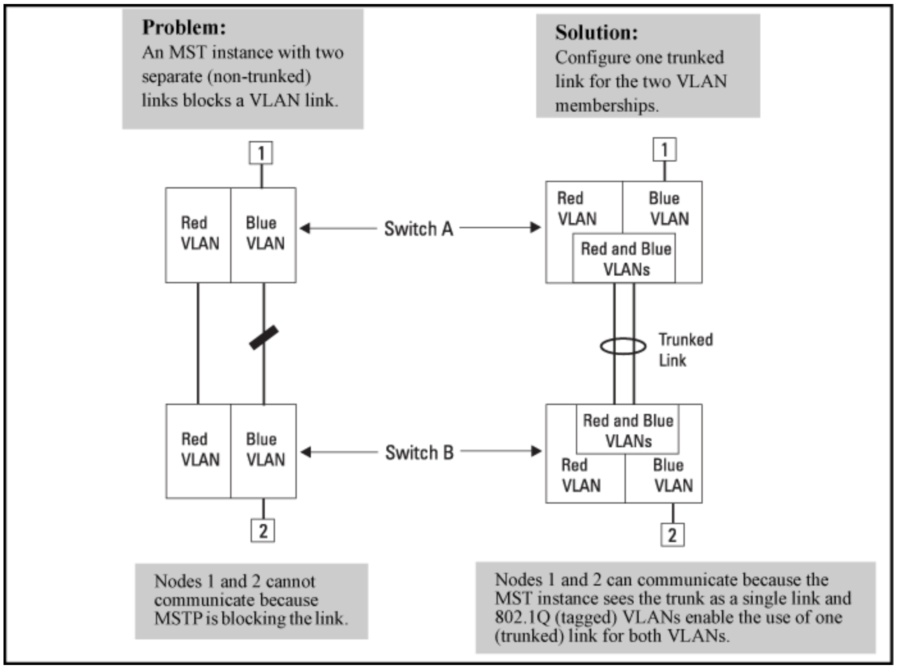
Regions, legacy STP and RSTP switches, and the Common Spanning Tree (CST)

The IST instance and any MST instances in a region exist only within that region. Where a link crosses a boundary between regions (or between a region and a legacy STP or RSTP switch), traffic is forwarded or blocked as determined by the Common Spanning Tree (CST). The CST ensures that there is only one active path between any two regions, or between a region and a switch running STP and RSTP. (See the figure in **MSTP structure** on page 96.)

MSTP operation with 802.1Q VLANs

As indicated in the preceding sections, within a given MST instance, a single spanning tree is configured for all VLANs included in that instance. This means that if redundant physical links exist in separate VLANs within the same instance, MSTP blocks all but one of those links. However, you can prevent the bandwidth loss caused by blocked redundant links for different VLANs in an instance by using a port trunk. The following example shows how you can use a port trunk with 802.1Q (tagged) VLANs and MSTP without unnecessarily blocking any links or losing any bandwidth.

Using a trunked link to support multiple VLAN connectivity within the same MST instance


Problem:
An MST instance with two separate (non-trunked) links blocks a VLAN link.
Solution:
Configure one trunked link for the two VLAN memberships.
1
Red VLAN
Blue VLAN
Switch A
Red VLAN
Blue VLAN
Red and Blue VLANs
Trunked Link
Red and Blue VLANs
Red VLAN
Blue VLAN
Switch B
Red VLAN
Blue VLAN
2
Nodes 1 and 2 cannot communicate because MSTP is blocking the link.
Nodes 1 and 2 can communicate because the MST instance sees the trunk as a single link and 802.1Q (tagged) VLANs enable the use of one (trunked) link for both VLANs.

NOTE:

All switches in a region should be configured with the VLANs used in that region, and all ports linking MSTP switches together should be members of all VLANs in the region. Otherwise, the path to the root for a given VLAN will be broken if MSTP selects a spanning tree through a link that does not include that VLAN.

MSTP compatibility with RSTP or STP

IEEE 802.1s MSTP includes RSTP functionality and is designed to be compatible with both IEEE 802.1D and 802.1w spanning tree protocols. Using the default configuration values, your switches will interoperate effectively with RSTP and STP devices. MSTP automatically detects when the switch ports are connected to non-MSTP devices in the spanning tree and communicates with those devices using 802.1D or 802.1w STP BPDU packets, as appropriate.

To enable effective interoperation with STP (802.1D) configured devices, however, you may need to adjust the default configuration values. Here are two such examples:

- The rapid state transitions employed by MSTP may result in an increase in the rates of frame duplication and misordering in the switched LAN. To allow the switch to support applications and protocols that may be sensitive to frame duplication and misordering, you can disable rapid transitions by setting the Force Protocol Version parameter to STP-compatible. The value of this parameter applies to all ports on the switch.
- One of the benefits of MSTP is the implementation of a larger range of port path costs, which accommodates higher network speeds. However, this can create some incompatibility between devices running the older 802.1D STP. You can adjust to this incompatibility by implementing the global spanning tree legacy-path cost command.

RSTP and MSTP implement a greater range of path costs than 802.1D STP, and use different default path cost values to account for higher network speeds. These values are shown in the following table.

Port type	802.1D STP path cost	RSTP and MSTP path cost
10 Mbps	100	2 000 000
100 Mbps	10	200 000
1 Gbps	5	20 000

Because the maximum value for the path cost allowed by 802.1D STP is 65535, devices running that version of spanning tree cannot be configured to match the values defined by MSTP, at least for 10 Mbps and 100 Mbps ports. In LANs where there is a mix of devices running 802.1D STP, RSTP, and MSTPs, you should reconfigure the devices so the path costs match for ports with the same network speeds.

Preconfiguring an MSTP regional topology

The MSTP VLAN configuration enhancement allows you to preconfigure an MSTP regional topology and ensure that the same VLAN ID-to-MSTI assignments exist on each MSTP switch in the region.

CAUTION: When this software version is installed, the prior VLAN ID-to-MSTI mappings do not change. However, this enhancement is not backward-compatible. If you install a software version earlier than this version, and you have configured MSTI entries instances mapped to VLANs, they will be removed from the configuration file when booting to the prior version of software. Do one of the following to install or reload a prior version of the software:

- Remove all MSTP mappings from the configuration file, then reconfigure the instance mapping after running the desired software version.
- Save the current configuration file before updating the software to a new version. If you later reload this older version of the software, use this configuration file when you reload the older version.

The default behavior of the `spanning-tree instance vlan` command changes so that, before a static VLAN is configured or a dynamic VLAN is learned on the switch, you can preconfigure its VLAN ID-to-MSTI mapping. Later, when the VLAN is created, it is automatically assigned to the MSTI to which it was previously mapped.

By supporting preconfigured VLAN ID-to-MSTI topologies, the VLAN configuration enhancement provides the following benefits:

- Scalability: In a network design in which you plan to use a large number of VLANs, you can preconfigure identical VLAN ID-to-MSTI mappings on all switches in a single, campus-wide MST region, regardless of the specific VLANs that you later configure on each switch. After the initial VLAN ID-to-MSTI mapping, you can decide on the exact VLANs that you need on each switch.All switches in a region must be configured with the same VLAN ID-to-MSTI mappings and the same MSTP configuration identifiers (region name and revision number).
- Flexibility: By preconfiguring identical VLAN ID-to-MSTI mappings on all switches in an MST region, you can combine switches that support different maximum numbers of VLANs.
- Network stability: You can reduce the interruptions in network connectivity caused by the regeneration of spanning trees in the entire network each time a configuration change in VLAN-to-MSTI mapping is detected on a switch. The negative impact on network performance is reduced if all newly created VLANs are pre-mapped to the correct MST instances. Later, VLAN creation and deletion are ignored by MSTP and no interruption in spanning tree traffic occurs.
- Usability: Dynamically learned GVRP VLANs can be mapped to MSTIs and support MSTP load balancing.

Preconfiguring VLANs in an MST instance

When configuring an MSTP regional topology, multiple spanning tree instances are created. Each MST instance provides a fully connected active topology for a particular set of VLANs.

Each switch in an MSTP region is configured with the following set of common parameters:

- Region name (`spanning-tree config-name`)
- Region revision number (`spanning-tree config-revision`)
- Identical VLAN ID-to-MSTI mapping (`spanning-tree instance vlan`)

Syntax:

```
spanning-tree instance 1..16 vlan vid [vid..vid]
```

```
no spanning-tree instance 1..16 vlan vid [vid..vid]
```

Configuring MSTP on the switch automatically configures the IST instance and places all statically and dynamically configured VLANs on the switch into the IST instance. This command creates a new MST instance (MSTI) and moves the VLANs specified from the IST to the MSTI.

You must map at least one VLAN to an MSTI when you create it. You cannot map a VLAN ID to more than one instance. You can create up to 16 MSTIs in a region.

The `no` form of the command removes one or more VLANs from the specified MSTI. If no VLANs are specified, the `no` form of the command deletes the specified MSTI.

When removing a VLAN from an MSTI, the VLAN returns to the IST instance, where it remains or is re-assigned to another MSTI configured in the region.

NOTE: The valid VLAN IDs to map to a specified MSTI are from 1 to 4094. The VLAN ID-to-MSTI mapping does not require a VLAN to be already configured on the switch. The MSTP VLAN enhancement allows preconfiguring MSTP topologies before the VLAN IDs associated with each instance exist on a switch.

When using preconfigured VLAN ID-to-MSTI topologies, ensure that MSTP switches remain in the same region by mapping all VLAN IDs used in the region to the same MSTIs on each regional switch.

Each MST instance supports a different set of VLANs. A VLAN that is mapped to an MST instance cannot be a member of another MST instance.

The MSTP VLAN configuration enhancement allows you to ensure that the same VLAN ID-to-MSTI assignments exist on each MSTP switch in a region. Before a static VLAN is configured or a dynamic VLAN is learned on the switch, use the `spanning-tree instance vlan` command to map VLANs to each MST instance in the region. Later, when the VLAN is created, the switch automatically assigns it to the MST instance to which you had previously mapped it.

Configuring MSTP instances with the VLAN range option (Example)

Using the `spanning-tree instance` command with the VLAN range option configures the entire range of VLANs, even if the range includes VLANs that are not currently present on the switch.

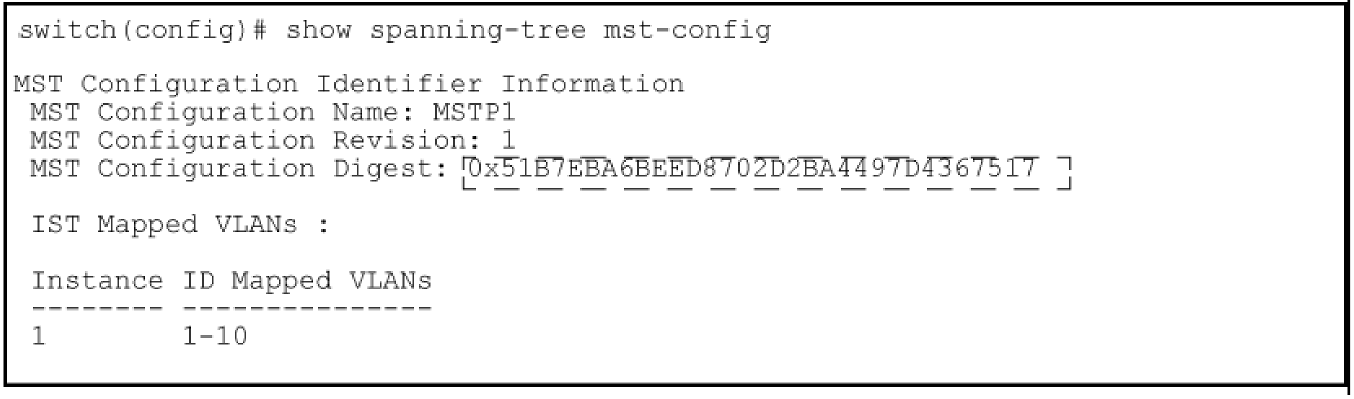
Mapping VLANs to MSTP Instance

If VLANs 1, 5, and 7 are currently present and you enter the following command, all the VLANs from 1 through 10 are included, even those VLANs that are not present.

```
switch(config)# spanning-tree instance 1 vlan 1-10
```

On switches other than those covered by this guide, only the VLANs that are present will be included, that is, only VLANs 1, 5, and 7. The switch will map these VLANs to MSTP Instance 1, which results in a Configuration Digest that is not the same as the Configuration Digest for the switches running this enhancement.

Below, the example shows an MSTP instance configured with the VLAN range option. All the VLANs are included in the instance whether they exist or not.

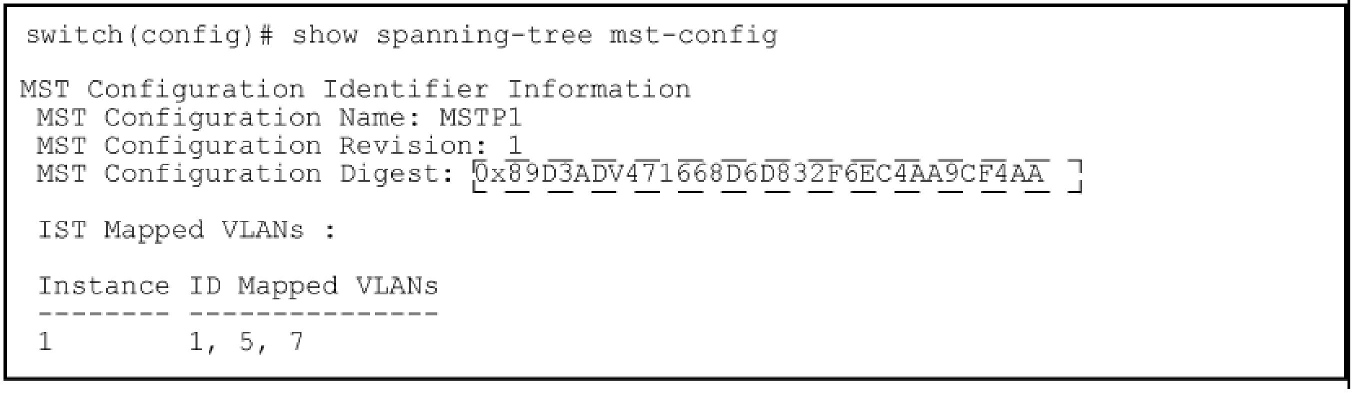


```
switch(config)# show spanning-tree mst-config

MST Configuration Identifier Information
 MST Configuration Name: MSTP1
 MST Configuration Revision: 1
 MST Configuration Digest: 0x51B7EBA6BEED8702D2BA4497D4367517

 IST Mapped VLANs :

 Instance ID Mapped VLANs
 --------- ---------------
 1         1-10
```

Configuration Digest value

The Configuration Digest value shown below is not the same as in the above example indicating that these switches do not operate in the same instance.

The Common Spanning Tree (CST) will still have the correct root associations.



```
 switch(config)# show spanning-tree mst-config

MST Configuration Identifier Information
 MST Configuration Name: MSTP1
 MST Configuration Revision: 1
 MST Configuration Digest: 0x89D3ADV471668D6D832F6EC4AA9CF4AA

 IST Mapped VLANs :

 Instance ID Mapped VLANs
 -------- ---------------
 1        1, 5, 7
```

Saving the current configuration before a software upgrade

Before updating to a new version of software, follow these steps:

Procedure

1. Enter the `show config files` command to display your current configuration files:

```
switch(config)# show config files

Configuration files:

 id | act pri sec | name
 ---+-------------+--------------------
  1 |  *   *   *  | config1
  2 |             | config2
  3 |             |
```

2. To save a configuration file for software version K.12.43, for example, type:

```
switch(config)# copy config config1 config configK1243.cfg
```

Choose any name for the saved configuration file that you prefer.

3. Display the configuration files as shown in the following example. Note the newly created configuration file listed.

```
switch(config)# show config files

Configuration files:

 id | act pri sec | name
 ---+-------------+----------------------
  1 |  *   *   *  | config1
  2 |             | config2
  3 |             | configK1243.cfg
```

4. Update the switch to the desired version, for example, K.12.51. Enter the `show flash` command to see the results. The switch is now running the software version K.12.51.

```
switch(config)# show flash

Image               Size(Bytes)    Date    Version   Build #:
-----               -----------  --------- -------   --------
Primary Image     : 6771179    04/17/08 K.12.51     304
Secondary Image : 7408949    11/06/08 K.12.43     123
Boot Rom Version: K.12.12
Default Boot     : Primary
```

5. To run the prior software version (K.12.43 in this example), type:

```
switch(config)# boot system flash secondary config configK1243.cfg
```

6. After rebooting, the switch is running software version K.12.43 and is using the configuration file that you saved for this software version, configK1243.cfg.

7. You can also save the K.12.43 configuration file on a TFTP server. To reload the K.12.43 version of the software again, reload the configuration file before doing the reload.

Types of Multiple Spanning Tree Instances

A multiple spanning tree network comprises separate spanning tree instances existing in an MST region. (There can be multiple regions in a network.) Each instance defines a single forwarding topology for an exclusive set of VLANs. By contrast, an STP or RSTP network has only one spanning tree instance for the entire network, and includes all VLANs in the network. (An STP or RSTP network operates as a single-instance network.) A region can include two types of STP instances:

- Internal spanning tree Instance (IST Instance) This is the default spanning tree instance in any MST region. It provides the root switch for the region and comprises all VLANs configured on the switches in the region that are not specifically assigned to Multiple Spanning Tree Instances (MSTIs, described below).Within a region, the IST instance provides a loop-free forwarding path for all VLANs associated with it. VLANs that are not associated with an MSTI are, by default, associated with the IST instance. Note that the switch automatically places dynamic VLANs (resulting from GVRP operation) in the IST instance. Dynamic VLANs cannot exist in an MSTI (described below).

- Multiple Spanning Tree Instance (MSTI)This type of configurable spanning tree instance comprises all static VLANs you specifically assign to it, and must include at least one VLAN. The VLANs you assign to an MSTI must initially exist in the IST instance of the same MST region. When you assign a static VLAN to an MSTI, the switch removes the VLAN from the IST instance. (Thus, you can assign a VLAN to only one MSTI in a given region.) All VLANs in an MSTI operate as part of the same single spanning tree topology. (The switch does not allow dynamic VLANs in an MSTI.)

CAUTION:

When you enable MSTP on the switch, the default MSTP spanning tree configuration settings comply with the values recommended in the IEEE 802.1s Multiple Spanning Tree Protocol (MSTP) standard. Inappropriate changes to these settings can result in severely degraded network performance. For this reason, HPE strongly recommends that changing these default settings be reserved only for experienced network administrators who have a strong understanding of the IEEE 802.1D/w/s standards and operation.

Planning an MSTP application

Before configuring MSTP, keep in mind the following tips and considerations:

- Ensure that the VLAN configuration in your network supports all the forwarding paths necessary for the desired connectivity. All ports connecting one switch to another within a region and one switch to another between regions should be configured as members of all VLANs configured in the region.
- Configure all ports or trunks connecting one switch to another within a region as members of all VLANs in the region. Otherwise, some VLANs could be blocked from access to the spanning tree root for an instance or for the region.
- Plan individual regions based on VLAN groupings. That is, plan on all MSTP switches in a given region supporting the same set of VLANs. Within each region, determine the VLAN membership for each spanning tree instance. (Each instance represents a single forwarding path for all VLANs in that instance.)
- Verify that there is one logical spanning tree path through the following:
  - Any interregional links
  - Any IST (Internal Spanning Tree) or Multiple Spanning Tree Instance within a region
  - Any legacy (802.1D or 802.1w) switch or group of switches. (Where multiple paths exist between an MST region and a legacy switch, expect the CST (Common Spanning Tree) to block all but one such path.)
- Determine the root bridge and root port for each instance.
- Determine the designated bridge and designated port for each LAN segment.
- Determine which VLANs to assign to each instance and use port trunks with 802.1Q VLAN tagging where separate links for separate VLANs would result in a blocked link preventing communication between nodes on the same VLAN. (See **MSTP operation with 802.1Q VLANs** .)
- Identify the edge ports connected to end nodes and enable the `admin-edge-port` setting for these ports. Leave the admin-edge-port setting disabled for ports connected to another switch, a bridge, or a half-duplex repeater.

IMPORTANT: When the Switch is configured in a stack, the number of configurable MSTIs has now been increased to 16, from 4. This is same as a standalone switch.

For the purposes of this guide, all examples assume that the Switch is standalone; therefore, a maximum of 16 instances are displayed.

Configuring MSTP at a glance

The general steps for configuring MSTP via the CLI are:

Procedure

1. Configure MSTP global parameters. This involves:

   a. Selecting MSTP as the spanning tree mode:

   ```
   spanning-tree mode mstp
   ```

   b. Clearing spanning tree debug counters:

   ```
   spanning-tree clear-debug-counters
   ```

   c. Specifying required parameters for MST region identity:

   ```
   Region Name:spanning-tree config-name Region Revision Number:spanning-tree
   config-revision
   ```

   d. Optionally, specifying MSTP parameter changes for region settings:

HPE recommends that you leave these parameters at their default settings for most networks. See the Caution below.

- The maximum number of hops before the MSTP BPDU (Bridge Protocol Data Unit) is discarded: `spanning-tree max-hops` (default: 20)
- Force-Version operation: `spanning-tree force-version`
- Forward Delay: `spanning-tree forward-delay`
- Hello Time (if it is the root device): `spanning-tree hello-time`
- Maximum age to allow for STP packets before discarding: `spanning-tree maximum-age`
- Device spanning tree priority. Specifies the priority value used along with the switch MAC address to determine which device is root. The lower a priority value, the higher the priority: `spanning-tree priority`

e. Enabling SNMP traps:

```
no spanning-tree trap [ errant-bpdu | loop-guard | new-root | root-guard ]
```

CAUTION: When you enable MSTP on the switch, the default MSTP spanning tree configuration settings comply with the values recommended in the IEEE 802.1s Multiple Spanning Tree Protocol (MSTP) standard. Inappropriate changes to these settings can result in severely degraded network performance. For this reason, HPE strongly recommends that changing these default settings be reserved only for experienced network administrators who have a strong understanding of the IEEE 802.1D/w/s standards and operation.

2. Configure per port parameters. HPE recommends that you use the default settings for these parameters and apply changes on a per-port basis only where a nondefault setting is clearly indicated by the circumstances of individual links. Other features you might consider include BPDU Filtering or BPDU Protection—these provide additional per-port control over spanning tree operations and security on the switch.

3. Configure MST instances. Configure one instance for each VLAN group that you want to operate as an active topology within the region to which the switch belongs. When you create the instance, you must include a minimum of one VID. You can add more VIDs later if desired. Use the following command:

```
spanning-tree instance n vlan vid
```

To move a VLAN from one instance to another, first use `no spanning-tree instance` *n* `vlan` *vid* to remove the mapping from the VLAN from the current instance, then add the VLAN to the other instance. (While the VLAN mapping is removed from an MSTI, it is associated with the region's IST instance.)

4. Configure the priority for each instance with the following command: `spanning-tree instance` *n* `priority` *n*

5. Configure MST instance port parameters. HPE recommends that you apply changes on a per-port basis only where a nondefault setting is clearly indicated by the circumstances of individual links. For example, you might want to set the path cost value for the *ist* or for the ports used by a specific MST instance. Use the following command:`spanning-tree instance <ist> | 1..64 <port-list> path-cost [ auto | 1..200000000 ]` Alternatively, leaving this setting at the default (auto) allows the switch to calculate the path-cost from the link speed.

6. Enable spanning tree operation on the switch with the `spanning-tree` command.

Configuring MSTP operation mode and global settings

The commands in this section apply at the switch (global) level.

Selecting MSTP as the spanning tree mode

Syntax:

```
spanning-tree mode mstp
```

Specifies that spanning tree will run in MSTP mode.

Clearing spanning tree debug counters

Syntax:

```
spanning-tree clear-debug-counters
```

Clears spanning tree debug counters.

Resetting the configuration name of the MST region in which a switch resides

Syntax:

```
spanning-tree config-name ascii-string
```

```
no spanning-tree config-name ascii-string
```

Resets the configuration name of the MST region in which the switch resides. This name can include up to 32 nonblank characters and is case-sensitive. On all switches within a given MST region, the configuration names must be identical. For more than one MSTP switch in the same MST region, the identical region name must be configured on all such switches. If the default configuration name is retained on a switch, it cannot exist in the same MST region with another switch.

The default name is a text string using the hexadecimal representation of the switch's MAC address.

The `no` form of the command overwrites the currently configured name with the default name.

NOTE: This option is available only when the switch is configured for MSTP operation. There is no defined limit on the number of regions you can configure.

Designating the revision number of the MST region for a switch

Syntax:

```
spanning-tree config-revision revision-number
```

Configures the revision number designated for the MST region in which you want the switch to reside. This setting must be the same for all switches residing in the same region. Use this setting to differentiate between region configurations in situations such as the following:

- Changing configuration settings within a region where you want to track the configuration versions you use
- Creating a new region from a subset of switches in a current region and want to maintain the same region name.
- Using the `pending` option to maintain two different configuration options for the same physical region.

This setting must be the same for all MSTP switches in the same MST region.

Range: 0 - 65535

Default: 0

NOTE:

This option is available only when the switch is configured for MSTP operation.

Setting the spanning tree compatibility mode

Syntax:

```
spanning-tree force-version [ stp-compatible | rstp-operation | mstp-operation ]
```

Sets the spanning tree compatibility mode. This command forces the switch to emulate behavior of earlier versions of spanning tree protocol, or return to MSTP behavior. The command is useful in test or debug applications, and removes the need to reconfigure the switch for temporary changes in spanning tree operation.

`stp-compatible`

The switch applies 802.1D STP operation on all ports.

`rstp-operation`

The switch applies 802.1w operation on all ports except those ports where it detects a system using 802.1D Spanning Tree. RSTP is Rapid Spanning Tree Protocol.

`mstp-operation`

The switch applies 802.1s MSTP operation on all ports where compatibility with 802.1D or 802.1w spanning tree protocols is not required.

NOTE:

Even when `mstp-operation` is selected, if the switch detects an 802.1D BPDU or an 802.1w BPDU on a port, it communicates with the device linked to that port using STP or RSTP BPDU packets. Also, if errors are encountered as described in **Configuring MSTP at a glance** on page 105, setting `force-version` to `stp-compatible` forces the MSTP switch to communicate out all ports using operations that are compatible with IEEE 802.1D STP.

NOTE:

When using MSTP rapid state transitions

Under some circumstances the rapid state transitions employed by MSTP can increase the rates of frame duplication and incorrect ordering in the switched LAN. To allow MSTP switches to support applications and protocols that may be sensitive to frame duplication and incorrect ordering, setting the Force Protocol Version (`force-version`) parameter to `stp-compatible` allows MSTP to operate with rapid transitions disabled. The value of this parameter applies to all ports on the switch.

Setting the time interval between listening, learning, and forwarding states

Syntax:

```
spanning-tree forward-delay
```

Sets the time the switch waits between transitions from listening to learning and from learning to forwarding states.

Range: 4 - 30

Default: 15 seconds

Setting spanning tree to operate in 802.1D legacy mode

Syntax:

```
spanning-tree legacy-mode
```

```
no spanning-tree legacy-mode
```

Forces spanning tree to operate in legacy (802.1D) mode.

Default: MSTP-operation.

The `no` form of this command returns the switch to the default 802.1s native mode (MSTP-operation.)

Setting spanning tree to operate with 802.1D legacy path cost values

Syntax:

```
spanning-tree legacy-path-cost
```

```
no spanning-tree legacy-path-cost
```

Forces spanning tree to operate with legacy (802.1D) path cost values.

Default: 802.1t.

The `no` form of the command returns the switch to the default 802.1t (not legacy) path cost values.

Specifying the time interval between BPDU transmissions

Syntax:

```
spanning-tree hello-time 1..10
```

If MSTP is running and the switch is operating as the CIST (Common and Internal Spanning Tree) root for your network, this command specifies the time in seconds between transmissions of BPDUs for all ports on the switch configured with the Global option (the default). This parameter applies in MSTP, RSTP, and STP modes.

During MSTP operation, you can override this global setting on a per-port basis with this command: `spanning-tree` *`port-list`* `hello-time` *`1..10`* .

Default: 2 seconds.

Setting the hop limit for BPDUs

Syntax:

```
spanning-tree max-hops hop-count
```

Resets the number of hops allowed for BPDUs in an MST region. When an MSTP switch receives a BPDU, it decrements the hop-count setting the BPDU carries. If the hop-count reaches zero, the receiving switch drops the BPDU.

The switch does not change the message-age and maximum-age data carried in the BPDU as it moves through the MST region and is propagated to other regions.

Range: 1 - 40

Default: 20

Setting the maximum age of received STP information

Syntax:

```
spanning-tree maximum age
```

Sets the maximum age time for received STP information before it is discarded.

Default: 20 seconds

Manipulating the pending MSTP configuration

Syntax:

```
spanning-tree pending [apply | config-name | config-revision | instance | reset]
```

Manipulates the pending MSTP configuration. The command is useful in test or debug applications, and enables rapid reconfiguration of the switch for changes in spanning tree operation.

`apply`

Applies pending MSTP configuration (swaps active and pending configurations).

`config-name`

Sets the pending MST region configuration name. Default is the switch's MAC address.

`config-revision`

Sets the pending MST region configuration revision number. Default is 0.

`instance`

Change pending MST instance configuration.

`reset`

Copies the active configuration to pending.

Setting the bridge priority for a region and determining the root switch

Syntax:

```
spanning-tree priority priority-multiplier
```

Every switch running an instance of MSTP has a Bridge Identifier, which is a unique identifier that helps distinguish this switch from all others. The switch with the lowest Bridge Identifier is elected as the root for the tree.

The Bridge Identifier is composed of a configurable priority component (2 bytes) and the bridge's MAC address (6 bytes). You can change the priority component provides flexibility in determining which switch will be the root for the tree, regardless of its MAC address.

This command sets the switch (bridge) priority for the designated region in which the switch resides. The switch compares this priority with the priorities of other switches in the same region to determine the root switch for the region. The lower the priority value, the higher the priority. If there is only one switch in the region, then that switch is the root switch for the region. The root bridge in a region provides the path to connected regions for the traffic in VLANs assigned to the region's IST instance. Traffic in VLANs assigned to a numbered STP instance in a given region moves to other regions through the root switch for that instance.

The priority range for an MSTP switch is 0-61440. However, this command specifies the priority as a multiplier (0 - 15) of 4096. That is, when you specify a priority multiplier value of 0 - 15, the actual priority assigned to the switch is: (priority-multiplier) x 4096

For example, with 2 as the priority-multiplier on a given MSTP switch, the Switch Priority setting is 8,192.

NOTE:

If multiple switches in the same MST region have the same priority setting, then the switch with the lowest MAC address becomes the root switch for that region.

Enabling SNMP traps

Syntax:

```
spanning-tree trap {errant-bpdu | loop-guard | new-root | root-guard}
```

```
no spanning-tree trap {errant-bpdu | loop-guard | new-root | root-guard}
```

Enables or disables SNMP traps for errant-BPDU, loop guard, new root, and root guard event notifications. This command is designed to be used in conjunction with the `spanning-tree bpdu-filter` command and the `bpdu-protection` command.

Parameters

`errant-bpdu`

Enables SNMP notification when an errant BPDU is received. Designed for use with BPDU filtering.

`loop-guard`

Enables SNMP notification when a loop guard inconsistency is detected. Designed for use with the Loop Guard option.

`new-root`

Enables SNMP notification when a new root is elected on any VLAN configured for MSTP on the switch.

`root-guard`

Enables SNMP notification when a root guard inconsistency is detected.

Default for all of the above options: Disabled

The `no` form of the command disables traps on the switch.

Configuring MSTP per-port parameters

In an MSTP topology, per-port parameters are set in the global configuration context. In most cases, HPE recommends that you use the default settings for these parameters and apply changes on a per-port basis only where a nondefault setting is clearly indicated by the circumstances of individual links. Some port parameters (such as `admin-edge-port`) affect all MSTI instances that consist of VLANs configured on the port. Other port parameters (such as `path-cost`) affect only the specified MST.

Enabling immediate transition to forwarding on end nodes

Syntax:

```
spanning-tree port-list admin-edge-port
```

```
no spanning-tree port-list admin-edge-port
```

Enables `admin-edge-port` on ports connected to end nodes. During spanning tree establishment, ports with `admin-edge-port` enabled transition immediately to the forwarding state. If a bridge or switch is detected on the segment, the port automatically operates as non-edge, not enabled.

Default: Disabled

If `admin-edge-port` is disabled on a port and `auto-edge-port` has not been disabled, the `auto-edge-port` setting controls the behavior of the port.

The no form of this command disables edge port operation on the specified ports.

Identifying edge ports automatically

Syntax:

```
spanning-tree port-list auto-edge-port
```

```
no spanning-tree port-list auto-edge-port
```

Enables automatic identification of edge ports for faster convergence. When enabled, the port looks for BPDUs for the first 3 seconds. If there are none, the port is classified as an edge port and immediately starts forwarding packets. If BPDUs are seen on the port, the port is classified as a non - edge port and normal STP operation commences on that port.

If admin-edge-port is enabled for a port, the setting for auto-edge-port is ignored whether set to yes or no.

If admin-edge-port is set to no, and auto-edge-port has not been disabled (set to no), then the auto-edge-port setting controls the behavior of the port.

CAUTION: Requires thorough knowledge of IEEE 802.1D/w/s standards and operation.

Spanning tree interprets a switch mesh as a single link. Because the switch automatically gives faster links a higher priority, the default MSTP parameter settings are usually adequate for spanning tree operation. Because incorrect MSTP settings can adversely affect network performance, do not change the MSTP settings from their default values unless you have a strong understanding of how spanning tree operates.

Default: Enabled

The no form of this command disables auto-edge-port operation on the specified ports.

Specifying the interval between BPDU transmissions

Syntax:

```
spanning-tree port-list hello-time [global | 1 - 10]
```

When the switch is the CIST root, this parameter specifies the interval (in seconds) between periodic BPDU transmissions by the designated ports. This interval also applies to all ports in all switches downstream from each port in the *port-list*.

A setting of global indicates that the ports in *port-list* on the CIST root are using the value set by the global spanning tree hello-time value.

When a given switch X is not the CIST root, the per-port hello-time for all active ports on switch X is propagated from the CIST root, and is the same as the hello-time in use on the CIST root port in the currently active path from switch X to the CIST root. When switch X is not the CIST root, then the upstream CIST root's port hello-time setting overrides the hello-time setting configured on switch X.

Default Per-Port setting: Use Global.

Default Global Hello-Time: 2.

Forcing a port to send RST/MST BPDUs

Syntax:

```
spanning-tree port-list mcheck
```

Forces a port to send RST/MST BPDUs for 3 seconds. This tests whether all STP bridges on the attached LAN have been removed and the port can migrate to native MSTP mode and use RST/MST BPDUs for transmission.

Determining which ports are forwarding ports by assigning port cost

Syntax:

```
spanning-tree port-list path-cost [auto | 1..200000000]
```

Assigns an individual port cost that the switch uses to determine which ports are forwarding ports in a given spanning tree. In the default configuration (auto) the switch determines a port's path cost by the port's type:

10 Mbps

`2000000`

100 Mbps

`200000`

1 Gbps

`20000`

Default: Auto

Informing the switch of the device type to which a port connects

Syntax:

```
spanning-tree port-list point-to-point-mac [true | false | auto]
```

Informs the switch of the type of device to which a specific port connects.

Parameters

`true`

(Default) Indicates a point-to-point link to a device such as a switch, bridge, or end-node.

`false`

Indicates a connection to a half-duplex repeater (which is a shared LAN segment).

`auto`

Causes the switch to set Force-False on the port if it is not running at full duplex.

Determining which port to use for forwarding

Syntax:

```
spanning-tree port-list priority priority-multiplier
```

MSTP uses this parameter to determine the port to use for forwarding. The port with the lowest priority number has the highest priority for use.

The range is 0 to 240, and is configured by specifying a multiplier from 0 - 15. When you specify a priority multiplier of 0 - 15, the actual priority assigned to the switch is:

(priority-multiplier) x 16

If you configure `2` as the priority multiplier on a given port, the actual Priority setting is 32. After specifying the port priority multiplier, the switch displays the actual port priority (and not the multiplier) in the `show spanning-tree` or `show spanning-tree` *`port-list`* displays.

You can view the actual multiplier setting for ports by executing `show running` and looking for an entry in this format:

```
spanning-treeport-list priority priority-multiplier
```

For example, configuring port A2 with a priority multiplier of `3` results in the following line in the `show running` output:

```
spanning-tree A2 priority 3
```

Denying a port the role of root port

Syntax:

```
spanning-treeport-list root-guard
```

When a port is enabled as `root-guard`, it cannot be selected as the root port even if it receives superior STP BPDUs. The port is assigned an "alternate" port role and enters a blocking state if it receives superior STP BPDUs.

A superior BPDU contains both "better" information on the root bridge and path cost to the root bridge, which would normally replace the current root bridge selection.

The superior BPDUs received on a port enabled as `root-guard` are ignored. All other BPDUs are accepted and the external devices may belong to the spanning tree as long as they do not claim to be the Root device.

Use this command on MSTP switch ports that are connected to devices located in other administrative network domains to:

- Ensure the stability of the core MSTP network topology so that undesired or damaging influences external to the network do not enter.
- Protect the configuration of the CIST root bridge that serves as the common root for the entire network.

Default: Disabled

Denying a port propagation change information

Syntax:

```
spanning-tree port-list tcn-guard
```

When enabled for a port, this causes the port to stop propagating received topology change notifications and topology changes to other ports.

Default: Disabled

Configure MST instance ports parameters

Syntax

```
spanning-tree instance 1-64 ethernet PORT-LIST
```

Description

Configure MST instance ports parameters.

Best practices

Follow the PORT-LIST with a '?' to get the list of all possible options.

Create a new instance or map VLAN(s) to an existing one

Syntax

```
spanning-tree instance ist | 1-64 vlan VLAN-ID
```

Description

Used to create a new instance or map VLAN(s) to an existing one. Each instance must have at least one VLAN mapped to it. The VLANs unmapped from other instances are automatically mapped to the IST instance. Only IST VLANs can be directly mapped to other instances. When VLANs are mapped to an instance, they are automatically unmapped from the instance they were mapped to before. Any MSTP instance can have all the VLANs configured in the switch.

Enable event logging

Syntax

```
no spanning-tree log state-transitions instance 1-64 | ist vlan
```

Description

By default port state change for IST is added in log.

Deleting an instance

Syntax

```
no spanning-tree instance <1-64>
```

Description

Deletes an instance. The IST instance cannot be deleted.

Configure an existent instance

Syntax

```
no spanning-tree instance <ist|1-64>
```

Description

Used to configure an existent instance.

Best Practices

Follow the syntax with a '?' to get a complete list of all the configurable parameters and sub-commands.

MSTP Config example

MSTP Config

```
VLAN 109
  ip addr 10.0.109.10/24
  tag 12
  exit

VLAN 110
  ip addr 10.0.110.10/24
  tag 12
  exit

Spanning-tree
Spanning-tree mode mstp
Spanning-tree config-name "MSTPRegion1"
Spanning-tree config-revision 1
Spanning-tree instance 1 VLAN 109
Spanning-tree instance 1 priority 4
Spanning-tree instance 2 VLAN 110
```

Downgrading to lower version build

The downgrade to lower version build will result in “stuck in boot” if more than 16 instances are created in the DUT.

Operating notes for the VLAN configuration enhancement

- Configuring MSTP on the switch automatically configures the Internal Spanning Tree (IST) instance and places all statically and dynamically configured VLANs on the switch into the IST instance. The spanning tree instance vlan command creates a new MST instance and moves the VLANs you specify from the IST to the MSTI.You must map a least one VLAN ID to an MSTI when you create it. You cannot map a VLAN ID to more than one instance. You can create up to 16 MSTIs in a region.
- The `no` form of the spanning tree instance vlan command removes one or more VLANs from the specified MSTI. If no VLANs are specified, the `no` form of the command deletes the specified MSTI.When you remove a VLAN from an MSTI, the VLAN returns to the IST instance, where it can remain or be reassigned to another MSTI configured in the region.
- If you enter the spanning tree instance vlan command before a static or dynamic VLAN is configured on the switch to preconfigure VLAN ID-to-MSTI mappings, no error message is displayed. Later, each newly configured VLAN that has already been associated with an MSTI is automatically assigned to the MSTI.This new default behavior differs from automatically including configured (static and dynamic) VLANs in the IST instance and requiring you to manually assign individual static VLANs to an MSTI.
- The valid VLAN IDs that you can map to a specified MSTI are from 1 to 4094. The VLAN ID-to-MSTI mapping does not require a VLAN to be already configured on the switch. The MSTP VLAN enhancement allows you to preconfigure MSTP topologies before the VLAN IDs associated with each instance exist on a switch.
- When you use preconfigured VLAN ID-to-MSTI topologies, ensure that MSTP switches remain in the same region by mapping all VLAN IDs used in the region to the same MSTIs on each regional switch.
- When you upgrade switch software to release K.13.XX and later, the existing MSTP topology configuration is automatically saved. All existing VLAN ID-to-MSTI assignments are maintained on a switch for uninterrupted MSTP network operation.

Configuring MST instance parameters

When you enable MSTP on the switch, a spanning tree instance is enabled automatically. The switch supports up to 16 configurable MST instances for each VLAN group that you want to operate as an active topology within the region to which the switch belongs. When creating an instance, you must include a minimum of one VID. You can add more VIDs later if desired.

Syntax:

```
spanning-tree instance 1..16 vlan vid [vid..vid]
```

```
no spanning-tree instance 1..16 vlan vid [vid..vid]
```

Configuring MSTP on the switch automatically configures the IST instance and places all statically and dynamically configured VLANs on the switch into the IST instance. This command creates a new MST instance (MSTI) and moves the VLANs you specify from the IST to the MSTI.

You must map at least one VLAN to an MSTI when you create it. You cannot map a VLAN ID to more than one instance. You can create up to 16 MSTIs in a region.

The `no` form of the command removes one or more VLANs from the specified MSTI. If no VLANs are specified, the `no` form of the command deletes the specified MSTI.

When you remove a VLAN from an MSTI, the VLAN returns to the IST instance, where it can remain or be re-assigned to another MSTI configured in the region.

NOTE: Starting in software release 13.x.x, you can enter the `spanning-tree instance vlan` command before a static or dynamic VLAN is configured on the switch to preconfigure VLAN ID-to-MSTI mappings. No error message is displayed. Later, each newly configured VLAN that has already been associated with an MSTI is automatically assigned to the MSTI.

This new default behavior differs from automatically including configured (static and dynamic) VLANs in the IST instance and requiring the manual assigning of individual static VLANs to an MSTI.

NOTE: The valid VLAN IDs that you can map to a specified MSTI are from 1 to 4094. The VLAN ID-to-MSTI mapping does not require a VLAN to be already configured on the switch. The MSTP VLAN enhancement allows you to preconfigure MSTP topologies before the VLAN IDs associated with each instance exist on a switch.

When using preconfigured VLAN ID-to-MSTI topologies, be sure that MSTP switches remain in the same region by mapping all VLAN IDs used in the region to the same MSTIs on each regional switch.

When upgrading switch software to release 13.x.x and later, the existing MSTP topology configuration is automatically saved. All existing VLAN ID-to-MSTI assignments are maintained on a switch for uninterrupted MSTP network operation.

Setting the bridge priority for an instance

Syntax:

```
spanning-tree instance 1..16 priority priority-multiplier
```

Sets the switch (bridge) priority for the designated instance. This priority is compared with the priorities of other switches in the same instance to determine the root switch. The lower the priority value, the higher the priority. If there is only one switch in the instance, then that switch is the root switch for the instance. The IST regional root bridge provides the path to instances in other regions that share one or more of the same VLANs.

The priority range for an MSTP switch is 0 - 61440. However, this command specifies the priority as a multiplier (0 - 15) of 4096. When a priority multiplier value is set from 0 - 15, the actual priority assigned to the switch for the specified MST instance is: (priority-multiplier) x 4096

For example, if you configure 5 as the priority-multiplier for MST Instance 1 on a given MSTP switch, the Switch Priority setting is 20,480 for that instance in that switch.

NOTE:

If multiple switches in the same MST instance have the same priority setting, the switch with the lowest MAC address becomes the root switch for that instance.

Assigning a port cost for an MST instance

Syntax:

```
spanning-tree instance ist | 1..16 port-list path-cost [auto | 1..200000000]
```

Assigns an individual port cost for the IST or for the specified MST instance.

For a given port, the path cost setting can be different for different MST instances to which the port may belong. The switch uses the path cost to determine which ports are the forwarding ports in the instance; that is, which links to use for the active topology of the instance and which ports to block.

The settings are either `auto` or in a range from 1 to 200,000,000. With the `auto` setting, the switch calculates the path cost from the link speed:

10 Mbps

`2000000`

100 Mbps

`200000`

1 Gbps

`20000`

Default

`Auto`

Setting the priority for a port in a specified MST instance

Syntax:

```
spanning-tree instance 1..16 port-list priority priority-multiplier
```

Sets the priority for the specified ports in the specified MST instance.

For a given port, the priority setting can be different for different MST instances to which the port may belong. The priority range for a port in a given MST instance is 0 - 255. However, this command specifies the priority as a multiplier (0 - 15) of 16. When you specify a priority multiplier of 0 - 15, the actual priority assigned to the switch is: (priority-multiplier) x 16

Setting priority for a port in a specified MST instance

If you configure 2 as the priority multiplier on a given port in an MST instance, then the actual Priority setting is 32x. After you specify the port priority multiplier in an instance, the switch displays the actual port priority and not the multiplier in the `show spanning-tree instance` *`1..16`* or `show spanning-tree` *`port-list`* `instance` *`1..16`* displays.

You can view the actual multiplier setting for ports in the specified instance by executing `show running` and looking for an entry in the following format:

```
spanning-tree instance 1..15 port-list priority priority-multiplier
```

For example, configuring port A2 with a priority multiplier of `3` in instance 1, results in this line in the `show running` output:

```
spanning-tree instance 1 A2 priority 3
```

Setting the priority for specified ports for the IST

Syntax:

```
spanning-tree port-list priority priority-multiplier
```

Sets the priority for the specified ports for the IST (Instance 0) of the region in which the switch resides.

The priority component of the port's Port Identifier is set. The Port Identifier is a unique identifier that helps distinguish this switch's ports from all others. It consists of the priority value with the port number extension—PRIORITY:PORT_NUMBER. A port with a lower value of Port Identifier is more likely to be included in the active topology.

This priority is compared with the priorities of other ports in the IST to determine which port is the root port for the IST instance. The lower the priority value, the higher the priority. The IST root port (or trunk) in a region provides the path to connected regions for the traffic in VLANs assigned to the region's IST instance.

The priority range for a port in a given MST instance is 0 - 240. However, this command specifies the priority as a multiplier (0 - 15) of 16. When you specify a priority multiplier of 0 - 15, the actual priority assigned to the switch is: (priority-multiplier) x 16

Setting priority for specified ports for an IST

Configuring 5 as the priority multiplier on a given port in the IST instance for a region creates an actual priority setting of 80. After specifying the port priority multiplier for the IST instance, the switch displays the actual port priority, not the multiplier, in the `show spanning-tree instance ist` or `show spanning-tree` *`port-list`* `instance ist` displays. You can view the actual multiplier setting for ports in the IST instance by executing `show running` and looking for an entry in this format:

```
spanning-tree port-list priority priority-multiplier
```

So configuring port A2 with a priority multiplier of `2` in the IST instance, results in this line in the `show running` output:

```
spanning-tree A2 priority 2
```

Enabling or disabling spanning tree operation

This command enables or disables spanning tree operation for any spanning tree protocol enabled on the switch. Before using to enable spanning tree, be sure that the right version is active on the switch.

Syntax:

```
no spanning-tree
```

Enables or disables spanning tree. Enabling spanning tree with MSTP configured, implements MSTP for all physical ports on the switch according to the VLAN groupings for the IST instance and any other configured instances.

Disabling MSTP removes protection against redundant loops that can significantly slow or halt a network.

This command simply turns spanning tree on or off. It does not change the existing spanning tree configuration.

NOTE: The convergence time for implementing MSTP changes can be disruptive to your network. To minimize such disruption, consider using the `spanning-tree pending` command.

Enabling an entire MST region at once or exchanging one region configuration for another

This operation exchanges the currently active MSTP configuration with the currently pending MSTP configuration, making it possible to implement a new MSTP configuration with minimal network disruption or to exchange MSTP configurations for testing or troubleshooting purposes.

When configuring or reconfiguring MSTP, the switch recalculates the corresponding network paths. This can have a ripple effect throughout your network as adjacent MSTP switches recalculate network paths to support the configuration changes invoked in a single switch. Although MSTP employs rapid spanning tree operation, the convergence time for implementing MSTP changes can be disruptive to your network. However, by using the `spanning-tree pending` feature, you can set up an MSTP on the switch and then invoke all instances of the new configuration at the same time, instead of one at a time.

Syntax:

```
no spanning-tree pending [apply | config-name | config-revision | instance | reset]
```

Exchanges the currently active MSTP configuration with the current pending MSTP configuration. Options are as follows:

`apply`

Exchanges the currently active MSTP configuration with the pending MSTP configuration.

`config-name`

Specifies the pending MST region name. Must be the same for all MSTP switches in the region. (Default: The switch's MAC address.)

`config-revision`

Specifies the pending MST region configuration revision number. Must be the same for all MSTP switches in the region. (Default: 0).

`instance`

Creates the pending instance and assigns one or more VLANs to the instance.

`reset`

Copies the switch's currently active MSTP configuration to the pending configuration. This is useful when you want to experiment with the current MSTP configuration while maintaining an unchanged version.

Creating a pending MSTP configuration

To create a pending MSTP configuration and exchange it with the active MSTP configuration:

Procedure

1. Configure the VLANs to include in any instances in the new region. When you execute the `pending` command, all VLANs configured on the switch will be assigned to a single pending IST instance unless assigned to other, pending MST instances. The `pending` command creates the region's IST instance automatically.
2. Configure MSTP as the spanning tree protocol, then execute `write mem` and reboot. The pending option is available only with MSTP enabled.
3. Configure the pending region *`config-name`* to assign to the switch.
4. Configure the pending *`config-revision`* number for the region name.

5. If you want an MST instance other than the IST instance, configure the instance number and assign the appropriate VLANs (VIDs) using the `pending instance1..16vlan [vid | vid-range] command.`
6. Repeat step 5 for each additional MST instance necessary.
7. To review your pending configuration, use the `show spanning-tree pending` command.
8. To exchange the currently active MSTP configuration with the pending MSTP configuration, use the `spanning-tree pending apply` command.

Viewing MSTP statistics

NOTE:

SNMP MIB Support for MSTP is a superset of the STP/802.1D and RSTP/802.1w protocols and uses the MIB objects defined for these two protocols.

Viewing global MSTP status

The following commands display the MSTP statistics for the connections between MST regions in a network.

Syntax:

```
show spanning-tree
```

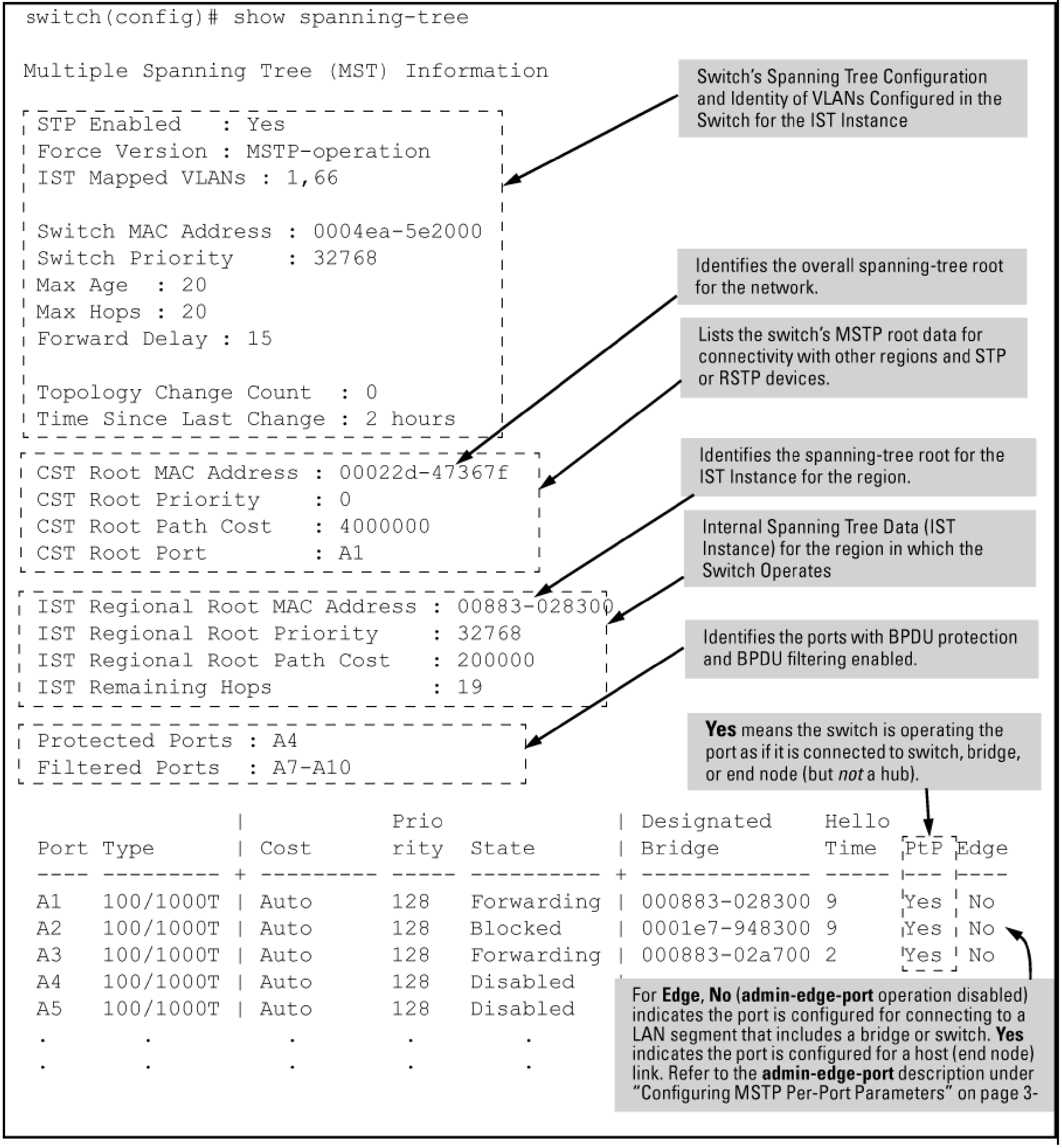
Displays the switch's global and regional spanning tree status, plus the per-port spanning tree operation at the regional level. Values for the following parameters appear only for ports connected to active devices: `Designated Bridge, Hello Time, PtP, and Edge`.

Syntax:

```
show spanning-tree port-list
```

Displays the spanning tree status for the designated ports. You can list data for a series of ports and port trunks by specifying the first and last port or trunk of any consecutive series of ports and trunks. For example, to display data for port A20-A24 and trk1, you would use this command: `show spanning-tree a20-a42,trk1`

Viewing a common spanning tree status



```
switch(config)# show spanning-tree

Multiple Spanning Tree (MST) Information

 STP Enabled   : Yes
 Force Version : MSTP-operation
 IST Mapped VLANs : 1,66

 Switch MAC Address : 0004ea-5e2000
 Switch Priority    : 32768
 Max Age  : 20
 Max Hops : 20
 Forward Delay : 15

 Topology Change Count  : 0
 Time Since Last Change : 2 hours

 CST Root MAC Address : 00022d-47367f
 CST Root Priority    : 0
 CST Root Path Cost   : 4000000
 CST Root Port        : A1

 IST Regional Root MAC Address : 00883-028300
 IST Regional Root Priority    : 32768
 IST Regional Root Path Cost   : 200000
 IST Remaining Hops            : 19

 Protected Ports : A4
 Filtered Ports  : A7-A10

                |           Prio              | Designated    Hello
 Port Type      | Cost      rity  State       | Bridge        Time  PtP  Edge
 ---- --------- + --------- ----- ---------- + ------------- ----- ---  ----
 A1   100/1000T | Auto      128   Forwarding | 000883-028300 9     Yes  No
 A2   100/1000T | Auto      128   Blocked    | 0001e7-948300 9     Yes  No
 A3   100/1000T | Auto      128   Forwarding | 000883-02a700 2     Yes  No
 A4   100/1000T | Auto      128   Disabled   |
 A5   100/1000T | Auto      128   Disabled
 .       .         .         .        .
 .       .         .         .        .
```

Switch's Spanning Tree Configuration and Identity of VLANs Configured in the Switch for the IST Instance

Identifies the overall spanning-tree root for the network.

Lists the switch's MSTP root data for connectivity with other regions and STP or RSTP devices.

Identifies the spanning-tree root for the IST Instance for the region.

Internal Spanning Tree Data (IST Instance) for the region in which the Switch Operates

Identifies the ports with BPDU protection and BPDU filtering enabled.

Yes means the switch is operating the port as if it is connected to switch, bridge, or end node (but *not* a hub).

For **Edge**, **No** (**admin-edge-port** operation disabled) indicates the port is configured for connecting to a LAN segment that includes a bridge or switch. **Yes** indicates the port is configured for a host (end node) link. Refer to the **admin-edge-port** description under "Configuring MSTP Per-Port Parameters" on page 3-

Viewing detailed port information

The following commands display the MSTP statistics for the connections between MST regions in a network.

Syntax:

```
show spanning-tree detail
```

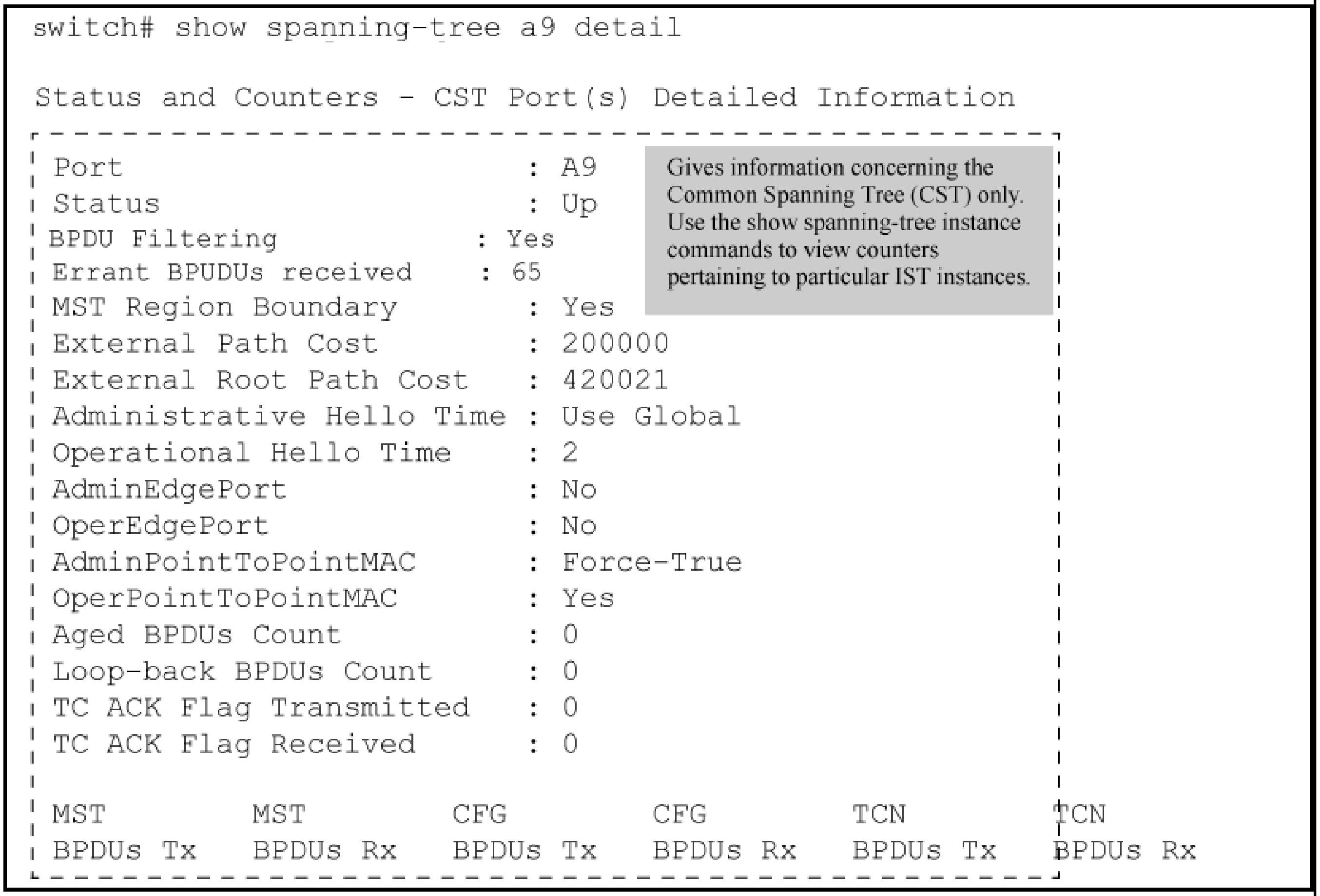
Displays additional parameters concerning the CST ports.

Syntax:

```
show spanning-tree port-list detail
```

Displays detailed spanning tree status for the designated ports.

Viewing port information



```
switch# show spanning-tree a9 detail

Status and Counters - CST Port(s) Detailed Information

 Port                          : A9
 Status                        : Up
BPDU Filtering              : Yes
 Errant BPUDUs received      : 65
 MST Region Boundary           : Yes
 External Path Cost            : 200000
 External Root Path Cost       : 420021
 Administrative Hello Time : Use Global
 Operational Hello Time        : 2
 AdminEdgePort                 : No
 OperEdgePort                  : No
 AdminPointToPointMAC          : Force-True
 OperPointToPointMAC           : Yes
 Aged BPDUs Count              : 0
 Loop-back BPDUs Count         : 0
 TC ACK Flag Transmitted       : 0
 TC ACK Flag Received          : 0

 MST         MST         CFG         CFG         TCN         TCN
 BPDUs Tx    BPDUs Rx    BPDUs Tx    BPDUs Rx    BPDUs Tx    BPDUs Rx
```

Gives information concerning the Common Spanning Tree (CST) only. Use the show spanning-tree instance commands to view counters pertaining to particular IST instances.

NOTE: This command gives information about the CST only. To view details of specific MST instances, use the `show spanning tree instance` commands.

Viewing status for a specific MST instance

The following commands display the MSTP statistics for a specified MST instance.

Syntax:

```
show spanning-tree instance [ist | 1..16]
```

Displays the MSTP statistics for either the IST instance or a numbered MST instance running on the switch.

Syntax:

```
show spanning-tree instance [ist | 1..16] detail
```

Displays status on all active ports for a specific instance of MSTP.

Syntax:

```
show spanning-tree port-list instance [ist | 1..16] detail
```

Displays status on specified ports for a specific instance of MSTP.

Viewing status for a specific instance of an MSTP

This shows how to display detailed status for all active ports for a specific instance of MSTP.

```
switch(config)# show spanning-tree instance 11
 MST Instance Information
  Instance ID : 11
  Mapped VLANs : 111,300
  Switch Priority        : 32768

  Topology Change Count   : 2
  Time Since Last Change  : 4 mins

 Regional Root MAC Address : 1cc1de-cfbc80
 Regional Root Priority    : 32768
 Regional Root Path Cost   : 400000
 Regional Root Port        : This switch is root
 Remaining Hops            : 20

                                                          Designated
 Port  Type      Cost      Priority Role       State      Bridge
 ----- --------- --------- -------- ---------- ---------- -------------
 1     10/100TX  200000    128      Root       Forwarding 1cc1de-cfbc80
 2     10/100TX  200000    128      Designated Forwarding 1cc1de-02a700
 3     10/100TX  Auto      112      Designated Forwarding 1cc1de-02a700
 4     10/100TX  Auto      128      Disabled   Disabled
 .        .       .         .          .          .
```

Viewing the MSTP configuration

MSTP configuration can be viewed at the global, per-instance, and regional level

Viewing the global MSTP configuration

This command displays the switch's basic and MST region spanning tree configuration, including basic port connectivity settings.

Syntax:

```
show spanning-tree config
```

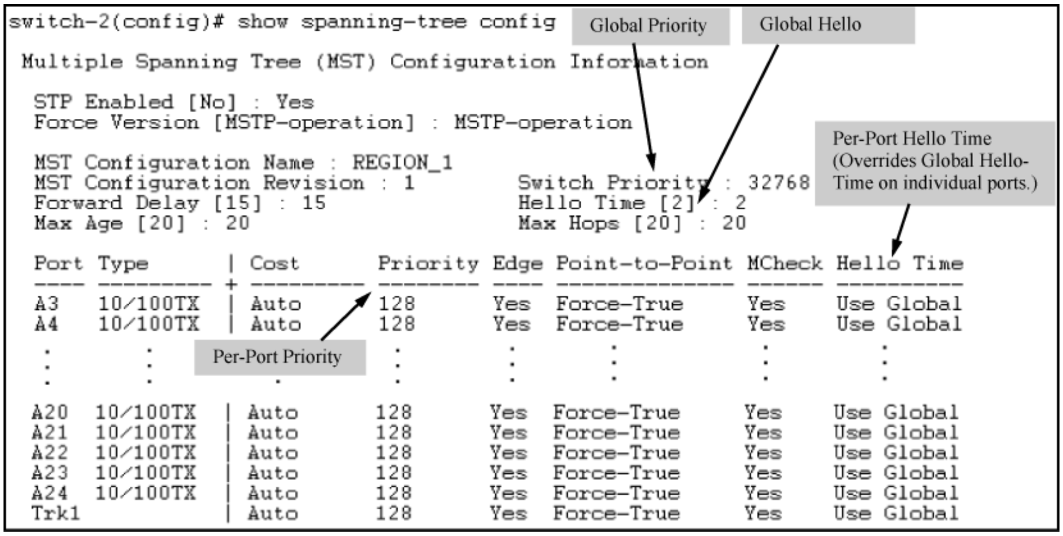
The upper part of this output shows the switch's global spanning tree configuration that applies to the MST region. The port listing shows the spanning tree port parameter settings for the spanning tree region operation configured by the `spanning-tree port-list` command.

Syntax:

```
show spanning-tree port-list config
```

This command shows the same data as the above command, but lists the spanning tree port parameter settings for only the specified port or trunk. You can list data for a series of ports and port trunks by specifying the first and last port or trunk of any consecutive series of ports and trunks. For example, to display data for port A20-A24 andtrk1, use the command: `show spanning-tree a20-a24,trk1 config`

Figure 13: *Viewing the switch's global spanning tree configuration*



```
switch-2(config)# show spanning-tree config

 Multiple Spanning Tree (MST) Configuration Information

  STP Enabled [No] : Yes
  Force Version [MSTP-operation] : MSTP-operation

  MST Configuration Name : REGION_1
  MST Configuration Revision : 1        Switch Priority : 32768
  Forward Delay [15] : 15               Hello Time [2] : 2
  Max Age [20] : 20                     Max Hops [20] : 20

  Port Type      | Cost      Priority Edge Point-to-Point MCheck Hello Time
  ---- --------- + --------- -------- ---- -------------- ------ ----------
  A3   10/100TX  | Auto      128      Yes  Force-True     Yes    Use Global
  A4   10/100TX  | Auto      128      Yes  Force-True     Yes    Use Global
   .       .                  .        .       .           .         .
   .       .                  .        .       .           .         .
   .       .          .       .        .       .           .         .

  A20  10/100TX  | Auto      128      Yes  Force-True     Yes    Use Global
  A21  10/100TX  | Auto      128      Yes  Force-True     Yes    Use Global
  A22  10/100TX  | Auto      128      Yes  Force-True     Yes    Use Global
  A23  10/100TX  | Auto      128      Yes  Force-True     Yes    Use Global
  A24  10/100TX  | Auto      128      Yes  Force-True     Yes    Use Global
  Trk1           | Auto      128      Yes  Force-True     Yes    Use Global
```

Viewing per-instance MSTP configurations

These commands display the per-instance port configuration and current state, along with instance identifiers and regional root data.

Syntax:

```
show spanning-tree config instance [ist | 1..16]
```

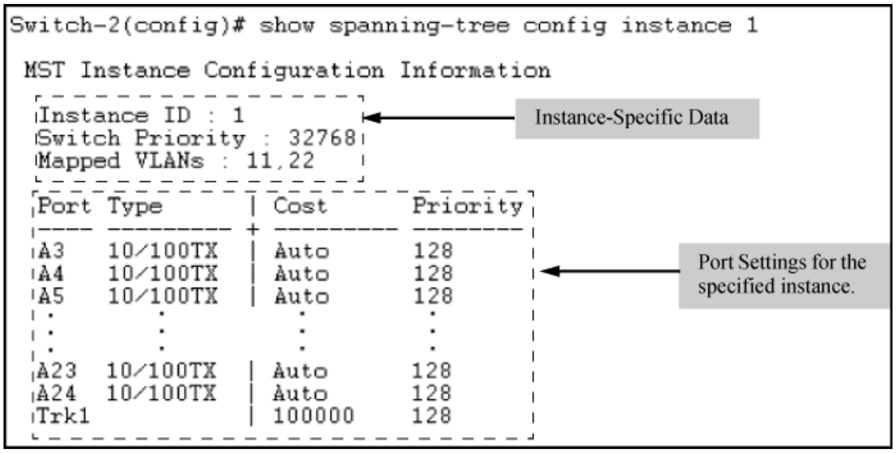
The upper part of this output shows the instance data for the ist or for the specified instance. The lower part of the output lists the spanning tree port settings for the specified instance.

Syntax:

```
show spanning-tree port-list config instance [ist | 1..16]
```

This command shows the same data as the preceding command, but lists the spanning tree port parameter settings for only the specified port or trunk. You can list data for a series of ports and port trunks by specifying the first and last port or trunk of any consecutive series of ports and trunks.

Viewing port data



```
Switch-2(config)# show spanning-tree config instance 1

 MST Instance Configuration Information

  Instance ID : 1
  Switch Priority : 32768
  Mapped VLANs : 11,22

  Port Type        | Cost       Priority
  ---- ---------- + ---------- --------
  A3   10/100TX   | Auto       128
  A4   10/100TX   | Auto       128
  A5   10/100TX   | Auto       128
  .       .          .          .
  .       .          .          .
  .       .          .          .
  A23  10/100TX   | Auto       128
  A24  10/100TX   | Auto       128
  Trk1            | 100000     128
```

To display data for ports A20-A24 and trk1, you would use the command:

```
switch(config)# show spanning-tree a20-a24,trk1 config instance 1
```

Viewing the region-level configuration

This command is useful for quickly verifying the allocation of VLANs in the switch's MSTP configuration, and for viewing the configured region identifiers.

Syntax:

```
show spanning-tree mst-config
```

NOTE:

The switch computes the MSTP Configuration Digest from the VID to MSTI configuration mappings on the switch itself. As required by the 802.1s standard, all MSTP switches within the same region must have the same VID to MSTI assignments, and any given VID can be assigned to either the IST or one of the MSTIs within the region. Thus, the MSTP Configuration Digest must be identical for all MSTP switches intended to belong to the same region. When comparing two MSTP switches, if their Digest identifiers do not match, they cannot be members of the same region.

Viewing a region-level configuration

```
switch(config)# show spanning-tree net-config

 MST Configuration Identifier Information

  MST Configuration Name : REGION_1
  MST Configuration Revision : 1
  MST Configuration Digest : 0xDAD6A13EC5141980B7EBDA71D8991E7C

  IST Mapped VLANs : 1,66

  Instance ID Mapped VLANs
  -------- ---------------
  1        11,22
  2        33,44,55
```

Viewing the pending MSTP configuration

This command displays the MSTP configuration the switch will implement if you execute the `spanning tree pending apply` command.

Syntax:

```
show spanning-tree pending [instance | mst-config]
```

instance [*1..16* | ist]

Lists region, instance ID, and VLAN information for the specified, pending instance.

mst-config

Lists region, IST instance VLANs, numbered instances, and assigned VLAN information for the pending MSTP configuration.

Viewing a pending configuration

```
switch(config)# show spanning-tree pending instance 3

 Pending MST Instance Configuration Information

  MST Configuration Name : New-Version_01
  MST Configuration Revision : 1
  Instance ID : 3
  Mapped VLANs : 3

switch(config)# show spanning-tree pending mst-config

 Pending MST Configuration Identifier Information

  MST Configuration Name : New-Version_01
  MST Configuration Revision : 1

  IST Mapped VLANs : 1,2,4-4094

  Instance ID Mapped VLANs
  ----------- ---------------------------------------
  3           3
```

MSTP operating rules

- All switches in a region must be configured with the same set of VLANs, the same MST configuration name and MST configuration number.
- Within a region, a VLAN can be allocated to either a single MSTI or to the region's IST instance.
- All switches in a region must have the same VID-to-MST instance assignment.
- There is one root MST switch per configured MST instance.
- Because boundary ports provide the VLAN connectivity between regions, all boundary ports on a region's root switch should be configured as members of all static VLANs defined in the region.
- There is one root switch for the Common and Internal Spanning Tree (CIST). At any given time, all switches in the network will use the per-port `hello-time` parameter assignments configured on the CIST root switch.
- Where multiple MST regions exist in a network, there is only one active, physical communication path between any two regions, or between an MST region and an STP or RSTP switch. MSTP blocks any other physical paths as long as the currently active path remains in service.

- Within a network, an MST region appears as a virtual RSTP bridge to other spanning tree entities (other MST regions, and any switches running 802.1D or 802.1w spanning tree protocols).
- Within an MSTI, there is one physical communication path between any two nodes, regardless of how many VLANs belong to the MSTI. Within an IST instance, there is also one spanning tree across all VLANs belonging to the IST instance.
- An MSTI comprises a unique set of VLANs and forms a single spanning tree instance within the region to which it belongs.
- A dynamic VLAN learned by GVRP will always be placed in the IST instance and cannot be moved to any configured MST instance.Starting in software release 13.X.X, dynamically learned GVRP VLANs can be mapped to MSTIs and support MSTP load balancing.
- In software release 13.X.X and later, you can preconfigure static and dynamic VLAN ID-to-MSTI mappings before the VLAN is created on the switch. Later, when the static VLAN ID is configured or a dynamic GVRP VLAN is learned, the VLAN is automatically associated with the preconfigured MSTI.
- Communication between MST regions uses a single spanning tree.
- If a port on a switch configured for MSTP receives a legacy (STP/802.1D or RSTP/802.1w) BPDU, it automatically operates as a legacy port. In this case, the MSTP switch interoperates with the connected STP or RSTP switch as a separate MST region.
- Within an MST region, there is one logical forwarding topology per instance, and each instance comprises a unique set of VLANs. Where multiple paths exist between a pair of nodes using VLANs belonging to the same instance, all but one of those paths will be blocked for that instance. However, if there are different paths in different instances, all such paths are available for traffic. Separate forwarding paths exist through separate spanning tree instances.
- A port can have different states (forwarding or blocking) for different instances (which represent different forwarding paths).
- MSTP interprets a switch mesh as a single link.

Troubleshooting an MSTP configuration

Cause

This section describes the `show spanning-tree` commands to use to monitor, troubleshoot, and debug the operation of a multiple-instance spanning tree configuration in a network.

The `show spanning-tree` commands described in this section allow for focusing on increasingly specific levels of operation. For example, you can display debug information for:

- All MST instances
- All ports used in one MST instance
- A specific port or several ports used in one MST instance

Also, you can display the change history for the root (bridge) switch used as the single forwarding path for:

- All MST regions, STP bridges, and RSTP bridges in an STP network
- All VLANs on MSTP switches in a region
- All VLANs on MSTP switches in an mst instance

Viewing the change history of root bridges

The show spanning-tree root-history command allows you to display change history information (up to 10 history entries) for a specified root bridge in any of the following MSTP topologies:

- Common Spanning Tree (cst):Provides connectivity in a bridged network between MST regions, STP LANs, and RSTP LANs.
- Internal Spanning Tree (ist):Provides connectivity within an MST region for VLANs associated with the default Common and Internal Spanning Tree (CIST) instance in your network (VLANs that have not been mapped to an MST instance).
- MST Instance (mst):Connects all static and (from release 13.X.X) dynamic VLANs assigned to a multiple spanning tree instance.

Syntax:

```
show spanning tree root-history [cst | ist | mst] instance-id
```

Displays the change history for the root bridge in the specified MSTP topology.

cst

Displays the change history for the root bridge of a spanning tree network, including MST regions and STP and RSTP bridges.

ist

Displays the change history for the root bridge in the IST instance of an MST region.

mst *instance-id*

Displays the change history for the root bridge in an MST instance, where *instance-id* is an ID number from 1 to 16.

Use the show spanning-tree root-history command to view the number and dates of changes in the assignment of a root bridge. Possible intrusion into your MST network may occur if an unauthorized external device gains access to a spanning tree by posing as the root device in a topology. To prevent an MST port connected to the device from being selected as the root port in a topology, use the spanning-tree root-guard command.

Sample output of the show spanning-tree root-history command for different MSTP topologies

The following examples show sample output of the show spanning-tree root-history command for different MSTP topologies. In each example, the root bridge ID is displayed in the format: *priority*: *mac-address*

Where:

- *priority*

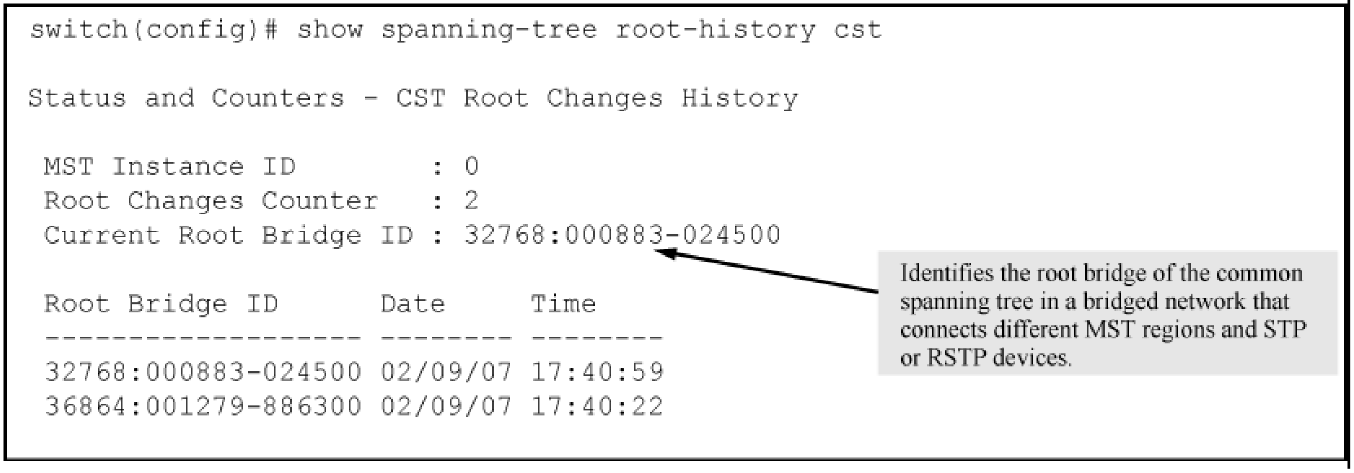
  is the MSTP switch priority calculated for one of the following:

  - The IST (regional) root switch using the spanning-tree priority command
  - An MSTI root switch using the spanning-tree instance priority command

- *mac-address*

  is the MAC address of the root (bridge) switch.

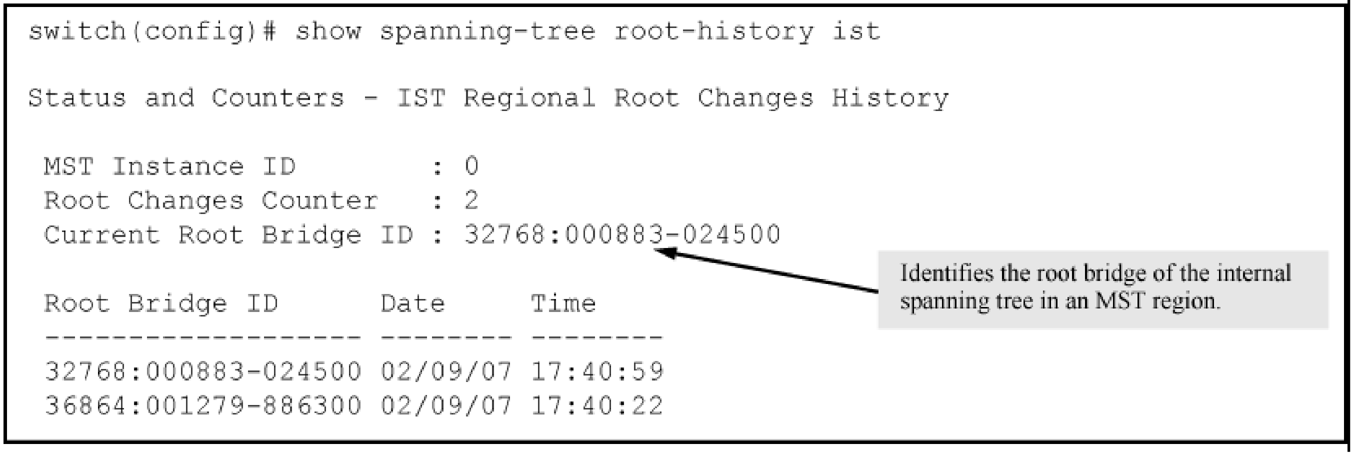
Viewing `show spanning-tree root-history` CST output



```
switch(config)# show spanning-tree root-history cst

Status and Counters - CST Root Changes History

 MST Instance ID        : 0
 Root Changes Counter   : 2
 Current Root Bridge ID : 32768:000883-024500

 Root Bridge ID       Date     Time
 -------------------- -------- --------
 32768:000883-024500 02/09/07 17:40:59
 36864:001279-886300 02/09/07 17:40:22
```

Identifies the root bridge of the common spanning tree in a bridged network that connects different MST regions and STP or RSTP devices.

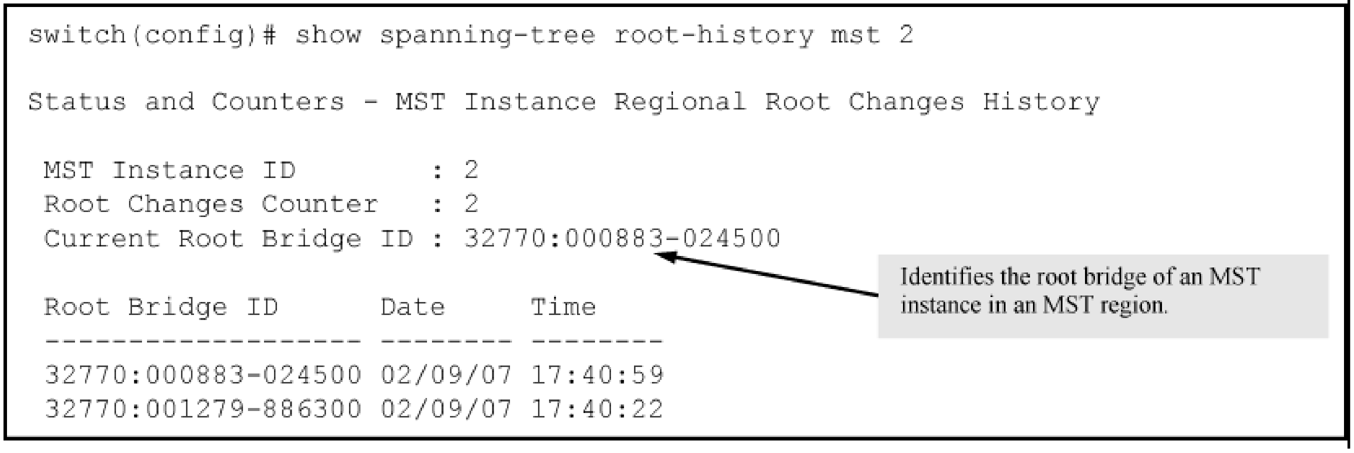
Viewing `show spanning-tree root-history` IST output



```
switch(config)# show spanning-tree root-history ist

Status and Counters - IST Regional Root Changes History

 MST Instance ID        : 0
 Root Changes Counter   : 2
 Current Root Bridge ID : 32768:000883-024500

 Root Bridge ID       Date     Time
 -------------------- -------- --------
 32768:000883-024500 02/09/07 17:40:59
 36864:001279-886300 02/09/07 17:40:22
```

Identifies the root bridge of the internal spanning tree in an MST region.

Viewing `show spanning-tree root-history` MSTI output



```
switch(config)# show spanning-tree root-history mst 2

Status and Counters - MST Instance Regional Root Changes History

 MST Instance ID        : 2
 Root Changes Counter   : 2
 Current Root Bridge ID : 32770:000883-024500

 Root Bridge ID       Date     Time
 -------------------- -------- --------
 32770:000883-024500 02/09/07 17:40:59
 32770:001279-886300 02/09/07 17:40:22
```

Identifies the root bridge of an MST instance in an MST region.

Enabling traps and viewing trap configuration

Syntax

```
spanning-tree trap {errant-bpdu | loop-guard | new-root | root-guard}
```

```
no spanning-tree trap {errant-bpdu | loop-guard | new-root | root-guard}
```

Enables SNMP traps. The `no` form of the command disables SNMP traps.

Syntax

```
show spanning-tree traps
```

Displays the current spanning tree trap configuration on the switch.

Viewing spanning tree traps in their default configuration

```
switch# show spanning-tree traps

 Status and Counters - STP Traps Information

 Trap Name              | Status
 ---------------------- + --------
 errant-bpdu            | Disabled
 new-root               | Disabled
 root-guard             | Disabled
 loop-guard             | Disabled
```

Viewing debug counters for all MST instances

The `show spanning-tree debug-counters` command allows you to display the aggregate values of all MSTP debug counters that are maintained on a switch. These aggregate values are a summary of the information collected from all ports and from all spanning tree instances that forward traffic on switch ports.

Use the displayed diagnostic information to globally monitor MSTP operation on a per-switch basis.

Syntax:

```
show spanning-tree debug-counters
```

Displays debug counters for MSTP activity on all ports configured for VLANs used in spanning tree instances.

Viewing output for debug counters

The following example shows sample output of the `show spanning-tree debug-counters` command for all ports.

```
switch(config)# show spanning-tree debug-counters

 Status and Counters - MSTP Bridge Common Debug Counters Information

  Counter Name                       Aggregated Value Collected From
  ---------------------------------- ---------------- --------------
  Invalid BPDUs                      0                CIST
  Errant BPDUs                       170927           CIST
  MST Config Error BPDUs             0                CIST
  Looped-back BPDUs                  0                CIST
  Starved BPDUs/MSTI MSGs            0                CIST/MSTIs
  Exceeded Max Age BPDUs             0                CIST
  Exceeded Max Hops BPDUs/MSTI MSGs 0                CIST/MSTIs
  Topology Changes Detected         2                CIST/MSTIs
  Topology Changes Tx               6                CIST/MSTIs
  Topology Changes Rx               4                CIST/MSTIs
  Topology Change ACKs Tx           0                CIST
  Topology Change ACKs Rx           0                CIST
  TCN BPDUs Tx                      0                CIST
  TCN BPDUs Rx                      0                CIST
  CFG BPDUs Tx                      0                CIST
  CFG BPDUs Rx                      0                CIST
  RST BPDUs Tx                      0                CIST
  RST BPDUs Rx                      0                CIST
  MST BPDUs/MSTI MSGs Tx            10               CIST/MSTIs
  MST BPDUs/MSTI MSGs Rx            341802           CIST/MSTIs
```

Viewing debug counters for one MST instance

The `show spanning-tree debug-counters instance` command allows you to display the aggregate values of all MSTP debug counters maintained on a switch for a specified spanning tree instance. These aggregate values are a summary of information collected from all ports that have VLANs assigned to the specified instance.

Use this command to troubleshoot the global MSTP diagnostic information displayed in `show spanning-tree debug-counters` command output when you suspect unauthorized MSTP activity in a specific MST instance.

Syntax:

`show spanning-tree debug-counters instance` *`instance-id`*

Displays debug counters for MSTP activity on all ports configured for VLANs in the specified MST instance.

The valid values for `instance` *instance-id* are 0 to 16, where 0 specifies the default MST (CIST) instance and 1 to 16 specify a multiple spanning tree (MST) instance.

Viewing bug counters for a CIST instance

The following example shows sample output of the `show spanning-tree debug-counters instance` command when applied to the Common and Internal Spanning Tree (CIST) instance (default MST instance 0) in the network.

```
switch(config)# show spanning-tree debug-counters instance 0

 Status and Counters - CIST Common Debug Counters Information

  MST Instance ID : 0

  Counter Name                      Aggregated Value Collected From
  --------------------------------- ---------------- --------------
  Invalid BPDUs                     0                Ports
  Errant BPDUs                      172603           Ports
  MST Config Error BPDUs            0                Ports
  Looped-back BPDUs                 0                Ports
  Starved BPDUs                     0                Ports
  Exceeded Max Age BPDUs            0                Ports
  Exceeded Max Hops BPDUs           0                Ports
  Topology Changes Detected         1                Ports
  Topology Changes Tx               3                Ports
  Topology Changes Rx               2                Ports
  Topology Change ACKs Tx           0                Ports
  Topology Change ACKs Rx           0                Ports
  TCN BPDUs Tx                      0                Ports
  TCN BPDUs Rx                      0                Ports
  CFG BPDUs Tx                      0                Ports
  CFG BPDUs Rx                      0                Ports
  RST BPDUs Tx                      0                Ports
  RST BPDUs Rx                      0                Ports
  MST BPDUs Tx                      5                Ports
  MST BPDUs Rx                      172577           Ports
```

Viewing debug counters for ports in an MST instance

The `show spanning-tree debug-counters instance ports` command displays the aggregate values of all MSTP debug counters maintained on one or more ports used by a specified spanning tree instance. These aggregate values are a summary of information collected from the specified ports that have VLANs assigned to the specified instance.

Use this command to troubleshoot at a finer level the more general MSTP diagnostic information displayed in the `show spanning-tree debug-counters instance` command output, when you suspect unauthorized MSTP activity on one or more MST ports in an MST instance.

Syntax:

`show spanning-tree debug-counters instance` *`instance-id`* `ports` *`port-list`*

Displays debug counters for MSTP activity on the specified ports configured for VLANs in the specified MST instance.

`instance` ***instance-id***

The valid values for *instance-id* are from 0 to 16, where 0 specifies the default MST (CIST) instance and 1 to 16 specify an MST instance.

`ports` ***port-list***

Specifies one or more MST ports or trunk ports. In the port list, enter a series of ports by separating the first and last ports in the series with a dash (-); for example, `a2-a8` or `trk1-trk3`. Separate individual ports and series of ports with a comma; for example, `a2-a8, a20, trk1, trk4-trk5`.

Viewing debug counters for a CIST and MST instance

The following example shows sample output of the `show spanning-tree debug-counters instance ports` command for both the CIST (default MST instance 0) and an MST instance (instance 2) on port A15.

```
switch(config)# show spanning-tree debug-counters instance 0 ports a15

 Status and Counters - CIST Port(s) Debug Counters Information

  MST Instance ID : 0
  Port : A15

  Counter Name                Value      Last Updated
  --------------------------- ---------- -----------------
  Invalid BPDUs               0
  Errant BPDUs                0
  MST Config Error BPDUs      0
  Looped-back BPDUs           0
  Starved BPDUs               0
  Exceeded Max Age BPDUs      0
  Exceeded Max Hops BPDUs     0
  Topology Changes Detected   1         02/09/07 17:40:59
  Topology Changes Tx         3         02/09/07 17:41:03
  Topology Changes Rx         2         02/09/07 17:41:01
  Topology Change ACKs Tx     0
  Topology Change ACKs Rx     0
  TCN BPDUs Tx                0
  TCN BPDUs Rx                0
  CFG BPDUs Tx                0
  CFG BPDUs Rx                0
  RST BPDUs Tx                0
  RST BPDUs Rx                0
  MST BPDUs Tx                5          02/09/07 17:41:03
  MST BPDUs Rx                173540     02/13/07 18:05:34
```

Viewing debug counters output for one port in an MST instance

The following example shows spanning tree debug-counters instance ports command output for one port in an MST instance.

```
switch(config)# show spanning-tree debug-counters instance 2 ports a15

 Status and Counters - MSTI Port(s) Debug Counters Information

   MST Instance ID : 2
   Port : A15

   Counter Name                Value      Last Updated
   --------------------------- ---------- -----------------
   Starved MSTI MSGs           0
   Exceeded Max Hops MSTI MSGs 0
   Topology Changes Detected   1          02/09/07 17:40:59
   Topology Changes Tx         3          02/09/07 17:41:03
   Topology Changes Rx         2          02/09/07 17:41:01
   MSTI MSGs Tx                5          02/09/07 17:41:03
   MSTI MSGs Rx                173489     02/13/07 18:03:52
```

Field descriptions in MSTP debug command output

The following table contains descriptions of the debugging information displayed in the output of `show spanning-tree debug-counters` commands.

Table 10: *MSTP debug command output: field descriptions*

Field	Displays the number of...
`Invalid BPDUs`	Received BPDUs that failed standard MSTP (802.1Q-REV/D5.0 14.4) validation checks and were dropped. This counter is maintained by the CIST (default MST instance 0) on a per-port basis.
`Errant BPDUs`	Received BPDUs that were dropped on a port that is configured to not expect BPDU packets. This counter is maintained by the CIST (MST instance, 0default MST instance 0 in the network) on a per-port basis and is incremented each time a BPDU packet is received on a port configured with the BPDU filter to ignore incoming BPDU packets (`spanning-tree bpdu-filter` command) or the BPDU protection feature to disable the port when BPDU packets are received (`spanning-tree bpdu-protection` command).

Table Continued

Field	Displays the number of...
`MST Config Error BPDUs`	BPDUs received from a neighbor bridge with inconsistent MST configuration information. For example, BPDUs from a transmitting bridge may contain the same MST configuration identifiers (region name and revision number) and format selector as the receiving bridge, but the value of the Configuration Digest field (VLAN ID assignments to regional IST and MST instances) is different. This difference indicates a probable configuration error in MST region settings on the communicating bridges. The received BPDU is still processed by MSTP. This counter is maintained by the CIST (default MST instance 0) on a per-port basis.
`Looped-back BPDUs`	Times a port has received self-sent BPDU packets as the result of an external loop condition in which the BPDUs were looped back to the originating transmission port. The received BPDU is still processed by MSTP and the port changes to a blocked state. This counter is maintained by the CIST (default MST instance 0) on a per-port basis.
`Starved BPDUs`	Times that no BPDUs are received within the scheduled interval (three times the Hello Time value configured with the `spanning-tree hello-time` command) from a downstream CIST-designated peer port on the CIST root, alternate, or backup port. As a result, the "starved" port triggers a spanning tree topology regeneration. This counter is maintained by the CIST (default MST instance 0) on a per-port basis.
`Starved MSTI MSGs`	Times that no BPDUs are received within the scheduled interval (three times the Hello Time value configured with the `spanning-tree hello-time` command) from a downstream MSTI-designated peer port on the MSTI root, alternate, or backup port. As a result, the "starved" port triggers a spanning tree topology regeneration. This counter is maintained by the CIST (default MST instance 0) on a per-port basis.
`Exceeded Max Age BPDUs`	Times that a BPDU packet is received from a bridge external to the MST region with a Message Age value greater than the configured value of the Max Age parameter (`spanning-tree maximum age` command). This may occur if the receiving bridge is located too far from the root bridge (beyond the configured size of the spanning tree domain on the root bridge) or if a BPDU packet with invalid root information is continuously circulating between bridges in a spanning tree domain and needs to be aged out. This counter is maintained by the CIST (default MST instance 0) on a per-port basis.

Table Continued

Field	Displays the number of...
`Exceeded Max Hops BPDUs`	Times that a BPDU packet is received from a bridge internal to the MST region with a CIST Remaining Hops value less than or equal to 1. This may occur if the receiving bridge is located too far from the CIST regional root bridge (beyond the configured size of the MST region on the CIST regional root bridge) or if a BPDU packet with invalid CIST regional root bridge information is continuously circulating between bridges in the MST Region and needs to be aged out. This counter is maintained by the CIST (default MST instance 0 in the region) on a per-port basis.
`Exceeded Max Hops MSTI MSGs`	Times that an MSTI MSG packet is received from a bridge internal to the MST region with an MSTI Remaining Hops value less than or equal to 1. This may occur if the receiving bridge is located too far from the MSTI regional root bridge (beyond the configured size of the MST region on the MSTI regional root bridge) or if a BPDU packet with invalid MSTI regional root bridge information is continuously circulating between bridges in an MST region and needs to be aged out. This counter is maintained on a per-MSTI per-port basis.
`Topology Changes Detected`	Times that a Topology Change event is detected by the CIST or MSTI port and the port triggers a topology change propagation throughout the network. A Topology Change event occurs when a non-edge port enters forwarding state. This counter is maintained on a per-CIST per-port and on a per-MSTI per-port basis.
`Topology Changes Tx`	Times that Topology Change information is propagated (sent out) through the port to the rest of the network.For a CIST port, the counter is the number of times that a CFG, RST, or MST BPDU with the TC flag set is transmitted out of the port.For an MSTI port, the counter is the number of times that an MSTI configuration message with the TC flag set is transmitted out of the port. This counter is maintained on a per-CIST per-port and on a per-MSTI per-port bases.
`Topology Changes Rx`	Times that Topology Change information is received from the peer port.For a CIST port, the counter is the number of times that a CFG, RST, or MST BPDU with the TC flag set is received.For an MSTI port, the counter is the number of times that an MSTI configuration message with the TC flag set is received. This counter is maintained on a per-CIST per-port and on a per-MSTI per-port basis.
`Topology Change ACKs Tx`	Times that the Topology Change acknowledgement is transmitted through the port (number of CFG, RST or MST BPDUs transmitted with the Topology Change Acknowledge flag set). This counter is maintained by the CIST (default MST instance 0) on a per-port basis.

Table Continued

Field	Displays the number of...
`Topology Change ACKs Rx`	Times the Topology Change acknowledgement is received on the port (number of CFG, RST or MST BPDUs received with the Topology Change Acknowledge flag set). This counter is maintained by the CIST (default MST instance 0) on a per-port basis.
`TCN BPDUs Tx`	Topology Change Notification BPDUs that are transmitted through the port. This counter is maintained by the CIST (default MST instance 0) on a per-port basis.
`TCN BPDUs Rx`	Topology Change Notification BPDUs that are received on the port. This counter is maintained by the CIST (default MST instance 0) on a per-port basis.
`CFG BPDUs Tx`	802.1D Configuration BPDUs that are transmitted through the port. This counter is maintained by the CIST (default MST instance 0) on a per-port basis.
`CFG BPDUs Rx`	802.1D Configuration BPDUs that are received on the port. This counter maintained by the CIST (default MST instance 0) on a per-port basis.
`RST BPDUs Tx`	802.1w RST BPDUs that are transmitted through the port. This counter is maintained by the CIST (default MST instance 0) on a per-port basis.
`RST BPDUs Rx`	802.1w RST BPDUs that are received on the port. This counter is maintained by the CIST (default MST instance 0) on a per-port basis.
`MST BPDUs Tx`	802.1s MST BPDUs that are transmitted through the port. This counter is maintained by the CIST (default MST instance 0) on a per-port basis.
`MST BPDUs Rx`	802.1s MST BPDUs that are received on the port. This counter is maintained by the CIST (default MST instance 0) on a per-port basis.

Table Continued

Field	Displays the number of...
`MSTI MSGs Tx`	Times that a configuration message for a specific MSTI was encoded in (802.1s) MST BPDUs that are transmitted through the port. This counter is maintained on a per-MSTI per-port basis.
`MSTI MSGs Rx`	Times that the MSTI detected a configuration message destined to the MSTI in (802.1s) MST BPDUs received on the port. This counter is maintained on a per-MSTI per-port basis.

Troubleshooting MSTP operation

Table 11: *Troubleshooting MSTP operation*

Problem	Possible cause
Duplicate packets on a VLAN, or packets not arriving on a LAN at all.	The allocation of VLANs to MSTIs may not be identical among all switches in a region.
A switch intended to operate in a region does not receive traffic from other switches in the region.	An MSTP switch intended for a particular region may not have the same configuration name or region revision number as the other switches intended for the same region. The MSTP configuration name (`spanning-tree config-name` command) and MSTP configuration revision number (`spanning-tree config-revision` command) must be identical on all MSTP switches intended for the same region. Another possible cause is that the set of VLANs and VLAN ID-to-MSTI mappings (`spanning-tree instance vlan` command) configured on the switch may not match the set of VLANs and VLAN ID-to-MSTI mappings configured on other switches in the intended region.

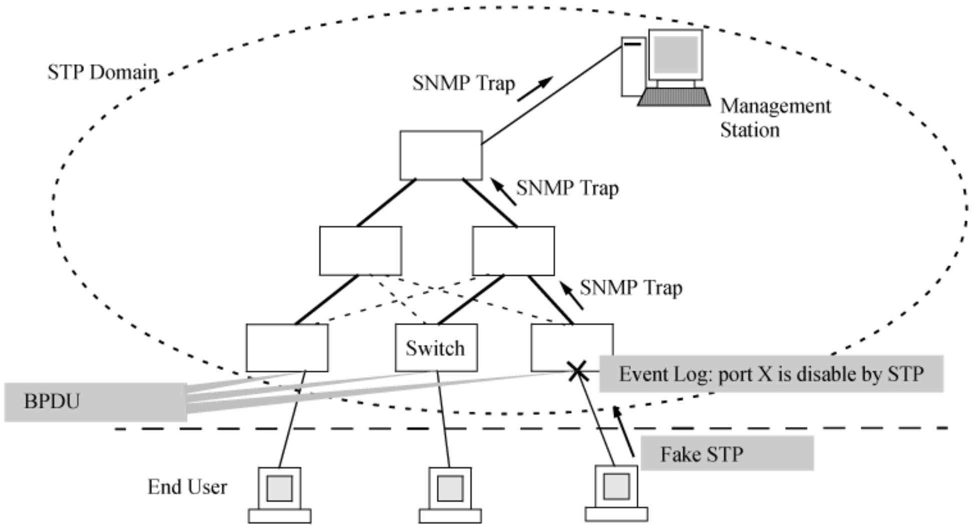
BPDU

BPDUs are data messages that are exchanged across the switches within an extended LAN that uses a spanning tree protocol topology. BPDU packets contain information on ports, addresses, priorities, and costs.

About BPDU protection

BPDU protection is a security feature designed to protect the active STP topology by preventing spoofed BPDU packets from entering the STP domain. In a typical implementation, BPDU protection would be applied to edge ports connected to end user devices that do not run STP. If STP BPDU packets are received on a protected port, the feature will disable that port and alert the network manager via an SNMP trap as shown below.

BPDU protection enabled at the network edge



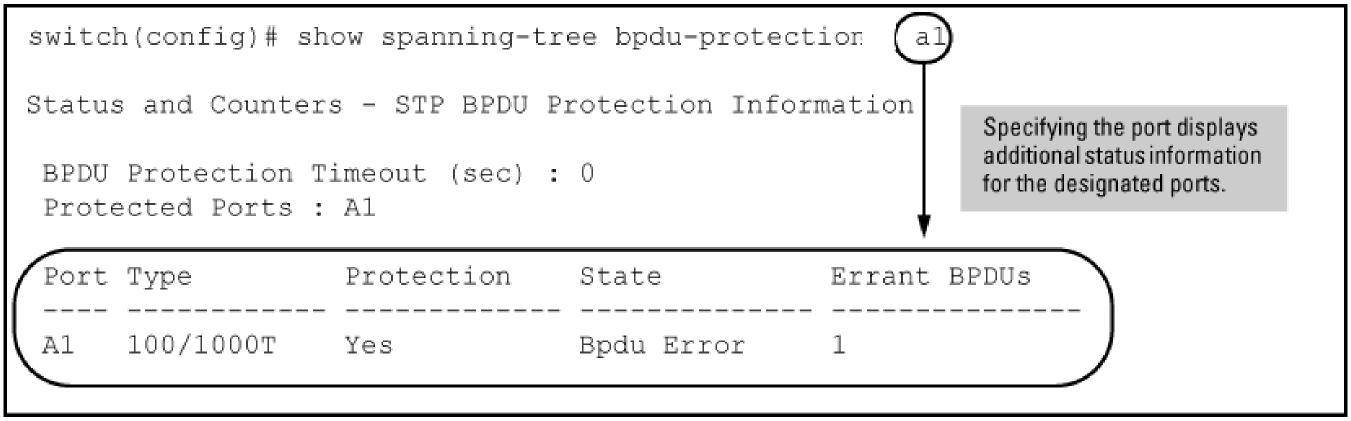
Viewing BPDU protection status

Syntax:

```
show spanning-tree bpdu-protection
```

Displays a summary listing of ports with BPDU protection enabled. To display detailed per port status information, enter the specific port numbers as shown here.

Figure 14: *Viewing BPDU protection status*

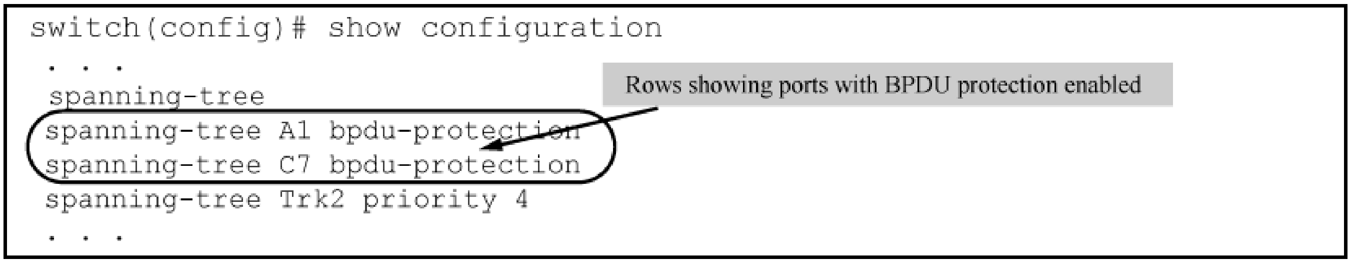


```
switch(config)# show spanning-tree bpdu-protection a1

Status and Counters - STP BPDU Protection Information

 BPDU Protection Timeout (sec) : 0
 Protected Ports : A1

 Port Type          Protection    State          Errant BPDUs
 ---- ------------ ------------- -------------- ---------------
 A1   100/1000T     Yes           Bpdu Error     1
```

Specifying the port displays additional status information for the designated ports.

BPDU protected ports are displayed as separate entries of the spanning tree category within the configuration file.

Figure 15: *Viewing BPDU filters using the* `show configuration` *command*



```
switch(config)# show configuration
 . . .
 spanning-tree
 spanning-tree A1 bpdu-protection
 spanning-tree C7 bpdu-protection
 spanning-tree Trk2 priority 4
 . . .
```

Rows showing ports with BPDU protection enabled

Configuring BPDU filtering

The STP BPDU filter feature allows control of spanning tree participation on a per-port basis. It can be used to exclude specific ports from becoming part of spanning tree operations. A port with the BPDU filter enabled will ignore incoming BPDU packets and stay locked in the spanning tree forwarding state. All other ports will maintain their role.

Syntax:

`spanning-tree [`*`port-list`*` | all] bpdu-filter`

`no spanning-tree [`*`port-list`*` | all] bpdu-filter`

Enables or disables the BPDU filter feature on specified port(s). This forces a port to always stay in the forwarding state and be excluded from standard STP operation.

Sample scenarios in which this feature may be used are:

- To have STP operations running on selected ports of the switch rather than every port of the switch at a time.
- To prevent the spread of errant BPDU frames.
- To eliminate the need for a topology change when a port's link status changes. For example, ports that connect to servers and workstations can be configured to remain outside of spanning tree operations.
- To protect the network from denial of service attacks that use spoofing BPDUs by dropping incoming BPDU frames. For this scenario, BPDU protection offers a more secure alternative, implementing port shut-down and a detection alert when errant BPDU frames are received.

CAUTION: Ports configured with the BPDU filter mode remain active (learning and forward frames); however, spanning tree cannot receive or transmit BPDUs on the port. The port remains in a forwarding state, permitting all broadcast traffic. This can create a network storm if there are any loops (that is, trunks or redundant links) using these ports. If you suddenly have a high load, disconnect the link and disable the bpdu-filter (using the `no` command).

Configuring BPDU filtering

To configure BPDU filtering on port a9, enter:

```
switch(config)# spanning-tree a9 bpdu-filter
```

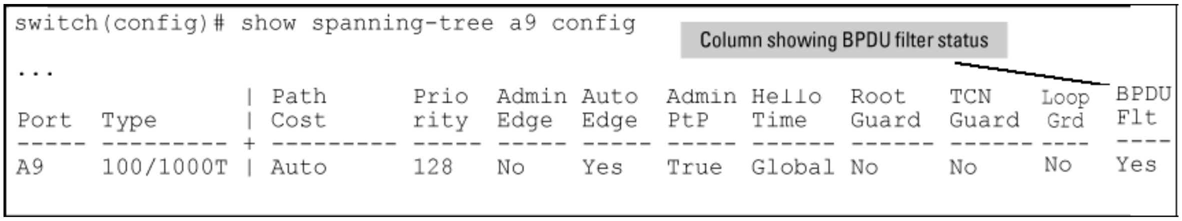
Viewing BPDU filtering

Syntax:

`spanning-tree show `*`port`*` configuration`

Displays the BPDU filter state.

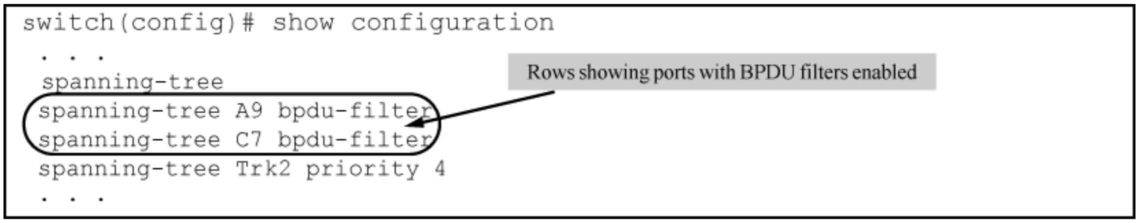
Viewing BPDU filter status using the `show spanning tree` command



```
switch(config)# show spanning-tree a9 config

...
                 | Path       Prio  Admin Auto  Admin Hello  Root   TCN    Loop BPDU
Port  Type       | Cost       rity  Edge  Edge  PtP   Time   Guard  Guard  Grd  Flt
----- ---------- + ---------- ----- ----- ----- ----- ------ ------ ------ ---- ----
A9    100/1000T  | Auto       128   No    Yes   True  Global No     No     No   Yes
```

Viewing BPDU filters using the `show configuration` command

BPDU filters per port are displayed as separate entries of the spanning tree category within the configuration file.

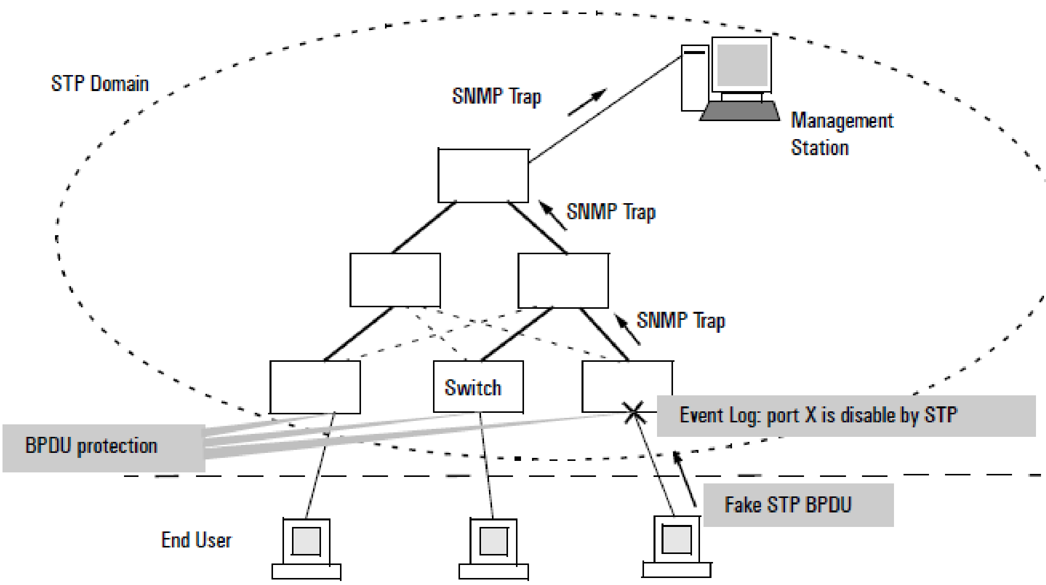


```
switch(config)# show configuration
 . . .
 spanning-tree
 spanning-tree A9 bpdu-filter
 spanning-tree C7 bpdu-filter
 spanning-tree Trk2 priority 4
 . . .
```

Configuring and managing BPDU protection

BPDU protection is a security feature designed to protect the active STP topology by preventing spoofed BPDU packets from entering the STP domain. In a typical implementation, BPDU protection would be applied to edge ports connected to end user devices that do not run STP. If STP BPDU packets are received on a protected port, the feature will disable that port and alert the network manager via an SNMP trap as shown in the following diagram.

Figure 16: *BPDU protection enabled at the network edge*



The following commands allow you to configure BPDU protection on VLANs for which the port is a member.

Syntax:

```
no spanning-tree port-list bpdu-protection
```

Enables/disables the BPDU protection feature on a port.

Default: Disabled.

Syntax:

```
no spanning-tree port-list bpdu-protection-timeout timeout
```

Configures the duration of time when protected ports receiving unauthorized BPDUs will remain disabled. The default value of 0 (zero) sets an infinite timeout (that is, ports that are disabled by `bpdu-protection` are not, by default, re-enabled automatically).

Default: 0

Range: 0 - 65535 seconds

Syntax:

```
no spanning-tree trap errant-bpdu
```

Enables/disables the sending of errant BPDU traps.

CAUTION: This command should only be used to guard edge ports that are not expected to participate in STP operations. Once BPDU protection is enabled, it will disable the port as soon as any BPDU packet is received on that interface.

Viewing BPDU protection status

Syntax:

```
show spanning-tree bpdu-protection [port-list]
```

Displays a summary listing of ports with BPDU protection enabled. To display detailed per-port status information, enter the specific port number(s). BPDU protected ports are displayed as separate entries of the spanning tree category within the configuration file.

Viewing BPDU protection status for specific ports

```
switch# show spanning-tree bpdu-protection 23-24

 Status and Counters - STP BPDU Protection Information

 BPDU Protection Timeout (sec)  : 0
 BPDU Protected Ports : 23-24


  Port   Type       Protection State              Errant BPDUs
  ------ ---------- ---------- ------------------ ------------
  23     100/1000T  Yes        Bpdu Error         1
  24     100/1000T  Yes                           0
```

Re-enabling a port blocked by BPDU protection

Ports disabled by BPDU Protection remain disabled unless BPDU Protection is removed from the switch or by configuring a nonzero BPDU protection timeout. For example, if you want to re-enable protected ports 60 seconds after receiving a BPDU, you would use this command:

```
switch(config)# spanning-tree bpdu-protection-timeout 60
```

Enabling and disabling BPDU protection

Syntax:

```
no spanning-tree port-list bpdu-protection
```

Enables or disables BPDU protection on specified port(s).

Syntax:

```
no spanning-tree port-list bpdu-protection-timeout timeout
```

Configures the duration in seconds when protected ports receiving unauthorized BPDUs will remain disabled. The default value of 0 (zero) sets an infinite timeout (that is, ports that are disabled by `bpdu-protection` are not, by default, re-enabled automatically).

Range: 0-65535 seconds

Default: 0

Syntax:

```
no spanning-tree trap errant-bpdu
```

Enables or disables the sending of errant BPDU traps.

CAUTION: This command should only be used to guard edge ports that are not expected to participate in STP operations. Once BPDU protection is enabled, it will disable the port as soon as any BPDU packet is received on that interface.

Configuring BPDU protection

To configure BPDU protection on ports 1 to 10 with SNMP traps enabled, enter:

```
switch(config)# spanning-tree 1-10 bpdu protection
switch(config)# spanning-tree trap errant-bpdu
```

The following steps will then be set in progress:

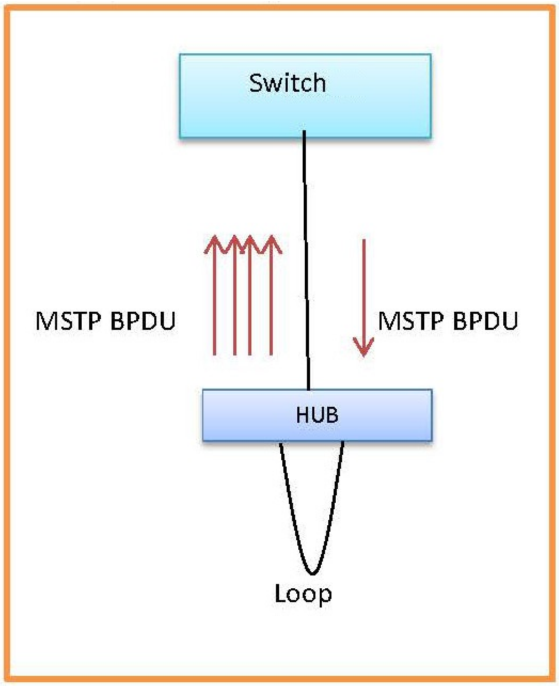
1. When an STP BPDU packet is received on ports 1-10, STP treats it as an unauthorized transmission attempt and shuts down the port that the BPDU came in on.
2. An event message is logged and an SNMP notification trap is generated.
3. The port remains disabled until re-enabled manually by a network administrator using the `interface` *port-list* `enable` command.

NOTE: To re-enable the BPDU-protected ports automatically, configure a timeout period using the `spanning-tree bpdu-protection-timeout` command.

Overview of MSTP BPDU throttling

When an STP enabled switch is hit by an MSTP BPDU storm, the CPU usage rises and the manageability of the switch goes down. In the following figure, the switch is connected to a HUB where there is a loop. The switch generates a single MSTP BPDU, which goes through the loop in the HUB and results in a BPDU storm eventually. Since all STP packets are taken to the CPU for processing, CPU usage goes high and the switch response slows down. The switch can become unmanageable as a result of this BPDU storm. BPDU throttling will take care of BPDU storms automatically through rate-limiting.

Figure 17: *MSTP BPDU path*



BPDU throttling is enabled when the spanning tree in MSTP mode is enabled. When spanning tree is enabled, all modules and members are assigned corresponding throttle values from the configuration. The default throttle value is 256.

An option is also provided to enabling/disabling BPDU throttling. This option is enabled by default if the switch does not support “V1 modules”. The spanning tree is enabled in MSTP mode by default.

Configuring MSTP BPDU throttling

The CLI allows you to configure MSTP BPDU throttling.

Configuring MSTP BPDU throttling

Syntax

```
no spanning-tree bpdu-throttle [Throttle-Value]
```

Configures BPDU throttling on a device. BPDU throttling limits the number of BPDUs that are sent to the switch’s CPU. The result prevents high CPU utilization on the switch when the network undergoes a broadcast storm or loop. The BPDU throttle value is in packets per second (pps). The valid BPDU throttle values are 64, 128, and 256 pps. The default throttle value is 256 pps.

Show MSTP BPDU configuration

The CLI allows you to show MSTP BPDU throttling configurations.

Syntax

```
show spanning-tree bpdu-throttle
```

Displays the configured throttle value.

Example

```
Show spanning-tree bpdu-throttle
BPDU Throttling State   : Enabled
BPDU Throttle value     : 256
```

Show running configuration

Syntax

```
show running configuration
```

Show running configuration will display any one of the following lines based on the configuration.

```
no spanning-tree bpdu throttle
spanning-tree bpdu throttle 128
spanning-tree bpdu throttle 64
```

PVST

PVST stands for Per-VLAN Spanning Tree. It allows for the creation of a spanning tree for each VLAN.

PVST protection and filtering

NOTE:

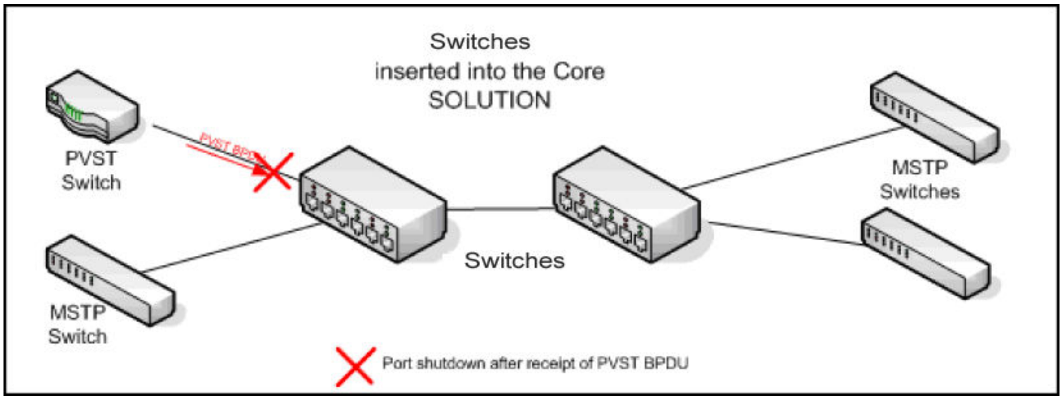
These options are available for switches that support the MSTP protocol only. They are not supported for switches running RSTP.

PVST protection

If a switch in the core of a network receives Per Vlan Spanning Tree (PVST) BPDUs and forwards the unrecognized PVST BPDUs on to MSTP-only switches, those switches then disconnect themselves from the network. This can create instability in the network infrastructure.

When the PVST protection feature is enabled on a port and a PVST BPDU is received on that port, the interface on which the PVST BPDU arrived is shut down, which isolates the sending switch from the rest of the network. An event message is logged and an SNMP notification trap is generated. The errant BPDU counter `SwitchStpPortErrantBpduCounter` is incremented. The PVST protection feature is enabled per-port.

Figure 18: *PVST switch being isolated after sending a PVST BPDU*



NOTE: This is similar to the BPDU Guard feature where BPDU protection is applied to edge ports connected to end user devices that do not run STP. If STP BPDU packets are received on a protected port, the feature will disable that port and alert the network manager via an SNMP trap.

PVST filtering

If you configure a port for PVST filtering instead of PVST protection, the port remains in operation but traps are still generated and the BPDU counter `SwitchStpPortErrantBpduCounter` is incremented.

CAUTION: Enabling the PVST filter feature allows the port to continuously forward packets without spanning tree intervention, which could result in loop formation. If this occurs, disable the port and then reconfigure it with these commands:

```
no spanning-tree port-list bpdu-filter
```

```
no spanning-tree port-list pvst-filter
```

Enabling and disabling PVST protection on ports

Syntax:

```
no spanning-tree port-list pvst-protection
```

Enables or disables PVST protection on the port(s) specified. The command indicates which ports are not expected to receive any PVST BPDUs.

Default: Disabled on all ports.

Enabling PVST protection

To enable the PVST protection feature on ports 4 through 8, enter:

```
switch(config)# spanning-tree 4-8 pvst-protection
```

To disable the PVST protection feature on a port, for example, port 4, enter:

```
switch(config)# no spanning-tree 4 pvst-protection
```

Enabling and disabling PVST filters on ports

Syntax:

```
no spanning-tree port-list pvst-filter
```

Enables or disables PVST filters on the port(s) specified. The command indicates which ports are not expected to receive any PVST BPDUs.

Default: Disabled on all ports

Enabling PVST filtering on a port

```
switch(config)# spanning-tree 8 pvst-filter

Warning: The BPDU filter allows the port to go into a continuousforwarding mode
and spanning-tree will not interfere, even if the port would cause a loop to form
in the network topology.
If you suddenly experience high traffic load, disable the port and reconfigure the
BPDU filter with the CLI command(s):
            "no spanning-tree PORT_LIST bpdu-filter"
            "no spanning-tree PORT_LIST pvst-filter"
```

Re-enabling a port manually

Syntax:

```
no spanning-tree bpdu-protection-timeout timeout
```

Configures the duration of time protected ports remain disabled. The default value of `0` sets an infinite timeout, so ports that are disabled are not re-enabled automatically.

NOTE: This is a GLOBAL command.

Range: 0 - 65535 seconds

Default: 0

You can also set the timeout in the MIB with this MIB object: `hpSwitchStpBpduProtectionTimeout`

It is also possible to use the following automatic re-enable timer command:

```
switch(config)# spanning-tree bpdu-protection-timeout 120
```

Viewing ports configured with PVST protection and filtering

Viewing all ports with PVST protection enabled

```
switch(config)# show spanning-tree pvst-protection

Status and Counters - PVST Port(s) BPDU Protection Information

BPDU Protection Timeout (sec) : 0
PVST Protected Ports : 5-6
```

Viewing all ports with PVST filtering enabled

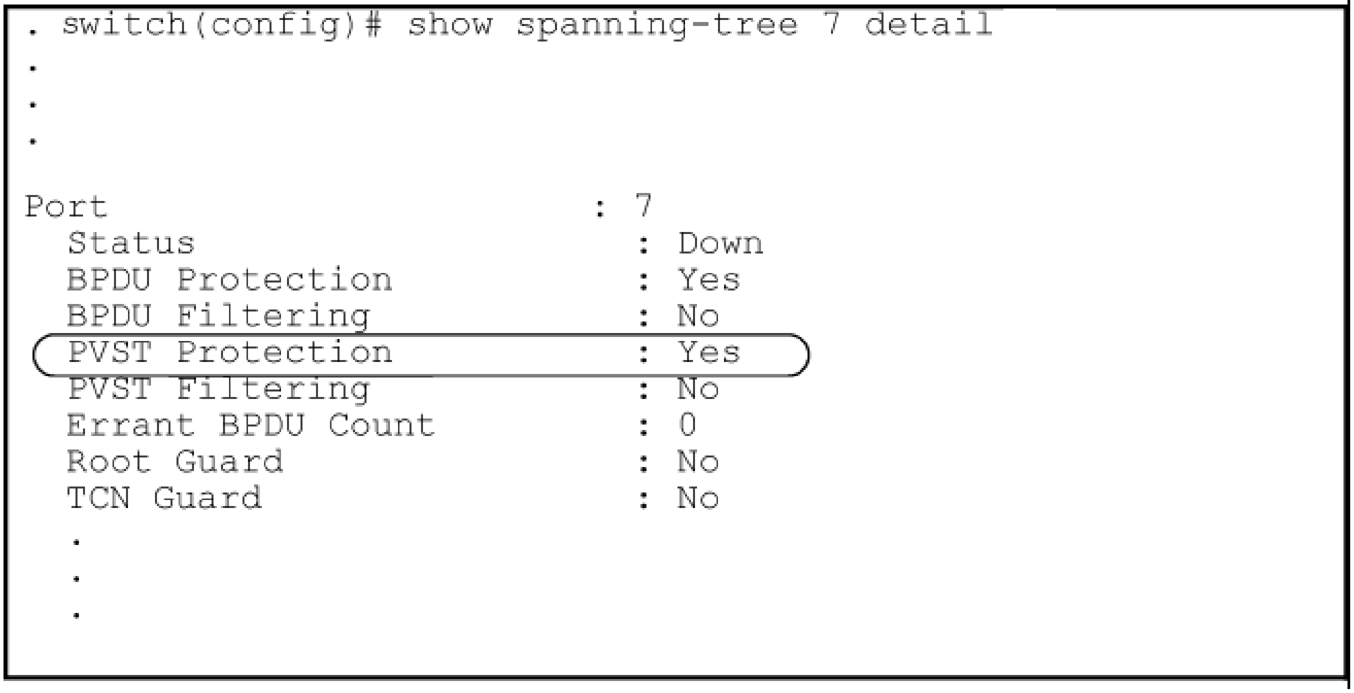
```
switch(config)# show spanning-tree pvst-filter
Status and Counters - PVST Port(s) BPDU Filter Information
PVST Filtered Ports : 8
```

Listing ports to see which have PVST protection or filtering enabled

Syntax:

```
show spanning-tree <port-list> detail
```

Viewing if PVST protection is enabled (Yes)



```
. switch(config)# show spanning-tree 7 detail
.
.
.

Port                      : 7
  Status                    : Down
  BPDU Protection           : Yes
  BPDU Filtering            : No
  PVST Protection           : Yes
  PVST Filtering            : No
  Errant BPDU Count         : 0
  Root Guard                : No
  TCN Guard                 : No
  .
  .
  .
```

In cases where spanning tree cannot be used to prevent loops at the edge of the network, loop protection may provide a suitable alternative. Loop protection operates in two modes:

Untagged

The default mode. This mode can be used to find loops in untagged downlinks.

Tagged VLAN

Finds loops on tagged VLANs. This mode can be used to detect loops in tagged-only uplinks where STP cannot be enabled.

The cases where loop protection might be chosen ahead of spanning tree to detect and prevent loops are as follows:

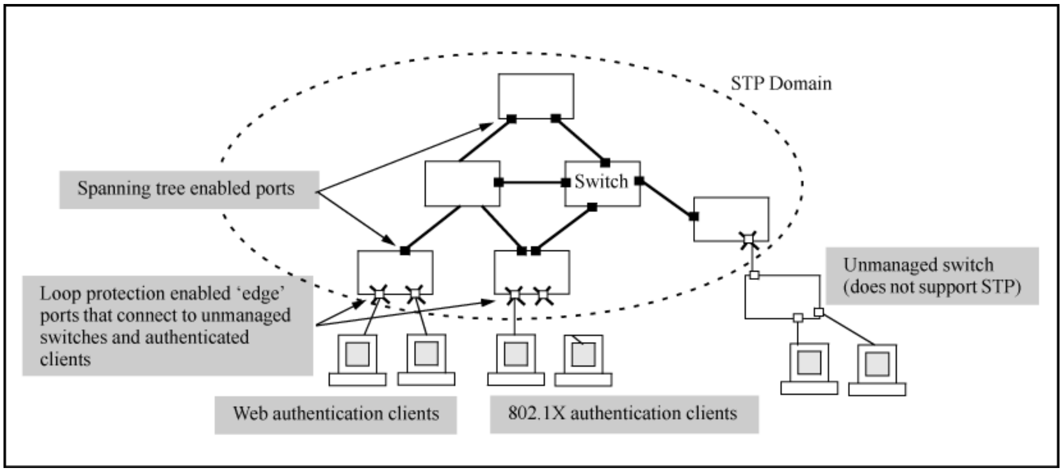
On ports with client authentication

When spanning tree is enabled on a switch that use 802.1X, Web authentication, and MAC authentication, loops may go undetected. For example, spanning tree packets that are looped back to an edge port will not be processed because they have a different broadcast/multicast MAC address from the client-authenticated MAC address. To ensure that client-authenticated edge ports get blocked when loops occur, you should enable loop protection on those ports.

On ports connected to unmanaged devices

Spanning tree cannot detect the formation of loops where there is an unmanaged device on the network that does not process spanning tree packets and simply drops them. Loop protection has no such limitation, and can be used to prevent loops on unmanaged switches.

Loop protection enabled in preference to STP


STP Domain
Spanning tree enabled ports
Switch
Loop protection enabled 'edge' ports that connect to unmanaged switches and authenticated clients
Unmanaged switch (does not support STP)
Web authentication clients
802.1X authentication clients

Configuring loop protection

Loop protection provides protection against loops by transmitting loop protocol packets out of ports on which loop protection has been enabled. When the switch sends out a loop protocol packet and then receives the same packet on a port that has a `receiver-action` of `send-disable` configured, it shuts down the port from which the packet was sent.

Syntax:

```
no loop-protect port-list [[receiver-action [[send-disable] | [no-disable]]] |
[transmit-interval 1-10] | [disable-timer 0-604800] | [trap loop-detected]] [mode]
[[port] | [vlan]] [vlan vid-list]
```

Description

Configures per-port loop protection on the switch.

Parameters

receiver-action send-disable | no-disable

Sets the action to be taken when a loop is detected on the specified ports. The port that receives the loop protection packet determines what action is taken. If `send-disable` is configured, the port that transmitted the packet is disabled. If `no-disable` is configured, the port is not disabled when a loop is detected.

NOTE: The port will not transmit loop protection packets unless it is a member of an untagged VLAN. If a port is only a member of tagged VLANs, the loop protection packets are not transmitted.

Default: `send-disable`

trap loop-detected

Configures loop protection traps for SNMP indicating when a loop has been detected on a port.

disable-timer *0-604800*

Configures how long (in seconds) a port is disabled when a loop has been detected. A value of zero disables the auto re-enable function.

Default: Timer is disabled

transmit-interval *1-10*

Configures the time in seconds between the transmission of loop protection packets.

Default: 5 seconds

{mode port | vlan}

Configures loop protection in port or VLAN mode.

vlan *vid-list*

Configures the VLANs on which loop-protect is enabled. Maximum number of loop-protected VLANs is 32.

Enabling loop protection in port mode

Follow these steps.

Procedure

1. Configure port mode with this command:

   ```
   switch(config)# loop-protect mode port
   ```

2. Enter the `loop-protect` command and specify the ports on which loop protection should be enabled. For example:

```
switch(config)# loop-protect 1-2
```

3. Optionally specify `receiver-action` of `send-disable` to shut down the port in the event of a loop. For example:

```
switch(config)# loop-protect 1-2 receiver-action send-disable
```

Enabling loop protection in VLAN mode

VLANs can be configured for loop protection only when operating in VLAN mode. When `loop-protect` is enabled for a VLAN and a `loop-protect` enabled interface is a member of that VLAN, loop protect packets are sent on that VLAN to detect loops.

To enable loop protection in VLAN mode:

Procedure

1. Configure VLAN mode with the command:

```
switch(config)# loop-protect mode vlan
```

2. Enter the `loop-protect` command and specify the VLANs on which loop protection should be enabled. For example:

```
switch(config)# loop-protect vlan 20,30
```

Changing modes for loop protection

When changing from VLAN mode to port mode, you are prompted with the message shown below. The VLANs will no longer be configured for loop protection.

Changing modes for loop protection

```
switch(config)# loop-protect mode port
Any Loop Protect enabled VLAN will be deleted. Do you want to continue [Y/N]?
N
```

Viewing loop protection status in port mode

Syntax:

```
show loop-protectport-list
```

Displays the loop protection status for ports. If no ports are specified, the information is displayed only for the ports that have loop protection enabled.

Viewing loop protection information for port mode

```
switch(config)# show loop-protect 1-2

 Status and Counters - Loop Protection Information

 Transmit Interval (sec)     : 5
 Port Disable Timer (sec)    : 5
 Loop Detected Trap          : Enabled
 Loop Protect Mode           : Port
 Loop Protect Enabled VLANs :


     Loop    Loop      Detected   Loop      Time Since   Rx           Port
Port Protect Detected  on VLAN    Count     Last Loop    Action       Status
---- ------- --------- --------- --------- ----------- ----------   ---------
1    Yes     Yes         NA       1         5s         send-disable Down
2    Yes     No          NA       0                    send-disable Up
```

Viewing loop protection status in VLAN mode

Syntax:

```
show loop-protect port-list
```

Displays the loop protection status for VLANs. If no ports are specified, the information is displayed only for the ports that have loop protection enabled.

Viewing loop protection information for VLAN mode

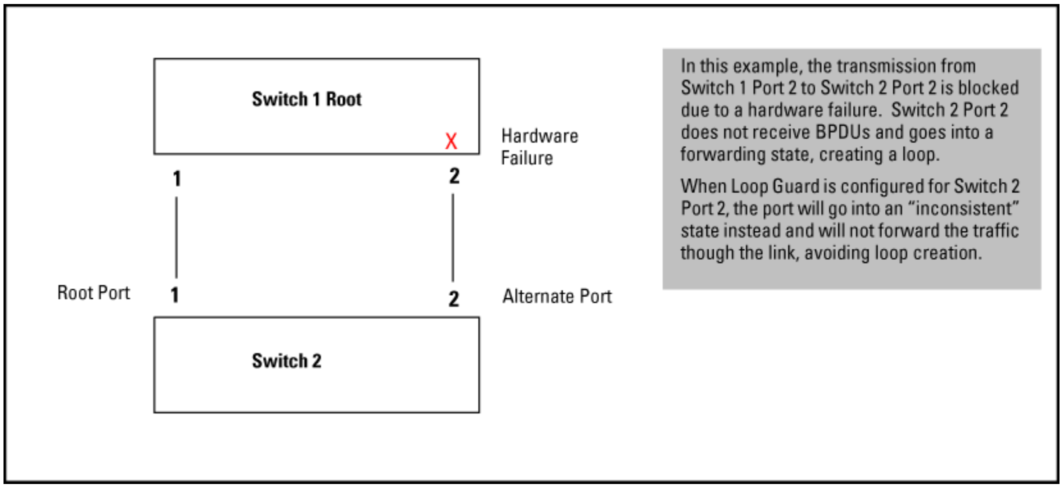
```
switch(config)# show loop-protect 1-2

 Status and Counters - Loop Protection Information

 Transmit Interval (sec)    : 5
 Port Disable Timer (sec)   : 5
 Loop Detected Trap         : Enabled
 Loop Protect Mode          : Vlan
 Loop Protect Enabled VLANs : 20,30


      Loop   Loop      Detected   Loop      Time Since   Rx           Port
Port Protect Detected  on VLAN    Count     Last Loop    Action       Status
---- ------- --------- --------- --------- ----------- ----------   ---------
1    Yes     Yes         20       1         45s        send-disable Down
2    Yes     No                   0                    send-disable Up
```

STP loop guard

Spanning Tree (STP) is used to ensure a loop-free topology over the LAN. Occasionally a hardware or software failure can cause STP to fail, creating STP/forwarding loops that can cause network failures where unidirectional links are used. The non-designated port transitions in a faulty manner because the port is no longer receiving STP BPDUs.

STP Loop Guard causes the non-designated port to go into the STP loop inconsistent state instead of the forwarding state. In the loop-inconsistent state, the port prevents data traffic and BPDU transmission through the link, therefore avoiding the loop creation. When BPDUs again are received on the inconsistent port, it resumes normal STP operation automatically. STP loop guard is best applied on blocking or forwarding ports.

Figure 19: *Loop creation with transmission failure*



In this example, the transmission from Switch 1 Port 2 to Switch 2 Port 2 is blocked due to a hardware failure. Switch 2 Port 2 does not receive BPDUs and goes into a forwarding state, creating a loop.

When Loop Guard is configured for Switch 2 Port 2, the port will go into an "inconsistent" state instead and will not forward the traffic though the link, avoiding loop creation.

Syntax:

`spanning-tree` *`port-list`* `loop-guard`

`no spanning-tree` *`port-list`* `loop-guard`

Enables STP loop guard on a particular port or ports. The `no` form of the command disables STP loop guard.

Default: Disabled.

Enabling spanning tree loop guard on Port 2 and Viewing the port's status

```
switch(config)# spanning-tree 2 loop-guard
```

```
switch(config)# show spanning-tree

 Multiple Spanning Tree (MST) Information

  STP Enabled   : Yes
  Force Version : MSTP-operation
  IST Mapped VLANs : 1-4094
  Switch MAC Address : 0024a8-d13a40
  Switch Priority    : 32768
  Max Age  : 20
  Max Hops : 20
  Forward Delay : 15

  Topology Change Count  : 1
  Time Since Last Change : 20 mins

  CST Root MAC Address : 001083-847000
  CST Root Priority    : 0
  CST Root Path Cost   : 60000
  CST Root Port        : 1

  IST Regional Root MAC Address : 0024a8-d13a40
  IST Regional Root Priority    : 32768
  IST Regional Root Path Cost   : 0
  IST Remaining Hops            : 20

  Root Guard Ports     :
```

```
  Loop Guard Ports     : 2
  TCN Guard Ports      :
  BPDU Protected Ports :
  BPDU Filtered Ports  :
  PVST Protected Ports :
  PVST Filtered Ports  :

                 |           Prio              | Designated    Hello
Port   Type      | Cost      rity State        | Bridge        Time PtP Edge
------ --------- + --------- ---- ------------ + -------------- ---- --- ----
1      100/1000T | 20000     128  Forwarding   | 001871-cdea00  2    Yes No
2      100/1000T | Auto      128  Inconsistent |
3      100/1000T | Auto      128  Disabled     |
4      100/1000T | Auto      128  Disabled     |
5      100/1000T | Auto      128  Disabled     |
6      100/1000T | Auto      128  Disabled     |
7      100/1000T | Auto      128  Disabled     |
8      100/1000T | Auto      128  Disabled     |
```

Viewing summary spanning tree configuration information

```
switch(config)# show spanning-tree config

 Multiple Spanning Tree (MST) Configuration Information

  STP Enabled [No] : Yes
  Force Version [MSTP-operation] : MSTP-operation
  Default Path Costs [802.1t] : 802.1t
  MST Configuration Name : 0024a8d13a40
  MST Configuration Revision : 0       Switch Priority : 32768
  Forward Delay [15] : 15              Hello Time [2] : 2
  Max Age [20] : 20                    Max Hops [20] : 20

               | Path      Prio Admin Auto Admin Hello  Root  Loop  TCN   BPDU
Port Type      | Cost      rity Edge  Edge PtP   Time   Guard Guard Guard Flt
---- --------- + --------- ---- ----- ---- ----- ------ ----- ----- ----- ---
1    100/1000T | Auto      128  No    Yes  True  Global No    No    No    No
2    100/1000T | Auto      128  No    Yes  True  Global No    Yes   No    No
3    100/1000T | Auto      128  No    Yes  True  Global No    No    No    No
4    100/1000T | Auto      128  No    Yes  True  Global No    No    No    No
5    100/1000T | Auto      128  No    Yes  True  Global No    No    No    No
6    100/1000T | Auto      128  No    Yes  True  Global No    No    No    No
.
.
.
```

Viewing detailed spanning tree configuration information

```
switch(config)# show spanning-tree detail

 Status and Counters - CST Port(s) Detailed Information
  Port                     : 1
  Status                   : Up
.
.
.

  Port                     : 2
  Status                   : Up
  BPDU Protection          : No
  BPDU Filtering           : No
  PVST Protection          : No
  PVST Filtering           : No
  Errant BPDU Count        : 0
  Root Guard               : No
  Loop Guard               : Yes
  TCN Guard                : No
  MST Region Boundary      : Yes
  External Path Cost       : 20000
  External Root Path Cost  : 40000
  Administrative Hello Time: Global
  Operational Hello Time   : 2
  AdminEdgePort            : No
  Auto Edge Port           : Yes
  OperEdgePort             : No
  AdminPointToPointMAC     : True
  OperPointToPointMAC      : Yes
  Aged BPDUs Count         : 0
  Loop-back BPDUs Count    : 0
  TC ACK Flag Transmitted  : 0
  TC ACK Flag Received     : 1

 MST        MST        CFG        CFG        TCN        TCN
 BPDUs Tx   BPDUs Rx   BPDUs Tx   BPDUs Rx   BPDUs Tx   BPDUs Rx
 ---------- ---------- ---------- ---------- ---------- ----------
 3          0          24354      1682       0          13
```

Viewing spanning tree configuration information for a single port

```
switch(config)# show spanning-tree 2

 Multiple Spanning Tree (MST) Information

  STP Enabled   : Yes
  Force Version : MSTP-operation
  IST Mapped VLANs : 1-4094
  Switch MAC Address : 0024a8-d13a40
  Switch Priority   : 32768
  Max Age : 20
  Max Hops : 20
  Forward Delay : 15

  Topology Change Count  : 1
  Time Since Last Change : 58 mins

  CST Root MAC Address : 001083-847000
  CST Root Priority   : 0
  CST Root Path Cost  : 60000
  CST Root Port       : 1

  IST Regional Root MAC Address : 0024a8-d13a40
  IST Regional Root Priority   : 32768
  IST Regional Root Path Cost  : 0
  IST Remaining Hops           : 20

  Root Guard Ports     :
  Loop Guard Ports     : 2
  TCN Guard Ports      :
  BPDU Protected Ports :
  BPDU Filtered Ports  :
  PVST Protected Ports :
  PVST Filtered Ports  :

                 |           Prio              | Designated    Hello
Port   Type      | Cost      rity State         | Bridge        Time PtP Edge
```

```
------ --------- + --------- ---- ------------ + ------------- ---- --- ----
2      100/1000T | Auto      128  Inconsistent |
```

Operating notes

- The `receiver-action` option can be configured on a per-port basis and can only be enabled after loop protection has been enabled on the port. All other configuration options (disable-timer, trap loop-detected, and transmit interval) are global.
- The `trap` option refers to an SNMP trap.
- Regardless of how the `receiver-action` and `trap` options are configured, all detected loops will be logged in the switch's event log.
- The `no loop-protect port` command will not remove a receive-action configuration line from the running configuration unless this option is set to `receive-action send-disable`.
- If `loop-protect` is enabled in port mode, it cannot also be enabled in VLAN mode, and vice-versa.

PVLAN introduction

A PVLAN (private VLAN) is an extension of a regular VLAN to help restrict traffic between users on the same VLAN.

The "private" in private VLAN refers to the restriction of the switch ports in the VLAN, called "private ports." Ports in a PVLAN can communicate only with a specified uplink port and with specified ports within the same VLAN.

A PVLAN consists of a regular VLAN that is partitioned into primary and secondary VLANs. The partitioned regular VLAN becomes the primary VLAN. Secondary VLANs are associated with the primary VLAN, have unique VLAN IDs, and have different types—isolated and community—that determine how and to where packets can be forwarded.

Private VLANs and regular VLANs can coexist on the same switch.

Typical uses for PVLANs include the following:

- Shared environments in which ports can be isolated from each other at the data link layer (for security, performance, or other reasons), while belonging to the same IP subnet. For example:
  - Hotels in which each room has a port for Internet access.
  - Networks configured to allow onsite access to vendors or contractors while keeping them isolated from the rest of the customer network.
  - ISP colocation in a data center.
- IP address conservation and efficient IP address assignment. Hosts in secondary VLANs remain in a separate broadcast domain, but belong to the same IP subnet. Hosts in secondary VLANs are assigned IP addresses based on the IP subnets associated with the primary VLAN.
- Backup networks, in which the backup server is in a PVLAN, and all the hosts using the backup server are configured on isolated secondary VLANs.

PVLAN structure

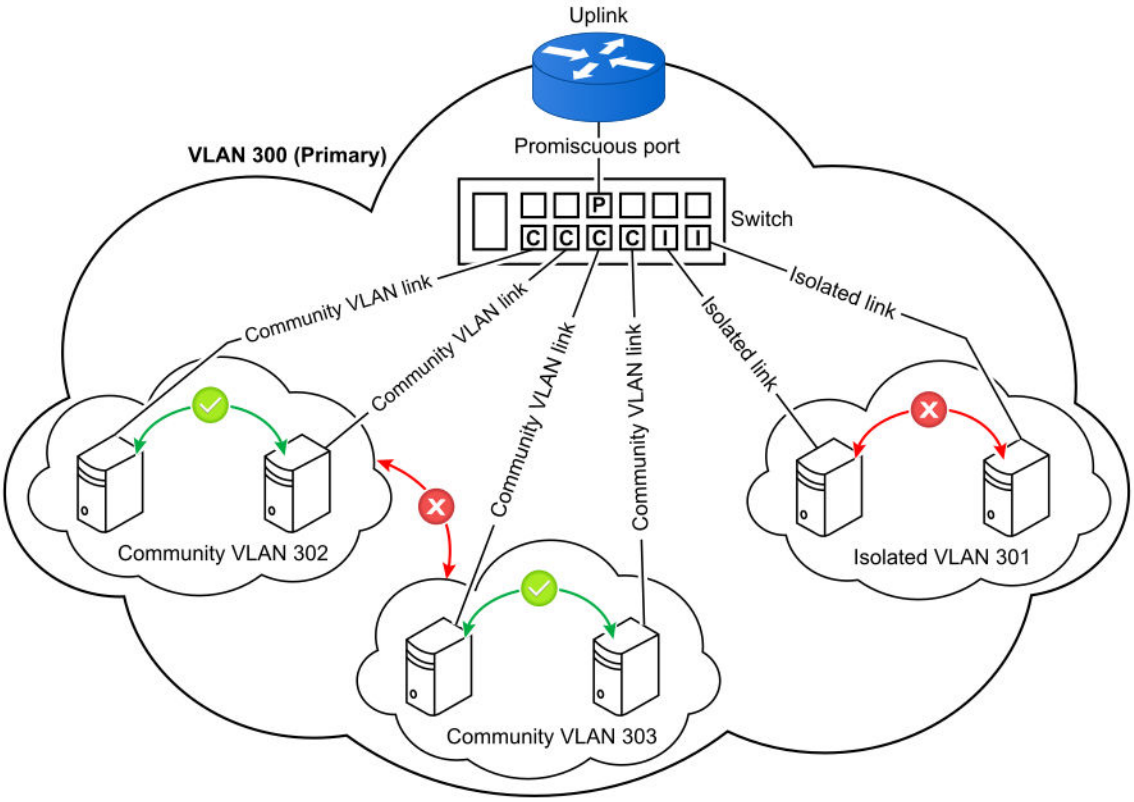
Primary VLAN

The primary VLAN is the standard VLAN that is partitioned to create the PVLAN. The primary VLAN delivers traffic downstream from the router to all mapped hosts, and contains the uplink from the connected devices to the router.

The secondary VLANs must use the primary VLAN to communicate with other secondary VLANs or to VLANs outside the PVLAN. This communication occurs through one or more promiscuous ports associated with the primary VLAN.

By default, all ports in the primary VLAN act as promiscuous ports.

Figure 20: *Primary VLAN in PVLAN*

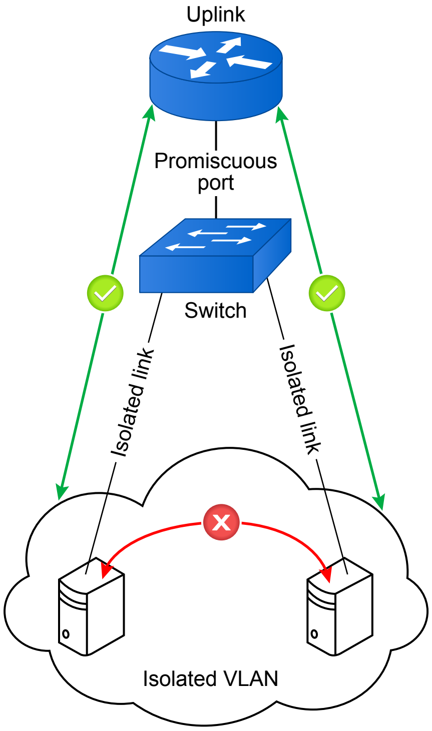


Secondary VLANs

Isolated VLAN

Ports in an isolated VLAN can communicate using Layer 2 with the promiscuous ports of the primary VLAN only. The ports that are associated with an isolated VLAN do not have Layer 2 connectivity between each other, but hosts can communicate with each other using a Layer 3 device.

Figure 21: *Isolated VLAN communications*

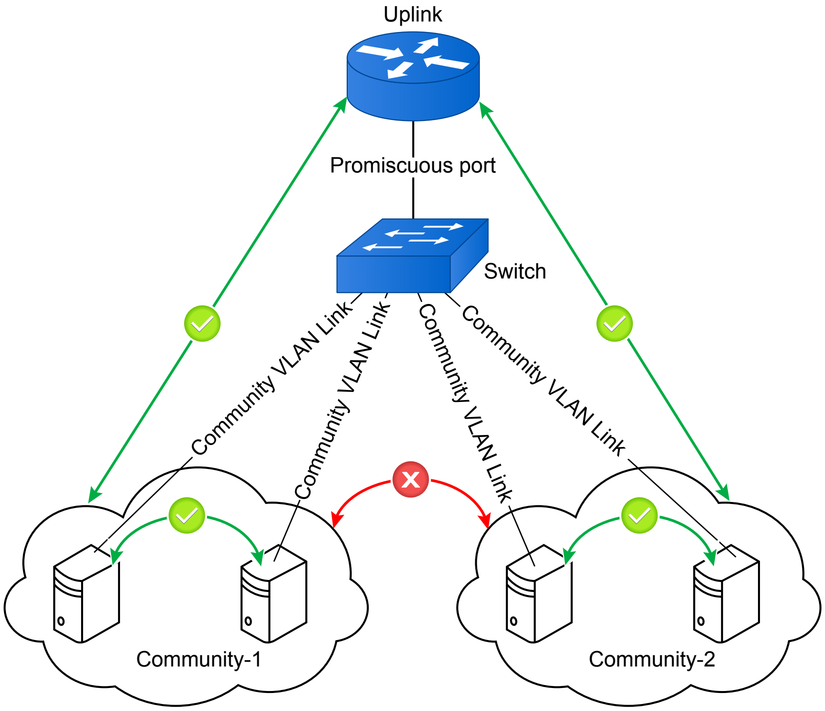


Community VLAN

The ports associated with a community VLAN can communicate using Layer 2 with each other and with the primary VLAN, but not directly with ports in other community VLANs.

For one community VLAN port to communicate with a different community VLAN port, the port traffic must go through the Layer 3 device.

Figure 22: *Community VLAN communications*



PVLAN port types

PVLANs (private VLANs) have the following types of ports:

- Promiscuous ports
- Community ports
- Isolated ports
- Interswitch link (ISL) ports (PVLAN member ports)

A port in a PVLAN is exactly one type.

Community port

A community port is a host access port in a community VLAN. Community ports can communicate on the Layer 2 level with the following ports:

- Other community ports in the same community VLAN
- Associated promiscuous ports in the primary VLAN

Community ports cannot communicate on the Layer 2 level with ports in any other community or isolated VLAN in the private VLAN.

Isolated ports

An isolated port is a host access port in an isolated VLAN. Isolated ports can communicate on the Layer 2 level with associated promiscuous ports in the primary VLAN only.

Isolated ports cannot communicate on the Layer 2 level with any other ports in the private VLAN (PVLAN) domain, including other ports in the same isolated VLAN.

However, if the switch that has the isolated port does not have a promiscuous port, traffic from the isolated VLAN ports can traverse through interswitch link (ISL) ports (PVLAN member ports) to a switch that has a promiscuous port.

Promiscuous ports

A promiscuous port is a switch port that is connected to a router, firewall, Level 3 switch, or other Common Gateway device. Promiscuous ports can communicate with all types of ports within a PVLAN (private VLAN) domain, including community ports and isolated ports.

- Promiscuous ports must be members of the primary VLAN.
- By default, every port in the primary VLAN is configured as a promiscuous port.

Interswitch link (ISL) ports (PVLAN member ports)

Interswitch link (ISL) ports (PVLAN member ports) connect multiple switches in a PVLAN domain.

Interswitch link (ISL) is a generic term to describe a connection between ports on different switches.

In ArubaOS-Switch software:

- Ports used for interswitch links in PVLANs are called "PVLAN member ports" because they automatically become members of all VLANs within the PVLAN (that is, the primary VLAN, the community VLANs and the isolated VLANs). They carry traffic from the primary VLAN and all secondary VLANs between switches.

  The port type displayed for a PVLAN member port in `show` commands is: `Member`
- PVLAN member ports are primary VLAN ports that are configured to remove the promiscuous port interface.

Traffic forwarding through interswitch links

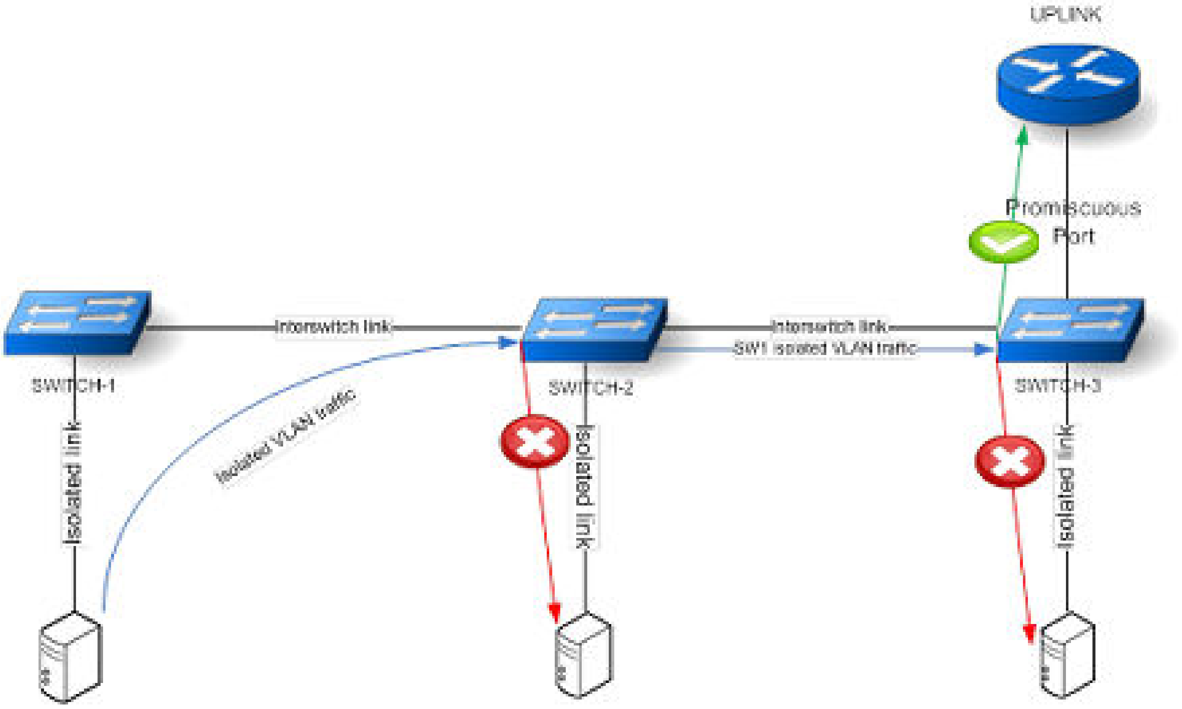
Consider a PVLAN network topology where secondary VLANs have been configured on a set of access switches but only one of these switches has uplink connectivity. The secondary VLAN traffic on switches that do not have an uplink configured must be forwarded through interswitch links to the device that has an uplink.

As secondary VLAN traffic traverses the interswitch links, all switches in the path use the secondary VLAN information carried in the frame to ensure that PVLAN traffic forwarding rules are preserved.

For example, for an isolated VLAN, each switch must ensure the following:

- Traffic from the isolated VLAN is forwarded to a switch that has a promiscuous port.
- Traffic from an isolated VLAN is not forwarded to any local community ports or to other isolated VLANs.

Figure 23: *Interswitch link traffic forwarding*



Example PVLAN Configuration

Consider the following PVLAN network scenario:

- VLAN 300 has been partitioned into a PVLAN (private VLAN), across two switches, consisting of the following VLANs and ports:
  - The primary VLAN (VLAN 300) with the following ports:
    - One promiscuous port connected to a router.
    - One port trunk configured as an interswitch link.
  - One isolated VLAN (VLAN 301) with two ports.
  - Two community VLANs (VLAN 302 and VLAN 303), each with two ports.
- The two switches also carry normal VLAN traffic for VLAN 5 and VLAN 6.

At the Layer 2 level in the PVLAN:

- The promiscous port can communicate with all the other ports, including ports in the isolated VLAN and through interswitch link (ISL) ports (PVLAN member ports).
- The community VLAN ports can communicate with the promiscuous port or other ports in the same community VLAN. The community VLAN ports cannot communicate with ports in any isolated VLAN or in other community VLANs. Traffic from community VLAN ports can traverse through interswitch link (ISL) ports (PVLAN member ports) to other ports belonging to the same community VLAN.
- The isolated VLAN ports can communicate with the promiscuous port only. However, traffic from the isolated VLAN ports can traverse through interswitch link (ISL) ports (PVLAN member ports) to a switch that has a promiscuous port.

Example commands

The following example lists the commands used to configure the PVLAN and interswitch links on one of the switches in the network scenario.

In the example, the number sign (#) delimits comments to you--the reader. Do not enter comments as commands.

```
# Configure the normal VLANs #
vlan 5
vlan 6
vlan 300

# Configure the normal VLAN host access ports #
vlan 5 untagged 1/11-1/12
vlan 6 untagged 1/13-1/14

# Configure VLAN 300 as a private VLAN #
vlan 300 private-vlan primary
vlan 300 private-vlan isolated 301
vlan 300 private-vlan community 302-303

# Configure port A2 as a promiscuous port #
# private-vlan promiscuous is the default configuration #
# for all primary VLAN ports, so you do not have to specify it explicitly  #
vlan 300 untagged A2

# Configure the isolated host access ports #
vlan 301 untagged 1/8-1/9

# Configure the community host access ports #
vlan 302 untagged 1/17-1/18
vlan 303 untagged 1/19-1/20

# Configure interswitch links A3 and A4 as a trunk port group to carry both PVLAN #
# and normal VLAN traffic #

trunk 1/A3,1/A4 trk1 lacp
vlan 300 tagged trk1
vlan 5 tagged trk1
vlan 6 tagged trk1
no interface trk1 private-vlan promiscous

# private-vlan promiscuous is the default configuration for primary VLAN ports #
# You configure ports as interswitch links by #
# removing the promiscuous port configuration #
```

Configuring PVLANs

Procedure

1. **Create the primary VLAN.**
2. **(Optional) Add an isolated VLAN.**
3. If the PVLAN contains multiple switches, **configure interswitch links.**

Creating a primary VLAN

Prerequisites

You must be in the global configuration context: `switch(config)#`

Procedure

Configure a VLAN as the primary VLAN in the private VLAN.

Use the following command:

```
vlan <VLAN-ID> private-vlan primary
```

<VLAN-ID>

Specifies the VLAN ID of an existing VLAN or a VLAN you are creating.

For example, the following command configures VLAN 300 as the primary VLAN:

```
switch(config)# vlan 300 private-vlan primary
```

Adding the isolated VLAN

Prerequisites

- The primary VLAN must exist in the configuration.
- The isolated VLAN must not exist in the configuration.
- You must be in the global configuration context: `switch(config)#`

Procedure

Add the isolated VLAN by entering the following command:

```
vlan <VLAN-ID> private-vlan isolated <Isolated-VLAN-ID>
```

<VLAN-ID>

Specifies the ID of the primary VLAN.

<Isolated-VLAN-ID>

Specifies the ID of the isolated VLAN.

For example, to add VLAN 301 as the isolated VLAN associated with VLAN 300, enter the following command:

```
switch(config)# vlan 300 private-vlan isolated 301
```

Adding community VLANs

Prerequisites

- The primary VLAN must exist in the configuration.
- The community VLANs must not exist in the configuration.
- You must be in the global configuration context: `switch(config)#`

Procedure

Add community VLANs by entering the following command:

`vlan <VLAN-ID> private-vlan community <VLAN-ID-LIST>`

<VLAN-ID>

Specifies the ID of the primary VLAN.

<VLAN-ID-LIST>

Specifies a list of the VLAN IDs of the community VLANs you are adding. Separate individual VLAN IDs with commas. Specify ranges of VLAN IDs with hyphens.

For example, enter the following command to add VLAN IDs 302 and 303 as community VLANs associated with primary VLAN 300:

```
switch(config)# vlan 300 private-vlan community 302,303
```

Adding ports to PVLANs

Prerequisites

- The PVLAN must exist in the configuration.
- You must be in the global configuration context: `switch(config)#`

Procedure

- Adding a port to a primary or secondary VLAN in a PVLAN is the same as adding a port to a regular VLAN.

  For example, to add port 1/17 as an untagged port to community VLAN 302, enter the following command:

  ```
  switch(config)# vlan 302 untagged 1/17
  ```
- To add promiscuous ports to the primary VLAN, add the ports to the primary VLAN.

  Because all primary VLAN ports act as promiscuous ports by default, you do not need to configure primary ports as promiscuous ports explicitly.
- To add interswitch link (ISL) ports (PVLAN member ports), you must change the configuration of ports in the primary VLAN. See **Configuring interswitch link (ISL) ports (PVLAN member ports)** on page 165.

Configuring interswitch link (ISL) ports (PVLAN member ports)

To configure an interswitch link, you must change the configuration of an existing promiscuous port or trunk group in the primary VLAN on each of the two switches in the link.

Prerequisites

- The primary VLAN must exist in the configuration.
- You must be in the global configuration context: `switch(config)#`

Procedure

- To configure a single port as an interswitch link (PVLAN member port):

  1. Choose the port to be used for the interswitch link and remove the existing (default) promiscuous port configuration setting for that interface.

     For example:

     ```
     switch-A(config)# no interface 1/A3 private-vlan promiscuous
     ```

- To configure a port trunk group as an interswitch link (PVLAN member):

  1. Choose the trunk group to be used for the interswitch link and remove the existing (default) promiscuous port configuration setting for that interface.

     For example:

     ```
     switch-A(config)# no interface trkl private-vlan promiscuous
     ```

Adding an interswitch link between two switches for VLAN 300

The interface you choose on the second switch does not have to match the interface you choose on the first switch.

Switch A commands:

```
switch-A(config)# vlan 300 tagged 1/A3
switch-A(config)# no interface 1/A3 private-vlan promiscous
```

Switch B commands:

```
switch-B(config)# vlan 300 tagged 1/A3
switch-B(config)# no interface 1/A3 private-vlan promiscous
```

You can also configure port trunk groups as an interswitch link. You can use either a static trunk or a static LACP trunk.

This example shows configuring a trunk group, adding the trunk group to the PVLAN as a tagged interface, and then configuring that trunk group as an interswitch link.

IMPORTANT: Configure port trunking **before** you connect the trunked links between switches. Otherwise, a broadcast storm could occur. (If you want to connect the ports before configuring them for trunking, you can temporarily disable the ports until the trunk is configured.) For more information about port trunking, see the *ArubaOS-Switch Management and Configuration Guide*.

```
switch-A(config)# switch-A(config)# trunk 1/A3,1/A4 trk1 lacp
switch-A(config)# vlan 300 tagged trk1
switch-A(config)# no interface trk1 private-vlan promiscous
```

Configuring promiscuous ports

By default, all ports in the primary VLAN of a private VLAN are configured as promiscuous ports. You do not need to configure an interface as promiscuous unless it has previously been configured as an interswitch link (ISL) port (PVLAN member port).

Prerequisites

You must be in the global configuration context: `switch(config)#`

Procedure

To configure an interface as a promiscuous port, enter the following command:

```
interface <port-list> private-vlan promiscuous
```

<port-list>

Specifies the list of ports, separated by commas for individual ports or hyphens for ranges of ports.

For example, the following command to configure interfaces 1/2 and 1/3 as promiscuous ports in the private VLAN to which they belong.

```
switch-A(config)# interface 1/2,1/3 private-vlan promiscuous
```

Rules for configuring PVLANs

- The default VLAN (VLAN 1) cannot be configured as a PVLAN.
- An existing VLAN cannot be configured as a secondary VLAN in a PVLAN.
- A secondary VLAN cannot be reconfigured as the primary VLAN and a primary VLAN cannot be reconfigured as a secondary VLAN.

  To change the VLAN types, you must reconfigure the primary VLAN as a normal VLAN, and then reconfigure the PVLAN.
- You cannot reconfigure or delete a primary VLAN until its associated secondary VLANs have been deleted.
- A PVLAN cannot be configured as a Management VLAN.
- For information about rules for PVLANs when used with other networking features, see **PVLAN Interaction with other features** on page 168.

Rules for assigning ports in PVLANs

- An access port—which is directly connected to a host—can belong to one secondary VLAN in the PVLAN only.
- Promiscuous ports can be members of the primary VLAN only.
- Ports can be one type only: promiscuous, community, isolated, or interswitch link (PVLAN member port)

Configuration limits for PVLANs

The following are configuration limits for PVLAN instances on a switch:

- Maximum number of primary VLANs: 16
- Maximum number of community VLANs in a PVLAN: 8
- Maximum number of isolated VLANs in a PVLAN: 1
- Maximum number of interswitch links: No configuration limit.

PVLAN Interaction with other features

Features you configure on the primary VLAN only

The following features are not permitted to be configured on secondary VLANs. The configuration of this feature on the primary VLAN is automatically applied to the secondary VLANs.

- Local-proxy ARP
- Proxy ARP
- DHCP snooping
- DHCPv6 Snooping
- UDP forwarder
- RA Guard
- ND Snooping
- DARPP
- DIPLD v4/v6
- IGMP Snooping
- MLD Snooping
- IPv4/v6 Address
- ND
- IP Directed Broadcast, DHCP Relay, UDP-broadcast-fwd
- Disable-Layer3
- Jumbo-MTU
- DHCP Server
- IP-Recv-MAC
- VRRP
- IRDP
- BGP
- OSPF
- OSPF3
- RIP
- MSTP vlan-instance map
- Smartlink

- Neighbor Discovery configurations
- Static routes
- Source routes
- Source interface
- Source VLAN for traceroute
- DT peer interface
- Ping (source VLAN configuration)
- Voice VLAN

Features to configure the same on all VLANs in a PVLAN

For the following features, Hewlett Packard Enterprise recommends that you apply the same configuration to the primary and all secondary VLANs. If you do not apply the same configuration to all the VLANs in the PVLAN domain, the feature might function in partial or suboptimal ways.

For example, locking down a MAC address (using the Static-MAC feature) on a port and a specific VLAN only restricts the MAC address on that VLAN. The client device with that MAC address can access other VLANs on the same port or through other ports. If you create a PVLAN by partitioning a VLAN that uses the Static-MAC feature, Hewlett Packard Enterprise recommends that you apply the same Static-MAC configurations to the secondary VLANs.

- DST-IP, L4-7 ACL
- DST-IP, L4-7 Policy Based Mirror
- QoS
- Static-MAC

Features not permitted on PVLANs

The following features are not permitted to be configured on PVLANs:

- Primary-vlan (PVLANs cannot be configured as the primary regular VLAN.)
- Management-vlan
- Protocol-vlan
- Level 3 mulitcast protocols and routing
- PVST
- RPVST
- LLDP radio ports
- Out Mirroring
- Distributed trunking, including Dynamic LACP trunk ports
- GVRP
- MVRP
- PIM

- QinQ
- VLAN source filtering
- BYOD onboarding VLAN
- OpenFlow
- Isolate-list
- Forbidden ports
- Untrusted port configuration on a primary VLAN port

Security interactions with PVLANs

DHCP and PVLANs

- Ports in the primary VLAN are trusted ports by default.
- The following features cannot be configured on a secondary VLAN:
  - DHCP snooping
  - DHCPv6 snooping
  - DHCP relay
  - DHCP server
  - DHCP client
  - DHCP server IPv4/IPv6 address configuration
  - IPv4/IPv6 gateway address configuration
  - Static IP binding in the DHCP binding database

  When you configure any of these features on a primary VLAN, the configuration automatically applies to all secondary VLANs associated with that primary VLAN.
- You can use the `show dhcp-snooping binding private-vlan` command to show DHCP snooping information for primary and secondary VLANs in the PVLAN.

Dynamic IP lockdown and PVLANs

When a PVLAN is configured, the output of the `show ip source-lockdown` command is affected as follows:

`show ip source-lockdown status`

Shows the port state as active even if the port is associated with a secondary VLAN and DHCP snooping is enabled on the corresponding primary VLAN.

`show ip source-lockdown bindings`

The VLAN field shows the primary VLAN only.

ARP and PVLANs

- Proxy ARP cannot be enabled on interfaces that are in a PVLAN.
- Local proxy ARP cannot be enabled on interfaces that are in a PVLAN.
- Dynamic ARP protect can be configured on primary VLANs only:
    - In `ip source binding` commands, you must specify the VLAN ID of the primary VLAN.
    - The `show arp-protect` command shows primary VLANs only.

Multicast interactions with PVLANs

The following Layer 2 multicast protocols are supported on PVLANs:

IGMPv2

You can enable IGMP on the primary VLAN in a PVLAN only. Use the MAC lookup mode of IGMP (`igmp lookup-mode mac`) only. IP lookup mode is not supported.

MLDv1 and MLDv2

You can enable MLD on the primary VLAN in a PVLAN only.

Filter consumption increases when IGMP or MLD is enabled on a PVLAN because a filter is allocated for the primary VLAN and for every secondary VLAN when a host connected to a PVLAN joins a multicast group.

PIM is not supported on PVLANs.

Showing PVLAN configurations

Procedure

- To show all configured PVLANs, enter the `show vlans private-vlan` command without specifying parameters.

```
switch(config)# show vlans private-vlan

 Configuration and Association - private VLANs

  primary  secondary  VLAN Type
  -------  ---------  ---------
  300      301        isolated
           302        community
           303        community
```

- To show information about a specific PVLAN, enter the `show vlans [<VLAN-ID>] private-vlan` command and specify the VLAN ID of a VLAN in the PVLAN.

  Example output if the VLAN is a primary VLAN:

```
switch(config)# show vlans 300 private-vlan

 Private VLAN Configuration Information: VLAN 300
  VLAN Type : primary

  Port Type          Ports
  ------------       ----
  Promiscuous        1/A2
  Member             Trk1
 Associated Secondary VLANs:

  VLAN ID   VLAN type    Access Ports
  -------   ---------    ------------
  301       isolated     1/8-1/9,Trk1
  302       community    1/17-1/18,Trk1
  303       community    1/19-1/20,Trk1
```

Example output if the VLAN is a secondary VLAN:

```
(config)# show vlans 301 private-vlan

 Private VLAN Configuration Information: VLAN 301
  VLAN Type      : isolated
  Access ports   : 1/8-1/9,Trk1

 Associated Primary VLAN: 300

  Port Type           Port
  -----------         ----
  Promiscuous          1/A2
  Member               Trk1
```

- To show the promiscuous ports of all PVLANs, enter the `show private-vlan promiscuous-ports` command.

```
switch(config)# show private-vlan promiscuous-ports

  primary VLAN    Port
  -------------  ------
  300             1/A2
```

Examples: `show vlans` command with PVLANs

When PVLANs (private VLANs) are configured, the output of the `show vlans` command includes information about the private VLANs.

Example output when a PVLAN is configured

```
(config)# show vlans 300

 VLAN ID : 300
 Name : VLAN300
 Status : Port-based
 Voice : No
 Jumbo : No
 Private VLAN: primary
 Associated Primary VID :  None
 Associated Secondary VIDs : 301-303

 Port Information   Mode       Unknown VLAN    Status
 ------------------ -------    --------------  --------
 1/A2               Untagged   Learn           Up
 Trk1               Tagged     Learn           Up
```

Example output for a secondary isolated VLAN

```
(config)# show vlans 301

 VLAN ID : 301
 Name : VLAN301
 Status : Port-based
 Voice : No
 Jumbo : No
 Private VLAN: isolated
 Associated Primary VID :  300
 Associated Secondary VIDs : None

 Port Information   Mode        Unknown VLAN    Status
 ------------------ -------     --------------  ---------
 1/8                Untagged    Learn           Up
 1/9                Untagged    Learn           Up
 Trk1               No          Learn           Up
```

Example: `show running-config` command for private VLANs

When PVLANs (private VLANs) are configured, the output of the `show running-config` command includes information similar to the following example:

```
switch# show running-config

Running configuration:

...

interface 1/A2

...

interface Trk1
...
   no private-vlan promiscuous
...

vlan 300
   name "VLAN300"
   private-vlan primary
   private-vlan isolated 301
   private-vlan community 302
   untagged 1/A2
   tagged Trk1
   no ip address
   exit

vlan 301
   name "VLAN301"
   untagged 1/8-1/9
   no ip address
   exit

vlan 302
   name "VLAN302"
   untagged 1/17-1/18
   no ip address
   exit

vlan 303
   name "VLAN303"
   untagged 1/19-1/20
   no ip address
   exit
```

Removing PVLANs from the configuration of the switch

Prerequisites

- The PVLAN (private VLAN) must exist in the configuration.
- You must be in the global configuration context: `switch(config)#`

Procedure

1. Remove all secondary community VLANs (if configured).

   For example:

   ```
   switch(config)# no vlan 300 private-vlan community 302,303
   ```

   All member ports of the deleted VLAN are moved to VLAN 1 automatically.
2. Remove the secondary isolated VLAN (if configured).

   For example:

   ```
   switch(config)# no vlan 300 private-vlan isolated 301
   ```

   All member ports of the deleted VLAN are moved to VLAN 1 automatically.
3. Remove the primary private VLAN configuration from the primary VLAN.

   For example:

   ```
   switch(config)# no vlan 300 private-vlan primary
   ```

   The primary VLAN is now configured as a regular VLAN and the private VLAN no longer exists.

PVLAN commands

interface private-vlan promiscuous

Syntax

```
interface <port-list> private-vlan promiscuous

no interface <port-list> private-vlan promiscuous
```

Description

Configures ports as promiscuous ports in private VLANs.

The `no` form of this command removes the promiscuous port configuration, which enables the ports to be configured as interswitch links.

Command context

Required context: `config`

Parameters

<port-list>

Specifies the ports. Separate individual ports with commas. Specify ranges of ports with hyphens.

Example

The following example configures interface 1/A2 as a promiscuous port:

```
switch(config)# interface 1/A2 private-vlan promiscuous
```

show private-vlan promiscuous-ports

Syntax

```
show private-vlan promiscuous-ports
```

Description

Shows a list of all the promiscuous ports of all the PVLANs (private VLANS).

Command context

Required context: `config`

Example

```
switch(config)# show private-vlan promiscuous-ports

  primary VLAN    Port
  -------------  ------
  300             1/A2
```

show vlans private-vlan

Syntax

```
show vlans [<VLAN-ID>] private-vlan
```

Description

Shows information about the configured PVLANs.

Command context

Required context: `config`

Parameters

<VLAN-ID>

Specifies the ID of a primary VLAN. When this parameter is specified, the `show` command displays information about the specified PVLAN only.

Example of showing all PVLANs (private VLANS)

In the following example, the values in the VLAN type column describe the secondary VLANs.

```
switch(config)# show vlans private-vlan

 Configuration and Association - private VLANs:

  primary  secondary  VLAN Type
  -------  ---------  ---------
  300      301        isolated
```

```
           302        community
           303        community
```

Example of showing information about a primary VLAN

```
switch(config)# show vlans 300 private-vlan

 Private VLAN Configuration Information: VLAN 300

  VLAN Type : primary

  Port Type         Ports
  ------------      ----
  Promiscuous       1/A2
  Member            Trk1

 Associated Secondary VLANs:

  VLAN ID   VLAN type    Access Ports
  -------   ---------    ------------
  301       isolated     1/8-1/9,Trk1
  302       community    1/17-1/18,Trk1
  303       community    1/19-1/20,Trk1
```

Example of showing information about a secondary VLAN

```
(config)# show vlans 301 private-vlan

 Private VLAN Configuration Information: VLAN 301
  VLAN Type     : isolated
  Access ports  : 1/8-1/9,Trk1

 Associated Primary VLAN: 300

  Port Type         Ports
  -----------       ----------
  Promiscuous        1/A2
  Member             Trk1
```

vlan private-vlan

Syntax

```
vlan <PRIMARY-VLAN-ID> private-vlan
  [primary | isolated <VLAN-ID> | community <VLAN-ID-LIST>]

no vlan <PRIMARY-VLAN-ID> private-vlan
  [primary | isolated <VLAN-ID> | community <VLAN-ID-LIST>]
```

Description

Creates new private VLAN or changes an existing VLAN into a private VLAN.

The `no` form of the command removes secondary VLANs from the configuration or removes the PVLAN from the configuration by reconfiguring the primary VLAN as a regular VLAN.

Command context

Required context: `config`

Parameters

<PRIMARY-VLAN-ID>

Specifies the VLAN ID of the primary VLAN in the PVLAN.

primary

Specifies that *<PRIMARY-VLAN-ID>* is the primary VLAN in a PVLAN.

isolated *<VLAN-ID>*

Specifies an isolated VLAN in the PVLAN, where *<VLAN-ID>* is the VLAN ID of the isolated VLAN.

community *<VLAN-ID-LIST>*

Specifies one or more community VLANs in the PVLAN, where *<VLAN-ID-LIST>* contains the VLAN IDs of the community VLANs. Separate individual VLAN IDs with commas. Specify ranges of VLAN IDs with hyphens.

Usage

The `no` form of this command removes the PVLAN configuration as follows:

- When used with the isolated or community parameter, removes the secondary VLAN from the configuration and assigns any ports to the default VLAN (VLAN 1).
- When used without the isolated or community parameters, removes the private VLAN configuration from the configuration and converts the primary VLAN into a regular VLAN.

Example

The following example configures a PVLAN with primary VLAN 300, isolated VLAN 301, community VLAN 302, and community VLAN 303:

```
switch(config)# vlan 300 private-vlan primary
switch(config)# vlan 300 private-vlan isolated 301
switch(config)# vlan 300 private-vlan community 302-303
```

Introduction to Quality of Service (QoS)

A Quality of Service (QoS) **network policy** refers to the network-wide controls available to:

- Ensure uniform and efficient traffic-handling throughout your network, while keeping the most important traffic moving at an acceptable speed, regardless of current bandwidth usage.
- Exercise control over the priority settings of inbound traffic arriving in and travelling through your network.

Adding bandwidth can be a good idea, but is not always feasible and does not completely eliminate the potential for network congestion. There will always be points in the network where multiple traffic streams merge or where network links change speed and capacity. The impact and number of these congestion points will increase over time as more applications and devices are added to the network.

When network congestion occurs, it is important to move traffic on the basis of relative importance. However, without QoS prioritization, less important traffic consumes network bandwidth and slows down or halts the delivery of more important traffic. Without QoS, most traffic received by the switch is forwarded with the same priority it had upon entering the switch. In many cases, such traffic is normal priority and competes for bandwidth with all other normal-priority traffic, regardless of its relative importance to your organization's mission.

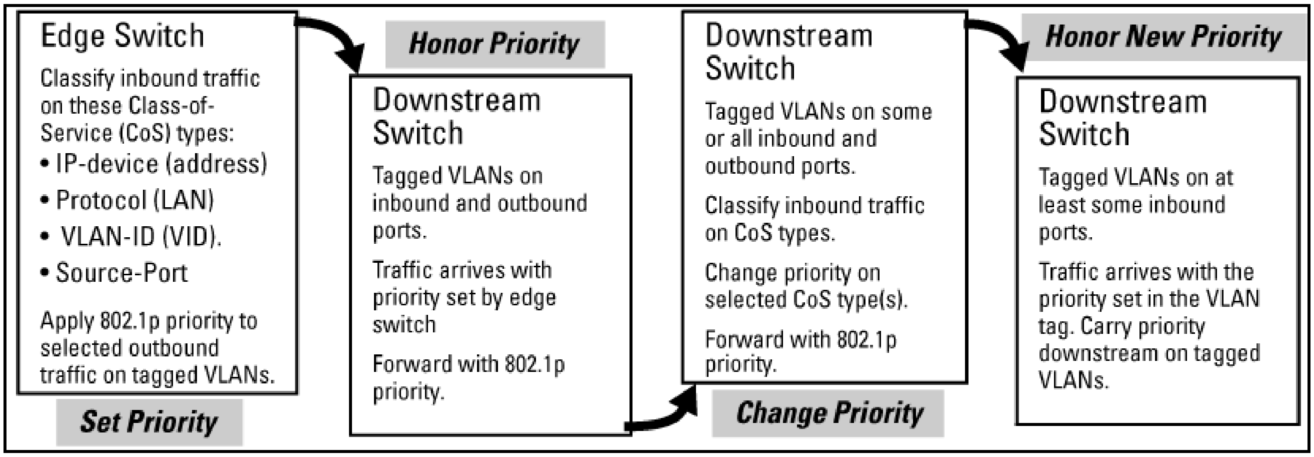
Using QoS to classify and prioritize network traffic

QoS is used to classify and prioritize traffic throughout a network. QoS enables you to establish an end-to-end traffic-priority policy to improve the control and throughput of important data. You can manage available bandwidth so that the most important traffic goes first. For example, you can use QoS to:

- Upgrade or downgrade traffic from various servers.
- Control the priority of traffic from dedicated VLANs or applications.

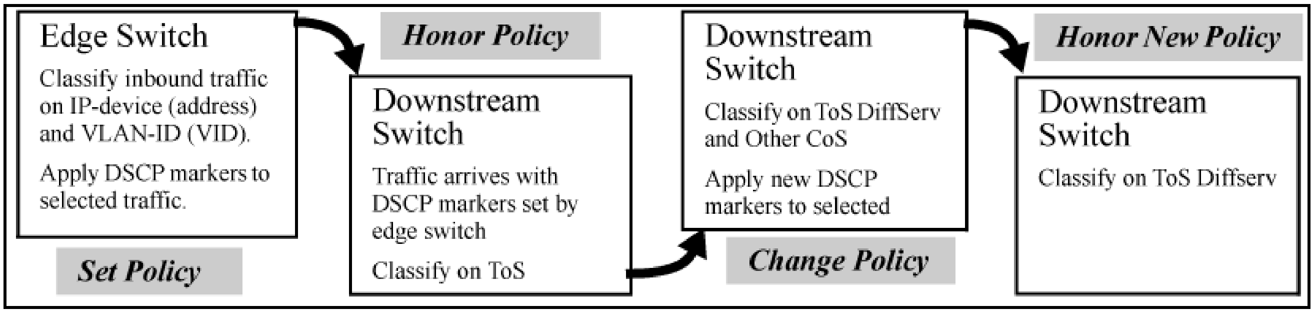
- Change the priorities of traffic from various segments of your network as your business needs change.
- Set priority policies in edge switches in your network to enable traffic-handling rules across the network.

Figure 24: *802.1p priority based on CoS (Class-of-Service) types and use of VLAN tags*



Edge Switch
Classify inbound traffic on these Class-of-Service (CoS) types:
• IP-device (address)
• Protocol (LAN)
• VLAN-ID (VID).
• Source-Port
Apply 802.1p priority to selected outbound traffic on tagged VLANs.
Set Priority
Honor Priority
Downstream Switch
Tagged VLANs on inbound and outbound ports.
Traffic arrives with priority set by edge switch
Forward with 802.1p priority.
Downstream Switch
Tagged VLANs on some or all inbound and outbound ports.
Classify inbound traffic on CoS types.
Change priority on selected CoS type(s).
Forward with 802.1p priority.
Change Priority
Honor New Priority
Downstream Switch
Tagged VLANs on at least some inbound ports.
Traffic arrives with the priority set in the VLAN tag. Carry priority downstream on tagged VLANs.

Figure 25: *Application of Differentiated Services Codepoint (DSCP) policies*



Edge Switch
Classify inbound traffic on IP-device (address) and VLAN-ID (VID).
Apply DSCP markers to selected traffic.
Set Policy
Honor Policy
Downstream Switch
Traffic arrives with DSCP markers set by edge switch
Classify on ToS
Downstream Switch
Classify on ToS DiffServ and Other CoS
Apply new DSCP markers to selected
Change Policy
Honor New Policy
Downstream Switch
Classify on ToS Diffserv

Applying QoS to inbound traffic at the network edge

At the edge switch, QoS classifies certain traffic types and in some cases applies a DSCP policy. At the next hop (downstream switch) QoS honors the policies established at the edge switch. Further downstream, another switch may reclassify some traffic by applying new policies, and yet other downstream switches can be configured to honor the new policies.

Preserving QoS in outbound traffic in a VLAN

QoS is implemented in the form of rules or policies that are configured on the switch. Although you can use QoS to prioritize traffic only while it moves through the switch, you derive the maximum benefit by using QoS in an 802.1Q VLAN environment (with 802.1p priority tags) or in an untagged VLAN environment (with DSCP policies in which QoS sets priorities that downstream devices can support without reclassifying the traffic).

Using QoS to optimize existing network resources

By prioritizing traffic, QoS supports traffic growth on the network while optimizing the use of existing resources—and delaying the need for further investments in equipment and services. QoS enables you to:

- Specify which traffic has higher or lower priority, regardless of current network bandwidth, or the relative priority setting of the traffic when it is received on the switch.
- Change (upgrade or downgrade) the priority of outbound traffic.
- Override "illegal" packet priorities set by upstream devices or applications that use 802.1Q VLAN tagging with 802.1p priority tags.

Overview of QoS settings

QoS settings operate on two levels:

- **Controlling the priority of outbound packets moving through the switch:** Configuring a new 802.1p priority value allows you to set the outbound priority queue to which a packet is sent. For example, you can configure an 802.1p priority of 0 through 7 for an outbound packet. When the packet is sent to a port, the QoS priority determines the outbound queue to which the packet is assigned as shown in the following table:

  Table 12: *802.1p priority settings and outbound queue assignment*

802.1p priority setting	Outbound port 8 queues	Outbound port 4 queues	Outbound port 2 queues
1	1	1	1
2	2		
0	3	2	
3	4		
4	5	3	2
5	6		
6	7	4	
7	8		

  (In an 802.1Q VLAN environment with VLAN-tagged ports, if QoS is **not** configured on the switch, but is configured on an upstream device, the priorities carried in the packets determine the forwarding queues in the switch.)

- **Configuring a priority for outbound packets and a service (priority) policy for use by downstream devices:**
  - **DSCP Policy:** This feature enables you to set a priority policy in outbound IP packets. (You can configure downstream devices to read and use this policy.) This method is not dependent on VLAN-tagged ports to carry priority policy to downstream devices, and can:
    - Change the codepoint (the upper 6 bits) in the ToS byte.
    - Set a new 802.1p priority for the packet.

    (Setting DSCP policies requires IPv4 inbound packets.)

  - **802.1p priority rules:** An outbound, VLAN-tagged packet carries an 802.1p priority setting that was configured (or preserved) in the switch. This priority setting ranges from 0 to 7, and can be used by downstream devices having up to eight outbound port queues. Thus, if packets within the switch move at the four priority levels shown in the table above, they still can carry an 802.1p priority that can be used by downstream devices having more or less than the four priority levels in the switches covered in this guide.

Also, if the packet enters the switch with an 802.1p priority setting, QoS can override this setting if configured with an 802.1p priority rule to do so.

NOTE: If your network uses only one VLAN (and therefore does not require VLAN-tagged ports), you can still preserve 802.1p priority settings in your traffic by configuring the ports as tagged VLAN members on the links between devices you want to honor traffic priorities.

Rule and policy limits: A large number of 802.1p priority rules and/or DSCP policies are allowed in any combination. For example, for the 2540 switch 6000 are allowed.

You can configure a QoS priority of 0 through 7 for an outbound packet. When the packet is then sent to a port, the QoS priority determines which outbound queue the packet uses:

Table 13: *QoS priority settings and operation*

802.1p priority setting	Outbound port 8 queues	Outbound port 4 queues	Outbound port 2 queues
1	1	1	1
2	2		
0	3	2	
3	4		
4	5	3	2
5	6		
6	7	4	
7	8		

If a packet is not in a VLAN-tagged port environment, then the QoS settings in the table above control only to which outbound queue the packet goes. Without VLAN tagging, no 802.1p priority is added to the packet for downstream device use. But if the packet is in a VLAN-tagged environment, then the above setting is also added to the packet as an 802.1p priority for use by downstream devices and applications (shown in the table below). In either case, an IP packet can also carry a priority policy to downstream devices by using DSCP-marking in the ToS byte.

Table 14: *Mapping switch QoS priority settings to device queues*

Priority setting	Outbound port queues in the switch	802.1p priority setting added to tagged VLAN packets exiting the switch	Queue assignment in downstream devices with:		
			8 queues	3 queues	2 queues
1	Queue 1	1 (low priority)	Queue 1	Queue 1	Queue 1
2		2	Queue 2		
0	Queue 2	0 (normal priority)	Queue 3	Queue 2	
3		3	Queue 4		

Table Continued

Priority setting	Outbound port queues in the switch	802.1p priority setting added to tagged VLAN packets exiting the switch	Queue assignment in downstream devices with:		
			8 queues	3 queues	2 queues
4	Queue 3	4 (medium priority)	Queue 5	Queue 3	Queue 2
5		5	Queue 6		
6	Queue 4	6 (high priority)	Queue 7		
7		7	Queue 8		

Classifiers for prioritizing outbound packets

NOTE:

Regarding using multiple criteria: Hewlett Packard Enterprise recommends that you configure a minimum number of the available QoS classifiers for prioritizing any given packet type. Increasing the number of active classifier options for a packet type increases the complexity of the possible outcomes and consumes switch resources.

Packet classifiers and evaluation order

The switches covered in this guide provide six types of globally-configured QoS classifiers (match criteria) to select packets for QoS traffic marking.

The switches covered in this guide provide six QoS classifiers (packet criteria) you can use to configure QoS priority.

Table 15: *Classifier search order and precedence*

Search order	Precedence	Global QoS classifier
1	1 (highest)	UDP/TCP application type (port)
2	2	Device priority (destination or source IP address)
3	3	IP type of service (ToS): precedence and DSCP bit sets (IP packets only)
4	4	IP protocol (IP, IPX, ARP, AppleTalk, SNA, and NetBeui)
5	5	VLAN ID
6	6	Incoming source-port on the switch
Default	7 (lowest)	The incoming 802.1p priority (present in tagged VLAN environments) is preserved if no global QoS classifier with a higher precedence matches.

Where multiple classifier types are configured, a switch uses the highest-to-lowest search order shown in the table to identify the highest-precedence classifier to apply to any given packet. When a match between a packet and a classifier is found, the switch applies the QoS policy configured for that classifier and the packet is handled accordingly.

NOTE: On the switches covered in this guide, if the switch is configured with multiple classifiers that address the same packet, the switch uses only the QoS configuration for the QoS classifier that has the highest precedence. In this case, the QoS configuration for another, lower-precedence classifier that may apply is ignored. For example, if QoS assigns high priority to packets belonging to VLAN 100, but normal priority to all IP protocol packets, since protocol priority (4) has precedence over VLAN priority (5), IP protocol packets on VLAN 100 will be set to normal priority.

Preparation for configuring QoS

Preserving 802.1p priority

QoS operates in VLAN-tagged and VLAN-untagged environments. If your network does not use multiple VLANs, you can still implement the 802.1Q VLAN capability for packets to carry their 802.1p priority to the next downstream device. To do so, configure ports as VLAN-tagged members on the links between switches and routers in your network infrastructure.

Table 16: *Summary of QoS capabilities*

Outbound packet options	Port membership in VLANs	
	Tagged	Untagged
Control port queue priority for packet types	Yes	Yes
Carry 802.1p priority assignment to next downstream device	Yes	No
Carry DSCP policy to downstream devices. The policy includes: • Assigning a ToS Codepoint • Assigning an 802.1p Priority to the Codepoint	Yes	Yes[2]

Steps for configuring QoS on the switch

Procedure

1. Determine the QoS policy you want to implement. This includes analyzing the types of traffic flowing through your network and identifying one or more traffic types to prioritize. In order of QoS precedence, these are:

   a. UDP/TCP applications

   b. Device Priority—destination or source IP address (Note that destination has precedence over source. See the table below.

   c. IP ToS Precedence Bits (Leftmost three bits in the ToS field of IP packets)

d. IP ToS Differentiated Service bits (Leftmost 6 bits in the ToS field of IP packets)

e. Layer 3 Protocol Priority

f. VLAN Priority (requires at least one tagged VLAN on the network)

g. Source-Port

h. Incoming 802.1p Priority (requires at least one tagged VLAN on the network). In a tagged VLAN environment, the incoming 802.1p priority is used as the default QoS classifier if no QoS classifier with a higher precedence matches

2. Select the QoS option you want to use. The following table lists the traffic types (QoS classifiers) and the QoS options you can use for prioritizing or setting a policy on these traffic types:

Table 17: *QoS marking supported by QoS classifiers*

Global QoS classifiers	**Type of QoS marking used to prioritize outbound traffic**	
	802.1p Priority[1] only	**DSCP policy[2]: DSCP codepoint with 802.1p priority**
UDP/TCP	Supported	Supported
IP Device	Supported	Supported
IP Precedence	Supported[3]	Not Supported
IP DiffServ	Supported	Supported
L3 Protocol	Supported	Not Supported
VLAN ID	Supported	Supported
Source Port	Supported	Supported

[1] When you configure only the 802.1p priority to mark packets that match a global QoS classifier, the selected traffic is prioritized and sent to the corresponding outbound port queue on the switch. VLAN-tagged ports are necessary to carry the 802.1p priority in a packet header to downstream devices.

[2] When you configure a DSCP policy to mark packets that match a global QoS classifier, the selected traffic is also prioritized according to the associated 802.1p priority and sent to the corresponding outbound port queue on the switch. VLAN-tagged ports carry the 802.1p priority in a packet header to downstream devices. In addition, you can configure downstream devices to read the DSCP value in IP packets and implement the service policy implied by the codepoint.

[3] When using a global QoS IP Precedence classifier, the 802.1p priority is automatically assigned to matching packets based on the IP precedence bit set in the packet header.

3. If you want 802.1p priority settings to be included in outbound packets, ensure that tagged VLANs are configured on the appropriate downstream links.

4. Determine the actual QoS configuration changes you will need to make on each QoS-capable device in your network in order to implement the desired policy. Also, if you want downstream devices to read and use DSCPs in IP packets from the switch, configure them to do so by enabling ToS Differentiated Service mode and making sure that the same DSCP policies are configured.

Using classifiers to configure QoS for outbound traffic

NOTE: In addition to the information in this section on the various QoS classifiers, see **QoS operating notes and restrictions** on page 232.

Viewing the QoS configuration

Examples of the `show qos` output are included with the example for each priority type.

Syntax:

```
show qos <priority-classifier>
```

`device-priority`: Displays the device priority table/configuration (priority based on the IP address).

`dscp-map`: Displays mappings between DSCP policy and 802.1p priority.

`port-priority`: Displays the current source-port priority configuration.

`protocol-priority`: Displays the protocol priority configuration.

`queue-config`: Displays the outbound port queue configuration information.

`resources`: Displays the resources used by the Policy Enforcement Engine.

`tcp-udp-port-priority`: Displays the TCP/UDP port priorities.

`type-of-service`: Displays the current type-of-service priority configuration. The display output differs according to the ToS option used:

- `IP Precedence`
- `Diffserve`

`vlan-priority`: Displays the current VLAN priority configuration.

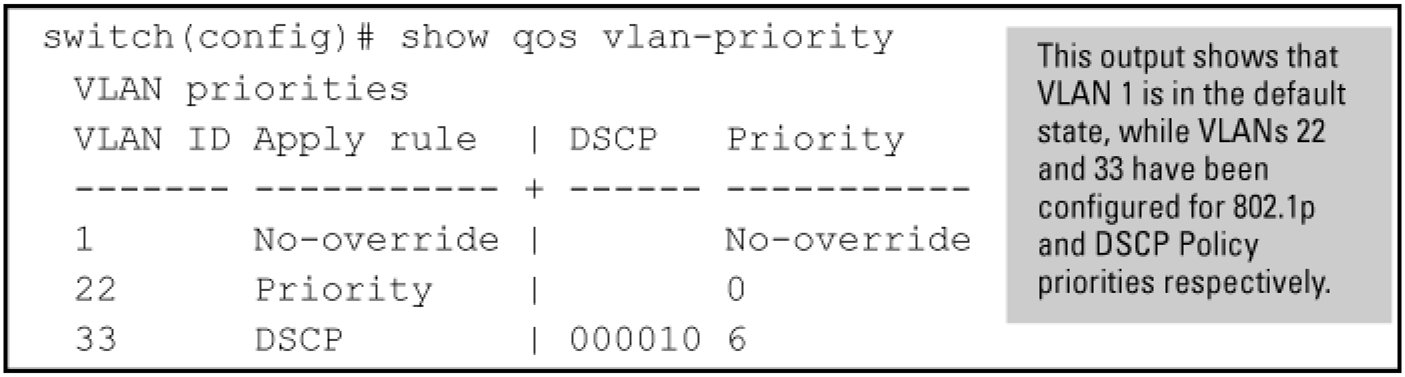
No override

By default, the `show` command outputs automatically list `No-override` for priority options that have not been configured. This means that if you do not configure a priority for a specific option, QoS does not prioritize packets to which that option applies, resulting in the `No-override` state.

- IP packets received through a VLAN-tagged port are managed using the 802.1p priority they carry in the 802.1Q field in their headers.
- VLAN-tagged packets received through an untagged port are handled by the switch with “normal” priority.

the `show qos vlan-priority` output shows the global QoS configurations on the switch that are configured with the VLAN ID classifier. Note that non-default 802.1p priorities have been configured for VLAN IDs 22 and 33; packets received on VLAN 1 are managed with the default settings, as described in the two bulleted items above.

Figure 26: *Output for the `show qos vlan-priority` command (example)*



```
switch(config)# show qos vlan-priority
 VLAN priorities
 VLAN ID Apply rule   | DSCP   Priority
 ------- ----------- + ------ -----------
 1       No-override |        No-override
 22      Priority    |        0
 33      DSCP        | 000010 6
```

This output shows that VLAN 1 is in the default state, while VLANs 22 and 33 have been configured for 802.1p and DSCP Policy priorities respectively.

Global TCP/UDP classifier

Global QoS classifier precedence: 1

When you use TCP or UDP and a layer 4 Application port number as a global QoS classifier, traffic carrying the specified TCP/UDP port numbers is marked with a specified priority level, without regard for any other QoS classifiers in the switch.

Options for assigning priority

Priority control options for TCP or UDP packets carrying a specified TCP or UDP port number include:

- 802.1p priority
- DSCP policy: Assigning a new DSCP and 802.1p priority

For a given TCP or UDP port number, you can use only one of the above options at a time. However, for different port numbers, you can use different options.

You can have up to 250 rules maximum for all TCP or UDP ports with assigned priorities.

TCP/UDP port number ranges

There are three ranges:

- Well-Known Ports: 0 – 1023
- Registered Ports: 1024 – 49151
- Dynamic and Private Ports: 49152 – 65535

For more information, including a listing of UDP/TCP port numbers, go to the **Internet Assigned Numbers Authority** (IANA) website at:

http://www.iana.org

Then click:

Protocol Number Assignment Services

P (under **Directory of General Assigned Numbers**)

Port Numbers

Assigning an 802.1p priority for a global TCP/UDP classifier

This option assigns an 802.1p priority to (IPv4) TCP or UDP packets as described below.

Syntax:

```
qos < udp-port | tcp-port > < tcp or udp port-number | range <start end> > priority < 0-7>
```

Configures an 802.1p priority for outbound packets having the specified TCP or UDP application port number. This priority determines the packet's queue in the outbound port to which it is sent. If the packet leaves the switch on a tagged port, it carries the 802.1p priority with it to the next downstream device. (Default: Disabled)

`port-number`: TCP/UDP port number from 1 to 65535.

`range <start end>` : Marks a range of TCP/UDP ports. If you specify a range, the minimum port number must precede the maximum port number in the range.

`priority <0-7>`: Marks the specified 802.1p priority in matching TCP or UDP packets.

Syntax:

```
no qos < udp-port | tcp-port > < tcp-udp port-number | range <start end> >
```

Deletes the specified UDP or TCP port number or range of numbers as QoS classifiers.

Syntax:

```
show qos tcp-udp-port-priority
```

Displays a listing of all TCP and UDP QoS classifiers currently in the running-config file.

Operating notes on using TCP/UDP port ranges

- Only six concurrent policies are possible when using unique ranges. The number of policies allowed is less if ACLs are also using port ranges.
- No ranges allowed that include any port numbers configured as part of another QoS application port number policy.
- An error message is generated if there are not enough hardware resources available when configuring a policy.
- The entire range of configured port numbers must be specified when using the `no` form of the command, for example:

```
switch(config)# qos udp-port range 1300 1399 dscp 001110
switch(config)# no qos range 1300 1399
```

The following example displays the following configuration for TCP and UDP port prioritization:

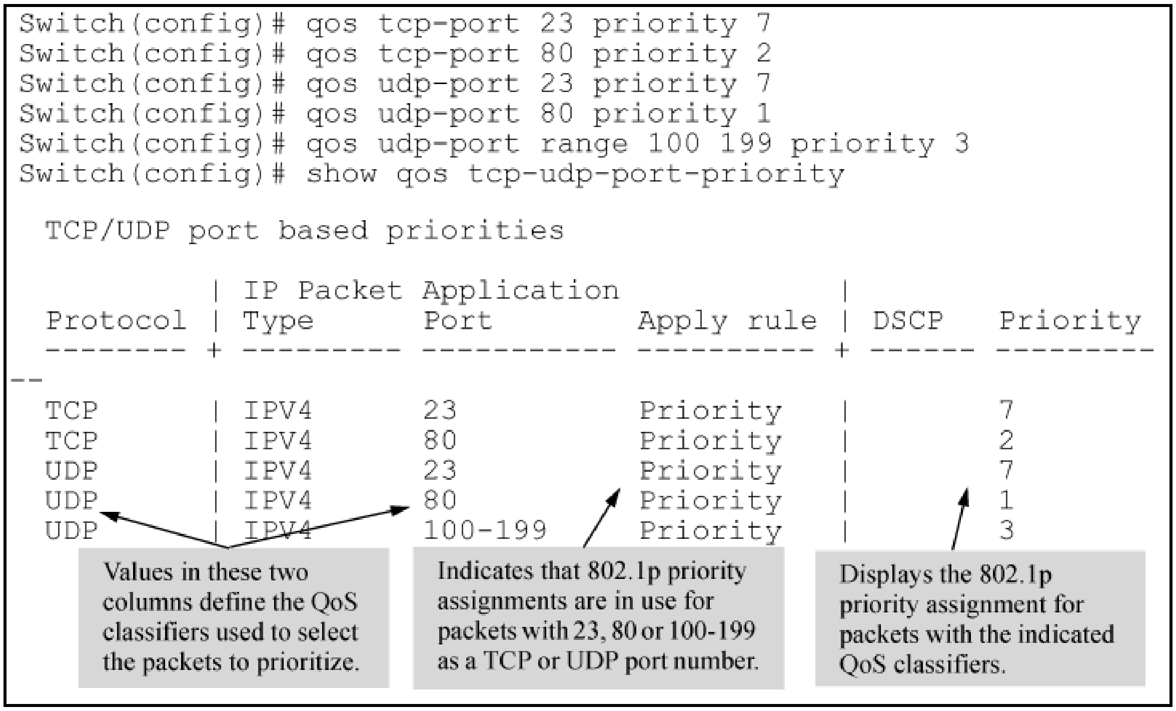
Configuration for TCP and UDP port prioritization

TCP/UDP port	802.1p priority for TCP	802.1p priority for UDP
TCP Port 23 (Telnet)	7	7
UDP Port 23 (Telnet)	7	7

Table Continued

TCP/UDP port	802.1p priority for TCP	802.1p priority for UDP
TCP Port 80 (World Wide Web HTTP)	2	2
UDP Port 80 (World Wide Web HTTP)	1	1

Figure 27: *Configuring 802.1p priority assignments on TCP/UDP ports*



```
Switch(config)# qos tcp-port 23 priority 7
Switch(config)# qos tcp-port 80 priority 2
Switch(config)# qos udp-port 23 priority 7
Switch(config)# qos udp-port 80 priority 1
Switch(config)# qos udp-port range 100 199 priority 3
Switch(config)# show qos tcp-udp-port-priority

 TCP/UDP port based priorities

          | IP Packet Application          |
 Protocol | Type      Port        Apply rule | DSCP   Priority
 -------- + --------- ----------- ---------- + ------ ---------
--
 TCP      | IPV4      23          Priority   |        7
 TCP      | IPV4      80          Priority   |        2
 UDP      | IPV4      23          Priority   |        7
 UDP      | IPV4      80          Priority   |        1
 UDP      | IPV4      100-199     Priority   |        3
```

Values in these two columns define the QoS classifiers used to select the packets to prioritize.

Indicates that 802.1p priority assignments are in use for packets with 23, 80 or 100-199 as a TCP or UDP port number.

Displays the 802.1p priority assignment for packets with the indicated QoS classifiers.

Assigning a DSCP policy for a global TCP/UDP classifier

This option assigns a previously configured DSCP policy (codepoint and 802.1p priority) to (IPv4) TCP or UDP packets having the specified port number. The switch does the following:

Procedure

1. Selects an incoming IP packet if the TCP or UDP port number it carries matches the port number specified in the TCP or UDP classifier (as shown in the figure in **Operating notes on using TCP/UDP port ranges**).
2. Overwrites (re-marks) the packet's DSCP with the new DSCP configured for matching packets.
3. Assigns the 802.1p priority associated with the new DSCP (see **Differentiated Services Codepoint (DSCP) mapping** on page 209).
4. Forwards the packet through the appropriate outbound port queue.

Creating a DSCP policy based on TCP/UDP port number classifiers

The following procedure creates a DSCP policy for IP packets carrying the selected TCP or UDP port-number classifier.

Procedure

1. Identify the TCP or UDP port-number classifier you want to use for assigning a DSCP policy.
2. Determine the DSCP policy for packets carrying the selected TCP or UDP port number or range of port numbers.
    a. Determine the DSCP you want to assign to the selected packets. (This codepoint will be used to overwrite (re-mark) the DSCP carried in packets received from upstream devices.)
    b. Determine the 802.1p priority you want to assign to the DSCP.
3. If necessary, use the `qos dscp-map` *`<codepoint>`* `priority <0-7>` command to configure the DSCP policy (codepoint and associated 802.1p priority) that you want to use to mark matching packets.

NOTE: Prerequisite: A DSCP codepoint must have a preconfigured 802.1p priority (0 - 7) before you can use the codepoint to mark matching packets. If a codepoint you want to use shows `No-override` in the Priority column of the DSCP Policy table (using the `show qos dscp-map` command), you must first configure a priority for the codepoint before proceeding (using the `qos dscp-map priority` command).

(Optional) This command is required only if an 802.1p priority is **not** already assigned to the specified *`<codepoint>`* in the DSCP Policy table. Valid values for a DSCP codepoint are as follows:

- A binary value for the six-bit codepoint from `000000` to `111111`.
- A decimal value from `0` (low priority) to `63` (high priority) that corresponds to a binary DSCP bit set
- An ASCII standard (hexadecimal) name for a binary DSCP bit set:

`af11` (001010)	`af42` (100100)
`af12` (001100)	`af43` (100110)
`af13` (001110)	`ef` (101110)
`af21` (010010)	`cs1` (001000) = precedence 1
`af22` (010100)	`cs2` (010000) = precedence 2
`af23` (010110)	`cs3` (011000) = precedence 3
`af31` (011010)	`cs4` (100000) = precedence 4
`af32` (011100)	`cs5` (101000) = precedence 5
`af33` (011110)	`cs6` (110000) = precedence 6

Table Continued

`af41` (100010)	`cs7` (111000) = precedence 7
`default` (000000)	

Enter **?** to display the list of valid codepoint entries.

When the switch applies the specified DSCP policy to a packet, the priority determines the packet's queue in the outbound port to which it is sent. If the packet leaves the switch on a tagged port, it carries the 802.1p priority with it to the next downstream device. For IP packets, the DSCP will be replaced by the codepoint specified in this command.

(Default: `No-override` for most codepoints.)

4. Configure the switch to assign the DSCP policy to packets with the specified TCP or UDP port number or range of port numbers.

```
no qos {udp-port | tcp-port} [ipv4 | ipv6 | ipv-all] {<port-number> | range
<port start><port end>} {dscp < codepoint> | priority <priority>}
```

```
no qos {udp-port | tcp-port} [ipv4 | ipv6 | ipv-all] {<port-number> | range
<port start><port end>} {dscp < codepoint> | priority <priority>}
```

Assigns a DSCP policy to outbound packets having the specified TCP or UDP application-port number or port range, and overwrites the DSCP in these packets with the assigned *<codepoint>* value, where:

- `port-number`: specifies a TCP/UDP port-number from 1 to 65535.
- `range` *<start end>*: specifies a range of TCP/UDP ports. If you specify a range, the minimum port number must precede the maximum port number in the range.
- `dscp` *<codepoint>*: overwrites the DSCP codepoint in the IPv4 ToS byte or IPv6 Traffic Class byte of matching packets with the specified value.Valid values for the DSCP codepoint are as follows:
    - A binary value for the 6-bit codepoint from `000000` to `111111`.
    - A decimal value from `0` (low priority) to `63` (high priority) that corresponds to a binary DSCP bit set
    - An ASCII standard name for a binary DSCP bit set

      Enter **?** to display the list of valid codepoint entries.

      The DSCP value you enter must be currently associated with an 802.1p priority in the DSCP Policy table. The 802.1p priority and determines the packet's queue in the outbound port to which it is sent. If the packet leaves the switch on a tagged port, it carries the 802.1p priority with it to the next downstream device.

      The default DSCP codepoint is `No-override`. The DSCP codepoint is not overwritten in matching packets.

The `no` form of the command deletes the specified UDP or TCP port number or range of port numbers as a QoS classifier. If you configured a range of port numbers as the QoS classifier, you must enter the entire range in the `no` command; you cannot remove part of a range.

Syntax

```
show qos tcp-udp-port-priority
```

Displays a listing of all TCP and UDP QoS classifiers currently in the running-config file.

Example:

This example shows how to assign the following DSCP policies to packets that match the specified TCP and UDP port applications:

Port Applications	DSCP Policies	
	DSCP	Priority
23-UDP	000111	7
80-TCP	000101	5
914-TCP	000010	1
1001-UDP	000010	1

1. Determine if the DSCP codepoints that you want to use to mark matching packets already have an 802.1p priority assigned, which could indicate use by existing applications (`show qos dscp-map` command).

   A DSCP codepoint must also have a priority configured before you can use it to mark matching packets.

```
switch(config)# show qos dscp-map

  DSCP -> 802.p priority mappings

  NOTE: ‘qos type-of-service diff-services’ must be configured
         before DSCP is honored on inbound traffic.

  DSCP CodePoint DSCP Value 802.1p tag   DSCP Policy name
  -------------- ---------- -----------  ----------------------
  000000         0          0            cs0
  000001         1          No-override
  000010         2          No-override
  000011         3          No-override
  000100         4          No-override
  000101         5          No-override
  000110         6          No-override
  000111         7          No-override
  001000         8          1            cs1
  001001         9          No-override
```

2. Configure the DSCP policies for the codepoints you want to use.

```
switch(config)# qos dscp-map af11 priority 3
switch(config)# qos dscp-map 13 priority 3
switch(config)# qos dscp-map af13 priority 3
switch(config)# write memory

switch(config)# show config
switch configuration:

; J9146 Configuration Editor; Created on release XX.15.XX

hostname “Switch”
time daylight-time-rule None
```

```
qos dscp-map af11 priority 3
qos dscp-map 13 priority 3
qos dscp-map af13 priority 3
...
```

3. Assign the DSCP policies to the selected TCP/UDP port applications and display the result.

```
switch(config)# qos udp-port 23 dscp 000111
switch(config)# qos tcp-port 80 dscp 000101
switch(config)# qos tcp-port 914 dscp 000010
switch(config)# qos udp-port range 1001 2000 dscp 000010

  TCP/UDP port based priorities

           | IP Packet Application            |
  Protocol | Type      Port        Apply rule | DSCP   Priority
  -------- + --------- ----------- ---------- + ------ -----------
  UDP      | IPV4      23          DSCP       | 8      7
  TCP      | IPV4      80          DSCP       | 6      5
  TCP      | IPV4      914         DSCP       | 3      1
  UDP      | IPV4      1001-2000   DSCP       | 3      1
```

The switch applies the DSCP policies in the above output to IP packets with the specified TCP/UDP port applications that are received in the switch. The switch manages the packets as follows:

- Overwrites the original DSCPs in the selected packets with the new DSCPs specified in the above policies.
- Assigns the 802.1p priorities in the above policies to the selected packets.

Global IP-device classifier

Global QoS classifier precedence: 2

The IP device option, which applies only to IPv4 packets, enables you to use up to 300 IP addresses (source or destination) as QoS classifiers.

Where a particular device-IP address classifier has the highest precedence in the switch for traffic addressed to or from that device, then traffic received on the switch with that address is marked with the IP address classifier’s configured priority level. Different IP device classifiers can have differing priority levels.

NOTE: QoS IP-device restriction: The switch does not allow a QoS IP-device priority for the Management VLAN IP address (if configured). If no Management VLAN is configured, then the switch does not allow configuring a QoS IP-device priority for the default VLAN IP address.

Options for assigning priority

The packet-marking options for global IP-device classifiers include:

- 802.1p priority
- DSCP policy: Assigning a new DSCP and 802.1p priority.

For a given IP address or subnet mask, you can assign only one of the above options at a time. However, for different IP addresses, you can use different options.

QoS IP Type-of-Service (ToS) policy and priority

Global QoS classifier precedence: 3

You can assign a maximum of 64 ToS rules. This feature applies only to IPv4 traffic and performs either of the following:

- **ToS IP-precedence mode:** All IP packets generated by upstream devices and applications include precedence bits in the ToS byte. Using this mode, the switch uses these bits to compute and assign the corresponding 802.1p priority.
- **ToS Differentiated Services (Diffserv) mode:** This mode requires knowledge of the codepoints set in IP packets by the upstream devices and applications. It uses the ToS codepoint in IP packets coming from upstream devices and applications to assign 802.1p priorities to the packets. You can use this option to do both of the following:
  - **Assign a new prioritization policy:** A "policy" includes both a codepoint and a corresponding 802.1p priority. This option selects an incoming IPv4 packet on the basis of its codepoint and assigns a new codepoint and corresponding 802.1p priority. (Use the `qos dscp-map` command to specify a priority for any codepoint; see **Differentiated Services Codepoint (DSCP) mapping** on page 209.)
  - **Assign an 802.1p priority:** This option reads the DSCP of an incoming IPv4 packet and, without changing this codepoint, assigns the 802.1p priority to the packet, as configured in the DSCP Policy Table (**Differentiated Services Codepoint (DSCP) mapping** on page 209). This means that a priority value of 0 – 7 must be configured for a DSCP before the switch will attempt to perform a QoS match on the packet's DSCP bits.

  Before configuring the ToS Diffserv mode, you must use the `qos dscp-map` command to configure the desired 802.1p priorities for the codepoints you want to use for either option. See **Differentiated Services Codepoint (DSCP) mapping** on page 209 for more information.

Unless IP-Precedence mode and Diffserv mode are both disabled (the default setting), enabling one automatically disables the other.

Assigning an 802.1p priority to IPv4 packets on the basis of the ToS precedence bits

If a device or application upstream of the switch sets the precedence bits in the ToS byte of IPv4 packets, you can use this feature to apply that setting for prioritizing packets for outbound port queues. If the outbound packets are in a tagged VLAN, this priority is carried as an 802.1p value to the adjacent downstream devices.

Syntax:

```
qos type-of-service ip-precedence
```

Causes the switch to automatically assign an 802.1p priority to all IPv4 packets by computing each packet's 802.1p priority from the precedence bits the packet carries. This priority determines the packet's queue in the outbound port to which it is sent. If the packet leaves the switch on a tagged port, it carries the 802.1p priority with it to the next downstream device.

(ToS IP Precedence Default: Disabled)

```
no qos type-of-service
```

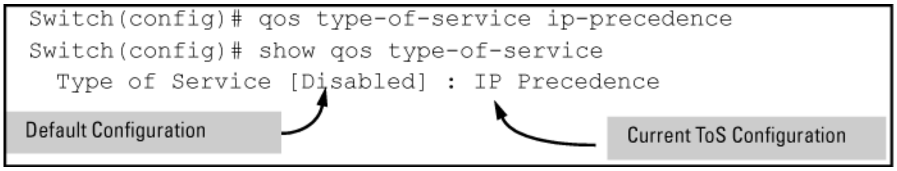
Disables all ToS classifier operation, including prioritization using the precedence bits.

```
show qos type-of-service
```

When the IP-precedence mode is enabled (or if neither ToS option is configured), this command displays the ToS configuration status. If the Diff-serv mode is enabled, codepoint data is displayed.

Using the IP-precedence classifier, prioritization of outbound packets relies on the IP-Precedence bit setting that IP packets carry with them from upstream devices and applications. To configure and verify this option:

Figure 28: *Enabling ToS IP-precedence prioritization*



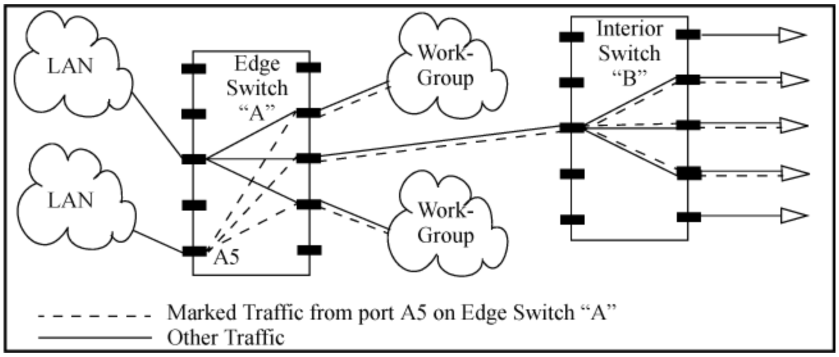
```
Switch(config)# qos type-of-service ip-precedence
Switch(config)# show qos type-of-service
  Type of Service [Disabled] : IP Precedence
```

To replace this option with the ToS diff-services option, configure diff-services as described below, which automatically disables IP-Precedence. To disable IP-Precedence without enabling the diff-services option, use this command: `no qos type-of-service`

Assigning an 802.1p priority to IPv4 packets on the basis of incoming DSCP

One of the best uses for this option is on an interior switch where you want to honor (continue) a policy set on an edge switch. That is, it enables you to select incoming packets having a specific DSCP and forward these packets with the desired 802.1p priority. For example, if an edge switch “A” marks all packets received on port 5 with a particular DSCP, you can configure a downstream (interior) switch “B” to handle such packets with the desired priority (regardless of whether 802.1Q tagged VLANs are in use).

Figure 29: *Interior switch B honors the policy established in edge switch A*



To do so, assign the desired 802.1p priority to the same codepoint that the upstream or edge switch assigns to the selected packets. When the downstream switch receives an IPv4 packet carrying one of these codepoints, it assigns the configured priority to the packet and sends it out the appropriate priority queue. (The packet retains the codepoint it received from the upstream or edge switch). You can use this option concurrently with the diffserv DSCP Policy option (described later in this section), as long as the DSCPs specified in the two options do not match.

NOTE: Regarding DSCP use: Different applications may use the same DSCP in their IP packets. Also, the same application may use multiple DSCPs if the application originates on different clients, servers, or other devices. Using an edge switch enables you to select the desired packets and mark them with predictable DSCPs that can be used by downstream switches to honor policies set in the edge switch.

When enabled, the switch applies direct 802.1p prioritization to all packets having codepoints that meet these prerequisites:

- The codepoint is configured with an 802.1p priority in the DSCP table. (Codepoints configured with `No-override` are not used.)
- The codepoint is not configured for a new DSCP policy assignment.

Thus, the switch does not allow the same incoming codepoint (DSCP) to be used simultaneously for directly assigning an 802.1p priority and also assigning a DSCP policy. For a given incoming codepoint, if you configure one option and then the other, the second overwrites the first.

To use this option:

Procedure

1. Identify a DSCP used to set a policy in packets received from an upstream or edge switch.
2. Determine the 802.1p priority (0–7) you want to apply to packets carrying the identified DSCP. (You can either maintain the priority assigned in the upstream or edge switch, or assign a new priority.)
3. Use `qos dscp-map <codepoint> priority <0-7>` to assign the 802.1p priority you want to the specified DSCP.
4. Enable `diff-services` if not already enabled.

Syntax:

```
qos type-of-service diff-services <codepoint>
```

Causes the switch to read the *<codepoint>* (DSCP) of an incoming IPv4 packet and, when a match occurs, assign a corresponding 802.1p priority, as configured in the switch's DSCP table (see **Differentiated Services Codepoint (DSCP) mapping**).

```
no qos type-of-service
```

Disables all ToS classifier operation.

```
no qos dscp-map <codepoint>
```

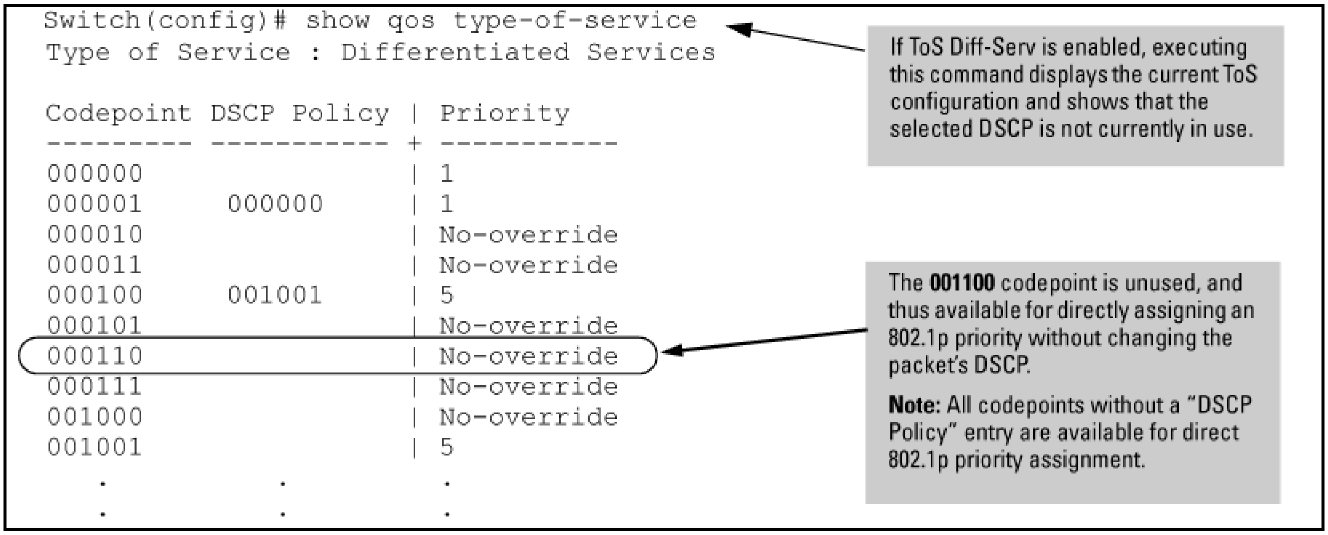
Disables direct 802.1p priority assignment to packets carrying the *<codepoint>* by reconfiguring the codepoint priority assignment in the DSCP table to `No-override`. If this codepoint is in use as a DSCP policy for another diffserv codepoint, you must disable or redirect the other diffserv codepoint's DSCP policy before you can disable or change the codepoint. For example, in **Figure 31: ToS configuration that enables both 802.1p priority and DSCP policy assignment** on page 196 you cannot change the priority for the 000000 codepoint until you redirect the DSCP policy for 000001 away from using 000000 as a policy. (See **Note on changing a priority setting** on page 211 and **Differentiated Services Codepoint (DSCP) mapping** on page 209.)

```
show qos type-of-service
```

Displays current Type-of-Service configuration. In diffserv mode it also shows the current direct 802.1p assignments and the current DSCP assignments covered later in this section.

For example, an edge switch “A” in an untagged VLAN assigns a DSCP of 000110 on IP packets it receives on port 6, and handles the packets with high priority (7). When these packets reach interior switch “B” you want the switch to handle them with the same high priority. To enable this operation, you would configure an 802.1p priority of 7 for packets received with a DSCP of 000110. ToS `diff-services` must be enabled as shown in the following images.

Figure 30: *Viewing the codepoints available for 802.1p priority assignments*



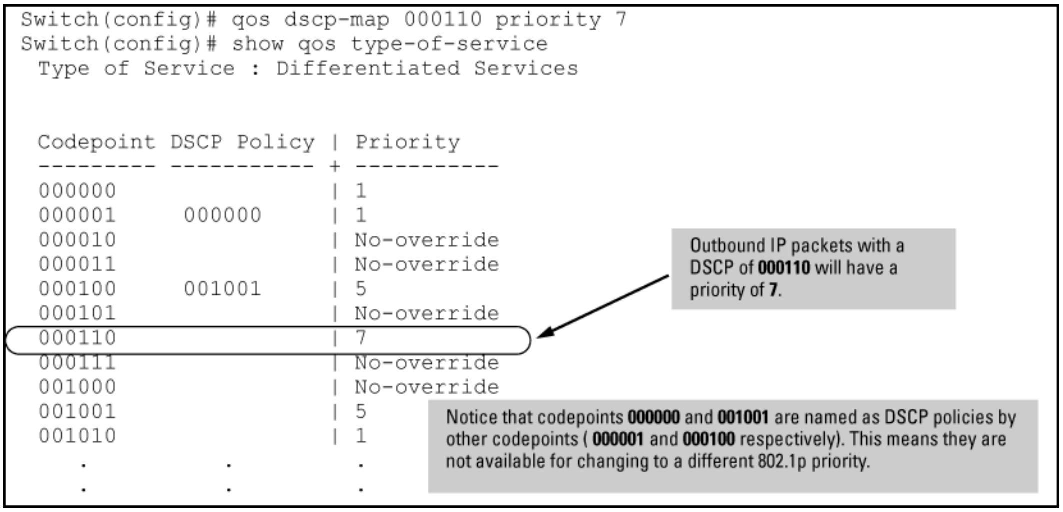
```
Switch(config)# show qos type-of-service
 Type of Service : Differentiated Services

 Codepoint DSCP Policy | Priority
 --------- ----------- + -----------
 000000                | 1
 000001     000000     | 1
 000010                | No-override
 000011                | No-override
 000100     001001     | 5
 000101                | No-override
 000110                | No-override
 000111                | No-override
 001000                | No-override
 001001                | 5
    .           .          .
    .           .          .
```

If ToS Diff-Serv is enabled, executing this command displays the current ToS configuration and shows that the selected DSCP is not currently in use.

The **001100** codepoint is unused, and thus available for directly assigning an 802.1p priority without changing the packet’s DSCP.

Note: All codepoints without a “DSCP Policy” entry are available for direct 802.1p priority assignment.

Figure 31: *ToS configuration that enables both 802.1p priority and DSCP policy assignment*



```
Switch(config)# qos dscp-map 000110 priority 7
Switch(config)# show qos type-of-service
  Type of Service : Differentiated Services


  Codepoint DSCP Policy | Priority
  --------- ----------- + -----------
  000000                | 1
  000001     000000     | 1
  000010                | No-override
  000011                | No-override
  000100     001001     | 5
  000101                | No-override
  000110                | 7
  000111                | No-override
  001000                | No-override
  001001                | 5
  001010                | 1
     .          .         .
     .          .         .
```

Outbound IP packets with a DSCP of **000110** will have a priority of **7**.

Notice that codepoints **000000** and **001001** are named as DSCP policies by other codepoints (**000001** and **000100** respectively). This means they are not available for changing to a different 802.1p priority.

Assigning a DSCP policy on the basis of the DSCP in IPv4 packets received from upstream devices

The preceding section describes how to forward a policy set by an edge (or upstream) switch. This option changes a DSCP policy in an IPv4 packet by changing its IP ToS codepoint and applying the priority associated with the new codepoint. (A DSCP policy consists of a differentiated services codepoint and an associated 802.1p priority.) You can use this option concurrently with the diffserv 802.1p priority option (above), as long as the DSCPs specified in the two options do not match.

To use this option to configure a change in policy:

Procedure

1. Identify the DSCP used to set a policy in packets received from an upstream or edge switch.
2. Create a new policy by using the `qos dscp-map` *`<code-point>`* `priority <0-7>` command to configure an 802.1p priority for the codepoint you will use to overwrite the DSCP that the packet carries from upstream.
3. Use the `qos type-of-service diff-services <` *`mapped to DSCP`* `> dscp <` *`mapped from DSCP`* `>` command to change the policy on packets coming from the edge or upstream switch with the specified incoming DSCP.

   Figure 29: Interior switch B honors the policy established in edge switch A on page 194 illustrates this scenario

Syntax:

```
qos type-of-service diff-services
```

Enables ToS Diff-serve QoS so that Diff-serve policy configurations can be applied to incoming packets that have matching codepoints.

Syntax:

```
qos type-of-service diff-services <current-codepoint> dscp <new-codepoint>
```

Configures the switch to select an incoming IP packet carrying the *`<current-codepoint>`* and then use the *`<new-codepoint>`* to assign a new, previously configured DSCP policy to the packet. The policy overwrites the *`<current-codepoint>`* with the *`<new-codepoint>`* and assigns the 802.1p priority specified by the policy.

Syntax:

```
no qos type-of-service
```

Disables all ToS classifier operation. Current ToS DSCP policies and priorities remain in the configuration and will become available if you re-enable ToS Diff-services.

Syntax:

```
no qos type-of-service [diff-services <codepoint>]
```

Deletes the DSCP policy assigned to the *`<codepoint>`* and returns the *`<codepoint>`* to the 802.1p priority setting it had before the DSCP policy was assigned, which is either a value from 0 - 7 or `No-override`.

Syntax:

```
show qos type-of-service
```

Displays a listing of codepoints with any corresponding DSCP policy reassignments for outbound packets. Also displays the 802.1p priority for each codepoint that does not have a DSCP remarking policy assigned to it.

Example

For example, suppose that you want to configure the following two DSCP policies for packets received with the indicated DSCPs.

Received DSCP	Policy DSCP	802.1p Priority	Policy Name (Optional)
001100	17	6	Level 6
001101	16	4	Level 4

1. Determine whether the DSCPs already have priority assignments, which could indicate use by existing applications. This is not a problem as long as the configured priorities are acceptable for all applications using the same DSCP. (See **Note on changing a priority setting** on page 211. Also, a DSCP must have a priority configured before you can assign any QoS classifiers to use it. See **Differentiated Services Codepoint (DSCP) mapping** on page 209.)
2. After configuring the DSCP policies for the codepoints you want to use, assign the policies to the codepoints in the selected packet type.

   An example of policy assignment to outbound packets on the basis of the DSCP in the packets received from upstream devices is shown below. The specified DSCP policies overwrite the original DSCPs on the selected packets, and use the 802.1p priorities previously configured.

```
switch(config)# qos type-of-service diff-services 001100 dscp 17
switch(config)# qos type-of-service diff-services 001101 dscp 16
switch(config)# show qos type-of-service
  Type of Service : Differentiated Services

  Codepoint DSCP Policy | Priority
  --------- ----------- + -----------
  000000                | No-override
  000001                | No-override
  000010                | No-override
  000011                | No-override
  000100                | No-override
  000101                | No-override
  000110                | No-override
  000111                | No-override
  001000    001011      | 7
  001001                | No-override
  001010                | 1
  001011                | 7
  001100    010001      | 6
  001101    010000      | 4
```

Details of QoS IP ToS

IP packets include a Type of Service (ToS) byte. The ToS byte includes:

- **A Differentiated Services Codepoint (DSCP):** This element is composed of the upper 6 bits of the ToS byte). There are 64 possible codepoints.
  - The default `qos` configuration includes some codepoints with 802.1p priority settings for Assured-Forwarding (af), Expedited Forwarding (ef, codepoint 101110), and Class Selector (cs). Others are unused (listed with `No-override` for a Priority).

  See **Figure 32: The ToS codepoint and precedence bits** on page 199 for an illustration of the default DSCP policy table.

  Using the `qos dscp-map` command, you can configure the switch to assign different prioritization policies to IPv4 packets having different codepoints. As an alternative, you can configure the switch to assign a new codepoint to an IPv4 packet, along with a corresponding 802.1p priority (0-7). To use this option in the simplest case, you would:

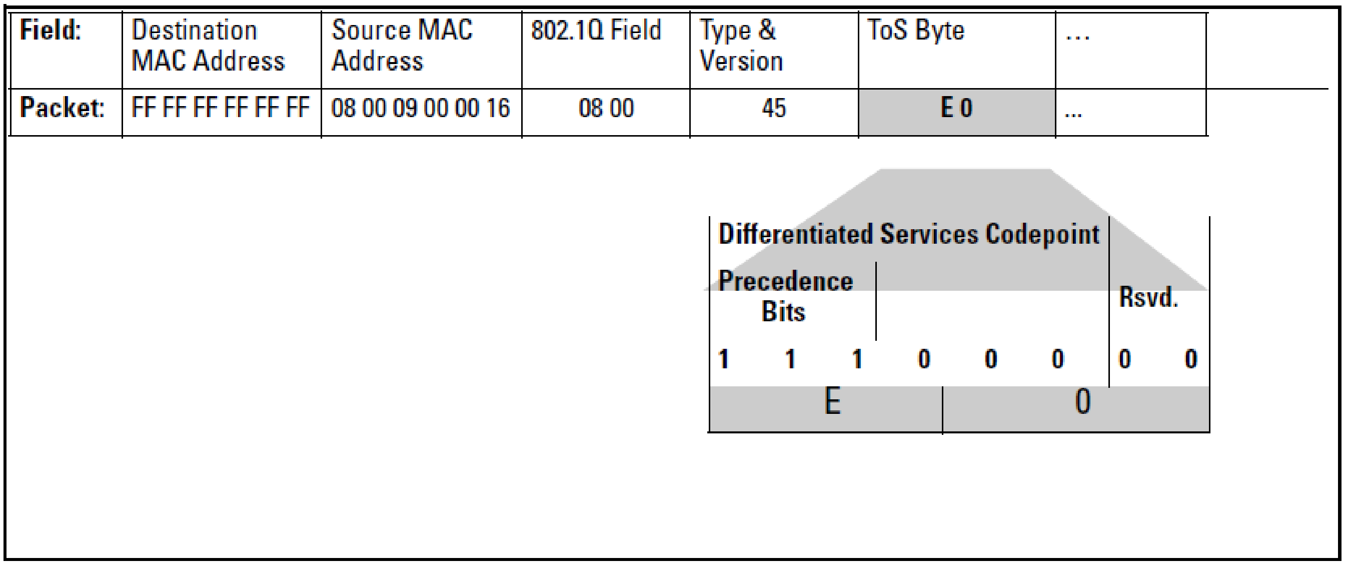
1. Configure a specific DSCP with a specific priority in an edge switch.
2. Configure the switch to mark a specific type of inbound traffic with that DSCP (and thus create a policy for that traffic type).
3. Configure the internal switches in your LAN to honor the policy.

(For example, you could configure an edge switch to assign a codepoint of 000001 to all packets received from a specific VLAN, and then handle all traffic with that codepoint at high priority.)For a codepoint listing and the commands for displaying and changing the DSCP Policy table, see **Differentiated Services Codepoint (DSCP) mapping** on page 209.

- **Precedence Bits:** This element is a subset of the DSCP and is composed of the upper 3 bits of the ToS byte. When configured to do so, the switch uses the precedence bits to determine a priority for handling the associated packet. (The switch does not change the setting of the precedence bits.) Using the ToS Precedence bits to prioritize IPv4 packets relies on priorities set in upstream devices and applications.

The following figure, shows an example of the ToS byte in the header for an IPv4 packet, and illustrates the diffserv bits and precedence bits in the ToS byte. (Note that the Precedence bits are a subset of the Differentiated Services bits.)

Figure 32: *The ToS codepoint and precedence bits*



Field:	Destination MAC Address	Source MAC Address	802.1Q Field	Type & Version	ToS Byte	...
Packet:	FF FF FF FF FF FF	08 00 09 00 00 16	08 00	45	E 0	...

Table 18: *How the switch uses the ToS configuration*

Outbound port	ToS option:	
	802.1p (value = 0 - 7)	**Differentiated services**
IP packet sent out an untagged port in a VLAN	Depending on the value of the IP Precedence bits in the packet's ToS field, the packet will go to one of eight outbound port queues in the switch. See the table in **Overview of QoS settings**.	For a given packet carrying a ToS codepoint that the switch has been configured to detect: • Change the codepoint according to the configured policy and assign the 802.1p priority specified for the new codepoint in the DSCP Policy Table (**Differentiated Services Codepoint (DSCP) mapping**). • Do not change the codepoint, but assign the 802.1p priority specified for the existing codepoint in the DSCP Policy Table (**Differentiated Services Codepoint (DSCP) mapping**). Depending on the 802.1p priority used, the packet will leave the switch through a queue as defined in the table in **Overview of QoS settings** on page 180.If `No-override` (the default) has been configured for a specified codepoint, then the packet is not prioritized by ToS and, by default, is sent to the "normal priority" queue.
IP packet sent out an untagged port in a VLAN	Same as above, plus the IP Precedence value (0 - 7) will be used to set a corresponding 802.1p priority in the VLAN tag carried by the packet to the next downstream device. See the table below.	Same as above, plus the Priority value (0 - 7) will be used to set a corresponding 802.1p priority in the VLAN tag carried by the packet to the next downstream device. Where `No-override` is the assigned priority, the VLAN tag carries a "0" (normal priority) 802.1p setting if not prioritized by other QoS classifiers.

Table 19: *ToS IP-precedence bit mappings to 802.1p priorities*

ToS byte IP precedence bits	Corresponding 802.1p priority	Service priority level
000	1	Lowest
001	2	Low
002	0	Normal
003	3	
004	4	

Table Continued

ToS byte IP precedence bits	Corresponding 802.1p priority	Service priority level
005	5	
006	6	
007	7	Highest

Global Layer-3 protocol classifier

Global QoS classifier precedence: 4

When a global Layer-3 Protocol classifier is configured as the highest-precedence classifier and the switch receives traffic carrying the specified protocol, matching packets are assigned the priority configured for the classifier.

Assigning a priority for a global Layer-3 protocol classifier

This global QoS packet-marking option assigns an 802.1p priority to outbound packets having the specified Layer-3 protocol.

Syntax:

```
qos protocol < ip | ipx | arp | appletalk | sna | netbeui> priority < 0 - 7 >
```

Configures an 802.1p priority for outbound packets having the specified protocol. This priority determines the packet's queue in the outbound port to which it is sent. If the packet leaves the switch on a tagged port, it carries the 802.1p priority with it to the next downstream device. You can configure one QoS classifier for each protocol type.

(Default: `No-override`)

Syntax:

```
no qos protocol < ip | ipx | arp | appletalk | sna | netbeui >
```

Disables use of the specified protocol as a QoS classifier and resets the protocol priority to `No-override`.

Syntax:

```
show qos protocol-priority
```

Lists the QoS protocol classifiers with their priority settings.

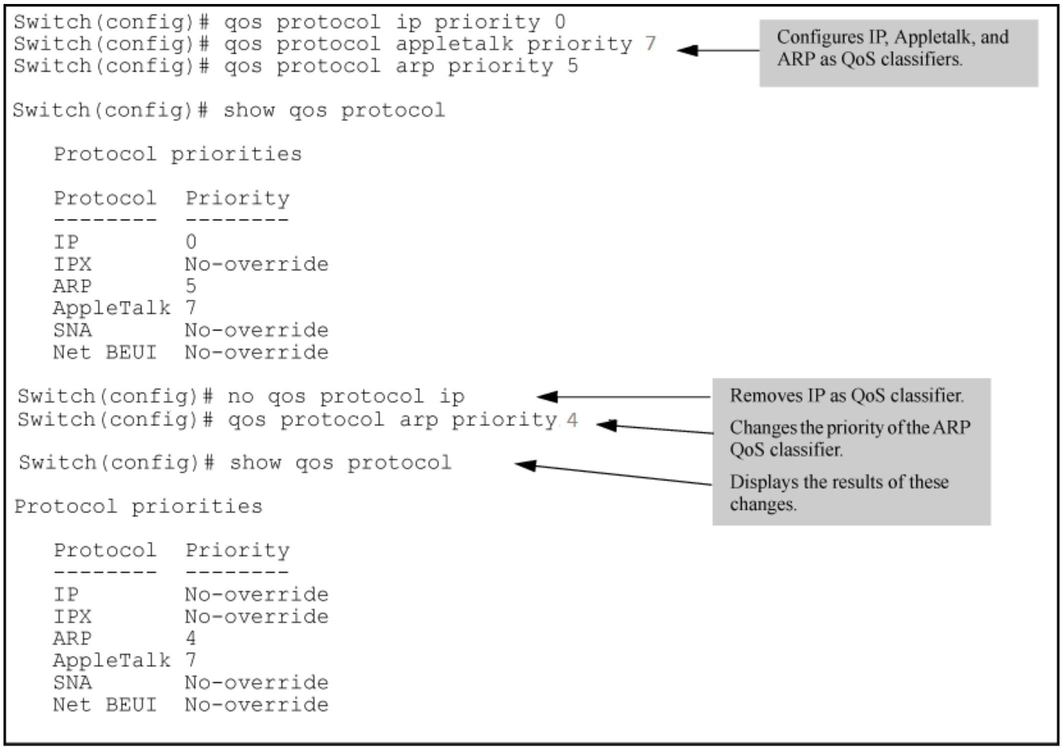
Configuring global Layer-3 protocol classifiers

To configure the following global Layer-3 protocol classifiers:

1. Configure QoS protocol classifiers with IP at 0 (normal), ARP at 5 (medium), and AppleTalk at 7 (high) and display the QoS protocol configuration.
2. Disable the QoS IP protocol classifier, downgrade the ARP priority to 4, and again display the QoS protocol configuration.

The following example shows the necessary configuration commands.

Figure 33: *Adding, viewing, removing, and changing QoS protocol classifiers*



```
Switch(config)# qos protocol ip priority 0
Switch(config)# qos protocol appletalk priority 7
Switch(config)# qos protocol arp priority 5

Switch(config)# show qos protocol

   Protocol priorities

   Protocol   Priority
   --------   --------
   IP         0
   IPX        No-override
   ARP        5
   AppleTalk  7
   SNA        No-override
   Net BEUI   No-override

Switch(config)# no qos protocol ip
Switch(config)# qos protocol arp priority 4

Switch(config)# show qos protocol

Protocol priorities

   Protocol   Priority
   --------   --------
   IP         No-override
   IPX        No-override
   ARP        4
   AppleTalk  7
   SNA        No-override
   Net BEUI   No-override
```

Configures IP, Appletalk, and ARP as QoS classifiers.

Removes IP as QoS classifier.

Changes the priority of the ARP QoS classifier.

Displays the results of these changes.

QoS VLAN-ID (VID) priority

Global QoS classifier precedence: 5

The QoS protocol option enables you to use up to 256 VIDs as QoS classifiers. Where a particular VLAN-ID classifier has the highest precedence in the switch for traffic in that VLAN, then traffic received in that VLAN is marked with the VID classifier’s configured priority level. Different VLAN-ID classifiers can have differing priority levels.

Options for assigning priority

Priority control options for packets carrying a specified VLAN-ID include:

- 802.1p priority
- DSCP policy: Assigning a new DSCP and 802.1p priority

(For operation when other QoS classifiers apply to the same traffic, see **Classifiers for prioritizing outbound packets** on page 182.)

NOTE: QoS with VID priority applies to static VLANs only, and applying QoS to dynamic VLANs created by GVRP operation is not supported. A VLAN must exist while a subject of a QoS configuration, and eliminating a VLAN from the switch causes the switch to clear any QoS features configured for that VID.

Assigning a priority based on VLAN-ID

This option assigns a priority to all outbound packets having the specified VLAN-ID (VID). You can configure this option by either specifying the VID ahead of the `qos` command or moving to the VLAN context for the VLAN you want to configure for priority.

Syntax:

```
vlan <vid> qos priority <0-7>
```

Configures an 802.1p priority for outbound packets belonging to the specified VLAN. This priority determines the packet's queue in the outbound port to which it is sent. If the packet leaves the switch on a tagged port, it carries the 802.1p priority with it to the next downstream device. You can configure one QoS classifier for each VLAN-ID.

(Default: `No-override`)

Syntax:

```
no vlan <vid> qos
```

Removes the specified VLAN-ID as a QoS classifier and resets the priority for that VLAN to `No-override`.

Syntax:

```
show qos vlan-priority
```

Displays a listing of the QoS VLAN-ID classifiers currently in the running-config file, with their priority data.

Procedure

1. For example, suppose that you have the following VLANs configured on the switch and want to prioritize them as shown:

```
switch(config)# show vlan

 Status and Counters - VLAN Information

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

  VLAN ID Name                         | Status     Voice Jumbo
  ------- ---------------------------- + ---------- ----- -----
  1       DEFAULT_VLAN                 | Static     No    No
  22      VLAN_22                      | Static     No    No
```

2. You would then execute the following commands to prioritize the VLANs by VID:

```
switch(config)# vlan 1 qos dscp 9
switch(config)# vlan 22 qos dscp 8

switch(config)# show qos vlan-priority

  VLAN priorities

  VLAN ID Apply rule  | DSCP   Priority
  ------- ----------- + ------ -----------
  1       DSCP        | 001001 7
  22      DSCP        | 001000 6
```

3. If you then decided to remove VLAN_22 from QoS prioritization:

   In this instance, `No-override` indicates that VLAN 22 is not prioritized by QoS.

```
switch(config)# no vlan 22 qos
switch(config)# show qos vlan

  VLAN priorities

  VLAN ID Apply rule   | DSCP   Priority
  ------- ----------- + ------ -----------
  1       DSCP        | 001001 7
  22      No-override |        No-override
```

Assigning a DSCP policy based on VLAN-ID

This option assigns a previously configured DSCP policy (codepoint and 802.1p priority) to outbound IP packets having the specified VLAN-ID (VID). The switch performs the following:

Procedure

1. Selects an incoming IP packet on the basis of the VLAN-ID it carries.
2. Overwrites the packet’s DSCP with the DSCP configured in the switch for such packets.
3. Assigns 802.1p priority configured in the switch for the new DSCP (see **Differentiated Services Codepoint (DSCP) mapping** on page 209).
4. Forwards the packet through the appropriate outbound port queue.

Steps for creating a policy based on VLAN-ID classifier:

1. Determine the VLAN-ID classifier to which you want to assign a DSCP policy.
2. Determine the DSCP policy for packets carrying the selected VLAN-ID:
   a. Determine the DSCP you want to assign to the selected packets. (This codepoint will be used to overwrite the DSCP carried in packets received through the source-port from upstream devices.)
   b. Determine the 802.1p priority you want to assign to the DSCP.
3. Configure the DSCP policy by using `qos dscp-map` to configure the priority for each codepoint (see **Differentiated Services Codepoint (DSCP) mapping** on page 209 for more information).
4. Configure the switch to assign the DSCP policy to packets with the specified VLAN-ID.

Syntax:

```
vlan <vid> qos dscp <codepoint>
```

Assigns a DSCP policy to packets carrying the specified VLAN-ID, and overwrites the DSCP in these packets with the assigned *<codepoint>* value. This policy includes an 802.1p priority and determines the packet’s queue in the outbound port to which it is sent. If the packet leaves the switch on a tagged port, it carries the 802.1p priority with downstream device.

(Default: `No-override`)

Syntax:

```
no vlan <vid> qos
```

Removes QoS classifier for the specified VLAN.

Syntax:

```
show qos vlan-priority
```

Displays a listing of the QoS VLAN-ID classifiers currently in the running-config file.

For example, suppose that you wanted to assign this set of priorities:

VLAN-ID	DSCP	Priority
40	15	7
30	16	5
20	17	1
1	17	1

Assign the DSCP policies to the selected VIDs and display the result.

An example of the completed VID-DSCP priority configuration is shown below.

```
switch(config)# vlan 1 qos dscp 17
switch(config)# vlan 20 qos dscp 17
switch(config)# vlan 30 qos dscp 16
switch(config)# vlan 40 qos dscp 15

switch(config)# show qos vlan-priority

 VLAN priorities

  VLAN ID Apply rule  | DSCP   Priority
  ------- ----------- + ------ -----------
  1       DSCP        | 010001 1
  20      DSCP        | 010001 1
  30      DSCP        | 010000 5
  40      DSCP        | 001111 7
```

In the example above, the switch will now apply the DSCP policies to packets received on the switch with the specified VLAN-IDs. This means the switch will:

- Overwrite the original DSCPs in the selected packets with the new DSCPs specified in the above policies.
- Assign the 802.1p priorities in the above policies to the appropriate packets.

QoS source-port priority

Global QoS classifier precedence: 6

The QoS source-port option enables you to use a packet's source-port on the switch as a QoS classifier. Where a particular source-port classifier has the highest precedence in the switch for traffic entering through that port, then traffic received from the port is marked with the source-port classifier's configured priority level. Different source-port classifiers can have different priority levels.

Options for assigning priority on the switch

Priority control options for packets from a specified source-port include:

- 802.1p priority
- DSCP policy: Assigning a new DSCP and 802.1p priority

(For operation when other QoS classifiers apply to the same traffic, see **Classifiers for prioritizing outbound packets** on page 182.)

Options for assigning priority from a RADIUS server

You can use a RADIUS server to impose a QoS source-port priority during an 802.1X port-access authentication session. See the RADIUS chapter in the *ArubaOS-Switch Access Security Guide* for your switch.

Assigning a priority based on source-port

This option assigns a priority to all outbound packets having the specified source-port. You can configure this option by either specifying the source-port ahead of the `qos` command or moving to the port context for the port you want to configure for priority. (If you are configuring multiple source-ports with the same priority, you may find it easier to use the `interface <port-list>` command to go to the port context instead of individually configuring the priority for each port.)

Syntax:

```
interface <port-list> qos priority <0-7>
```

Configures an 802.1p priority for packets entering the switch through the specified (source) ports. This priority determines the packet queue in the outbound ports to which traffic is sent. If a packet leaves the switch on a tagged port, it carries the 802.1p priority with it to the next downstream device. You can configure one QoS classifier for each source-port or group of source-ports.

(Default: `No-override`)

Syntax:

```
no interface <port-list> qos
```

Disables use of the specified source-ports for QoS classifiers and resets the priority for the specified sourceports to `No-override`.

Syntax:

```
show qos port-priority
```

Lists the QoS port-priority classifiers with their priority data.

For example, suppose that you want to prioritize inbound traffic on the following source-ports:

Source-port	Priority
1–3	2
4	3

You would then execute the following commands to prioritize traffic received on the above ports.

Configuring and displaying source-port QoS priorities

```
switch(config)# interface e 1-3 qos priority 2
switch(config)# interface e 4 qos priority 3
switch(config)# show qos port-priority

  Port priorities

  Port Apply rule   | DSCP  Priority    Radius Override
  ---- ----------    ----  --------    ---------------
  1    Priority    |       2           No-override
  2    Priority    |       2           No-override
  3    Priority    |       2           No-override
  4    Priority    |       3           No-override
  5    No-override |       No-override No-override
  .    .                   .           .
  .    .                   .           .
```

If you then decided to remove port 1 from QoS prioritization:

Returning a QoS-prioritized VLAN to “No-override” status

In this instance, `No-override` indicates that port 1 is not prioritized by QoS.

```
switch(config)# no interface 1 qos
switch(config)# show qos port-priority

  Port priorities

  Port Apply rule   | DSCP  Priority    Radius Override
  ---- ----------    ----  --------    ---------------
  1    No-override |       No-override No-override
  2    Priority    |       2           No-override
  3    Priority    |       2           No-override
  4    Priority    |       3           No-override
  5    No-override |       No-override No-override
  .    .                   .           .
  .    .                   .           .
```

Assigning a DSCP policy based on the source-port

This option assigns a previously configured DSCP policy (codepoint and 802.1p priority) to outbound IP packets (received from the specified sourceports). That is, the switch:

Procedure

1. Selects an incoming IP packet on the basis of its source-port on the switch.
2. Overwrites the packet’s DSCP with the DSCP configured in the switch for such packets.
3. Assigns 802.1p priority configured in the switch for the new DSCP (see **Differentiated Services Codepoint (DSCP) mapping** on page 209).
4. Forwards the packet through the appropriate outbound port queue.

Steps for creating a policy based on source-port classifier:

NOTE: You can select one DSCP per source-port. Also, configuring a new DSCP for a source-port automatically overwrites (replaces) any previous DSCP or 802.1p priority configuration for that port.)

1. Identify the source-port classifier to which you want to assign a DSCP policy.
2. Determine the DSCP policy for packets having the selected source-port:
   a. Determine the DSCP you want to assign to the selected packets. (This codepoint will be used to overwrite the DSCP carried in packets received through the source-port from upstream devices.)
   b. Determine the 802.1p priority you want to assign to the DSCP.
3. Configure the DSCP policy by using `qos dscp-map` to configure the priority for each codepoint (see **Differentiated Services Codepoint (DSCP) mapping** on page 209 for more information).
4. Configure the switch to assign the DSCP policy to packets from the specified source-port.

Syntax:

```
interface <port-list> qos dscp <codepoint>
```

Assigns a DSCP policy to packets from the specified sourceports, and overwrites the DSCP in these packets with the assigned *<codepoint>* value. This policy includes an 802.1p priority and determines the packet’s queue in the outbound port to which it is sent. If the packet leaves the switch on a tagged port, it carries the 802.1p priority with it to the next downstream device.

(Default: `No-override`)

Syntax:

```
no interface <port-list> qos
```

Removes QoS classifier for the specified source-ports.

Syntax:

```
show qos port
```

Displays a listing of all source-port QoS classifiers currently in the running-config file.

For example, suppose that you wanted to assign this set of priorities that have been configured on the switch:

Source-port	DSCP	Priority
2	15	7
1, 3	16	5
4, 5	17	1

Assign the DSCP policies to the selected source-ports and display the result.

An example of the completed source-port DSCP-priority configuration is shown below

```
switch(config)# int 4,5
switch(eth-4,5)# qos dscp 17
switch(eth-4,5)# int 1,3
switch(eth-1,3)# qos dscp 16
switch(eth-1,3)# int 2
switch(eth-2)# qos dscp 15
switch(eth-2)# show qos port-priority

Port priorities

  Port Apply rule  | DSCP   Priority    Radius Override
  ---- ----------- + ------ ----------- ---------------
  1    DSCP        | 010000 5           No-override
  2    DSCP        | 001111 7           No-override
  3    DSCP        | 010000 5           No-override
  4    DSCP        | 010001 1           No-override
  5    DSCP        | 010001 1           No-override
  6    No-override |        No-override No-override
  7    No-override |        No-override No-override
  .         .         .          .           .
  .         .         .          .           .
```

RADIUS override field

During a client session authenticated by a RADIUS server, the server can impose a port priority that applies only to that client session. For more information, see the RADIUS chapter in the *ArubaOS-Switch Access Security Guide* for your switch.

Differentiated Services Codepoint (DSCP) mapping

A codepoint must have an 802.1p priority assignment (0 - 7) before you can configure a policy for prioritizing packets. If a codepoint you want to use shows `No-override` in the `Priority` column of the DSCP map (`show qos dscp-map`), then you must assign a 0 - 7 priority before proceeding (`qos dscp-map priority` command).

The DSCP Policy Table associates an 802.1p priority with a specific ToS byte codepoint in an IPv4 packet. This enables you to set a LAN policy that operates independently of 802.1Q VLAN-tagging.

A partial display of the default DSCP Policy Table is show in the table below.

You can use the following command to list the current DSCP Policy table.

Syntax:

```
show qos dscp-map
```

Displays the DSCP Policy Table.

Table 20: *Partial display from the default DSCP Policy Table*

DSCP CodePoint	DSCP Value	802.1p tag	DSCP Policy name
000000	0	0	cs0
000001	1	0	
000010	2	0	
000011	3	0	
000100	4	0	

Table Continued

DSCP CodePoint	DSCP Value	802.1p tag	DSCP Policy name
000101	5	0	
000110	6	0	
000111	7	0	
001000	8	1	cs1
001001	9	1	
001010	10	1	af11
001011	11	1	
001100	12	1	af12
001101	13	1	
001110	14	1	af13
001111	15	1	
010000	16	2	cs2
010001	17	2	
010010	18	2	af21
010011	19	2	

Default priority settings for selected codepoints

In a few cases, such as 001010 (af21) and 001100 (af43), a default policy (implied by the DSCP standards for Assured-Forwarding and Expedited-Forwarding) is used.

You can change the priorities for the default policies by using `qos dscp-map` *`<codepoint>`* `priority <0-7>` . (These policies are not in effect unless you have either applied the policies to a QoS classifier or configured QoS Type-of-Service to be in `diff-services` mode.)

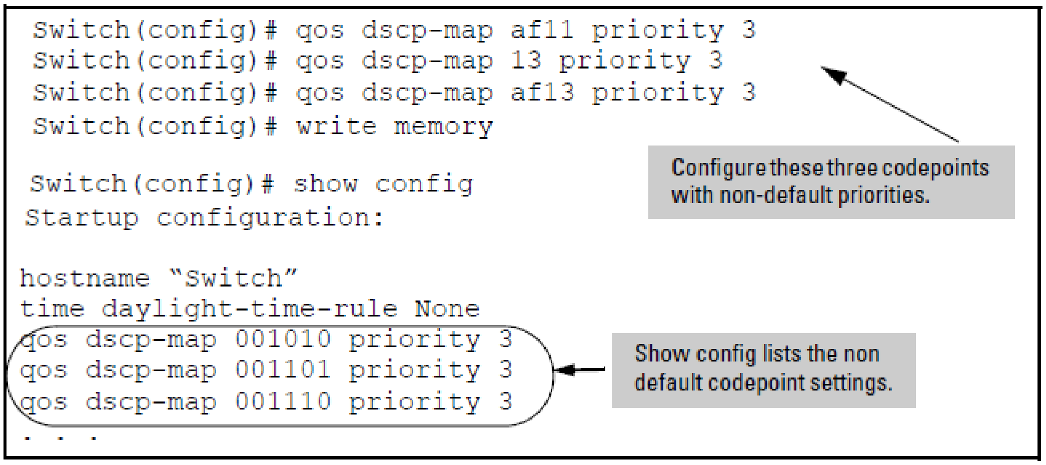
Quickly listing non-default codepoint settings

The DSCP Policy Table in **Differentiated Services Codepoint (DSCP) mapping**, lists the switch's default codepoint/priority settings. If you change the priority of any codepoint setting to a non-default value and then execute `write memory`, the switch will list the non-default setting in the `show config` display. For example, in the default configuration, the following codepoint settings are true:

Codepoint	Default priority
001100	1
001101	No-override
001110	2

If you change all three settings to a priority of 3, and then execute `write memory`, the switch will reflect these changes in the `show config` listing:

Figure 34: *Example of show config listing with non-default priority settings in the DSCP table*



```
Switch(config)# qos dscp-map af11 priority 3
Switch(config)# qos dscp-map 13 priority 3
Switch(config)# qos dscp-map af13 priority 3
Switch(config)# write memory

Switch(config)# show config
Startup configuration:

hostname "Switch"
time daylight-time-rule None
qos dscp-map 001010 priority 3
qos dscp-map 001101 priority 3
qos dscp-map 001110 priority 3
. . .
```

Configure these three codepoints with non-default priorities.

Show config lists the non default codepoint settings.

Effect of `No-override`: In the QoS Type-of-Service differentiated services mode, a `No-override` assignment for the codepoint of an outbound packet means that QoS is effectively disabled for such packets. That is, QoS does not affect the packet queuing priority or VLAN tagging.

In this case, the packets are handled as follows (as long as no other QoS feature creates priority assignments for them):

802.1Q status	Outbound 802.1p priority
Received and Forwarded on a tagged port member of a VLAN.	Unchanged
Received on an Untagged port member of a VLAN; Forwarded on a tagged port member of a VLAN.	0 (zero)—”normal”
Forwarded on an Untagged port member of a VLAN.	None

Note on changing a priority setting

If a QoS classifier is using a policy (codepoint and associated priority) in the DSCP Policy table, you must delete or change this usage before you can change the priority setting on the codepoint. Otherwise the switch blocks the change and displays this message:

```
Cannot modify DSCP Policy < codepoint > - in use by other
qos rules.
```

In this case, use `show qos <classifier>` to identify the specific classifiers using the policy you want to change; that is:

```
show qos device-priority

show qos port-priority

show qos tcp-udp-port-priority

show qos vlan-priority

show qos type-of-service
```

For example, suppose that the 000001 (dscp 1) codepoint has a priority of 6, and several classifiers use the 000001 codepoint to assign a priority to their respective types of traffic. If you wanted to change the priority of codepoint 000001, you would do the following:

1. Identify which QoS classifiers use the codepoint.
2. Change the classifier configurations by assigning them to a different DSCP policy, or to an 802.1p priority, or to `No-override`.
3. Reconfigure the desired priority for the 000001 (dscp 1) codepoint.
4. Either reassign the classifiers to the 000001 (dscp 1) codepoint policy or leave them as they were after step 2, above.

Changing the priority setting on a policy when one or more classifiers are currently using the policy (example)

Suppose that codepoint 1 is in use by one or more classifiers. If you try to change its priority, you see a result similar to the following:

Error message for changing the priority on a DSCP policy

```
switch(config)# qos dscp-map 1 priority 2
Cannot modify DSCP Policy 1 - in use by other qos rules.
```

In this case, you would use steps similar to the following to change the priority.

1. Identify which classifiers use the codepoint you want to change. The following example shows a search to identify classifiers using a codepoint you want to change.

```
switch(config)# show qos device-priority

 Device priorities

 Device Address Apply Rule | DSCP Priority
 -------------- ---------- - ---- -----------
 10.26.50.104   DSCP       | 1    6

switch(config)# show qos port-priority

 Port priorities

 Port Apply rule  | DSCP  Priority     Radius Override
 ---- ----------    ----- ----------   ---------------
 1    No-override |       No-override  No-override
 2    No-override |       No-override  No-override
 3    DSCP        | 1     6            No-override
 4    No-override |       No-override  No-override
 .
 .
```

```
.
switch(config)# show qos tcp-udp-port-priority

 TCP/UDP port based priorities

          | IP Packet  Application
 Protocol | Type       Port       Apply rule | DSCP    Priority
 -------- + ---------- ---------- ---------- + ------ ---------
 UDP      | IPv4       1260       DSCP       | 1       6
```

2. Change the classifier configurations by assigning them to a different DSCP policy, or to an 802.1p priority, or to `No-override`. For example:

   a. Delete the policy assignment for the `device-priority` classifier. (That is, assign it to `No-override`.)

   ```
   switch(config)# no qos device-priority 10.26.50.104
   ```

   b. Create a new DSCP policy to use for re-assigning the remaining classifiers.

   ```
   switch(config)# qos dscp-map 5 priority 6
   ```

   c. Assign the `port-priority` classifier to the new DSCP policy.

   ```
   switch(config)# int 3 qos dscp 5
   ```

   d. Assign the `udp-port 1260` classifier to an 802.1p priority.

   ```
   switch(config)# qos udp-port 1260 priority 2
   ```

3. Reconfigure the desired priority for the 000001 (dscp 1) codepoint.

   ```
   switch(config)# qos dscp-map 000001 priority 4
   ```

4. You could now re-assign the classifiers to the original policy codepoint or leave them as currently configured.

Traffic Policing

Perform traffic policing on the incoming packets as defined in "RFC 2698: Two rate three color marker (trCRM)" using a meter by specifying the commit and peak-rates. If the traffic exceeds the commit-rate, you can only change the priority of the packets by remarking (DSCP). For traffic exceeding peak-rate, you can either remark again or drop the packets.

Traffic policing is supported only on IPv4 and IPv6 traffic classes and not on MAC class.

Traffic rates

The **RFC-2698: A two rate three color marker**, provides you an option to categorize the incoming traffic based on the following two rates (in kilobits per second):

Commit Information Rate (CIR)

Specifies the bandwidth limit for guaranteed traffic. Once it exceeds the limit, you can only perform DSCP remarking (changing priority).

Peak Information Rate (PIR)

Specifies the bandwidth limit for peak traffic. Once it exceeds the limit, you can either remark the DSCP again or drop the incoming traffic.

You can configure the meter with the preceding traffic rates using the CLI. The show command output provides the administrator a statistical overview of the metered packets.

Traffic policy configuration

Use the `class` command to configure a traffic policy by specifying a meter with commit and peak-rates. Traffic policing helps the administrator to monitor the traffic flow from the configured value. Execute the `class` command from the user policy context. Set the user policy from the config context with the `policy user <policy-name>` command before executing the `class` command.

class

Syntax

```
[<SEQ-NUM>] class { ipv4 | ipv6 } <CLASS-NAME>
                             { action meter
                               commit-rate kbps <1-10000000> dscp-remark
                               peak-rate kbps <1-10000000> {drop | dscp-remark}}
no [<SEQ-NUM>] class { ipv4 | ipv6 } <CLASS-NAME>
                             { action meter
                               commit-rate kbps <1-10000000> dscp-remark
                               peak-rate kbps <1-10000000> {drop | dscp-remark}}
```

Description

Allows you to configure the traffic as per two rate three color policy specified in RFC 2698.

The `no` form of this command removes the metered traffic rate entries from the running configuration output.

Command context

`config`, `config-class`, and `policy-qos`.

Parameters

`commit-rate`

Configures the commit-rate as per RFC 2698.

`commit-rate kbps`*`<1-10000000>`*

Configures the commit-rate in Kb/s. Range: 1-10000000.

`dscp-remark`

Remarks the DSCP for traffic that exceeds the commit or peak-rate.

`peak-rate`

Configures the peak-rate as per RFC 2698.

`peak-rate kbps`*`<1-10000000>`*

Configures the peak-rate in Kb/s. Range: 1-10000000.

`drop`

Drops the traffic which exceeds the peak-rate.

Usage

If the traffic on the interface on which the policy is applied exceeds:

- For the commit-rate, its DSCP is remarked to the configured value.
- For the peak-rate, traffic is either dropped or DSCP remarked again.

Examples

```
switch(config)# class
ipv4                  Classify traffic based on IPv4 information.
ipv6                  Classify traffic based on IPv6 information.
mac                   Classify traffic based on Ethernet header information.
resequence            Renumber the entries in the class.
switch(config)# class ipv4 class1
switch(config-class)# 10
ignore                Create a rule to ignore specified packets.
match                 Create a rule to match specified packets.
remark                Add a comment to the class.
gre                   Match GRE packets.
esp                   Match ESP packets.
ah                    Match AH packets.
ospf                  Match OSPF packets.
pim                   Match PIM packets.
vrrp                  Match VRRP packets.
alias-src             Specify the netdestination to control incoming packets.
icmp                  Match ICMP packets.
igmp                  Match IGMP packets.
ip                    Match IP packets.
sctp                  Match SCTP packets.
tcp                   Match TCP packets.
udp                   Match UDP packets.
switch(config-class)# 10 match ip any any
ip-dscp               Match a specified IP DSCP value.
precedence            Match a specified IP precedence value.
tos                   Match a specified IP Type of Service value.
vlan                  Match a specified VLAN ID.
<cr>
switch(config-class)# 10 match ip any any
switch(config-class)#
switch(config)#
switch(config)# show run

Running configuration:

; JL557A Configuration Editor; Created on release #WC.16.06.0000x
; Ver #13:03.f8.1c.9b.3f.bf.bb.ef.7c.59.fc.6b.fb.9f.fc.ff.ff.37.ef:49
hostname "switch-name"
module 1 type jl557a
class ipv4 "class1"
     10 match ip 0.0.0.0 255.255.255.255 0.0.0.0 255.255.255.255
   exit
policy qos "qos-policy-1"
   exit
policy qos "qos-policy1"
   exit
interface 2
   speed-duplex auto-100
   exit
interface 48
   speed-duplex auto-100
   exit
snmp-server community "public" unrestricted
vlan 1
   name "DEFAULT_VLAN"
   untagged 1-52
   ip address dhcp-bootp
   ipv6 enable
   ipv6 address dhcp full
   exit
switch(policy-qos)# 10 class ipv4
ASCII-STR             Enter an ASCII string.
switch(policy-qos)# 10 class ipv4 class1 action
    dscp                      Specify an IP DSCP.
 ip-precedence         Specify the IP precedence.
 meter                 Configure a two-rate-three-color-marker as specified in
                       RFC 2698.
 priority              Specify the priority.
 rate-limit            Configure rate limiting for all traffic.
switch(policy-qos)# 10 class ipv4 class1 action meter
switch(policy-qos)# 10 class ipv4 class1 action meter commit-rate kb
switch(policy-qos)# 10 class ipv4 class1 action meter commit-rate kbps
<1-10000000>          Enter an integer number.
switch(policy-qos)# 10 class ipv4 class1 action meter commit-rate kbps 1000

dscp-remark           Remark the DSCP for traffic which exceeds the commit-rate.
switch(policy-qos)# 10 class ipv4 class1 action meter commit-rate kbps 1000

dscp-remark           Remark the DSCP for traffic which exceeds the commit-rate.
switch(policy-qos)# 10 class ipv4 class1 action meter commit-rate kbps 1000 dscp
switch(policy-qos)# 10 class ipv4 class1 action meter commit-rate kbps 1000 dscp-remark
af11                  Match DSCP AF11 (001010)
af12                  Match DSCP AF12 (001100)
af22                  Match DSCP AF22 (010100)
af23                  Match DSCP AF23 (010110)
```

```
af42                  Match DSCP AF42 (100100)
af43                  Match DSCP AF43 (100110)
cs5                   Match DSCP CS4 (101000)
cs6                   Match DSCP CS6 (110000)
ef                    Match DSCP EF (101110)
switch(policy-qos)# 10 class ipv4 class1 action meter commit-rate kbps 1000 dscp-remark cs7

switch(policy-qos)# 10 class ipv4 class1 action meter commit-rate kbps 1000 dscp-remark cs7 peak-rate
dscp-remark           Remark the DSCP for traffic which exceeds the peak-rate.
switch(policy-qos)# 10 class ipv4 class1 action meter commit-rate kbps 1000 dscp-remark cs7 peak-rate kbps 1000 drop
The commit rate must be less than the peak rate.
switch# 10 class ipv4 class1 action meter commit-rate kbps 1000 dscp-remark cs7 peak-rate kbps 2000

drop                  Drop the traffic which exceeds the peak-rate.
dscp-remark           Remark the DSCP for traffic which exceeds the peak-rate.
switch(policy-qos)# 10 class ipv4 class1 action meter commit-rate kbps 1000 dscp-remark cs7 peak-rate kbps 2000 drop

switch(config)#
switch(config)# show run

; JL557A Configuration Editor; Created on release #WC.16.06.0000x
; Ver #13:03.f8.1c.9b.3f.bf.bb.ef.7c.59.fc.6b.fb.9f.fc.ff.ff.37.ef:49
hostname "switch-name"
module 1 type jl557a
class ipv4 "class1"
     10 match ip 0.0.0.0 255.255.255.255 0.0.0.0 255.255.255.255
   exit
policy qos "qos-policy-1"
     10 class ipv4 "class1" action meter commit-rate kbps 1000 dscp-remark cs7
peak-rate kbps 2000 drop
   exit
policy qos "qos-policy1"
   exit
interface 2
   speed-duplex auto-100
   exit
interface 48
   speed-duplex auto-100
   exit
snmp-server community "public" unrestricted
```

show statistics

Syntax

`show statistics policy` *`POLICY-NAME-STR`* `port` *`PORT-NUM`*

Description

Displays the statistics for a configured policy applied at a port.

Command context

`config`

Parameters

`POLICY-NAME-STR`

To view the statistics, specify the policy name.

`port` ***`PORT-NUM`***

To view the port statistics, specify either inward or outward bound port number. Currently, traffic policing is supported only for inward direction.

Usage

`show statistics policy qos-policy-1 port 1 in help`

For QoS policies with the action meter, the interpretation of the statistics is as follows:

Total

Total number of packets or bytes that matches with the class statement.

Total - Exceeded CIR

Number of packets or bytes that are not metered (which are in the green band as per RFC 2698).

Exceeded CIR - Exceeded PIR

Number of packets or bytes that are remarked due to violation of CIR (which are in the yellow band as per RFC 2698).

Exceeded PIR

Number of packets or bytes that are dropped or remarked due to violation of PIR (which are in the red band as per RFC 2698).

Example

```
switch(config)# interface 1 service-policy qos-policy-1
in                    Apply the policy to inbound packets on the port.
out                   Apply the policy to outbound packets on the port.
switch(config)# interface 1 service-policy qos-policy-1 out
The QoS policy with the action 'meter' cannot be applied to a port in the
outbound direction.
switch(config)# interface 1 service-policy qos-policy-1
in
switch(config)# show statistics policy qos-policy-1
port                  The port to show statistics for.
vlan                  The VLAN to show statistics for.
switch(config)# show statistics policy qos-policy-1 port
[ethernet] PORT-NUM   Enter a port name.
switch(config)# show statistics policy qos-policy-1 port
1
switch(config)# show statistics policy qos-policy-1 port 1 in
Hit Counts for Policy qos-policy-1 in the last 39 seconds

  Total

      10 class ipv4 "class1" action meter commit-rate kbps 1000 dscp-remark cs7
peak-rate kbps 2000 drop

(     713 )      10 match ip 0.0.0.0 255.255.255.255 0.0.0.0 255.255.255.255
Exceeded PIR |                    0 |                    0


switch(config)# show statistics policy qos-policy-1
switch(config)# interface 1 service-policy qos-policy-1 in help
Usage: [no] interface [ethernet] <PORT-LIST> [...]
       [no] interface loopback <Num> [...]
       [no] interface tunnel <Num> [...]

Description: Enter the Interface Configuration Level, or execute one
             command for that level. With no optional parameters
             specified, the "interface" command enters the
             Interface Configuration Context Level for execution of
             multiple configuration changes to the same interface.

             "interface ?" will show a list of all valid commands.

switch(config)# interface 1 service-policy q
switch(config)# interface 1 service-policy qos-policy-1 in help
switch(config)# show statistics policy qos-policy-1 port 1 in help
Usage: show statistics policy <POLICY-ID> port <PORT> {in|out}
       show statistics policy <POLICY-ID> vlan <VLAN-ID> {in|out}
```

Scenarios

Scenario 1

The commit-rate is greater or equal to peak rate, an error is displayed when the commit-rate exceeds the peak-rate value.

```
switch(policy-qos)# 10 class ipv4 class1 action meter commit-rate kbps 1000 dscp-remark cs7 peak-rate kbps 1000 drop
The commit rate must be less than the peak rate.
```

Scenario 2

If you apply a policy action meter on a MAC class, an error "Only IPv4 and IPv6 classes are supported" appears.

Scenario 3

If the commit-rate and peak-rate are not in the range <1 to 10000000>, an error that the rates are not within the configured values appears.

Scenario 4

If you apply a policy with action meter on an interface with 'out' direction, an error "QoS policy with action meter cannot be applied for a port with outbound direction" appears.

```
switch(config)# interface 1 service-policy qos-policy-1
in                    Apply the policy to inbound packets on the port.
out                   Apply the policy to outbound packets on the port.
switch(config)# interface 1 service-policy qos-policy-1
out
The QoS policy with the action 'meter' cannot be applied to a port in the
outbound direction.
```

Scenario 5

Meter classifier action is supported only on Ethernet ports. An error "QoS policy with classifier action meter is applicable only on Ethernet ports" appears.

```
switch(vlan-1)# service-policy test in
QoS policy with classifier action meter is applicable only on Ethernet ports.
```

Scenario 6

When you configure a class with action meter for a policy applied on trunk or VLAN, an error "Class with meter action cannot be configured for the policy applied on trunk or VLAN" appears.

Scenario 7

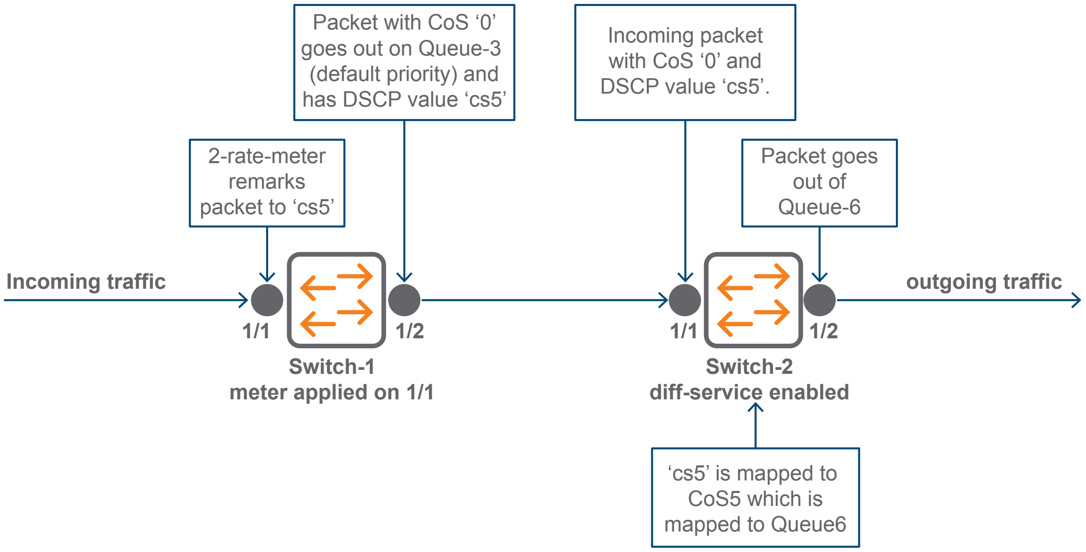
Remarking of 802.1q priority (CoS) field with the corresponding DSCP value is not supported. If attempted, the CoS field of the packet will be set to zero and only the DSCP field of the packet is remarked. The following behavior is noted due to this restriction:

Meter has a higher precedence over ACLs

Occurs if you apply CoS through QoS ACL on the same port as two rate meter. The CoS value of the packet is set to zero as CIR/PIR DSCP are applied through a meter.

Packets that are remarked by meter are always egressed out of the default-priority queue

Occurs if the egress queue is selected based on the CoS value and ASIC limitation.


Packet with CoS ‘0’ goes out on Queue-3 (default priority) and has DSCP value ‘cs5’
Incoming packet with CoS ‘0’ and DSCP value ‘cs5’.
2-rate-meter remarks packet to ‘cs5’
Packet goes out of Queue-6
Incoming traffic
outgoing traffic
1/1
1/2
1/1
1/2
Switch-1
meter applied on 1/1
Switch-2
diff-service enabled
‘cs5’ is mapped to CoS5 which is mapped to Queue6

In the preceding illustration, the packets that are remarked by the meter are always egressed out of the default-priority queue, Q3. The egress queue is selected based on CoS value and the ASIC limitation. The CoS packet is marked as zero and the queue corresponding to CoS value of zero is selected.

Metered outcome is undetermined if multiple meters are applied on the same port

Occurs when you meter the traffic on a port using a QoS policy, and apply a new meter on the same port for that class of traffic using a user-policy. Due to ASIC limitation, the outcome of the metered rate is not determined.

The following example shows difference in rate limit when a QoS policy and user policy is applied on the same port:

```
‘policy-qos’ is a QoS policy
‘policy-user’ is an user policy
‘c1’ is a class which matches all IP traffic

switch(config)# show policy policy-qos

Statements for policy "policy-qos"
policy qos "policy-qos"
     10 class ipv4 "c1" action rate-limit kbps 2000
   exit

switch(config)# show policy policy-user

Statements for policy "policy-user"
policy user "policy-user"
     10 class ipv4 "c1" action rate-limit kbps 1000
   exit

switch(config)# show class ipv4 c1

Statements for class IPv4 "c1"
class ipv4 "c1"
     10 match ip 0.0.0.0 255.255.255.255 0.0.0.0 255.255.255.255
   exit

The QoS Policy, policy-qos is applied to the interface
```

```
switch(config)# show run int 1

Running configuration:

interface 1
   service-policy "policy-qos" in
   untagged vlan 10
   aaa port-access mac-based
   exit

In the preceding example, we observe that the rate at which traffic
is limited is as per the configuration (2000 kbps).
We also observe that port 1 is enabled with mac-based authentication.
As part of mac-based authentication, user-policy is applied to the client.

The following show command displays the client details.


A mac-auth client is applied on the same port with user policy, policy-user
switch(config)# show port-access client detail

 Port Access Client Status Detail

  Client Base Details :
   Port             : 1                      Authentication Type : mac-based
   Client Status    : authenticated          Session Time        : 145 seconds
   Client Name      : 00000087b9fe           Session Timeout     : 0 seconds
   MAC Address      : 000000-87b9fe
   IP               : n/a

 User Role Information

   Name                              : authRole
   Type                              : local
   Reauthentication Period (seconds) : 0
   Untagged VLAN                     : 10
   Tagged VLANs                      :
   Captive Portal Profile            :
   Policy                            : policy-user

Statements for policy "policy-user"
policy user "policy-user"
     10 class ipv4 "c1" action rate-limit kbps 1000
   exit


Statements for class IPv4 "c1"
class ipv4 "c1"
     10 match ip 0.0.0.0 255.255.255.255 0.0.0.0 255.255.255.255
   exit

   Tunnelednode Server Redirect      : Disabled
   Secondary Role Name               :

 After the port is applied with the meter specified in user policy, policy-user (1000 kbps),
 the rate of traffic varies between 0 kbps and 1000 kbps.
```

You can overcome the preceding restrictions using the following work-around:

- Stop the incoming traffic or the lower the rate to less than 100 kbps for a couple of seconds after you apply the second meter on the port.
- Before the traffic flow, apply the meters on the interface.

Restrictions

Traffic policing comes with the following restrictions:

- Does not support MAC classes.
- Cannot configure burst size even though RFC 2698 allows you to specify committed burst size and peak burst size. Incorrect burst sizes can either lead to excessive traffic loss, or poor rate-limiting thus reducing the performance.
- Cannot configure rates in packets-per-second.
- Exceeded commit-rate packets are only DSCP remarked.
- Operates only in color blind mode.
- Applicable only for QoS policies and not PBR or mirror policies.
- Cannot configure using Next Gen WEBUI or switch menu.
- QoS policy containing a two rate meter can only be applied on individual physical interfaces, and not on logical interfaces (VLANs or LAG).
- The Egress ACLs do not support DSCP remarking. As DSCP remarking is the only supported action for commit-rate violation, traffic policing cannot be enabled on an outward interface.
- If you apply CoS through QoS ACL on the same port as two rate meter, the meter has a higher precedence over ACLs. The CoS value of the packet is set to zero as CIR/PIR DSCP is applied through a meter.

IP Multicast (IGMP) interaction with QoS

IGMP high-priority-forward causes the switch to service the subscribed IP multicast group traffic at high priority, even if QoS on the switch has relegated the traffic to a lower priority. This does not affect any QoS priority settings, so the QoS priority is honored by downstream devices. However, QoS does take precedence over IGMP normal-priority traffic.

The switch's ability to prioritize IGMP traffic for either a normal or high priority outbound queue overrides any QoS criteria, and does not affect any 802.1p priority settings the switch may assign. For a given packet, if both IGMP high priority and QoS are configured, the QoS classification occurs and the switch marks the packet for downstream devices, but the packet is serviced by the high-priority queue when leaving the switch.

IGMP high priority	QoS configuration affects packet	Switch port output queue	Outbound 802.1p setting (requires tagged VLAN)
Not Enabled	Yes	Determined by QoS	Determined by QoS
Enabled	See above paragraph.	High	As determined by QoS if QoS is active.

QoS messages in the CLI

Message	Meaning
DSCP Policy < **decimal-codepoint** > not configured	You have attempted to map a QoS classifier to a codepoint for which there is no configured priority (`No-override`). Use the `qos dscp-map` command to configure a priority for the codepoint, then map the classifier to the codepoint.
Cannot modify DSCP Policy < **codepoint** > - in use by other qos rules.	You have attempted to map a QoS classifier to a codepoint that is already in use by other QoS classifiers. Before remapping the codepoint to a new priority, you must reconfigure the other QoS classifiers so that they do not use this codepoint. You can have multiple QoS classifiers use this same codepoint as long as it is acceptable for all such classifiers to use the same priority.

Configuring traffic templates

In order to define mappings of 802.1p priorities to queues, traffic class groups (traffic groups) are defined as part of a traffic template. A traffic group consists of a set of 802.1p priorities that are mapped to the same egress queue. A traffic template consists of a set of traffic groups that cover all priorities 0- 7. The number of traffic groups allowed within a traffic template is equal to the number of queues configured on a platform, although a queue may have no priorities mapped to it.

For example, if you want to configure a switch with a priority to queue mapping that matches a connected networking device's queue configuration, and the neighbor device has 3 queues configured, all priorities should be mapped to the 3 highest priority queues on the switch in a manner that matches the configuration of the neighbor device.

Once defined, the traffic template is then assigned in its entirety to all ports of the switch.

Two predefined traffic templates are provided that correspond to the IEEE 802.1p traffic group specification (default-tcgt) and the 802.1q update (dot1q.tcgt). These predefined templates may be applied as is, or they may be used as the basis for defining a custom template.

Displaying traffic template information

To display a summary of all traffic templates, enter the `show qos traffic-template` command.

List of the currently defined traffic templates and their status

```
switch(config)# show qos traffic-template

System default template: default-tcgt

Template Name                        Status      Queues
-------------                        ------      ------
example                              Valid          4
dot1q-tcgt (predefined)              Valid          4
default-tcgt (predefined)            Active         4
```

To display detailed information about a single traffic template, enter the `show qos traffic-template <template-name>` command.

List of the currently defined traffic templates and their status

```
switch(config)# show qos traffic-template default-tcgt

  Template Name: default-tcgt (predefined)
  Status       : Active

Queue-no  Priorities     Name
--------  -------------  --------------------
1         1,2            background-tcg
2         0,3            best-effort-tcg
3         4,5            controlled-load-tcg
4         6,7            control-tcg
```

Note: The Name column shows the descriptive names assigned to the traffic group to help identify their purpose.

Creating a traffic template

A traffic template can be created, modified, or deleted by entering this command in the global configuration context.

Syntax:

```
qos traffic-template <template-name>
```

```
no qos traffic-template <template-name>
```

Enter the Traffic Template Configuration level or execute one command for that level. With no additional parameters specified, the `qos traffic-template` command changes the context to Traffic Template Configuration level for execution of configuration changes to the named template.

If *<template-name>* does not exist already, the template will be created as a copy of default-tcgt and the traffic template context is entered. The maximum length is 40 characters.

If *<template-name>* already exists, the traffic template context for that template is entered and you can make modifications to the template.

```
qos traffic-template <template-name> <copy-from-name>
```

When executed, a new template is created with the name *<template-name>* as a copy of the existing template named *<copy-from-name>*.

NOTE: You cannot modify any predefined traffic templates.

```
no qos traffic-template <template-name>
```

Deletes the specified traffic template. Predefined templates and the currently active template may not be deleted.

```
show qos traffic-template [<template-name>]
```

When entered without the name of a specific traffic template, displays a list of the current traffic templates and their status - active, valid, or invalid.

When executed with the name of a specific traffic template, detailed information for that traffic template is displayed.

After executing the `qos traffic-template <template-name>` command in the global configuration context, you are in traffic template context and can begin modifying a newly created template or an existing template.

Creating a new traffic template and entering traffic template configuration context

```
switch(config)# qos traffic-template example

switch(cfg-tcgt-example[1])# show qos traffic-template example

 Template Name: example
 Status : Valid

Queue-no  Priorities     Name
--------  -------------  --------------------
1         1,2            background-tcg
2         0,3            best-effort-tcg
3         4,5            controlled-load-tcg
4         6,7            control-tcg
```

[1] Traffic template context

You should always check the traffic template status using the `show qos traffic-template` command to make sure that the template is valid. Invalid templates cannot be applied to ports. A traffic template will be invalid if the priorities are not mapped to an egress queue. This can happen, for example, if you delete a priority from a queue and do not reassign it to another queue.

If a template has been applied, it is considered to be in use and cannot be deleted. The `show qos queue-config` command displays the currently active traffic template.

Displaying the currently active traffic template

```
switch(config)# show qos queue-config

 Egress Queue Configuration

 Number of Queues: 8
 Traffic Template: default-tcgt

        802.1p
 Queue Priority
 ----- --------
 1       1
 2       2
 3       0
 4       3
 5       4
 6       5
 7       6
 8       7
```

Configuring traffic groups within a traffic template

When in the traffic template context, you can modify traffic groups within that template by changing which priorities are assigned to what queues, as well as assigning a name to each group.

Syntax:

```
map-traffic-group <queue-num> priority <priority> | [name <namestring>]
```

```
no map-traffic-group <queue-num> priority <priority> | [name <namestring>]
```

Allows configuration of traffic groups with a traffic template.

The `no` form of the command removes a priority from the currently mapped queue.

`<queue-num>`: Ranges from 1 to the number of active queues, which can vary from 1 to 8 queues. The number of queues is configured with the `qos queue-config` command.

`priority <0-7>`: Specifies an 802.1p priority to assign this queue. This command may be repeated to assign multiple priorities to the same queue.

`name` *`<namestring>`*: Assigns a documentary label to the traffic group. Maximum length is 40 characters.

The `no` form of the command removes the descriptive name from the group.

Moving a priority from one traffic group to another

Typically, when modifying a traffic template, priorities are reassigned to different queues. When moving a priority from one traffic group to another, you must first delete that priority from the traffic group it is currently mapped to (`no map-traffic-group` *`<queue-num>`* `priority` *`<priority>`* and then add it the desired traffic group.

NOTE: Before changing the traffic groups' priority mappings from the defaults, it is important to examine the current policies that act on 802.1p priorities to ensure that the policies best serve the behavior desired on the network, including DSCP, VLAN, interface, or protocol rules.

Removing priorities from a traffic group

```
switch(cfg-tcgt-example)# no map-traffic-group 2 priority 0
```

After removing the priority from the currently mapped queue, the template becomes invalid because priority 0 is no longer mapped to any queue.

Mapping the priority to a new queue results in a valid traffic template again.

Invalid traffic template because a priority is unmapped

```
switch(cfg-tcgt-example)# show qos traffic-template example

TRAFFIC-TEMPLATE: example
Status          : Invalid

Queue-no  Priorities     Name
--------  -------------  --------------------
1         1,2            background-tcg
2         3              best-effort-tcg
3         4,5            controlled-load-tcg
4         6,7            control-tcg
```

Valid traffic template with remapped queue

```
switch(cfg-tcgt-example)# map-traffic-group 1 priority 0

switch(cfg-tcgt-example)# show qos traffic-template example

Template Name: example
Status       : Valid

Queue     Priorities     Name
```

```
--------  -------------  --------------------
1         1,2            background-tcg
2         3,0            best-effort-tcg
3         4,5            controlled-load-tcg
4         6,7            control-tcg
```

After modifying a traffic template, you must apply it to the switch to activate the new mapping. See **Applying a traffic template** on page 226.

Applying a traffic template

After creating a traffic template with the desired queue assignments, you must apply it. The same traffic templates is applied to the all ports on the switch. A reboot is required for the new template to take effect.

Syntax:

```
qos queue-config traffic-template <template-name>
```

Applies the specified traffic template to all the ports and reboots the switch.

Applying a traffic template to a switch

```
switch(config)# qos queue-config traffic-template example
 This command will modify the current running configuration,
 execute 'write memory' to replace the startup configuration,
 and then reboot.

Egress queues will be configured as follows:
 4-queues
 Queue servicing: weighted-round-robin
 simple-red disabled
 Traffic template: example

Do you want to save current configuration [y/n/^C]? y
Device will be rebooted, do you want to continue [y/n]? y
```

Port QoS Trust Mode

The Port QoS Trust feature restricts which packet QoS information may be used to determine inbound queue servicing and any priority information to be permitted into the local hop.

Port QoS Trust Mode configuration allows preservation or removal of the inbound QoS priorities carried in Layer 2 (the VLAN cos or Priority CodePoint (PCP) value, known as the 802.1p priority tag) and/or in Layer 3 (the IP-ToS byte, in IP-Precedence or IP-Diffserv mode). The different modes let the customer trust all, some, or no packet priority fields.

The per-port configuration enables the customer to trust some sources or devices and not others. This feature is mutually exclusive with any active port-priority configuration.

Configuration commands

qos trust

Syntax

```
qos trust [default|dot1p|dscp|ip-prec|none|device [none|<DEVICE-TYPE>]]
```

Description

Set the QoS Trust Mode configuration for the port.

Parameters

default

Trust 802.1p priority and preserve DSCP or IP-ToS.

device *<DEVICE-TYPE>*

On approved devices, trust IP-ToS Differentiated-Services in IP packets, and use the DSCP-MAP to remark the 802.1p priority. If the DSCP codepoint does not have an associated priority, the priority will be remarked to 0. On unapproved devices, trust 802.1p priority and preserve any IP- ToS values.

dot1p

Trust 802.1p priority and preserve DSCP or IP-ToS.

dscp

Trust IP-ToS Differentiated-Services in IP packets, and use the DSCP-MAP to remark the 802.1p priority. If the DSCP codepoint does not have an associated 802.1p priority, the priority will be remarked to 0.

ip-precedence

Trust IP-ToS IP-Precedence mode in IP packets and remark the 802.1p priority.

none

Do not trust either the 802.1p priority or the IP-ToS values.

QoS trust devices

aruba-ap

Aruba Access point device.

none

Clear all trusted devices from port.

NOTE:

Both SNMP and the CLI will verify that the current QoS Port Priority and desired QoS Trust Mode configuration are not mutually exclusive (and conversely).

qos dscp-map

Syntax

```
qos dscp-map <CODEPOINT> priority <PRIORITY> [name <NAME> | default | legacy]
```

Description

Modifies DSCP mapping.

Parameters

default

Returns switch to the fully mapped factory-default configuration.

legacy

Restore the legacy default behavior (partial mapping) used in earlier code releases.

Show commands

show qos trust

Syntax

```
show qos trust [device] <PORT>
```

Description

Shows port-based QoS trust configuration

Parameters

device

Show list of trusted devices per-port.

<port>

Show trusted devices on a single port.

Usage

```
show qos trust [device | [ethernet <PORT-LIST> ]
```

show qos trust

```
switch# show qos trust

Port-based qos Trust Configuration

  Port    Trust Mode   | Device Trust State ---- --- ----
  A1      Default      |
  A2      Default      |
  A3      Device**     | Trusted
  A4      IP-Prec      |
  A5      Dot1p        |
  A5      None         |
  A5      DSCP         |
  A5      Device**     |
  A5      Dot1p        |

** For a list of trusted devices per-port, use the command show qos trust device.
To show trusted devices on a single port, use the command show qos trust device <PORT>.
```

show qos trust device

```
switch# show qos trust device

Port-Based QoS Trust Configuration

  Port    Trusted Devices
  ------- ---------------
  A1      aruba-ap
  A2      aruba-ap
  A4      aruba-ap
```

show qos trust device <PORT>

```
switch# show qos trust device <PORT>

Port A4 QoS Trust Configuration
   Current state: Trusted

   Trusted Devices: aruba-ap
```

QoS queue configuration

QoS queue configuration allows you to reduce the number of outbound queues that all switch ports will use to buffer packets for 802.1p user priorities. By default, there are eight priority queues or traffic classes. Using this feature, you can reconfigure the switch to an eight-queue mode, four-queue mode or two-queue mode to increase the available bandwidth per queue.

Use the following commands to change the number of queues per port and display the current priority queue configuration on the switch.

Syntax:

```
qos queue-config < 2-queues | 4-queues | 8-queues>
```

Configures the number of outbound priority queues for all ports on the switch using one of the following options: 2-queues, 4-queues, or 8-queues.

(Default: 8-queues)

CAUTION: This command will execute a `write memory` followed by an immediate reboot, replacing the Startup configuration with the content of the current Running configuration.

The new configuration will:

- Remove any previously configured bandwidth-min output settings
- Set the new number of outbound port queues

If you select anything but ‘yes’ for this operation, the operation is aborted and a message stating `Operation aborted` appears.

Syntax:

```
show qos queue-config
```

Displays the current qos queue configuration.

Mapping of outbound port queues

This table shows the mapping of 802.1p priorities to outbound port queues:

Table 21: *Mapping 802.1p priorities to outbound port queues*

802.1p priority	8 Queues (default)	4 Queues	2 Queues
1 (lowest)	1	1	1
2	2		
0 (normal)	3	2	
3	4		
4	5	3	2
5	6		
6	7	4	
7 (highest)	8		

Configuring the number of priority queues

To change the number of outbound priority queues for all ports on the switch, use the `qos queue-config` command.

CAUTION: The `qos queue-config` command executes a `write memory` followed by an immediate reboot, replacing the Startup configuration with the contents of the current Running configuration.

Example:

To change the number of outbound priority queues for all ports on the switch from four queues to two:

- Specify the number of outbound priority queues to be configured using the `qos queue-config` command.

  ```
  switch(config)# qos queue-config 2-queues
  ```

  A caution message is displayed (see the Caution note above) concluding with the following prompt:

  ```
  This command will modify the current running configuration,
   execute 'write memory' to replace the startup configuration,
   and then reboot.

   Egress queues will be configured as follows:
    Number of Queues : 2

  Do you want to save the current configuration (y/n)? y
  ```

- Type **y** to continue.
- A second confirmation prompt appears:

```
This will reboot the system.

Continue (y/n)?
```

- Type **Yes** to initiate a write memory followed by an immediate reboot. (If you enter **Cancel** at either of the two prompts, the command is aborted and the current queue configuration is maintained on the switch).
- The changes will be committed to the startup configuration and the switch will reboot automatically with the new priority queue changes in effect.

Viewing the QoS queue configuration

Syntax:

```
show qos queue-config
```

Displays the current priority queue configuration per queue.

Viewing QoS queue configuration

```
switch# show qos queue-config

 Outbound Port Queue Configuration

        802.1p
 Queue  Priority
 ------ --------
   1      0-3
   2      4-7
```

QoS port egress-queue drop counters

Port egress-queue drop counters help customers debug network behavior and performance issues.

Egress-queue drop counters can be used to detect congestion on outbound ports, and help troubleshoot the network.

Syntax:

```
show interfaces queues... | config | custom ... | display | port-utilization |
transceiver ... | status ... | tunnel ... | ethernet PORT-LIST
```

Show port configuration and status information.

show interfaces ...

```
switch(vlan-2)#show interface queues 1
Status and Counters - Port Counters for port 1

Name  :
MAC Address      : b05ada-96e0df
Link Status      : Up
Port Enabled     : Yes
Port Totals (Since boot or last clear) :

Rx Packets      : 7                    Tx Packets      : 28
Rx Bytes        : 448                  Tx Bytes        : 6,132
Rx Drop Packets : 0                    Tx Drop Packets : 0
Rx Drop Bytes   : 0                    Tx Drop Bytes   : 0

Egress Queue Totals (Since boot or last clear) :
   Tx Packets      Dropped Packets  Tx Bytes         Dropped Bytes
Q1 8               0                1,126
```

```
0
Q2 0               0                0
0
                                                                             0
Q3 2               0                136             0
Q4 0               0                0               0
Q5 0               0                0               0
Q6 0               0                0               0
Q7 0               0                0               0
Q8 18              0                4,870           0
```

QoS operating notes and restrictions

- **All switches:** For explicit QoS support of IP subnets, Hewlett Packard Enterprise recommends forcing IP subnets onto separate VLANs and then configuring VLAN-based classifiers for those VLANs.
- **For devices that do not support 802.1Q VLAN-tagged ports:** For communication between these devices and the switch, connect the device to a switch port configured as `Untagged` for the VLAN in which you want the device's traffic to move.
- **Port tagging rules:** For a port on the switch to be a member of a VLAN, the port must be configured as either `Tagged` or `Untagged` for that VLAN. A port can be an untagged member of only one VLAN of a given protocol type. Otherwise, the switch cannot determine which VLAN should receive untagged traffic.
- **Not supported:** Use of an inbound 802.1p packet priority as a classifier for remapping a packet's outbound priority to different 802.1p priority. For example, where inbound packets carry an 802.1p priority of 1, QoS cannot be configured use this priority as a classifier for changing the outbound priority to 0.
- **Monitoring shared resources:**The QoS feature shares internal switch resources with several other features. For information on determining the current resource availability and usage, see “Monitoring Resources” in the *Management and Configuration Guide* for your switch.

Introduction to Stack Management

Stacking feature is available on Aruba 2930M switches. Only ring or chain topology is supported on these switches. For information on stacking topologies, see the *Installation and Getting Started Guide* corresponding to your switch.

For more information about the supported power supplies on 2930M switches, see the *Power over Ethernet (PoE/PoE+) Planning and Implementation Guide* for your switch.

You cannot stack a 2930M with a 2920 switch. You can only stack similar switches (For example, a 2930M with another 2930M).

NOTE: This feature is different from the stacking feature that is implemented on some other Networking switches. The other feature is implemented via the front-panel networking cables and it does not have the high bandwidth and redundancy features of the 2930M stacking.

The stacking feature for the 2930M switches allows you to connect up to 10 switches and have them act as a single high-bandwidth switch for both data and management.

One switch in the stack is designated as “Commander” and one switch is elected to be the “Standby”. The other switches are designated “Member”. The Commander is responsible for the overall management of the stack. The Standby provides redundancy for the stack and takes over stack management operations if the Commander fails, or if a Commander failover is forced by an administrator. The Members are not part of the overall stack management, however, they must manage their local subsystems and ports to operate correctly as part of the stack. The Commander and Standby are also responsible for their own local subsystems and ports.

Switch Stack Management (stacking) enables you to use a single IP address and standard network cabling to manage a group of up to 10 total switches in the same IP subnet (broadcast domain). Using stacking for these switches enables you to:

- Simplify management of small workgroups or wiring closets while scaling your network to handle increased bandwidth demand.
- Add switches to your network without having to first perform IP addressing tasks.
- Reduce the number of IP addresses needed in your network.
- Reduce downtime with high availability in the event of a failure.

NOTE: In the default configuration, stacking is enabled on these switches. However, if a 2930M switch is powered on and it does not have a Stacking Module installed, stacking is disabled. If a Stacking Module is subsequently installed in the switch, stacking must be enabled from the switch CLI (in the configuration context) by entering the following command:

```
switch(config)# stacking enable
```

Configuring a stack

Creating a stack

Ways to create a stack include:

- Deterministic method: By the sequence in which the switches are booted. You choose which member becomes Commander.
- Plug-and-go method: Ensure that stacking is enabled on all the switches, and then connect them together in the desired stacking topology. The plug-and-go method lets stacking decide which member is the Commander.

Using a deterministic method

1. Install a Stacking Module into a 2930M switch and then boot the switch. Follow the procedure described in the *Switch Installation and Getting Started Guide* corresponding to your switch.

   NOTE: You cannot use a 2920 stacking module on a 2930M switch and vice versa.

2. Make sure that stacking is enabled for the switch:

   a. Enter the `show stacking` command.

   b. If stacking is disabled, enter `stacking enable` (in global config context). This command causes the switch to reboot.

3. When the switch finishes booting, enter the `show stacking` command again. The switch now has the status of Commander. It has a Member ID of 1 (one) and a default priority of 128. An example of the `show stacking` output is shown below.

```
switch(config)# show stacking
Stack ID         : NO ID - will merge upon connectivity

MAC Address      : e0071b-e641ca
Stack Topology   : No Stack Formed
Stack Status     : No Stack Formed
Split Policy     : One-Fragment-Up
Uptime           : 0d 0h 12m
Software Version : WC.16.05.0000x

 Mbr
 ID  Mac Address       Model                                 Pri Status
 --- ----------------- ------------------------------------- --- ---------
*1   e0071b-e641c0     Aruba JL321A 2930M-48G Switch         128 Commander
```

4. To generate a stack ID, enter the following command:

```
switch(config)#stacking set-stack
```

5. (Optional) To have this switch retain its function as Commander through stack boots and other situations, you can increase its priority. The switch with the highest priority becomes Commander when all the switches are booted simultaneously. The default priority is 128. The priority can be set to any value between 1 and 255. To increase the switch's stacking priority, enter the following command:

```
switch(config)# stacking member 1 priority 255
```

6. (Optional) Preconfigure (provision) the stack for the other switches that become members of the stack. You can assign a member number and a priority by entering the following command for each switch:

```
switch(config)# stacking member N type JxxxxA [mac MAC-Addr]
```

   where:

- `N` is the stacking member number for the switch.
- `JxxxxA` is the product number of the switch (required). Any of the 2930M models can be installed and can assume this provisioned configuration. If you specify a value for this parameter, then only a switch of this specific model type can assume this provisioned configuration.
- (Optional) `MAC-Addr` can be specified if you want a specific switch to assume this provisioned configuration. If this value is entered, then the type value for the switch that has this MAC address must be correct, or a configuration error is logged and the switch is not allowed to join the stack.

7. Connect the stacking cables to the module ports for the desired stacking topology. For example, plug port 1 and 2 in a ring.
8. Install Stacking Modules into the other switches that will be members of the stack, but do not boot them yet.

NOTE: At a minimum, a ring topology must be created. A chain topology is not recommended because any hardware or software failure in the stack results in lost ports, which increases the amount of time for the recovery of full stack operation due to multiple reboots. See the *Installation and Getting Started Guide* for supported topologies corresponding to your switch.

9. Boot the Standby and Member switches. The second switch that is booted becomes the Standby. The remaining switches become Members when booted.
10. When all of the switches are booted, enter the `show stacking` command to confirm that the stack is operating correctly. The following example shows ten switches in a ring topology.

Viewing output for ten switches in a ring topology

```
switch(config)# show stacking
MAC Address      : 941882-d83c49
Stack Topology   : Ring
Stack Status     : Active
Split Policy     : One-Fragment-Up
Uptime           : 0d 0h 6m
Software Version: WC.16.04.0000x

Mbr
 ID  Mac Address       Model                                Pri  Status
 --- ----------------- ----------------------------------- ----- ----------
 *1  941882-d83c40     Aruba JL321A 2930M-48G Switch        128  Commander
  2  941882-dd3480     Aruba JL320A 2930M-24G-PoE+ Switch   128  Member
  3  941882-d91a40     Aruba JL319A 2930M-24G Switch        128  Member
  4  941882-d9a240     Aruba JL321A 2930M-48G Switch        128  Member
  5  941882-d9e900     Aruba JL319A 2930M-24G Switch        128  Member
  6  941882-da0f40     Aruba JL319A 2930M-24G Switch        128  Member
  7  941882-dc05c0     Aruba JL322A 2930M-48G-PoE+ Switch   128  Member
  8  941882-dc9340     Aruba JL322A 2930M-48G-PoE+ Switch   128  Member
  9  941882-db0080     Aruba JL321A 2930M-48G Switch        128  Standby
 10 941882-dccf80      Aruba JL320A 2930M-24G-PoE+ Switch   128  Member
```

Using the plug-and-go method

1. Install a Stacking Module into each 2930M switch that will be in the stack, as described in the *Installation and Getting Started Guide* for the switch, but do not connect them together with stacking cables yet.
2. Make sure that stacking is enabled for each switch.

a. You can determine this by connecting a console to each switch and entering the command `show stacking` from the switch CLI.

b. If stacking is disabled, enter the command `stacking enable`. This command causes the switch to reboot.

NOTE: By default, stacking is enabled on all the 2930M switches when a Stacking Module is installed before the switch is powered up for the first time, but if the switches were powered up without a Stacking Module installed, then stacking is disabled.

If you are connecting stacking cables during/after switch boot, then multiple stacks can form (plug-and-go method).

3. Connect the stacking cables between the switches to form the desired stacking topology, then power on all switches.

When the switches that are stacked together complete booting up, one of the switches is elected as the Commander, one of the switches is elected as the Standby, the remaining switches become Members of the stack, and the stack becomes fully operational.

To find out the roles of the switches in the stack, connect a console to any of the switches and enter the `show stacking` command. You can use the MAC address and other information in the display to determine the roles of each of the switches.

Stack formation in Enhanced or Standard Secure Mode

To form a stack in Enhanced secure mode, the switch software supports only three scenarios:

Scenario 1

If independent switches in Standard secure mode are connected to form a stack, the resultant stack is in Standard secure mode.

Scenario 2

If independent switches in Enhanced secure mode are connected to form a stack, the resultant stack is in Enhanced secure mode.

Scenario 3

Following are the steps when independent switches in Standard secure mode are connected to form a stack in Enhanced secure mode:

1. Form a stack with one switch as commander and other as standby.
2. Check if the commander switch is of highest priority.

   If the priority is not set, use the command:

   ```
   stacking member <member_number> priority <priority_number>
   ```

3. Execute the command:

   ```
   secure-mode enhanced
   ```

   After reboot, the stack is in Enhanced secure mode.

4. To add switches to the stack, follow the preceding steps.

   If the newly added switch is in Standard secure mode, the switch will reboot in Enhanced secure mode to form a stack in Enhanced secure mode.

Adding a switch to a stack as a new member

Networking stacking allows for switches to be added to the stack while the stack is operational (as long as the maximum number of 10 switches in the stack is not exceeded).

1. Provision the stack for the new switch by entering the following command:

   ```
   switch(config)# stacking member N type JxxxxA [mac MAC-Addr]
   ```

   where:

   - `N` is the stacking member number for the switch.
   - `JxxxxA` is the product number of the switch (required). Any of the 2930M models can be installed and assume this provisioned configuration. If you specify a value for this parameter, then only a switch of this specific model type can assume this provisioned configuration.
   - (Optional) `MAC-Addr` can be specified if you want a specific switch to assume this provisioned configuration. If this value is entered, then the type value for the switch that has this MAC address must be correct, or a configuration error is logged and the switch is not allowed to join the stack.
   - (Optional) You can preconfigure a priority for the member switch by entering this command:

     ```
     switch(config)# stacking <member-ID> priority <1-255>
     ```

     Where:

     - <member-ID> is the stacking member number for the switch.
     - 1–255 is the priority (but it must be less than the priority assigned to the Commander. The priority for the Standby should be the second highest in the stack. The member switches can be left at the default priority value of 128.)

     NOTE: You must configure the type and the priority separately.

   NOTE: When the new switch has been provisioned in the stack, a complete configuration can be applied to the switch even before it is physically connected to the stack, connected to the network, and powered up.

2. Power on the new switch. The new switch does not become a member of the stack unless stacking has been enabled on the switch.
3. Install a Stacking Module into the new switch, connect the switch into the stack via the stacking cables, and form the desired stacking topology.
4. When the switch has finished booting, establish a console session with it and, from the config context, issue the command to enable stacking:

   ```
   switch(config)# stacking enable
   ```

   This causes the switch to reboot. When the reboot is complete, the switch is a member of the stack with the attributes that you provisioned for it.

5. Confirm that the switch is now a member of the stack by issuing a `show stacking` command via a console session with any of the switches in the stack. The command output should show that the new switch is a Standby or Member of the stack with the member number and priority that you assigned to it.

When you add the switch to the stack, the following occurs:

- The Stack Revision Number is incremented by one.
- The Commander verifies that the new switch has the same switch software as the other switches in the stack, and downloads the software to the new switch if it does not. When downloading new software, there will be an automatic reboot during this process.
- A stack ID is assigned, even if the switch is later disconnected from the stack.
- The member's console shows a change in hostname as Aruba-Stack-2930M. The console is redirected.
- The OOBM IP address for that member is no longer reachable.
- Ethernet port address is renamed from 1 to 1/1, 2/1,3/1 ...10/1 depending on the member number or member port number.

Removing a switch from the stack

You can remove a switch from the stack to be redeployed in another part of the network. The procedures vary depending on whether the switch is the Commander of the stack or not.

Removing a Member or the Standby

1. Establish a console session with the stack via direct console cable connection or telnet. If using the console cable, connect it to the Standby.
2. Enter the following command to remove the switch from the stack:

   ```
   switch(config)# stacking member N remove
   ```

   This causes the switch to lose its complete configuration and to be removed from the stack configuration. A subsequent `show stacking` command issued to the stack will show that the removed switch no longer exists in the stack.
3. Power down the removed switch.
4. Disconnect the stacking cables from the removed switch and from the other switches in the stack.

Replacing a faulty member without losing the configuration

You can replace a faulty member from the stack without affecting the switch configuration using the preconfigure command. The following example shows how to change or drop the MAC address of the faulty member in a four member stack using the preconfigure command.

```
switch(config)# stacking member 4 type JL321A mac-address e0071b-def840
This will save the current configuration. Continue [y/n]?  y

switch(config)# show stack

Stack ID          : 0100e007-1bdab580

MAC Address       : e0071b-dab58a
Stack Topology    : Chain
Stack Status      : Active
Split Policy      : One-Fragment-Up
Uptime            : 0d 0h 34m
Software Version  : WC.16.04.0000x

Mbr
 ID  Mac Address       Model                                 Pri  Status
 ---  -------------    -----------------------------------    ---  ----------
 *1  e0071b-dab580     Aruba JL322A 2930M-48G-PoE+ Switch    128  Commander
  2  e0071b-de7dc0     Aruba JL321A 2930M-48G Switch          128  Member
```

```
  3  e0071b-de8d00     Aruba JL321A 2930M-48G Switch          128  Standby
  4  e0071b-def840     Aruba JL321A 2930M-48G Switch          128  Not Joined
```

After the fourth member boots up:

```
switch(config)# show stack

Stack ID          : 0100e007-1bdab580

MAC Address       : e0071b-dab58a
Stack Topology    : Ring
Stack Status      : Active
Split Policy      : One-Fragment-Up
Uptime            : 0d 0h 43m
Software Version : WC.16.04.0000x

Mbr
 ID  Mac Address       Model                                  Pri Status
 --- ----------------- -------------------------------------- --- ----------
 *1  e0071b-dab580     Aruba JL322A 2930M-48G-PoE+ Switch     128 Commander
  2  e0071b-de7dc0     Aruba JL321A 2930M-48G Switch          128 Member
  3  e0071b-de8d00     Aruba JL321A 2930M-48G Switch          128 Standby
  4  e0071b-def840     Aruba JL321A 2930M-48G Switch          128 Member
```

NOTE: Use the command `stack member <id> remove reboot` to reboot a stack when a stack member is in inconsistent state, and when a failure is also detected.

Removing the Commander

1. Establish a console session with the stack via direct console cable connection or telnet. If using the console cable, connect it to a switch other than the Commander
2. Enter the following command to force the Commander status over to the Standby switch:

   ```
   switch(config)# redundancy switchover
   ```

   This results in the Standby switch taking the role of the Commander and a new Standby being selected from the remaining member switches. The former Commander becomes a Member of the stack.
3. To remove the former Commander from the stack:

   ```
   switch(config)# stacking member N remove
   ```

   where `N` is the member number of the former Commander
4. Power down the removed switch.
5. Disconnect the stacking cables from the removed switch and from the other switches in the stack.

Renumbering stack members

If you did not provision the stack for the switches when you first created the stack, it is possible that members did not acquire the desired member numbers. The stack members can be renumbered.

A four member stack is used in the following example with switches A, B, C, and D. These switches inadvertently acquired member numbers 1, 3, 2, and 4, respectively. Switch B acquired member number 3 and switch C acquired member number 2. The stack should have A=member 1, B=member 2, C=member 3, and D=member 4.

1. In the global config context, enter the `remove` command option for switch B (member 3) and switch C (member 2):

   ```
   switch(config)# stack member 3 remove
   ```

```
switch(config)# stack member 2 remove
```

This command clears the MAC address of the member 2 configuration to allow switch C's MAC address to be entered in the next command, without a duplicate MAC address occurring in the stack.

All configurations on the removed member switch are deleted, not just the stacking configuration.

2. Enter the following command:

```
switch(config)# stack member 2 type <J-number> mac-address <B's MAC address>

switch(config)# stack member 3 type <J-number> mac-address <C's MAC address>
```

3. Reboot switch B (new member 2) and then switch C (new member 3).
4. To confirm that each switch now has the desired member number, enter the `show stacking` command.

Restoring the operation of a stack

Restoring the operation of a stack includes restoring stack operation after disconnecting a power cord or a stacking cable, replacing a failed stack member or a failed stacking module.

Restoring operation after disconnecting a power cord

If a power cord becomes disconnected from one of the switches in the stack, the stack operation is affected. The stacking status of the switch that lost power is "Missing." Its record is retained in the stack configuration. The effect of the power loss depends on the role of the switch in the stack.

- If the Commander loses power, the Standby switch takes over as the Commander and one of the member switches in the stack is elected as the new Standby.
- If the Standby loses power, one of the member switches in the stack is elected as the new Standby.
- For any switch that loses power, all the network ports and stacking ports are non-operational until power is restored to the switch and it rejoins the stack. This affects the stacking topology.
- Reconnecting the power cord restores the operation of the switch, however, if the switch was either the Commander or the Standby, then it returns in a different role if the topology has 3 or more members. In a 2-member stack, a Standby that reboots will rejoin as Standby.

Restoring operation after disconnecting a stacking cable

If a stacking cable becomes disconnected from one of the switches in the stack, the effect depends on the stacking topology that is being used:

- Ring—There is little effect. The stack topology is temporarily changed to a chain topology. To recover, simply reconnect the stacking cable; the ring topology and the previous stack configuration is restored.
- Chain—The following occurs:
  - The smaller section (fragment) of the stack that results from the disconnection becomes `Inactive` (the `Stack Status` value shown in the output of the show stacking command is `Inactive`.
  - If the two resulting fragments are the same size, the fragment that contains the Commander will be `Active`, and the other fragment becomes `Inactive`.
  - Both fragments will have a Commander and a Standby selected (if there is more than one switch in each fragment).
  - When the stacking cable is reconnected to reform the chain:

- The Commander and Standby of the Active fragment retain those roles for the resulting stack. If the original Commander was not in that fragment, then the stack will have a new Commander when the stack is reformed.
- The switches in the Inactive fragment reboot and assume their new roles in the reformed chain.

Replacing a failed stack member

If a Stack Member fails, the effect on the stack depends on which member failed.

- If the Commander fails, the Standby switch takes over as the Commander and one of the Member switches in the stack is elected as the new Standby. All network ports and stacking ports on the failed switch become non-operational.
- If the Standby fails, one of the Member switches in the stack is elected as the new Standby. All network ports and stacking ports on the failed switch become non-operational.
- If a Member fails, all network ports and stacking ports on that switch become non-operational.

If a Stack Member fails:

1. Physically remove the Stack Member from the stack.
2. Replace the failed Stack Member.

NOTE: Hewlett Packard Enterprise recommends using the same type (product or "J" number) switch as a replacement since all configuration information is retained.

- If you are using the same type switch as a replacement:
  1. Provision the new switch using the `stacking member N` command.
  2. Reconnect all Ethernet ports as they were on the failed switch.

- If you are using a different type switch as a replacement:
  1. Remove the failed switch from the stack configuration using the `stacking member N remove` command.
  2. Provision the new switch using the `stacking member N` command.
  3. Reconnect Ethernet ports and create a new stack configuration on the new switch.

If the replacement switch uses a different version of software, it will be updated automatically to match the software version running on the stack.

Replacing a failed stacking module

Replacing a failed stacking module is simpler than replacing a Stack Member since the switch configuration itself does not change. In this case, there is no need to re-provision the switch as a member of the stack. After you replace the stacking module, if the switch that experienced the module failure was Commander or Standby, the election of a new Commander and Standby is the same.

Merging stack fragments

When two fragments have the same stack-id, the merge of the fragments is almost always allowed regardless of the merge policy. The Commander and Standby of the merged stack are selected based on the election rules. All of the switches in the previously inactive fragment or fragments reboot, and then join the Active fragment as Members.

If both fragments are Inactive, then an election process occurs. The two (or more) Commanders in the fragments are compared. The Commander is selected using the following criteria:

1. Highest Stack Rev
2. If the stack rev is the same for both, then choose the switch with the highest configured priority
3. If the priorities are the same for both, then choose the switch with the highest OS revision
4. If the OS revisions are the same, then choose the switch with the longest uptime
5. If the uptimes are the same, then choose the switch with the lowest MAC address

Modifying the stack topology

You can increase the efficiency and redundancy of the stack by adding stacking cables to create a stacking mesh instead of a ring. This modification can be performed while the switches are powered on and the stack is operating. After connecting the cables, enter the `show stack` command. The `Stack Topology field` value displays the new topology.

Downloading new software to the stack

The stack is essentially a single switch with the Commander unit controlling the management functionality, so the process of loading new software is identical to the process for a standalone switch. For example: 2930M. See the *Management and Configuration Guide* for your switch.

After new software is loaded on the Commander, the Commander installs the software on all the stack members. The loading process can take some time.

To load the new software:

1. Load the new software onto the Commander via TFTP or Xmodem.
2. Once the new software is loaded, establish a console session with the stack and enter the following command:

   ```
   switch# boot system
   ```

   This causes the entire stack to be rebooted. Each unit is booted from its image unless you specify otherwise with options to this command. Make sure that you boot from the image to which you downloaded (that is, primary or secondary). If you add a new member to an existing stack, the Commander updates the new switch’s software to match the current stack software. Multiple versions of software are not supported across stack members.
3. Confirm that the new software has been loaded on each stack member by entering the `member-context` command for each member. From the stack member context, you can see the switch software version that is running on that switch by entering the `show flash` or `show version` command.

   Syntax:

   ```
   member-context stack-member
   ```

   Sets the CLI context so that subsequent commands apply to the stack member that is specified.

Monitoring stacking

Use the following commands to monitor the status and configuration of the stack.

Syntax:

```
show stacking
```

Shows the current state of the stack.

Viewing `show stacking` output

```
switch(config)# show stacking

Stack ID         : 0100e007-1bdab580

MAC Address      : e0071b-dab58a
Stack Topology   : Ring
Stack Status     : Active
Split Policy     : One-Fragment-Up
Uptime           : 0d 0h 45m
Software Version : WC.16.04.0000x

 Mbr
 ID  Mac Address       Model                                 Pri Status
 --- -------------     ------------------------------------- --- ----------
 *1  e0071b-dab580     Aruba JL322A 2930M-48G-PoE+ Switch    128 Commander
  2  e0071b-de7dc0     Aruba JL321A 2930M-48G Switch         128 Member
  3  e0071b-de8d00     Aruba JL321A 2930M-48G Switch         128 Standby
  4  e0071b-def840     Aruba JL321A 2930M-48G Switch         128 Member
```

If stacking is disabled on the switch, the `show stacking` command displays this message:

```
Stacking is disabled.
```

Possible values for the various parameters are:

- Stack Topology: Chain, Ring, Unknown
- Stack Status: Active, Fragment Active, Fragment Inactive
- Pri (Priority): <1 - 255>
- Status: Commander, Standby, Member, Standby Booting, Booting, Missing, Not Joined, Failed

Syntax:

```
show stacking detail
```

Shows the same output as the `show stacking` command, but with much more information about each device's stack port and connectivity, CPU state, uptime, and so on.

Viewing `show stacking detail` output

```
switch(config)# show stacking detail

Stack ID         : 0100e007-1bdab580

MAC Address      : e0071b-dab58a
Stack Topology   : Ring
Stack Status     : Active
Split Policy     : One-Fragment-Up
Uptime           : 0d 0h 46m
Software Version : WC.16.04.0000x

Name             : Aruba-Stack-2930M
Contact          :
Location         :
```

```
Member ID         : 1
Mac Address       : e0071b-dab580
Type              : JL322A
Model             : Aruba JL322A 2930M-48G-PoE+ Switch
Priority          : 128
Status            : Commander
ROM Version       : WC.17.01.0001
Serial Number     : ff ff ff ff ff ff ff ff ff ff
Uptime            : 0d 0h 47m
CPU Utilization   : 2%
Memory - Total    : 331,407,872 bytes
Free              : 170,619,327 bytes
Stack Ports -
#1 : Active, Peer member 2
#2 : Active, Peer member 4

MAC Address       : e0071b-dab58a
Stack Topology    : Ring
Stack Status      : Active
Split Policy      : One-Fragment-Up
Uptime            : 0d 0h 46m
Software Version  : WC.16.04.0000x

Name              : Aruba-Stack-2930M
Contact           :
Location          :


Member ID         : 1
Mac Address       : e0071b-dab580
Type              : JL322A
Model             : Aruba JL322A 2930M-48G-PoE+ Switch
Priority          : 128
Status            : Commander
ROM Version       : WC.17.01.0001
Serial Number     : ff ff ff ff ff ff ff ff ff ff
Uptime            : 0d 0h 47m
CPU Utilization   : 2%
Memory - Total    : 331,407,872 bytes
Free              : 170,619,327 bytes
Stack Ports -
#1 : Active, Peer member 2
#2 : Active, Peer member 4



Member ID         : 2
Mac Address       : e0071b-de7dc0
Type              : JL321A
Model             : Aruba JL321A 2930M-48G Switch
Priority          : 128
Status            : Member
ROM Version       : WC.17.01.0001
Serial Number     :
Uptime            : 0d 0h 46m
CPU Utilization   : 0%
Memory - Total    : 331,407,872 bytes
Free              : 198,676,837 bytes
Stack Ports -
#1 : Active, Peer member 3
#2 : Active, Peer member 1
```

```
Member ID         : 3
Mac Address       : e0071b-de8d00
Type              : JL321A
Model             : Aruba JL321A 2930M-48G Switch
Priority          : 128
Status            : Standby
ROM Version       : WC.17.01.0001
Serial Number     :
Uptime            : 0d 0h 46m
CPU Utilization   : 0%
Memory - Total    : 331,407,872 bytes
Free              : 186,097,709 bytes

Member ID         : 2
Mac Address       : e0071b-de7dc0
Type              : JL321A
Model             : Aruba JL321A 2930M-48G Switch
Priority          : 128
Status            : Member
ROM Version       : WC.17.01.0001
Serial Number     :
Uptime            : 0d 0h 46m
CPU Utilization   : 0%
Memory - Total    : 331,407,872 bytes
Free              : 198,676,837 bytes
Stack Ports -
#1 : Active, Peer member 3
#2 : Active, Peer member 1



Member ID         : 3
Mac Address       : e0071b-de8d00
Type              : JL321A
Model             : Aruba JL321A 2930M-48G Switch
Priority          : 128
Status            : Standby
ROM Version       : WC.17.01.0001
Serial Number     :
Uptime            : 0d 0h 46m
CPU Utilization   : 0%
Memory - Total    : 331,407,872 bytes
Free              : 186,097,709 bytes
Stack Ports -
#1 : Active, Peer member 4
#2 : Active, Peer member 2



Member ID         : 4
Mac Address       : e0071b-def840
Type              : JL321A
Model             : Aruba JL321A 2930M-48G Switch
Priority          : 128
Status            : Member
ROM Version       : WC.17.01.0001
Serial Number     :
Uptime            : 0d 0h 7m
CPU Utilization   : 0%
Memory - Total    : 331,407,872 bytes
Free              : 198,678,297 bytes
Stack Ports -
```

```
#1 : Active, Peer member 1
#2 : Active, Peer member 3
```

Syntax:

`show stacking member` *`stack-member-list`*

Shows detailed information about switches in the *`stack-member-list`* only.

Syntax:

`show stacking stack-ports member`*`stack-member`*

Shows the current state of the stacking ports of the specified member. If a member is not specified, the command shows the state of the ports of all physically present devices in the stack.

Viewing `show stacking stack-ports` output

```
switch(config)# show stacking stack-ports
  Member Stacking Port State    Peer Member Peer Port
  ------ ------------- -------- ----------- ---------
  1      1             Up       2           2
  1      2             Up       4           1
  2      1             Up       3           2
  2      2             Up       1           1
  3      1             Up       4           2
  3      2             Up       2           1
  4      1             Up       1           2
  4      2             Up       3           1
```

If you specify specific stack members in the command, then the stacking port information for those members displays.

Troubleshooting stacking

Troubleshoot OOBM and split stack issues

If all the OOBM ports in the stack are in the same VLAN, you can use the `show oobm` commands to view the current state of all the switches. For example, if you have a five-member chain and the link between member 3 and 4 fails, a stack split will occur with an active fragment on members 1-2-3 and an inactive fragment on members 4 and 5.

There is one IP address for the active fragment. This can be statically set by assigning an IP address to the global OOBM port.

If the stack splits, you can connect to the Active Fragment using the global OOBM IP address and then enter the `show oobm discovery` command to see if this active fragment has discovered any other members that are connected using the OOBM LAN.

In the following five member chain example, connect using the global IP address of 10.0.102.173. Once logged on, enter the `show stack` command.

Viewing stacking member status before split

```
switch# show stack

Stack ID         : 02009418-82db0080

MAC Address      : 941882-db0089
Stack Topology   : Chain
Stack Status     : Active
Split Policy     : One-Fragment-Up
Uptime           : 0d 0h 5m
```

```
Software Version : WC.16.04.0000x

 Mbr
 ID  Mac Address       Model                                 Pri Status
 --- ----------------- ------------------------------------- --- ----------
 *1  941882-db0080     Aruba JL321A 2930M-48G Switch         128 Commander
  2  941882-dc05c0     Aruba JL322A 2930M-48G-PoE+ Switch    128 Member
  3  941882-dc9340     Aruba JL322A 2930M-48G-PoE+ Switch    128 Member
  4  941882-dccf80     Aruba JL320A 2930M-24G-PoE+ Switch    128 Member
  5  941882-dd3480     Aruba JL320A 2930M-24G-PoE+ Switch    128 Standby
```

Viewing stacking member status after split

```
switch# show stack

Stack ID         : 02009418-82db0080

MAC Address      : 941882-db0089
Stack Topology   : Chain
Stack Status     : Fragment Active
Split Policy     : One-Fragment-Up
Uptime           : 0d 0h 7m
Software Version : WC.16.04.0000x

 Mbr
 ID  Mac Address       Model                                 Pri Status
 --- ----------------- ------------------------------------- --- ---------------
  1  941882-db0080     Aruba JL321A 2930M-48G Switch         128 Commander
 *2  941882-dc05c0     Aruba JL322A 2930M-48G-PoE+ Switch    128 Member
  3  941882-dc9340     Aruba JL322A 2930M-48G-PoE+ Switch    128 Standby Booting
  4  941882-dccf80     Aruba JL320A 2930M-24G-PoE+ Switch    128 Missing
  5  941882-dd3480     Aruba JL320A 2930M-24G-PoE+ Switch    128 Missing



switch# show stack

Stack ID         : 02009418-82db0080

MAC Address      : 941882-dd34a2
Stack Topology   : Chain
Stack Status     : Fragment Inactive
Split Policy     : One-Fragment-Up
Uptime           : 0d 0h 7m
Software Version : WC.16.04.0000x

 Mbr
 ID  Mac Address       Model                                 Pri Status
 --- ----------------- ------------------------------------- --- ---------------
  1  941882-db0080     Aruba JL321A 2930M-48G Switch         128 Missing
  2  941882-dc05c0     Aruba JL322A 2930M-48G-PoE+ Switch    128 Missing
  3  941882-dc9340     Aruba JL322A 2930M-48G-PoE+ Switch    128 Missing
  4  941882-dccf80     Aruba JL320A 2930M-24G-PoE+ Switch    128 Missing
 *5  941882-dd3480     Aruba JL320A 2930M-24G-PoE+ Switch    128 Commander
```

Enter `show oobm discovery` to see if the members have been discovered using OOBM.

Viewing oobm discovery

```
Switch# show oobm discovery

Active Stack Fragment(local)
```

```
IP Address  : 10.0.102.173


Mbr Mac Address        Status
ID
--- ------------------ ----------
1   941882-d83c40      Commander
2   941882-d91a40      Member
3   941882-d9a240      Member

Inactive Stack Fragment(discovered)

IP Address  : 10.0.102.114


Mbr Mac Address        Status
ID
--- ------------------ ----------
5   941882-d9e900      Commander
4   941882-da0f40      Member



Switch# show oobm discovery

Inactive Stack Fragment(local)

IP Address  : 10.0.102.114

Mbr Mac Address        Status
ID
--- ------------------ ----------
5   941882-d9e900      Commander
4   941882-da0f40      Member

Active Stack Fragment(discovered)

IP Address  : 10.0.102.173

Mbr Mac Address        Status
ID
--- ------------------ ----------
1   941882-d83c40      Commander
2   941882-d91a40      Member
3   941882-d9a240      Member
```

Members 4 and 5 are up, but is an inactive fragment. It has an addressable IP address, which can be used to connect to this fragment.

Connecting to a stack fragment

```
switch# telnet 10.0.102.162 oobm

Aruba JL320A 2930M-24G-PoE+ Switch
Software revision WC.16.04.0000x

(C) Copyright 2017 Hewlett Packard Enterprise Development LP

                      RESTRICTED RIGHTS LEGEND
Confidential computer software. Valid license from Hewlett Packard Enterprise
Development LP required for possession, use or copying. Consistent with FAR
12.211 and 12.212, Commercial Computer Software, Computer Software
Documentation, and Technical Data for Commercial Items are licensed to the
U.S. Government under vendor's standard commercial license.
```

```
We'd like to keep you up to date about:
  * Software feature updates
  * New product announcements
  * Special events
Please register your products now at:  www.hpe.com/networking/register

Press any key to continue
```

Enter the `show stacking` command.

Viewing missing stack members

```
switch# show stacking

Stack ID         : 02009418-82db0080

MAC Address      : 941882-dd34a2
Stack Topology   : Chain
Stack Status     : Fragment Inactive
Split Policy     : One-Fragment-Up
Uptime           : 0d 0h 9m
Software Version : WC.16.04.0000x

 Mbr
 ID  Mac Address       Model                                 Pri Status
 --- ----------------- ------------------------------------- --- ---------------
  1  941882-db0080     Aruba JL321A 2930M-48G Switch         128 Missing
  2  941882-dc05c0     Aruba JL322A 2930M-48G-PoE+ Switch    128 Missing
  3  941882-dc9340     Aruba JL322A 2930M-48G-PoE+ Switch    128 Missing
  4  941882-dccf80     Aruba JL320A 2930M-24G-PoE+ Switch    128 Missing
  5  941882-dd3480     Aruba JL320A 2930M-24G-PoE+ Switch    128 Commander
```

Confirm by entering the `oobm discovery` command. Member 4 is down.

Confirming stack member 4 is down

```
switch# show oobm discovery

Inactive Stack Fragment(local)

IP Address  : 10.0.102.162

Mbr Mac Address         Status
ID
--- ------------------ ----------
5   941882-dd3480      Commander

Active Stack Fragment(discovered)

IP Address  : 10.0.102.218

Mbr Mac Address         Status
ID
--- ------------------ ----------
1   941882-db0080      Commander
2   941882-dc05c0      Member
3   941882-dc9340      Member
```

Using fault recovery/troubleshooting tools

Stacking provides tools and logging information to aid in troubleshooting problems specific to stacking. Problems may include:

- Installation/deployment issues
- Problems with initial stack creation
    - Problems with adding or removing members
    - Booting an existing stack

Stacking failures encountered while running an existing stack

The tools used in troubleshooting problems are:

- Event Log
- Show commands
    - Show stacking
    - Show system
    - Show boot history

- Show tech
- LEDs

Troubleshooting installation and deployment issues

Cause

Installation and deployment issues include the initial deployment or creation of a stack, adding additional members or removing members from a stack.

Problem:

When using the Deterministic method, one or more of the statically provisioned members did not join the stack.

Possible reasons a switch does not join a stack are:

- The switch being added is already a member of another stack and has a different stack ID.
- The maximum number of switches is already configured.
- The switch being added has been statically provisioned. The MAC address matches, but the switch type does not.
- There is a problem with the stack cable.
- The stack cables are connected in a way that creates an unsupported topology.
- Stack module failure.

Solution:

Perform a diagnostic analysis.

Troubleshooting issues with adding or removing members in the stack

Cause

Various problems described below could be causing issues with adding or removing members in the stack.

Problem:

Cannot add a new switch to an existing stack.

Solution:

Identify root cause. Possible reasons for a member not joining an existing stack are:

- The switch being added has already been a member of another stack and has a different stack ID.
- The maximum number of switches is already configured.
- The switch being added has been statically provisioned, but switch type and MAC address in the configuration do not match the switch being added.
- There is a problem with the stack cable.
- There is a problem with the stack physical cabling (illegal topology).

Problem:

The entire stack does not come up after a boot.

Solution:

There are several reasons why all members do not join the stack:

- There is a problem with the stack cable.
- Physical cabling was changed.
- Stack booted on incorrect configuration.
- One or more of the switches has a hardware problem (for example, bad power supply, back stacking module, corrupt flash).

Problem:

One or more of the members keeps rebooting and does not join the stack.

Possible reasons:

- An unresponsive member.
- Heartbeat loss—a stack that has a member no longer in the stack or a member failing after joining the stack.
- Illegal topology.

Problem:

After initial boot sequence, the activity and Link LEDs of an interface are not on and the ports are not passing traffic.

Solutions:

- Identify the “inactive fragment” and provide alternatives for recovery.
- Verify that all OOBMs are connected so that there is uninterrupted access.

Problem:

After a reboot, the selected Command or Standby are not the expected switches.

Solutions:

Check to see if the log files provide a reason why the Commander and Standby were chosen and which rule they matched.

Troubleshooting a strictly provisioned, mismatched MAC address

Cause

When switches are strictly provisioned, it is possible to enter an incorrect type or incorrect MAC address. If this occurs, the switch does not match the intended configuration entry and stacking attempts to add this switch as a new “plug-and-go” switch. If the stacking configuration already has 10 switches, then the “plug-and-go” fails.

The following example shows a stack with 9 members. There is a new JL319A switch that is supposed to be member 4; however, the MAC address was mis-typed, therefore, there is an “opening” for a plug-and-go at member 10. It will join as member 10.

Action

1. This shows the stack before boot.

   Viewing a stack with 9 members

```
switch(config)# show stack

Stack ID         : 02009418-82d83c40

MAC Address      : 941882-d83c49
Stack Topology   : Chain
Stack Status     : Active
Split Policy     : One-Fragment-Up
Uptime           : 0d 14h 27m
Software Version : WC.16.04.0000x


 Mbr
 ID  Mac Address       Model                                 Pri Status
 --- ----------------- ------------------------------------- --- ----------------
 *1  941882-d83c40     Aruba JL321A 2930M-48G Switch         150 Commander
  2  9cdc71-f576c0     Aruba JL324A 2930M-24SR-PoE+ Switch   128 Member
  3  941882-d91a40     Aruba JL319A 2930M-24G Switch         128 Standby
  4  9cdc71-f2bbc6     Aruba JL324A 2930M-24SR-PoE+ Switch   128 Not Joined
  5  941882-d9a240     Aruba JL321A 2930M-48G Switch         128 Not Joined
  6  941882-d9e900     Aruba JL319A 2930M-24G Switch         128 Not Joined
  7  9cdc71-f8f480     Aruba JL323A 2930M-40G-8SR-PoE+ Sw... 128 Not Joined
  8  941882-da0f40     Aruba JL319A 2930M-24G Switch         128 Not Joined
  9  9cdc71-f8d400     Aruba JL323A 2930M-40G-8SR-PoE+ Sw... 128 Not Joined
```

2. This shows that, after booting, the switch is joined as member 10.

   Viewing a member joining the stack

```
switch(config)# show stack

Stack ID         : 02009418-82d83c40

MAC Address      : 941882-d83c49
Stack Topology   : Chain
Stack Status     : Active
Split Policy     : One-Fragment-Up
Uptime           : 0d 14h 44m
Software Version : WC.16.04.0000x

Mbr
 ID  Mac Address       Model                                 Pri Status
 --- ----------------- ------------------------------------- --- ----------------
 *1  941882-d83c40     Aruba JL321A 2930M-48G Switch         150 Commander
  2  9cdc71-f576c0     Aruba JL324A 2930M-24SR-PoE+ Switch   128 Member
  3  941882-d91a40     Aruba JL319A 2930M-24G Switch         128 Standby
  4  9cdc71-f2bbc6     Aruba JL324A 2930M-24SR-PoE+ Switch   128 Not Joined
  5  941882-d9a240     Aruba JL321A 2930M-48G Switch         128 Not Joined
  6  941882-d9e900     Aruba JL319A 2930M-24G Switch         128 Not Joined
  7  9cdc71-f8f480     Aruba JL323A 2930M-40G-8SR-PoE+ Sw... 128 Not Joined
  8  941882-da0f40     Aruba JL319A 2930M-24G Switch         128 Not Joined
  9  9cdc71-f8d400     Aruba JL323A 2930M-40G-8SR-PoE+ Sw... 128 Not Joined
  10 9cdc71-f54940     Aruba JL324A 2930M-24SR-PoE+ Switch   128 Member
```

To correct this issue:

a. Write down the correct MAC address.

b. Remove the member that was added using plug-and-go with the strictly provisioned, mismatched MAC address as shown in the following example.

c. Update the strictly provisioned entry with the correct MAC address.

d. Boot the switch.

Removing a member and updating the entry with a MAC address

```
switch(config)# stacking member 10 remove
The specified stack member will be removed from the stack and
its configuration will be erased. The resulting configuration
will be saved. The stack member will be shutdown. Continue [y/n]? y

Aruba-Stack-2930M(config)# stacking member 4 type JL324A mac-address 9cdc71-f2bbc0
This will save the current configuration. Continue [y/n]?  y
```

3. This shows that member 4 has joined the stack.

Viewing that member 4 joined the stack

```
switch(config)# show stack

Stack ID         : 02009418-82d83c40

MAC Address      : 941882-d83c49
Stack Topology   : Chain
Stack Status     : Active
Split Policy     : One-Fragment-Up
Uptime           : 0d 14h 50m
Software Version : WC.16.04.0000x

Mbr
 ID  Mac Address       Model                                 Pri Status
 --- ----------------- ------------------------------------- --- ---------------
 *1  941882-d83c40     Aruba JL321A 2930M-48G Switch         150 Commander
  2  9cdc71-f576c0     Aruba JL324A 2930M-24SR-PoE+ Switch   128 Member
  3  941882-d91a40     Aruba JL319A 2930M-24G Switch         128 Standby
  4  9cdc71-f2bbc6     Aruba JL324A 2930M-24SR-PoE+ Switch   128 Not Joined
  5  941882-d9a240     Aruba JL321A 2930M-48G Switch         128 Not Joined
  6  941882-d9e900     Aruba JL319A 2930M-24G Switch         128 Not Joined
  7  9cdc71-f8f480     Aruba JL323A 2930M-40G-8SR-PoE+ Sw... 128 Not Joined
  8  941882-da0f40     Aruba JL319A 2930M-24G Switch         128 Not Joined
  9  9cdc71-f8d400     Aruba JL323A 2930M-40G-8SR-PoE+ Sw... 128 Not Joined

boot member 4


switch(config)# show stack

Stack ID         : 02009418-82d83c40
```

```
MAC Address      : 941882-d83c49
Stack Topology   : Chain
Stack Status     : Active
Split Policy     : One-Fragment-Up
Uptime           : 0d 14h 50m
Software Version : WC.16.04.0000x

Mbr
 ID  Mac Address       Model                                 Pri Status
 --- ----------------- ------------------------------------- --- ---------------
 *1  941882-d83c40     Aruba JL321A 2930M-48G Switch         150 Commander
  2  9cdc71-f576c0     Aruba JL324A 2930M-24SR-PoE+ Switch   128 Member
  3  941882-d91a40     Aruba JL319A 2930M-24G Switch         128 Standby
  4  9cdc71-f2bbc0     Aruba JL324A 2930M-24SR-PoE+ Switch   128 Member
  5  941882-d9a240     Aruba JL321A 2930M-48G Switch         128 Not Joined
  6  941882-d9e900     Aruba JL319A 2930M-24G Switch         128 Not Joined
  7  9cdc71-f8f480     Aruba JL323A 2930M-40G-8SR-PoE+ Sw... 128 Not Joined
  8  941882-da0f40     Aruba JL319A 2930M-24G Switch         128 Not Joined
  9  9cdc71-f8d400     Aruba JL323A 2930M-40G-8SR-PoE+ Sw... 128 Not Joined
```

Troubleshoot a mismatched stack-ID

This is an example of a stack that has two members with three more members that have been strictly provisioned, following the deterministic method of initial installation.

Viewing a stack with 3 unjoined switches

```
switch# show stack

Stack ID         : 01009418-82d9e900

MAC Address      : 941882-da0f62
Stack Topology   : Chain
Stack Status     : Active
Split Policy     : One-Fragment-Up
Uptime           : 0d 0h 3m
Software Version : WC.16.04.0000x

 Mbr
 ID  Mac Address       Model                                 Pri Status
 --- ----------------- ------------------------------------- --- ---------------
  1  941882-d83c40     Aruba JL321A 2930M-48G Switch         128 Not Joined
  2  941882-d91a40     Aruba JL319A 2930M-24G Switch         128 Not Joined
  3  941882-d9a240     Aruba JL321A 2930M-48G Switch         128 Not Joined
 *4  941882-da0f40     Aruba JL319A 2930M-24G Switch         128 Commander
  5  941882-d9e900     Aruba JL319A 2930M-24G Switch         128 Standby
```

When powering on switch 3, it does not join the stack.

The stack ports for the new switch appear online, however, the `show stacking` command shows that the switch has not been recognized.

Viewing the switch is not recognized

```
switch(config)# show stack

Stack ID         : 01009418-82d9a240

MAC Address      : 941882-d9a249
Stack Topology   : Chain
Stack Status     : Active
Split Policy     : One-Fragment-Up
```

```
Uptime           : 0d 0h 1m
Software Version : WC.16.04.0000x



 Mbr
 ID  Mac Address       Model                                  Pri Status
 --- ----------------- -------------------------------------- --- ---------------
 *3  941882-d9a240     Aruba JL321A 2930M-48G Switch          128 Commander
```

The `show stacking` command does not show that the member is “Not Joined.”

A log file indicates that a “topo /hello” was seen from a switch that was not part of the current stack ID. The console of the switch that should have been member 3 shows the following example output.

Viewing output from the “not joined” switch

```
switch(config)# show stacking stack-ports member 4,5

Member 4

  Member Stacking Port State    Peer Member Peer Port
  ------ ------------- -------- ----------- ---------
  4      1             Down     0           0
  4      2             Up       5           1

Member 5

  Member Stacking Port State    Peer Member Peer Port
  ------ ------------- -------- ----------- ---------
  5      1             Up       4           2
  5      2             Down     0           0
```

The output is different if you have an inactive fragment, since this switch can have the configuration from an old stack. In this case, it might be inactive and show ‘missing’ switches from the old configuration. The stack-id value does not match the stack ID of the Aruba Stack 2930M stacking factory reset.

```
switch(config)# stacking factory-reset
Configuration will be deleted and device rebooted,continue [y/n]?
Y
```

To join this switch to the other stack, execute the `stacking factory-reset` command to erase all of the stale stacking configuration information. This command automatically reboots the switch and on its subsequent boot, the switch is able to join the new stack.

Troubleshoot stacking problems using the show logging command

The `show logging` command troubleshoots problems in stacking.

Syntax

```
show logging <a|b|r|s|t|m|p|e|w|i|d|command|filter|option-str|substring ...]>
```

The options `a|r|substring` can be used in combination with an event class option.

a

Instructs the switch to display all recorded log events, which include events from previous boot cycles.

b

Display log events as time since boot instead of date/time format.

r

Instructs the switch to display recorded log events in reverse order (most recent first.)

s

Display AMM and SMM log events.

t

Display log events in granularity in 10 milli seconds.

command

Instructs the switch to display command logs.

substring

Instructs the switch to display only those events that match the substring.

The remaining event class options are listed in order of severity - lowest severity first. The output of the command is confined to event classes of equal or higher severity.

Only one of options `d|i|w|e|p|m` can be used in the command at a time.

The `a|r` and substring options may be used in combination with an event class option.

m

Display major type of messages.

e

Display error event class.

p

Display major and error type of messages.

w

Display major, error, and warning type of messages.

I

Display major, error, warning, and information.

d

Display major, error, warning, information, and debug messages.

filter

Display log filter configuration and status information.

Option-str

Filter events shown.

show logging example output

```
switch# show logging -e
Keys:   W=Warning   I=Information
         M=Major     D=Debug E=Error
----  Event Log listing: Events Since Boot  ----
M 07/05/17 18:52:33 00064 system: ST4-CMDR: Reboot of Member ID 4, User reboot
M 07/05/17 18:52:33 02796 chassis: ST4-CMDR: Internal power supply 1 inserted.
            Total fault count: 0.
M 07/05/17 18:52:33 02797 chassis: ST4-CMDR: Internal power supply 1 is OK.
            Total fault count: 0.
M 07/05/17 18:52:50 02796 chassis: ST5-STBY: Internal power supply 1 inserted.
```

```
           Total fault count: 0.
M 07/05/17 18:52:50 02797 chassis: ST5-STBY: Internal power supply 1 is OK.
           Total fault count: 0.
----  Bottom of Log : Events Listed = 5  ----
```

Troubleshooting a strictly provisioned, mismatched type

Cause

When the MAC address matches a strictly provisioned configuration, it either matches the configured type and succeeds, or it does not match the type and fails. This is because the MAC address is unique and you cannot have duplicate MAC addresses.

The log messages indicate that this was the type of failure. The information in the log message helps you correct the configuration.

Action

1. The switch that fails to join automatically reboots. Execute the `show stacking` command to view the mis-configured entry.

   Viewing the mis-configured entry

   ```
   switch(config)# show stack

   Stack ID         : 01009418-82d83c40

   MAC Address      : 941882-d83c4a
   Stack Topology   : Chain
   Stack Status     : Active
   Split Policy     : One-Fragment-Up
   Uptime           : 0d 0h 8m
   Software Version : WC.16.04.0000x



    Mbr
    ID  Mac Address       Model                                 Pri Status
    --- ----------------- ------------------------------------- --- ---------------
    *1  941882-d83c40     Aruba JL321A 2930M-48G Switch         128 Commander
     2  941882-d91a40     Aruba JL319A 2930M-24G Switch         128 Member
     3  941882-d9a240     Aruba JL321A 2930M-48G Switch         128 Member
     4  941882-d9e900     Aruba JL319A 2930M-24G Switch         128 Standby
     5  941882-da0f40     Aruba JL321A 2930M-48G Switch         128 Not Joined
   ```

2. The configuration entry for member 5 matches a JL319A switch that will be added, however, it will fail because it is configured as a JL321A switch. failures

   The following example shows the log entries with the failure to join the stack.

   ```
   I 07/05/17 19:21:51 02556 chassis: ST4-STBY: Stack port 1 cable inserted.
   W 07/05/17 19:22:02 03277 stacking: ST1-CMDR: Member 5 (941882-da0f40) cannot
               join stack due to incorrect product id: JL319A
   I 07/05/17 19:22:02 02558 chassis: ST4-STBY: Stack port 1 is now on-line.
   W 07/05/17 19:23:32 03258 stacking: ST1-CMDR: Provisioned switch with Member ID
               5 removed due to loss of communication
   I 07/05/17 19:23:32 03272 stacking: ST1-CMDR: Stack fragment active
   I 07/05/17 19:24:11 05225 activate: ST1-CMDR: Loading security certificates and
               synchronizing time with NTP.
   W 07/05/17 19:24:17 05220 activate: ST1-CMDR: Unable to resolve the Activate
               server address device.arubanetworks.com.
   ```

```
----  Bottom of Log : Events Listed = 192  ----
switch#
```

3. You cannot re-type the configuration command with the same MAC address, member ID, and a different J-number. Remove the configuration and then reconfigure this switch member entry.

   Removing a stack member and reconfiguring

```
switch# stacking member 5 remove
The specified stack member configuration will be erased. The
resulting configuration will be saved. Continue [y/n]?   y
switch(config)# stacking member 5 type JL319A mac-address 941882-da0f40
```

4. Boot the switch with the matching MAC/Type.

   Viewing joined stack members

```
switch(config)# show stack

Stack ID          : 01009418-82d83c40

MAC Address       : 941882-d83c4a
Stack Topology    : Chain
Stack Status      : Active
Split Policy      : One-Fragment-Up
Uptime            : 0d 0h 43m
Software Version  : WC.16.04.0000x

 Mbr
 ID  Mac Address       Model                                 Pri Status
 --- ----------------- ------------------------------------- --- ---------------
  1  941882-d83c40     Aruba JL321A 2930M-48G Switch         128 Commander
  2  941882-d91a40     Aruba JL319A 2930M-24G Switch         128 Member
  3  941882-d9a240     Aruba JL321A 2930M-48G Switch         128 Member
 *4  941882-d9e900     Aruba JL319A 2930M-24G Switch         128 Standby
  5  941882-da0f40     Aruba JL319A 2930M-24G Switch         128 Member
```

Troubleshooting maximum stack members exceeded

Cause

This failure can happen if you have an active stack that has already reached its maximum number of members. It can also happen when the maximum number of switches is reached with a combination of active members and strictly provisioned members.

Since one of the suggested initial deployment techniques is a deterministic method using strictly provisioned entries, this failure example demonstrates what occurs if the maximum number of members is reached by strictly provisioning 10 members. At least one of these configuration entries has an incorrect MAC addresses. Similar to the mismatched MAC address example, the stack attempts to “plug-and-go” to add the switch, however, since the maximum number of membership has already been reached, the switch cannot join the stack.

Action

1. The following example shows the `show stacking` output before the switch attempts to join.

   Viewing stack members before the join

```
switch(config)# show stack

Stack ID          : 02009418-82d83c40

MAC Address       : 941882-d83c49
```

```
Stack Topology   : Chain
Stack Status     : Active
Split Policy     : One-Fragment-Up
Uptime           : 0d 15h 3m
Software Version : WC.16.04.0000x

Mbr
 ID  Mac Address   Model                                Pri Status
 ---------------------------------------------------------------------
 *1  941882-d83c40 Aruba JL321A 2930M-48G Switch        150 Commander
  2  9cdc71-f576c0 Aruba JL324A 2930M-24SR-PoE+ Switch 128 Member
  3  941882-d91a40 Aruba JL319A 2930M-24G Switch        128 Standby
  4  9cdc71-f2bbc0 Aruba JL324A 2930M-24SR-PoE+ Switch 128 Member
  5  941882-d9a240 Aruba JL321A 2930M-48G Switch        128 Not Joined
  6  941882-d9e900 Aruba JL319A 2930M-24G Switch        128 Not Joined
  7  9cdc71-f8f480 Aruba JL323A 2930M-40G-8SR-PoE+ Sw..128 Not Joined
  8  941882-da0f40 Aruba JL319A 2930M-24G Switch        128 Not Joined
  9  9cdc71-f8d400 Aruba JL323A 2930M-40G-8SR-PoE+ Sw..128 Not Joined
  10 9cdc71-f54940 Aruba JL324A 2930M-24SR-PoE+ Switch 128 Not Joined
```

2. When a switch that does not match the MAC addresses attempts to join, that switch reboots when the maximum configuration is detected. The active stack logs the following:

   ```
   W 10/07/00 06:01:11 03253 stacking: ST3-CMDR: Maximum number of switches in the
   stack has been reached. Cannot add 1cc1de-4da900 type JL324A.
   ```

3. The failure can be due to one of the strictly provisioned entries being incorrect. To correct this entry, reboot the switch. If there are already 10 switches in the stack, you cannot add additional switches at this time.

Troubleshooting a bad cable

Cause

Bad cables can cause the stack port to flap or go down completely. If there are an excessive number of port flaps, the port is disabled and the following log message appears:

```
W 10/06:00 23:23:16 03260 chassis: ST4-CMDR: Stack port 1 disabled due to
excessive errors. Check cable. To reenable use 'stacking member 4 stack-port 1
enable'.
```

Action

1. When this occurs, the `show stacking stack-ports` command shows the port with a status of “Disabled”.

   Viewing a disabled stack port

   ```
   switch(config)# show stacking stack-ports

   Member Stacking      Port State Peer Member Peer Port
   ------ ------------- --------   ----------- -----------
   1      1             Up         2            2
   1      2             Down       0            0
   2      1             Up         3            2
   2      2             Up         1            1
   3      1             Up         4            2
   3      2             Up         2            1
   4      1             Disabled   0            0
   4      2             Up         3            1
   5      1             Down       0            0
   5      2             Up         4            1

   switch(config)#
   ```

2. The following example shows member 3, port 2, which should be connected to member 4, port 1. The ports are down because the cable is bad or disconnected.

   Viewing that two ports are down due to a bad connection

```
switch# show stacking stack-ports

Member Stacking Port State    Peer Member Peer Port
------ ------------- -------- ----------- ---------
1      1             Up       2           2
1      2             Down     0           0
2      1             Up       3           2
2      2             Up       1           1
3      1             Down     0           0
3      2             Up       2           1
4      1             Up       5           2
4      2             Down     0           0
5      1             Down     0           0
5      2             Up       4           1
```

3. The solution in both cases is to ensure that the cable is firmly connected at both ends. If the problem continues, replace the cable. It is possible that there could be a problem with the stack port itself. In this case, validation of this issue requires the installation of a known good cable to see if that cable also fails.

   The port state is not UP until both ends of the cable are connected and the cable has been validated as a genuine cable.

   To view the statistics on the physical port, execute the `show tech` command in member-context 4. The following examples show the types of information displayed.

   Viewing `show tech` output

```
Port Number : 1 State : Available
Last Event  : Available Start Req : 1
NE Present  : 1 HPID Good : 1
HPID Fails  : 0 FE Present : 1
Rem Dev Rdy : 1
ESSI Link   : 1 ESSI Good : 1
ESSI Fails  : 0 ESSI TX En : 1
ICL Good    : 1 ICL Enabled : 1
30 Stack Management for 2930M
LP Local RDY: 1 LP Rem RDY : 1
LP DONE     : 1 ICL FailCnt : 0 (10 second interval)
ICL FailCnt : 0 (10 minute interval)
NE Presence HW : 1
FE Presence HW : 1
Rem Dev Rdy HW : 1
Local Dev Rdy HW : 1
Asserted NE Presence HW : 1
Asserted FE Presence HW : 1
Asserted Rem Dev Rdy HW : 1
Phy Frame Errors : 0
Invalid Status Errors      : 0
Invalid Packet Type Errors : 0
Incomplete Packet Errors   : 0
Checksum Errors            : 0
ESSI Flow Out This Port (HW) : 0x2
```

   Viewing trace information for a port

```
Trace for Port 1
[ 0] [Info ] Start Request Received (Empty) [0]
[ 1] [Info ] Waiting for Stack Module Good (Empty) [0]
[ 2] [Info ] Stack Module Good Received (Empty) [0]
[ 3] [Info ] Cable Insertion Detected (Empty) [1]
[ 4] [Info ] Re-enable NE Present Int [487]
[ 5] [Info ] Starting Cable HPID Validation (Inserted) [488]
[ 6] [Info ] Skipping Cable HPID Validation (Inserted) [488]
[ 7] [Info ] Far End Insertion Detected (Valid) [988]
[ 8] [Info ] Polling for ESSI phy link up (Valid) [988]
[ 9] [Info ] ESSI Link Up [9988]
[10] [Info ] ESSI Link Good (Valid) [9988]
[11] [Info ] ESSI Linked at 9988 ms [9988]
[12] [Info ] Remote Device Ready Detected (Valid) [10898]
[13] [Info ] ICL Change Request Enable (Cable Ready) [10898]
[14] [Info ] Detected Remote Ready Drop. (Cable Ready) [12651]
[15] [Info ] ICL Good. Behind. Partner ready. (Cable Ready) [12988]
[16] [Info ] ICL GOOD received at 2091 ms [12988]
[17] [Info ] Partner LP ready. (Cable Ready) [13980]
[18] [Info ] Set Device Ready. (Cable Ready) [13987]
[19] [Info ] ESSI Link Verified [13988]
[20] [Info ] ESSI Able to Transmit (Cable Ready)
```

Troubleshooting when a switch crashes and reboots

Cause

Although the switch software is highly reliable, a switch in the stack can experience a software issue that results in the crash and reboot of that switch. This crash can happen in the software running on the CPU in the management CPU or on the software running on the CPUs in the interfaces. In either case, crash information is generated and the switch is rebooted.

The resiliency of the stack is determined by the stacking topology, however, in all cases, the interfaces/ports on the switch that crashes are brought down and a reboot of that switch occurs.

The following table describes how the stack reacts to the crashing switch, depending on what role the switch had when the crash occurred. The assumption in this table is that the topology is a resilient topology (that is, a mesh or ring).

Stacking role	Description
Commander	• The standby takes over as the new Commander • A new standby is elected • Crashing switch writes core file to local stable storage • Crashing switch reboots and joins the stack • Core file and crash information for this switch is available from the Commander
Standby	• A new standby is elected • Crashing switch writes core file to local stable storage • Crashing switch reboots and joins the stack • Core file and crash information for this switch is available from the Commander
Member	• Crashing switch writes core file to local stable storage • Crashing switch reboots and joins the stack • Core file and crash information for this switch is available from the Commander

After a switch crashes, you can collect data to help understand why the crash occurred. The information is a combination of crash data, crash log, and core-dump files. The `show tech` command displays logs of events that happened right before the crash.

Troubleshooting an unexpected Commander or Standby switch selection

Action

Viewing the running configuration with priority

When a switch stack is established and a boot/reboot of the stack is performed, the Commander and Standby are selected based on the configured switch priority. There are other rules in the election process that can override this priority.

```
switch(config)# show running-config

Running configuration:

; Stack_WC Configuration Editor; Created on release #WC.16.04.0000x
; Ver #10:9b.3f.bf.bb.ef.7c.59.fc.6b.fb.9f.fc.ff.ff.37.ef:ad

stacking
   member 1 type "JL321A" mac-address 941882-d83c40
   member 2 type "JL319A" mac-address 941882-d91a40
   member 3 type "JL321A" mac-address 941882-d9a240
   member 3 priority 200
   member 3 flexible-module A type JL078A
   member 4 type "JL319A" mac-address 941882-d9e900
   member 4 priority 175
   member 4 flexible-module A type JL078A
   member 5 type "JL319A" mac-address 941882-da0f40
   member 5 flexible-module A type JL083A
   exit
hostname "Switch"
snmp-server community "public" unrestricted
oobm
   ip address dhcp-bootp
   member 1
-- MORE --, next page: Space, next line: Enter, qu      ip address dhcp-bootp
      exit
   member 2
      ip address dhcp-bootp
      exit
   member 3
      ip address dhcp-bootp
      exit
   member 4
      ip address dhcp-bootp
      exit
   member 5
      ip address dhcp-bootp
      exit
   exit
vlan 1
   name "DEFAULT_VLAN"
   untagged 1/1-1/48,2/1-2/24,3/1-3/48,3/A1,4/1-4/24,4/A1,5/A1-5/A4
   ip address dhcp-bootp
   exit
```

On a boot of the stack, member 3 becomes a Commander and member 4 becomes a Standby, based on priority. If this were a chain with member 1 at one end of the chain and member 5 at the other end, the number of hops between switches will be part of the election process.

Managing interactions with other switch features

This section describes about how to manage the interactions with other switch features.

Managing SSH or Telnet sessions

Switches in a non-stacking configuration support up to six sessions running SSH or Telnet concurrently. However, if stacking is configured, each stacking connection reduces the number of sessions available. For example, five connections into the stack leaves only one session available for SSH or Telnet.

Managing switch-level configuration interactions

In a stack, the Commander functions as a single switch and the Standby and Members function as additional network ports for that switch. Switch configuration is performed in the same manner as for any other switch, as described in these manuals:

- *ArubaOS-Switch Software Basic Operation Guide*
- *ArubaOS-Switch Software Management and Configuration Guide*
- *ArubaOS-Switch Access Security Guide*
- *ArubaOS-Switch Software Multicast and Routing Guide*
- *ArubaOS-Switch Software Access Security Guide*
- *ArubaOS-Switch Software IPv6 Configuration Guide*

Managing port-level configuration interactions

For features that are configured on specific switch ports in a stack, the configuration procedures are the same as for stand-alone switches, but the port designations for the ports in the stack are modified. Each port is identified by the stack member ID of its switch, followed by a slash and then the port number as it is shown on the switch. For example, for a switch with stack member ID 3, port 10 on that switch would be identified as port 3/10.

Viewing `show interfaces brief` output for port 3/10

```
switch(config)# show interfaces brief 3/10

 Status and Counters - Port Status

                         | Intrusion                           MDI  Flow Bcast
  Port          Type     | Alert     Enabled Status Mode       Mode Ctrl Limit
  ------------ --------- + --------- ------- ------ ---------- ---- ---- -----
  3/10          100/1000T | No        Yes     Down   1000FDx         off  0
```

Similarly, CLI commands requiring specific port (interface) numbers on a 2930M switch configured for stacking require the modified port designations. For example, to enter the port context for port 10 on stack member 2, type:

```
switch(config)# interface 2/10
switch(eth-2/10)#_
```

In the output containing designated port numbers for a 2930M switch configured for stacking, the port numbers are likewise listed in the modified format.

Viewing `show interfaces config` output

```
switch(config)# show interfaces config

 Port Settings

  Port   Type      | Enabled Mode         Flow Ctrl MDI
  ------ --------- + ------- ------------ --------- ----
  1/1    100/1000T | Yes     Auto         Disable   Auto
  1/2    100/1000T | Yes     Auto         Disable   Auto
  1/3    100/1000T | Yes     Auto         Disable   Auto
   .
   .
   .
  2/1    100/1000T | Yes     Auto         Disable   Auto
  2/2    100/1000T | Yes     Auto         Disable   Auto
  2/3    100/1000T | Yes     Auto         Disable   Auto
  2/4    100/1000T | Yes     Auto         Disable   Auto
.
.
.
```

Attempting to enter a CLI command for a port on a stack member without using the modified port number format generates a “Module not present...” message such as the following:

```
switch(config)# interface 10
Module not present for port or invalid port: 10
```

LACP support

LACP trunking can support up to 60 trunks in a stacking configuration, each with up to eight links (ports) per trunk.

```
switch(config)# trunk 1/1,1/2,1/3,1/4,1/5,1/6,1/7,1/8  trk60 lacp
switch(config)# trunk 1/9 trk60 lacp
An attempt to exceed the maximum number (8) of ports in a trunk group.


switch(config)# trunk 1/2
trk1                  Trunk group 1
trk2                  Trunk group 2
trk3                  Trunk group 3
trk4                  Trunk group 4
trk5                  Trunk group 5
trk6                  Trunk group 6
trk7                  Trunk group 7
trk8                  Trunk group 8
trk9                  Trunk group 9
trk10                 Trunk group 10
trk11                 Trunk group 11
trk12                 Trunk group 12
trk13                 Trunk group 13
trk14                 Trunk group 14
trk15                 Trunk group 15
trk16                 Trunk group 16
trk17                 Trunk group 17
trk18                 Trunk group 18
trk19                 Trunk group 19
trk20                 Trunk group 20
trk21                 Trunk group 21
trk22                 Trunk group 22
trk23                 Trunk group 23
trk24                 Trunk group 24
trk25                 Trunk group 25
trk26                 Trunk group 26
trk27                 Trunk group 27
trk28                 Trunk group 28
trk29                 Trunk group 29
trk30                 Trunk group 30
trk31                 Trunk group 31
trk32                 Trunk group 32
trk33                 Trunk group 33
trk34                 Trunk group 34
trk35                 Trunk group 35
trk36                 Trunk group 36
trk37                 Trunk group 37
trk38                 Trunk group 38
trk39                 Trunk group 39
trk40                 Trunk group 40
trk41                 Trunk group 41
trk42                 Trunk group 42
trk43                 Trunk group 43
trk44                 Trunk group 44
trk45                 Trunk group 45
trk46                 Trunk group 46
trk47                 Trunk group 47
trk48                 Trunk group 48
trk49                 Trunk group 49
trk50                 Trunk group 50
trk51                 Trunk group 51
trk52                 Trunk group 52
trk53                 Trunk group 53
trk54                 Trunk group 54
```

```
trk55                  Trunk group 55
trk56                  Trunk group 56
trk57                  Trunk group 57
trk58                  Trunk group 58
trk59                  Trunk group 59
trk60                  Trunk group 60
```

Managing OOBM ports

Each OOBM port of a member is assigned one MAC address from that Member's manufacturing allocated range. The OOBM port also can be assigned an IP address (IPv4 /v6/DHCP/Manual/Auto-Config/LinkLocal). The Commander's OOBM IP address (called the Global IP address) is used for managing the Commander through the OOBM port.

After switchover/failover of control from the Commander to the Standby, the OOBM port IP address of the new Commander is the Global IP address. This change in address causes some undesirable behavior (after failover):

- When using DHCP or DHCPv6, the new Commander requests a new lease and typically receives a new network address (IPv4 or IPv6). With OOBM high availability (HA), it will seem as if a new link has come up requesting a network address.
- IPV6 link-local or auto-config addresses will change.

Using a static IP address avoids these issues. During failover, it will be as if the IP address is reconfigured. All ARP entries are updated automatically.

For more information on OOBM operation, see the *ArubaOS-Switch Management and Configuration Guide* for your switch.

Understanding stacking election

Electing a Commander

When the Commander of the stack is not identified, the stack undergoes a Commander Election process. This occurs when the entire stack is rebooted simultaneously, such as during a building power failure recovery, or when the stack becomes split and the Commander is isolated in the Inactive fragment, requiring the Active fragment to elect a new Commander.

All of the switches go through discovery and election at the same time. There is an election timer that is set for 60 seconds, and if there are no new switches discovered during that timeout period, the switches in the stack enter the election phase.

During the election process, for each group of switches that has the same STACK-ID (they should all be the same), these steps occur:

1. The switches with the highest Stack Revision are discovered.
2. The switch with the highest configured priority is selected as the Commander.
3. If there are switches with the same “highest” priority, the switch that was the previous Commander is selected.
4. If no switches were previous commanders, the switch that was the previous Standby is selected.
5. If none of the above conditions apply, the switch with the lowest MAC Address is selected as the Commander.

Electing a Standby

The Standby switch is selected by the Commander following the same rules used to elect the Commander. Like the Commander, the Standby switch is not changed unless a failure occurs (for example, the Standby switch fails, or the Commander fails and the Standby becomes the Commander).

NOTE:

Since the Commander will update the revision software and set the stack IDs of all the switches, this information will be the same for all Standby switch contenders.

The criteria used by the Commander to select the Standby is in this priority order:

1. A switch with the same system revision software as the commander is available. This speeds up the initial boot since the stack will not have to wait for the standby to be updated. (If this were not the case, then the selected Standby would need to be reloaded with the new system and rebooted, resulting in the selection of a new Standby. This process would continue until either the original Standby was rebooted or a Standby was chosen that already had the correct system revision).
2. For all switches with the Commander's revision software, the switch with the highest priority that is not the current Commander will become the Standby switch.

   NOTE:

   It is possible for the Standby to have a higher priority than the Commander, if the priority of the Standby was increased after the Commander becomes the Commander. (The Commander is not changed unless it fails or is on the Inactive fragment side of a stack that becomes split).
3. If there are two or more switches whose priority is equally high, then the Commander will look at the topology of the stack and pick a switch that is the most hops away from the Commander. This will increase the probability that the Commander and Standby switch will be in different stack fragments should a failure occur.
4. If the priorities and hop counts of the contenders are the same, then the switch with the lowest MAC address is selected as the Standby switch.

Flexible Uplink Modules

The uplink speed and capacity can be changed by adding or removing the uplink modules. The Flexible Uplink Modules (FUP) that are supported on these switches are JL078A, JL081A, and JL083A modules. The speed and the number of ports depend on the selected uplink module.

All physically inserted flexible modules are auto-detected when you boot a 2930M switch with the default configuration (factory reset).

Flexible Ports

The following types of flexible modules are supported on the 2930M switch series:

- Aruba 2930M 1-port QSFP+ 40GbE Module (JL078A)
- Aruba 2930M 4 HPE Smart Rate PoE+ Module (JL081A)
- Aruba 2930M 4-port 100M/1G/10G SFP+ MACsec Module (JL083A)

NOTE: Only copper and fibre optic media types are supported. Auto media selection, such as detecting 1G SFP in 10G SFP+ FlexPort, is also supported on 4x10G SFP+ FlexPort.

Naming conventions for FUP

FUP uses the same naming convention as the expansion module ports in Aruba 2920 switch.

Naming FUP with stacking enabled

For switches with stacking enabled, the naming convention for FUP is:

`1/A1, 1/A2...1/AN, 2/A1,2/A2...2/AN...10/A1,10/A2...10/AN`

Where:

`N` represents number of ports in the flexible module.

`A` represents the flexible module names.

`1, 2...10` prefix represents the member number that slot ports belong to.

Naming FUP in standalone

For switches with stacking disabled (standalone), the naming convention for FUP is:

`A1, A2, A3...AN`

Where:

`N` represents number of ports in the flexible module.

`A` represents the flexible module names.

Provisioning FUP

Use the CLI commands to provision the FUP for both standalone and with stacking enabled.

Provisioning FUP with stacking enabled

Use the CLI command to provision an FUP for an already provisioned or an existing member configured with the specified flexible module type defined by the corresponding J-number.

Syntax

```
stacking member <stack-member> flexible-module <SLOT-ID> type <J-NUMBER>
```

Description

Specifies the stack member to be configured.

Parameters

- `stack-member`

  Specify the `stack-member` in the range 1–10.

- `SLOT-ID`

  Specify an alphabetic device slot identifier A.

- `J-NUMBER`

  Configure the type of the flexible module to be provisioned.

    JL078A
    JL081A
    JL083A

Preconfigure a flexible module before installing, as it allows the ports to be preconfigured and ready to use. If the modules are not preconfigured, the default configuration is used. If a module other than the preconfigured module type is inserted, the module is not activated.

Example output

```
switch(config)# stacking member
 <1-10>                   Enter a stack member-ID for the 'member'
                          command/parameter
 switch(config)# stacking member 1
 flexible-module       Pre-configure a flexible module before installing it.
 priority              Assigns a priority to the specified stack member.
 remove                Erases the stack member's configuration.
 shutdown              Shut down the specified stack member.
 type                  Configure the family of the switch being provisioned.


 switch(config)# stacking member 1 flexible-module
 SLOT-ID               Enter an alphabetic device slot identifier.


 switch(config)# stacking member 1 flexible-module A
 remove                Erases the flexible module configuration.
 type                  Configure the type of the flexible module being
                       provisioned

 switch(config)# stacking member 1 flexible-module A type
 JL078A
 JL081A
 JL083A

 switch(config)# stacking member 1 flexible-module A type JL078A
 switch(config)# show running-config

Running configuration:

; Stack_WC Configuration Editor; Created on release #WC.16.04.0000x
; Ver #10:0b.bf.bf.bb.ef.7c.59.fc.6b.fb.9f.fc.ff.ff.37.ef:bd

stacking
   member 1 type "JL322A" mac-address e0071b-dab580
   member 1 flexible-module A type JL078A
   member 2 type "JL321A" mac-address e0071b-de7dc0
   member 3 type "JL321A" mac-address e0071b-de8d00
   member 3 flexible-module A type JL081A
   member 4 type "JL321A" mac-address e0071b-def840
   exit
hostname "Switch"
snmp-server community "public" unrestricted
oobm
   ip address dhcp-bootp
   member 1
      ip address dhcp-bootp
      exit
   member 2
      ip address dhcp-bootp
      exit
   member 3
      ip address dhcp-bootp
      exit
   member 4
      ip address dhcp-bootp
      exit
   exit
vlan 1
   name "DEFAULT_VLAN"
   untagged 1/1-1/48,1/A1,2/1-2/48,3/1-3/48,3/A1-3/A4,4/1-4/48
   ip address dhcp-bootp
   exit
```

Provisioning FUP with stacking disabled

Use the CLI command to provision an FP module in a standalone environment, with the specified flexible module type defined by the corresponding J-number. Preconfigure a flexible module before installing, as it allows the ports to be preconfigured and ready to use. If the modules are not preconfigured, the default configuration is used. If a module other than the preconfigured module type is inserted, the module is not activated.

Syntax

```
flexible-module <SLOT-ID> {type <J-NUMBER>}
```

Description

Specify a flexible module to be configured.

Parameters

- `SLOT-ID`

  Specify an alphabetic device slot identifier A.

- `J-NUMBER`

  Configure the type of the flexible module to be provisioned.

  JL078A
  JL081A
  JL083A

Example output

```
switch(config)# flexible-module
SLOT-ID                Enter an alphabetic device slot identifier

switch(config)# flexible-module A
remove                 Erases the flexible module configuration.
type                   Configure the type of the flexible module being
                       provisioned


switch(config)# flexible-module A type
JL078A
JL081A
JL083A

switch(config)# flexible-module A type JL078A

switch(config)# show running-config

Running configuration:

; JL322A Configuration Editor; Created on release #WC.16.05.0000x
; Ver #10:9b.3f.bf.bb.ef.7c.59.fc.6b.fb.9f.fc.ff.ff.37.ef:ad

hostname "Aruba-2930M-48G-PoEP"
module 1 type jl322a
flexible-module A type JL081A
include-credentials
snmp-server community "public" unrestricted
snmpv3 engineid "00:00:00:0b:00:00:e0:07:1b:e5:6b:00"
oobm
   ip address dhcp-bootp
   exit
vlan 1
```

```
   name "DEFAULT_VLAN"
   untagged 1-48,A1-A4
   ip address dhcp-bootp
   exit
```

Unprovisioning FUP

Use the CLI commands to remove the configured FUP for both standalone and with stacking enabled.

Unprovisioning FUP with stacking enabled

Use the CLI command to remove a provisioned FP module.

Syntax

`Stacking member <STACK-MEMBER> flexible-module <SLOT-ID> remove`

Description

Specify a flexible module to be removed.

Parameters

`Stacking member <STACK-MEMBER> flexible-module <SLOT-ID> remove`

Remove the configuration for the specified flexible module.

- `stack-member`

  Specify the `stack-member` in the range 1–10.

- `SLOT-ID`

  Specify an alphabetic device slot identifier A.

Example output

```
switch(config)# show running-config

Running configuration:

; Stack_WC Configuration Editor; Created on release #WC.16.04.0000x
; Ver #10:0b.bf.bf.bb.ef.7c.59.fc.6b.fb.9f.fc.ff.ff.37.ef:bd

stacking
   member 1 type "JL322A" mac-address e0071b-dab580
   member 1 flexible-module A type JL078A
   member 2 type "JL321A" mac-address e0071b-de7dc0
   member 3 type "JL321A" mac-address e0071b-de8d00
   member 3 flexible-module A type JL081A
   member 4 type "JL321A" mac-address e0071b-def840
   exit
hostname "Switch"
snmp-server community "public" unrestricted
oobm
   ip address dhcp-bootp
   member 1
      ip address dhcp-bootp
      exit
   member 2
      ip address dhcp-bootp
      exit
   member 3
      ip address dhcp-bootp
      exit
   member 4
      ip address dhcp-bootp
      exit
   exit
vlan 1
   name "DEFAULT_VLAN"
   untagged 1/1-1/48,1/A1,2/1-2/48,3/1-3/48,3/A1-3/A4,4/1-4/48
   ip address dhcp-bootp
   exit

   switch(config)#  stacking member 1 flexible-module A remove
   switch(config)#  show running-config

Running configuration:

; Stack_WC Configuration Editor; Created on release #WC.16.04.0000x
; Ver #10:0b.bf.bf.bb.ef.7c.59.fc.6b.fb.9f.fc.ff.ff.37.ef:bd

stacking
   member 1 type "JL322A" mac-address e0071b-dab580
   member 2 type "JL321A" mac-address e0071b-de7dc0
   member 3 type "JL321A" mac-address e0071b-de8d00
   member 3 flexible-module A type JL081A
   member 4 type "JL321A" mac-address e0071b-def840
   exit
hostname "Switch"
snmp-server community "public" unrestricted
oobm
   ip address dhcp-bootp
   member 1
      ip address dhcp-bootp
      exit
   member 2
      ip address dhcp-bootp
      exit
   member 3
      ip address dhcp-bootp
      exit
   member 4
      ip address dhcp-bootp
      exit
   exit
vlan 1
   name "DEFAULT_VLAN"
   untagged 1/1-1/48,2/1-2/48,3/1-3/48,3/A1-3/A4,4/1-4/48
   ip address dhcp-bootp
   exit
```

Unprovisioning FUP with stacking disabled

Use the CLI command to remove an already provisioned FP module and its associated configurations.

Syntax

```
flexible-module <SLOT-ID> {type <J-NUMBER> | remove}
```

Description

Specify a flexible module, for which configuration has to be removed.

Parameters

`SLOT-ID`

Specify an alphabetic device slot identifier A.

Example output

```
switch(config)#  flexible-module A remove

switch(config)# show running-config

Running configuration:

; JL322A Configuration Editor; Created on release #WC.16.05.0000x
; Ver #10:9b.3f.bf.bb.ef.7c.59.fc.6b.fb.9f.fc.ff.ff.37.ef:ad

hostname "Aruba-2930M-48G-PoEP"
module 1 type jl322a
include-credentials
snmp-server community "public" unrestricted
snmpv3 engineid "00:00:00:0b:00:00:e0:07:1b:e5:6b:00"
oobm
   ip address dhcp-bootp
   exit
vlan 1
   name "DEFAULT_VLAN"
   untagged 1-48
   ip address dhcp-bootp
   exit
```

Saving power by turning off FUPs

Use the CLI commands to save power by turning `off` unutilized FUPs.

Saving power by turning off FUP with stacking enabled

Use the CLI command to power `off` the flexible module for a stacking member to save power.

Syntax

```
savepower flexible-module <SLOT-ID> member <STACK-MEMBER>
```

Description

Power flexible modules `off`.

Parameters

- `SLOT-ID`

  Specify an alphabetic device slot identifier A.

- `stack-member`

  Specify the `stack-member` to be configured in the range 1–10.

NOTE: You cannot configure `savepower` for flexible modules on a preprovisioned stack member. Also, configuring LED status for flexible modules is not supported.

Example output

```
switch(config)# show modules

Status and Counters - Module Information


  Stack ID       : 0100e007-1bdab580
  Member
  ID     Slot     Module Description                  Serial Number    Status
  ------ -------- ----------------------------------- ---------------- -------
  1      STK      Aruba JL325A 2p Stacking Module                      Up
  2      STK      Aruba JL325A 2p Stacking Module                      Up
  3      A        Aruba JL081A 4p 10GbE XGT Module    SG6ZGZ4226       Up
  3      STK      Aruba JL325A 2p Stacking Module                      Up
  4      STK      Aruba JL325A 2p Stacking Module                      Up

 switch(config)# savepower flexible-module
 SLOT-ID         Enter an alphabetic device slot identifier.
switch(config)# savepower flexible-module A member 3
This command will shut down the specified modules.  Ports on those
modules will no longer pass traffic and any management traffic to
ports on those modules will be interrupted.  This command may take
up to 2 minutes to power down all specified modules.  Please use the
event log to monitor progress.

Continue (y/n)? y

switch(config)# show modules

Status and Counters - Module Information


  Stack ID       : 0100e007-1bdab580

  Member
  ID     Slot     Module Description                  Serial Number    Status
  ------ -------- ----------------------------------- ---------------- -------
  1      STK      Aruba JL325A 2p Stacking Module                      Up
  2      STK      Aruba JL325A 2p Stacking Module                      Up
  3      A        Aruba  Expansion Module                              Save...
  3      STK      Aruba JL325A 2p Stacking Module                      Up
  4      STK      Aruba JL325A 2p Stacking Module                      Up

  switch(config)# show running-config

Running configuration:

; hpStack_WC Configuration Editor; Created on release #WC.16.04.0000x
; Ver #10:0b.bf.bf.bb.ef.7c.59.fc.6b.fb.9f.fc.ff.ff.37.ef:bd

stacking
   member 1 type "JL322A" mac-address e0071b-dab580
   member 2 type "JL321A" mac-address e0071b-de7dc0
   member 3 type "JL321A" mac-address e0071b-de8d00
   member 3 flexible-module A type JL081A
   member 4 type "JL321A" mac-address e0071b-def840
   exit
hostname "Aruba-Stack-2930M"
snmp-server community "public" unrestricted
oobm
   ip address dhcp-bootp
   member 1
      ip address dhcp-bootp
      exit
   member 2
      ip address dhcp-bootp
      exit
   member 3
      ip address dhcp-bootp
      exit
   member 4
```

```
      ip address dhcp-bootp
      exit
   exit
vlan 1
   name "DEFAULT_VLAN"
   untagged 1/1-1/48,2/1-2/48,3/1-3/48,3/A1-3/A4,4/1-4/48
   ip address dhcp-bootp
   exit
savepower flexible-module A member 3
```

Saving power by turning off FUPs with stacking disabled

Use the CLI command to turn `off` a flexible module with stacking disabled to save power. The flexible module configuration remains unchanged during a savepower operation.

Syntax

`savepower flexible-module <SLOT-ID>`

Description

Power flexible modules `off`.

Option

`SLOT-ID`

Specify an alphabetic device slot identifier A.

Example output

```
switch(config)# savepower flexible-module A

switch(config)# savepower flexible-module A
This command will shut down the specified modules.  Ports on those
modules will no longer pass traffic and any management traffic to
ports on those modules will be interrupted.  This command may take
up to 2 minutes to power down all specified modules.  Please use the
event log to monitor progress.

Continue (y/n)? y
switch(config)# show modules

Status and Counters - Module Information

  Chassis: 2930M-48G-PoE+  JL322A         Serial Number:   SG6JQN02R


  Slot  Module Description                         Serial Number    Status
  ----- ------------------------------------------ ---------------- ----------
  A     Aruba  Expansion Module                                     SavePower

  switch(config)# show running-config

Running configuration:

; JL322A Configuration Editor; Created on release #WC.16.05.0000x
; Ver #10:9b.3f.bf.bb.ef.7c.59.fc.6b.fb.9f.fc.ff.ff.37.ef:ad

hostname "Aruba-2930M-48G-PoEP"
module 1 type jl322a
flexible-module A type JL078A
include-credentials
snmp-server community "public" unrestricted
snmpv3 engineid "00:00:00:0b:00:00:e0:07:1b:e5:6b:00"
oobm
```

```
   ip address dhcp-bootp
   exit
vlan 1
   name "DEFAULT_VLAN"
   untagged 1-48,A1
   ip address dhcp-bootp
   exit
savepower flexible-module A
```

Disabling savepower by turning on FUPs

Use the CLI commands to power `on` FUPs in `savepower` status.

Turning ON FUPs in savepower status with stacking enabled

Syntax

```
no savepower flexible-module <SLOT-ID> member <STACK-MEMBER>
```

Description

Disable save power for FUP in a stack.

Parameters

- `SLOT-ID`

  Specify an alphabetic device slot identifier A.

- `stack-member`

  Specify the `stack-member` to be configured in the range 1–10.

Example output

```
switch(config)# no savepower flexible-module A member 3
switch(config)# show modules

Status and Counters - Module Information


  Stack ID       : 0100e007-1bdab580

  Member
  ID     Slot     Module Description                  Serial Number    Status
  ------ -------- ----------------------------------- ---------------- -------
  1      STK      Aruba JL325A 2p Stacking Module                        Up
  2      STK      Aruba JL325A 2p Stacking Module                        Up
  3      A        Aruba JL081A 4p 10GbE XGT Module    SG6ZGZ4226       Booting
  3      STK      Aruba JL325A 2p Stacking Module                        Up
  4      STK      Aruba JL325A 2p Stacking Module                        Up

  switch(config)# show modules

Status and Counters - Module Information


  Stack ID       : 0100e007-1bdab580

  Member
  ID     Slot     Module Description                  Serial Number    Status
  ------ -------- ----------------------------------- ---------------- -------
  1      STK      Aruba JL325A 2p Stacking Module                        Up
  2      STK      Aruba JL325A 2p Stacking Module                        Up
```

```
 3       A        Aruba JL081A 4p 10GbE XGT Module    SG6ZGZ4226       Up
 3       STK      Aruba JL325A 2p Stacking Module                      Up
 4       STK      Aruba JL325A 2p Stacking Module                      Up
```

Turning ON FUPs in savepower status with stacking disabled

Syntax

`no savepower flexible-module <SLOT-ID>`

Description

Disable save power for a standalone FUP.

Option

`SLOT-ID`

Specify an alphabetic device slot identifier A.

Example output

```
switch(config)# no savepower flexible-module A
```

Changing flexible modules on a running stack

You can change a flexible module in a running stack under the following scenarios:

- **Inserting a flexible module into a running stack** on page 277
- **Booting with flexible module configuration, no flexible modules inserted** on page 278
- **Removing the flexible modules** on page 278
- **Replacing a flexible module** on page 278

There are various options available for each of the above scenarios that helps resolve troubleshooting issues.

Inserting a flexible module into a running stack

You can boot a stack or standalone either with the default or flexible module configuration under various scenarios such as:

Scenario 1

A stack or standalone booted with the default configuration, but flexible modules are not inserted at the time of boot. Hot-Insert a flexible module compatible with the 2930M SKU. The flexible modules are booted and the flexible ports are available for use. The `show modules` command displays the flexible module status as `UP`.

Scenario 2

A stack or standalone booted with the flexible module configuration, but flexible modules are not inserted at the time of boot. Hot-Insert a flexible module of the same type as configured. The flexible modules are booted and flexible ports are available for use. The `show modules` command displays the status of flexible module as `UP`.

Scenario 3

A stack or standalone booted with the flexible module configuration, but flexible modules are not inserted at the time of boot. Hot-Insert a flexible module other than the configured. The `show modules` command displays `Failed State` in the output due to a mismatch. RMON has a log entry for `Misconfigured module`. The flexible ports are unavailable for use.

Scenario 4

A stack or standalone booted with the flexible module configuration, and same type of flexible modules inserted as configured at the time of boot. The flexible modules are booted and the flexible ports are available for use. The `show modules` command displays status of flexible module as `UP`.

Scenario 5

A stack or standalone booted with flexible module configuration and flexible modules other than the configured are inserted at the time of boot. The `show modules` command displays `Failed State` in the output due to a mismatch. RMON has a log entry for `Misconfigured module`. The flexible ports are unavailable for use.

Booting with flexible module configuration, no flexible modules inserted

Procedure

Boot a stack or standalone with the flexible module configuration and no flexible modules inserted at the time of boot. Use the `unprovision` command to remove the existing flexible module, and then Hot-Insert a flexible module that is compatible with 2930M SKU. The flexible modules are booted and the flexible ports are available for use. The `show modules` displays the status of inserted flexible module as `UP`.

Removing the flexible modules

The following scenarios are available when you boot a stack or standalone with the flexible module configuration, followed by removing flexible modules of either same or different type than configured:

Scenario 1

Perform a Hot-Removal of the inserted flexible module. The `show modules` command displays the status of flexible module as `Unknown`.

Scenario 2

Perform a Hot-Removal of the inserted flexible module followed by a Hot-Insert of either same or different type. The flexible ports are unavailable for use.

NOTE: Hewlett Packard Enterprise recommends unprovisioning or provisioning flexible modules as Hot-Swap is not supported in the current release.

Replacing a flexible module

Although, Hot-Swap is not supported in the current release, you can still replace a flexible module on a running stack or standalone. To make the flexible module operational, follow these steps:

Procedure

1. Run the `unprovision` command.
2. Perform a Hot-Removal of the installed flexible module.
3. Hot-Insert a flexible module of any supported type.

The flexible modules are booted and the flexible ports are available for use. The `show modules` command displays the status of inserted flexible module as `UP`.

Saving power for FUPs

You can save power by turning `off` the unused flexible modules under the following scenarios:

Procedure

1. **Booting a switch with no inserted flexible modules, and with or without flexible module configuration** on page 279
2. **Changing flexible modules in savepower status in a running stack** on page 279
3. **Booting a switch with flexible modules inserted, and flexible modules in savepower status** on page 279

Booting a switch with no inserted flexible modules, and with or without flexible module configuration

The following are the various scenarios:

Scenario 1

The `savepower` command is executed, but the `show modules` does not display the flexible modules.

Scenario 2

Insert a flexible module, and then execute `savepower`. The flexible modules are powered `off`. The `show modules` displays the flexible modules status as `savepower`.

Scenario 3

Insert a flexible module, execute `savepower`, and then run `no savepower`. The flexible modules are powered `on` again.

Scenario 4

Execute `savepower`, and then perform Hot-Insert. The `show modules` displays the flexible modules status as `savepower`, but flexible ports are unavailable for use.

Changing flexible modules in savepower status in a running stack

Scenario 1

Unprovision the flexible module. The `show modules` does not display the flexible modules.

Scenario 2

Perform a Hot-Removal of the flexible module. The `show modules` displays the flexible modules status as `Unknown`. The `savepower` configuration is retained for the slot.

NOTE:

In a stacking environment, execute member `remove` on which the `savepower` command is issued. The `show modules` does not display the flexible modules. The `savepower` command is cleared from the configuration.

Scenario 3

Execute `savepower`, and then perform Hot-Insert. The `show modules` displays the flexible modules status as `savepower`, but flexible ports are unavailable for use.

Booting a switch with flexible modules inserted, and flexible modules in savepower status

The `show modules` displays the flexible modules status as `savepower`. The flexible ports are unavailable for use.

Overview of RPVST+

NOTE: For information on configuring basic and multiple instance spanning tree, see **Multiple instance spanning tree operation**.

RPVST+ is a proprietary spanning tree implementation that extends RSTP (802.1w) to run a separate spanning tree for each VLAN on the switch, and ensures that only one active, loop-free path exists between any two nodes on a given VLAN.

Table 22: *RPVST scalability*

Platform	Maximum Allowed RPVST Enabled VLANs	Recommended Maximum Virtual Ports	Maximum Allowed Virtual Ports (x is the number of logical ports in the system)
2930	128	400	2000 + x

Where x is the sum of all physical ports and logical interface, such as Trk1 if configured.

The following shows how x is calculated using the 2920 switch as an example, you can substitute the actual values for your switch as shown in the table to do a similar calculation. In a stack of 4 with 2x2920-24 and 1 trunk interface Trk1 configured, x will be 24+24+1=49. Therefore the maximum allowed vPorts is 299.

```
Switch# show spanning-tree system-limits rapid-pvst

 Spanning Tree Information

  STP Enabled              : No
  Mode                     : MSTP
  RPVST Enabled VLANs      : 1,4,20-23

  Switch MAC Address                   : 40a8f0-0df69e
  Count of RPVST Enabled GVRP VLANs    : 0
  Count of RPVST Enabled VLANs         : 6
  Maximum Allowed RPVST Enabled VLANs  : 64
  Count Of Total Virtual Ports         : 51
  Maximum Allowed Virtual Ports        : 299

                       Current         Operational     Recommended Maximum
  Ports                Virtual Ports   Virtual Ports   Virtual Ports
  -------------------- --------------- --------------- --------------------
  Member 1/1-24        31              27              250
  Member 2/1-24        31              24              250
```

Configuring RPVST+ at a glance

The general steps for configuring RPVST+ via the CLI are:

Procedure

1. Select RPVST+ as the active spanning tree mode by entering the following command: `spanning-tree mode rapid-pvst`

   To begin with the default RPVST+ configuration (recommended), go to step 6.

2. Configure global spanning tree parameters.
3. Configure per-VLAN parameters.
4. Configure per-port per-VLAN parameters. These commands affect RPVST+ operation on traffic associated with the specified VLAN(s) through the specified port(s).
5. Configure per-port parameters. These commands affect RPVST+ operation for all traffic through the specified port(s).
6. Use one of the following commands to enable RPVST+ spanning tree operation on the switch:

   a. One or more selected VLANs:

      `spanning-tree vlan vid-list`

   b. One or more selected VLANs:

      `spanning-tree vlan vid-list`

   c. The first 400 VLANs:

      `spanning-tree`

      Any VLANs in excess of the first 400 would have RPVST+ disabled. In this case, use the `no` spanning-tree vlan *vid-list* command to change the mix of RPVST+ enabled and disabled VLANs.

Additional configuration options include:

- **Allowing traffic on VLAN ID (PVID) mismatched links** on page 286
- **Configuring STP loop guard** on page 287

Selecting RPVST+ as the spanning tree mode

Syntax:

```
spanning-tree mode [mstp | rapid-pvst]
```

```
no spanning-tree mode [mstp | rapid-pvst]
```

Specifies that spanning tree will run in MSTP (default) or RPVST+ mode.

To view Mode, use the `show run` command. This will eliminate confusion if there is an RPVST configuration but MSTP is running. This will lead to a change in the existing factory default setting.

RPVST+ parameters can be configured even if the mode is MSTP and vice versa. This command does not enable/disable spanning tree. It sets the mode which is operational once spanning tree is enabled using `spanning-tree enable`.

The `no` form of the command changes the spanning tree mode to the default mode (MSTP) .

Configuring global spanning tree

Syntax:

```
spanning-tree extend system-id
```

Creates a unique bridge identifier for each VLAN by adding the VLAN ID (vid) value to the priority field of the bridge identifier in every RPVST+ BPDU.

Syntax

```
no spanning-tree log state-transitions [instance <instance-id> cst]
```

- The command enables or disables event logging for port-block events.
- List of VLAN identifiers
- Range: <instance-id> 1–16

  ```
  [vlan <vid-list>]
  ```

Syntax:

```
no spanning-tree ignore-pvid-inconsistency
```

Causes the switch to ignore per-VLAN ID inconsistencies when the ports on both ends of a point-to-point link are untagged members of different VLANs, thus allowing RPVST+ to run on the mismatched links. On a given switch, affects all ports belonging to VLANs on which RPVST+ is enabled.

Default: Disabled

NOTE: The `no` form of this command is ineffective when there is a PVID inconsistency between a VLAN1 and any non-VLAN1 member because VLAN1 uses IEEE BPDUs to form a spanning tree topology.

Syntax:

```
no spanning-tree bpdu-protection-timeout timeout
```

- Configures the duration of time when protected ports receiving unauthorized BPDUs will remain disabled. The default value of 0 (zero) sets an infinite timeout (that is, ports that are disabled by `bpdu-protection` are not, by default, re-enabled automatically).
- Default: 0
- Range: 0 - 65535 seconds

Configuring per-VLAN spanning tree

Syntax:

```
spanning-tree vlan <vid-list> hello-time 1...10
```

Specifies the time in seconds between transmissions of BPDUs on the specified VLAN(s) when the switch is root for those VLAN(s).

Default: 2

Range: 1 - 10

Syntax:

```
spanning-tree vlan <vid-list>forward-delay 4...30
```

Sets the time in seconds the switch waits before transitioning from listening to learning and from learning to forwarding states.

Default: 15

Range: 4 - 30

Syntax:

```
spanning-tree vlan vid-listmaximum age 6...40
```

Sets the maximum age in seconds of received STP information before it is discarded for specified VLAN(s).

Default: 20

Range: 6 - 40

NOTE: `Maximum age` must be within the following bounds:

- greater than or equal to 2x (`hello-time` +1)
- less than or equal to 2x (`forward-delay` - 1)

Syntax:

```
spanning-tree vlan vid-list priority 0...15
```

Sets the switch (bridge) priority for the designated VLAN. The switch compares this priority with the priorities of other switches on the same VLAN to determine the RPVST+ root switch for the VLAN. The lower the priority value, the higher the priority. The switch with the lowest Bridge Identifier on the VLAN is elected as the RPVST+ root switch for that VLAN.

The Bridge Identifier is composed of a configurable Priority (2 bytes) and the switch's MAC address (6 bytes). You can change the Priority provides flexibility for determining which switch on the VLAN will be the root for RPVST+, regardless of its MAC address.

The priority range for an RPVST+ switch is 0-61440. However, this command specifies the priority as a multiplier (0 - 15) of 4096. That is, when you specify a priority multiplier value of 0 - 15, the actual priority assigned to the switch is: (priority-multiplier) x 4096.

For example, if you configure "2" as the priority-multiplier on a given RPVST+ switch, then the Switch Priority setting for the specified VLAN is 8,192.

NOTE: If multiple switches on the same VLAN have the same priority setting, then the switch with the lowest MAC address becomes the root switch for that VLAN.

Syntax:

```
spanning-tree vlan vid-list root {primary | secondary}
```

```
no spanning-tree vlan vid-list root {primary | secondary}
```

Specifies the switch as the primary or secondary root bridge for the specified VLAN(s). Otherwise, by default, the root bridge for each VLAN will be determined by the lowest MAC address in that topology.

The `no` form of the command returns the determination of root to the lowest MAC address criterion.

Configuring per-port per-VLAN spanning tree

Syntax

```
spanning-tree pathcostrapid-pvst | mstp[8021d | 8021t | proprietary]
```

```
no spanning-tree pathcostrapid-pvst | mstp[8021d | 8021t | proprietary]
```

Specify a standard to use when calculating the default pathcost.

Default: 8021t

NOTE: All devices in the network should be configure to use same pathcost mode for proper functioning.

Syntax:

```
spanning-tree port port-#: vlan vid-list path-cost {auto | [1...200000000]}
```

```
no spanning-tree port port-#: vlan vid-list path-cost {auto | [1...200000000]}
```

Sets the path cost for a single port on the specified VLAN(s). If the port is a member of more than one VLAN, the `path-cost` applies only where the port has traffic for the VLAN(s) specified.

Default: auto

Range: 1 - 200000000

The `no` form of the command returns `path-cost` to its default setting.

Syntax:

```
spanning-tree port port-number vlan vid-list priority <0-15> path-costauto | <Path-Cost>
```

```
no spanning-tree port port-number vlan vid-list priority <0-15> path-costauto | <Path-Cost>
```

Sets the port priority for the specified VLANs. The value is in the range of 0-240 divided into steps of 16 that are numbered 0 to 15. The default is step 16.

The per-port per-VLAN priority is used to help choose the root port for a switch on the specified VLAN if there are multiple links to the root switch.

Default: 8

Range 0 - 15

The `no` form of the command sets the priority to its default value.

Configuring per-port spanning tree

Syntax:

```
spanning-tree port-list admin-edge-port
```

```
no spanning-tree port-list admin-edge-port
```

Enables `admin-edge-port` on ports connected to end nodes. During spanning tree establishment, ports with `admin-edge-port` enabled transition immediately to the forwarding state. If a bridge or switch is detected on the segment, the port automatically operates as non-edge, not enabled.

If `admin-edge-port` is disabled on a port and `auto-edge-port` has not been disabled, the `auto-edge-port` setting controls the behavior of the port.

Default: No - disabled

The `no` form of the command disables edge-port operation on the specified ports.

Syntax:

```
spanning tree port-list auto-edge-port
```

```
no spanning tree port-list auto-edge-port
```

Enables or disables the automatic identification of edge ports. The port will look for BPDUs for 3 seconds. If there are none, it begins forwarding packets. If `admin-edge-port` is enabled for a port, the setting for auto-edge-port is ignored whether set to yes or no. If `admin-edge-port` is set to No, and `auto-edge- port` has not been disabled (set to No), then the auto-edge-port setting controls the behavior of the port.

Default: Yes - enabled

The `no` form of the command disables `auto-edge-port operation` on the specified ports.

Syntax:

```
no spanning tree port-list bpdu-filter
```

Enables or disables BPDU filtering on the specified port(s). The `bpdu-filter` option forces a port to always stay in the forwarding state and be excluded from standard STP operation.

Default: Disabled

Syntax:

Enables or disables BPDU protection on the specified port(s).

Syntax:

```
spanning tree port-list point-to-point-mac [true | false | auto]
```

Informs the switch of the type of device to which a specific port connects.

`true` (default)

Indicates a point-to-point link to a device such as a switch, bridge, or end-node.

`false`

Indicates a connection to a hub (which is a shared LAN segment).

`auto`

Causes the switch to set Force-False on the port if it is not running at full duplex. (Connections to hubs are half-duplex.)

Syntax:

```
spanning tree port-list root-guard
```

This feature is available in RPVST+ only. When a port is enabled as `root-guard`, it cannot be selected as the root port even if it receives superior STP BPDUs. The port is assigned an “alternate” port role and enters a blocking state if it receives superior STP BPDUs. (A superior BPDU contains “better” information on the root bridge and path cost to the root bridge, which would normally replace the current root bridge selection.)

The superior BPDUs received on a port enabled as `root-guard` are ignored. All other BPDUs are accepted and the external devices may belong to the spanning tree as long as they do not claim to be the Root device. Use the following command on RPVST+ switch ports that are connected to devices located in other administrative network domains to ensure the stability of the core RPVST+ network topology so that undesired or damaging influences external to the network do not enter.

Default: Disabled.

Syntax:

```
spanning-tree port-list tcn-guard
```

When `tcn-guard` is enabled for a port, it causes the port to stop processing or propagating received topology change notifications and topology changes to other ports.

Default: Disabled.

Enabling or disabling RPVST+ spanning tree

With the spanning tree mode set to RPVST+, you can do either of the following:

- Enable or disable RPVST+ on all VLANs on the switch.
- Enable or disable RPVST+ on specified VLANs that are RPVST+-enabled on the switch.

Syntax:

```
no spanning-tree [enable | disable]
```

To globally enable RPVST+ on all VLANs on the switch, use either of the following:

```
spanning-tree [enable]

no spanning-tree disable
```

To globally disable RPVST+ on all VLANs on the switch, use any of the following:

```
no spanning-tree

spanning-tree disable

no spanning-tree enable
```

NOTE: This status will always be shown in `show run` to let you know whether the spanning-tree is enabled. Having spanning tree present but not enabled will lead to a change in the existing factory default settings.

NOTE: This command overrides the per-VLAN enable/disable command (below).

Syntax:

```
spanning-tree vlan vid list [enable | disable]
```

To enable RPVST+ on one or more VLANs on the switch, use either of the following:

```
spanning-tree vlan vid list enable

no spanning-tree vlan vid list disable
```

To disable RPVST+ on one or more VLANs on the switch, use any of the following:

```
no spanning-tree vlan vid list

spanning-tree vlan vid listdisable

no spanning-tree vlan vid list enable
```

Allowing traffic on VLAN ID (PVID) mismatched links

When RPVST+ is running in the default configuration on a link where there is a VLAN ID mismatch, PVST blocks the link, resulting in traffic on the mismatched VLANs being dropped. However, there can be instances where traffic passing between mismatched VLANs on a link is desirable. When enabled on the switch, the `ignore-pvid-inconsistency` command allows this behavior. That is, where the ports on both ends of a point-to-point link are untagged members of different VLANs, enabling `ignore-pvid-inconsistency` enables RPVST+ to process untagged RPVST+ BPDUs belonging to the peer's untagged VLAN as if it was received on the current device's untagged VLAN.

Syntax:

```
no spanning-tree ignore-pvid-inconsistency
```

Causes the switch to ignore per-VLAN ID inconsistencies when the ports on both ends of a point-to-point link are untagged members of different VLANs, thus allowing RPVST+ to run on the mismatched links. On a given switch, this affects all ports belonging to VLANs on which RPVST+ is enabled.

Default: Disabled

RPVST+ behavior

Table 23: *RPVST+ behavior with ignore-pvid-inconsistency enabled*

Switch “A” Port on VLAN X	Switch “B” Peer port on VLAN Y	RPVST+ behavior with ignore-pvid-inconsistency enabled
Untagged on VLAN 10	Untagged on VLAN 10	Forward
Untagged on VLAN 10	Untagged on VLAN 20	Forward[11,]
Untagged on VLAN X	Tagged on VLAN X	Drop
Untagged on VLAN X	Tagged on VLAN Y	Drop (traffic from both VLANs)
Tagged on VLAN X	Tagged on VLAN X	Forward[1]
Tagged on VLAN X	Tagged on VLAN Y	Drop (traffic from both VLANs)

[1] Forwarding state applies if the link has not been blocked by RPVST+ as a redundant link.

NOTE: The `no spanning-tree ignore-pvid-inconsistency` command is ineffective when there is a PVID inconsistency between a VLAN1 and any non-VLAN1 member because VLAN1 uses IEEE BPDUs to form a spanning tree topology.

Configuring STP loop guard

Spanning tree is used to ensure a loop-free topology over the LAN. Occasionally a hardware or software failure can cause STP to fail, creating STP/ forwarding loops that can cause network failures where unidirectional links are used. The non-designated port transitions in a faulty manner because the port is no longer receiving STP BPDUs.

STP Loop Guard causes the non-designated port to go into the STP loop inconsistent state instead of the forwarding state. In the loop-inconsistent state, the port prevents data traffic through the link, therefore avoiding the loop creation. When BPDUs again are received on the inconsistent port, it resumes normal STP operation automatically.

Syntax:

```
spanning-tree port-list loop-guard
```

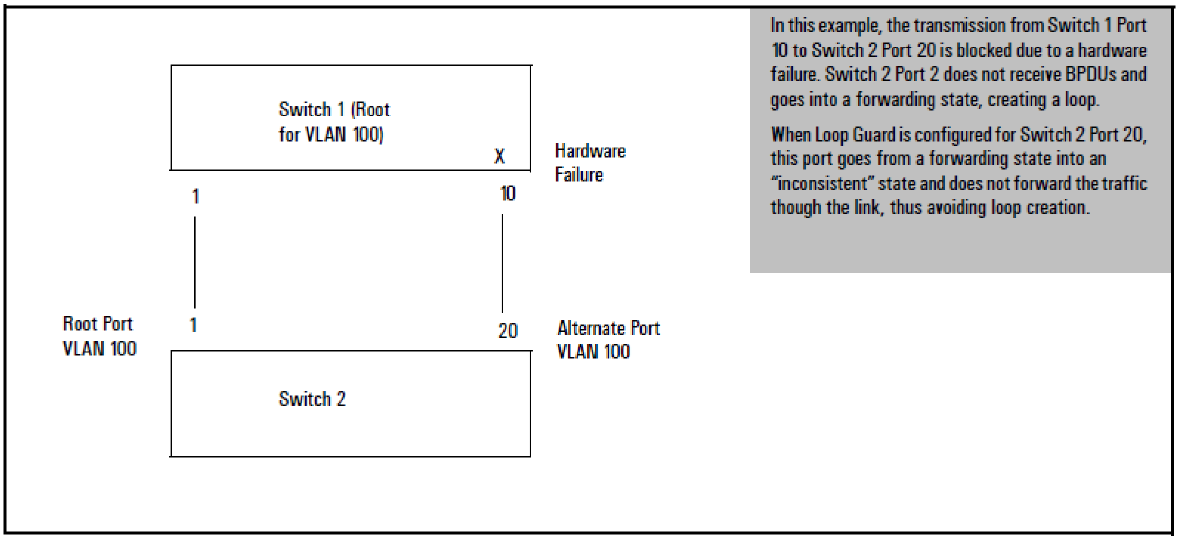
```
no spanning-tree port-list loop-guard
```

Enables STP Loop Guard on a particular port or ports. STP Loop Guard is best applied on blocking or forwarding ports.

The `no` form of the command disables STP Loop Guard.

Default: Disabled

Figure 35: *Loop creation with transmission failure*



In this example, the transmission from Switch 1 Port 10 to Switch 2 Port 20 is blocked due to a hardware failure. Switch 2 Port 2 does not receive BPDUs and goes into a forwarding state, creating a loop.

When Loop Guard is configured for Switch 2 Port 20, this port goes from a forwarding state into an "inconsistent" state and does not forward the traffic though the link, thus avoiding loop creation.

Before configuring loop guard

Before configuring Loop Guard on port 20, the status of VLAN 20 appears as follows:

```
switch(config)# show spanning-tree vlan 20

 Spanning Tree Information

  STP Enabled                [No] : Yes
  Mode                            : RPVST
  Extended System ID              : Enabled
  Ignore PVID Inconsistency       : Disabled
  Switch MAC Address              : 002347-c651c0


  VLAN ID                         : 20
  RPVST Enabled                   : Enabled


  Root MAC Address                : 0024a8-d13a40
  Root Priority                   : 32,768
  Root Path Cost                  : 20,000
  Root Port                       : 1
  Operational Hello Time (secs)   : 2
  Topology Change Count           : 2
  Time Since Last Change          : 9 secs

                                                           Designated
  Port   Type       Cost   Priority  Role       State      Bridge
  -----  ---------- ------ --------- ---------- ---------- ---------------
   1     100/1000T  20000  128       Root       Forwarding 0024a8-d13a40
   20    10/100TX   200000 128       Alternate  Blocking   002347-587b80
```

After configuring loop guard

By executing `spanning-tree 20 loop-guard`, loop guard has been configured on port 20 of Switch 2:

```
switch(config)# show spanning-tree

 Spanning Tree Information

  STP Enabled          [No] : Yes
  Mode                      : RPVST
  Extended System ID        : Enabled
  Ignore PVID Inconsistency : Disabled
  RPVST Enabled VLANs       : 20


  Switch MAC Address        : 002347-c651c0
  Root Guard Ports          :
  Loop Guard Ports          : 20
  TCN Guard Ports           :
  BPDU Protected Ports      :
  BPDU Filtered Ports       :
  Auto Edge Ports           : 1-24
  Admin Edge Ports          :

  VLAN  Root Mac        Root       Root       Root                 Hello
  ID    Address         Priority   Path-Cost  Port                 Time(sec)
  ----- --------------- ---------- ---------- -------------------- ---------
  100   0024a8-d13a40   32,768     20,000     1                    2
```

Switch ceasing to send BPDUs

With switch 1 ceasing to send BPDUs through port 20 to switch 2, port 20 goes into the “inconsistent” state and ceases to forward traffic, as displayed in the following `show spanning-tree` output for VLAN 20.

```
switch(config)# show spanning-tree vlan 20

 Spanning Tree Information

  STP Enabled              [No] : Yes
  Mode                          : RPVST
  Extended System ID            : Enabled
  Ignore PVID Inconsistency     : Disabled
  Switch MAC Address            : 002347-c651c0


  VLAN ID                       : 20
  RPVST Enabled                 : Enabled


  Root MAC Address              : 0024a8-d13a40
  Root Priority                 : 32,768
  Root Path Cost                : 20,000
  Root Port                     : 1
  Operational Hello Time (secs) : 2
  Topology Change Count         : 3
  Time Since Last Change        : 42 hours

                                                        Designated
  Port  Type       Cost   Priority  Role       State      Bridge
  ----- ---------- ------ --------- ---------- ---------- ---------------
  1     100/1000T  20000  128        Root       Forwarding 0024a8-d13a40
  20    10/100TX   200000 128        Alternate  Inconsi... 002347-587b80
```

Viewing configuration file with loop guard enabled

The following example displays `show spanning-tree config` output with loop guard enabled on Port 20:

```
switch(config)# show spanning-tree config

 Spanning Tree Information

  STP Enabled          [No] : Yes
  Mode                      : RPVST
  Extended System ID        : Enabled
  Ignore PVID Inconsistency : Disabled
  RPVST Enabled VLANs       : 100


  Switch MAC Address        : 002347-c651c0

  Root Guard Ports          :
  Loop Guard Ports          : 20
  TCN Guard Ports           :
  BPDU Protected Ports      :
  BPDU Filtered Ports       :
  Auto Edge Ports           : 1-24
  Admin Edge Ports          :

                Max Age Forward    Hello     Admin Root
  VLAN Priority (sec)   Delay(sec) Time(sec) Bridge
  ---- -------- ------- ---------- --------- ----------------
  100  32768    20      15         2         Not Configured
```

About RPVST+

RPVST+ is a proprietary spanning tree implementation that extends RSTP (802.1w) to run a separate spanning tree for each VLAN on the switch, and ensures that only one active, loop-free path exists between any two nodes on a given VLAN.

Comparing spanning tree options

Without spanning tree, having more than one active path between a pair of nodes causes loops in the network, which can result in duplication of messages, leading to a “broadcast storm” that can bring down the network.

The 802.1D spanning tree protocol operates without regard to a network's VLAN configuration, and maintains one common spanning tree throughout a bridged network. This protocol maps one loop-free, logical topology on a given physical topology. This results in the least optimal link utilization and longest convergence times.

The 802.1s multiple spanning tree protocol (MSTP) uses multiple spanning tree instances with separate forwarding topologies. Each instance is composed of one or more VLANs, which significantly improves network link utilization and the speed of reconvergence after a failure in the network’s physical topology. However, MSTP requires more configuration overhead and is more susceptible to dropped traffic due to misconfiguration.

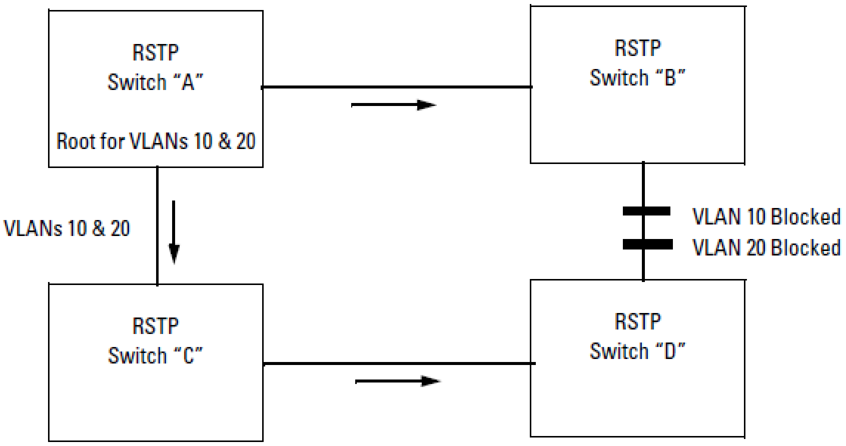
Rapid spanning tree protocol (RSTP) requires less configuration overhead, provides faster convergence on point-to-point links, and speedier failure recovery with predetermined, alternate paths. The switches covered by this guide, use the IEEE Rapid Per-VLAN spanning tree Protocol (RPVST) standard. RPVST was introduced as an enhancement to Rapid spanning tree Protocol (RSTP) to improve the link utilization issue and require less configuration overhead. Basically, RPVST+ is RSTP operating per-VLAN in a single layer 2 domain. VLAN tagging is applied to the ports in a multi-VLAN network to enable blocking of redundant links in one VLAN while allowing forwarding over the same links for non-redundant use by another VLAN. Each RPVST+ tree can have a different root switch and therefore can span through different links. Since different VLAN traffic can take different active paths from multiple possible topologies, overall network utilization increases.

Another major advantage of RPVST+ is that it localizes topology change propagation to individual VLANs. Since there is a separate spanning tree for each VLAN, topology changes affecting a particular VLAN are propagated only inside that VLAN. The switch flushes the MAC addresses learned only on the affected VLAN and other VLAN traffic is not disturbed. This minimizes the network flooding caused by the spanning tree topology changes. This is a significant improvement in the case of a large, flat, layer 2 network. In a network having a large number of per-VLAN spanning tree instances, RPVST+ can result in an increased load on the switch's CPU.

Understanding how RPVST+ operates

RPVST+ applies one RSTP tree per-VLAN. Each of these RSTP trees can have a different root switch and span the network through shared or different links. As shown in the following diagram, since the active paths for traffic on different VLANs can use the same for different links, multiple topologies are possible, and overall network utilization increases.

Figure 36: *RSTP forming a single spanning tree across all VLANs*



The topology has four switches running RSTP. Switch "A" is the root switch. To prevent a loop, RSTP blocks the link between switch "B" and switch "D". There are two VLANs in this network (VLAN 10 and VLAN 20). Since RSTP does not have VLAN intelligence, it forces all VLANs in a layer 2 domain to follow the same spanning tree. There will not be any traffic through the link between switch "B" and switch "D" and hence the link bandwidth gets wasted. On the other hand, RPVST+ runs different spanning trees for different VLANs. Consider the following diagrams.

Figure 37: *RPVST+ creating a spanning tree for VLAN 10*

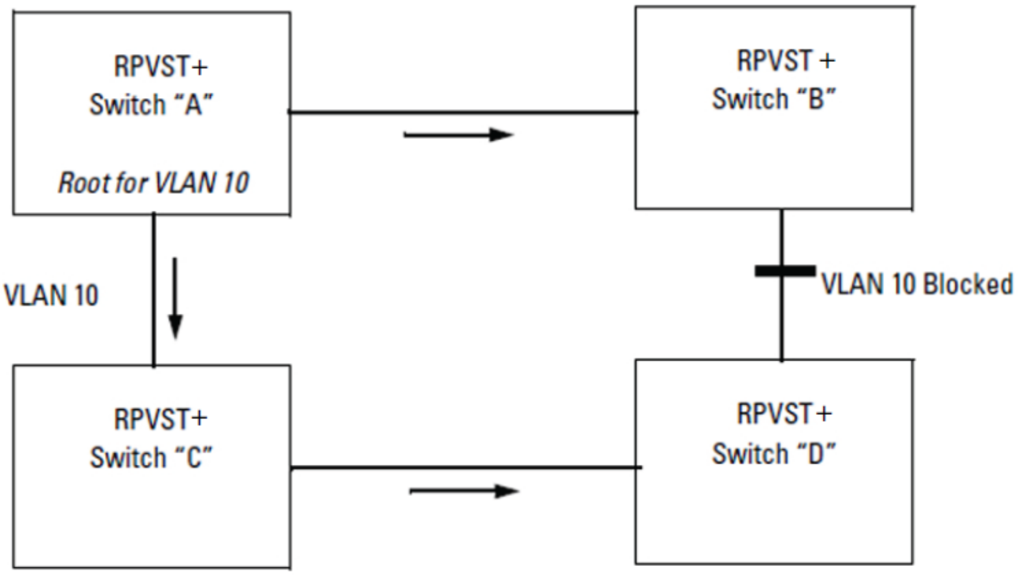
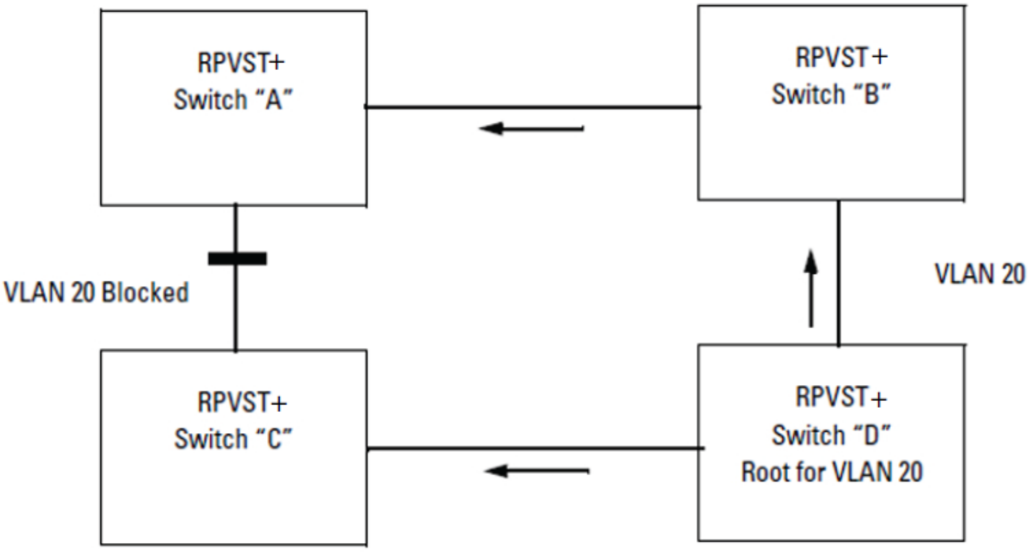


Figure 38: *RPVST+ creating a spanning tree for VLAN 20*

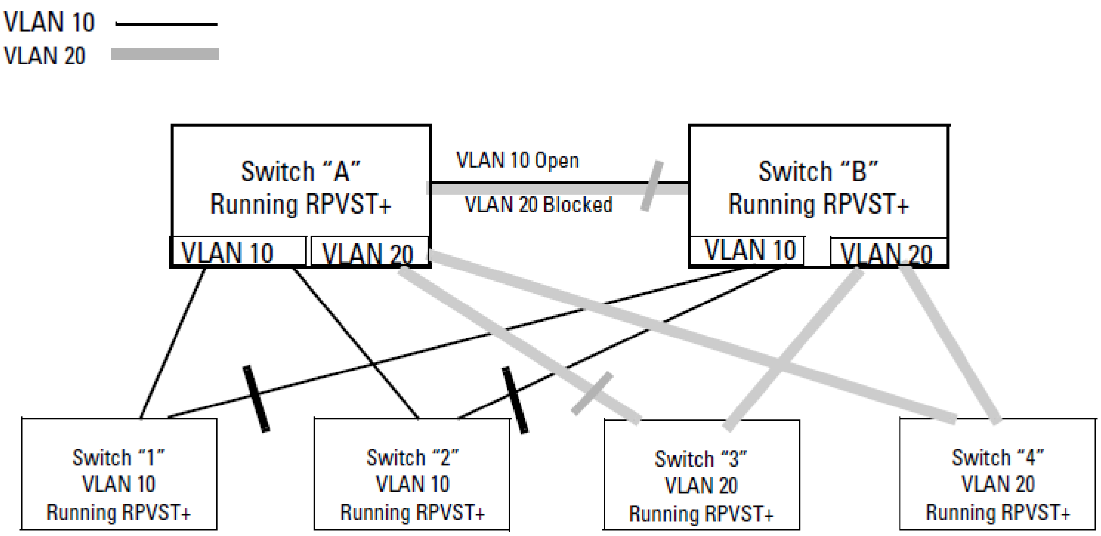


The two topologies above are the same as the first topology, but now the switches run RPVST+ and can span different trees for different VLANs. Switch "A" is the root switch for the VLAN 10 spanning tree and switch "D" is the root switch for the VLAN 20 spanning tree. The link between switch "B" and switch "D" is only blocked for VLAN 10 traffic but VLAN 20 traffic goes through that link. Similarly the link between switch "A" and switch "C" is blocked only for VLAN 20 traffic but VLAN 10 traffic goes through that link. Here, traffic passes through all the available links, and network availability and bandwidth utilization increase.

Another major advantage of RPVST+ is that it localizes topology change propagation. Since there is a separate spanning tree for each VLAN, topology changes affecting a particular VLAN are propagated only inside that VLAN. The switch flushes the MAC addresses learned only on the affected VLAN, the traffic on other VLANs is not disturbed. This minimizes the network flooding due to spanning tree topology changes. This is a significant improvement in the case of a large, flat, layer 2 network.

The following figure shows a further example of shared links and redundant path-blocking in a network running RPVST+.

Figure 39: *Sample RPVST+ network*



Working with the default RPVST+ configuration

In the factory default configuration, spanning tree operation is disabled. Configuring the spanning tree mode as RPVST+ on a switch and then enabling spanning tree automatically creates a spanning tree instance for each VLAN on the switch. Configuration with default settings is automatic, and in many cases does not require any adjustments. This includes operation with spanning tree regions in your network running STP, MSTP, or RSTP. Also, the switch retains its currently configured spanning tree parameter settings when spanning tree is disabled. Thus, if you disable, then later re-enable spanning tree, the parameter settings will be the same as before spanning tree was disabled.

CAUTION:

The switch automatically senses port identity and type, and automatically defines spanning tree parameters for each type, and parameters that apply across the switch. Although these parameters can be adjusted, HPE strongly recommends leaving these settings in their default configurations unless the proposed changes have been supplied by an experienced network administrator who has a strong understanding of RPVST+ operation.

RPVST+ operating notes

Recommended application

RPVST+ is ideal in networks having less than 100 VLANs. In networks having 100 or more VLANs, MSTP is the recommended spanning tree choice due to the increased load on the switch CPU.

VLAN membership

A port will be part of a given VLAN spanning tree only if the port is a member of that VLAN.

RPVST+ interoperates with RSTP and MSTP on VLAN 1

Because a switch running RPVST+ transmits IEEE spanning tree BPDUs, it can interoperate with IEEE RSTP and MSTP spanning tree regions, and opens or blocks links from these regions as needed to maintain a loop-free topology with one physical path between regions.

NOTE: RPVST+ interoperates with RSTP and MSTP only on VLAN 1.

Single spanning tree applications

One spanning tree variant can be run on the switch at any given time. On a switch running RPVST+, MSTP cannot be enabled. However, any MSTP-specific configuration settings in the startup configuration file will be maintained.

Exclusions

The following features cannot run concurrently with RPVST+:

- Features that dynamically assign ports to VLANs:
  - GVRP
  - RADIUS-based VLAN assignments (802.1X, WebAuth, MKAC auth
  - Auth-VID/UnAuth-VID configuration on interfaces
  - MAC-Based VLANs
  - LLDP Radio Port VLAN
- Switch Meshing
- QinQ
- Protocol VLANs
- Distributed Trunking
- Filter Multicast in rapid-PVST mode (The multicast MAC address value cannot be set to the PVST MAC address 01:00:0c:cc:cc:cd.)

GVRP

Spanning tree mode cannot be set to RPVST+ when GVRP is enabled, and GVRP cannot be enabled when RPVST+ is enabled.

RPVST+ operating limits

Virtual ports (vPorts) on a switch are determined by the number of physical ports on the switch, plus other factors. Exceeding the recommended number of vPorts can result in dropped BPDUs.

Allowing traffic on per-VLAN ID (PVID) mismatched links

The switch generates an Event Log message for a VID mismatch on an active RPVST+ VLAN only if `ignore-pvid-inconsistency` is disabled (the default).

If `ignore-pvid-inconsistency` is enabled on multiple switches connected by hubs, there could be more than two VLANs involved in PVID mismatches that will be ignored by RPVST+.

If there is an actual misconfiguration of port VLAN memberships in a network, then enabling ignore-pvid-inconsistency prevents RPVST+ from detecting the problem. This could result in packet duplication in the network because RPVST+ would not converge correctly.

Viewing RPVST+ statistics and configuration

NOTE:

RPVST+ is a superset of the STP/802.1D and RSTP/802.1w protocols, and uses the RPVST+ MIB (hpicfRpvst).

Viewing global and VLAN spanning tree status

Syntax:

```
show spanning-tree
```

Displays the switch’s global and VLAN spanning tree status.

Viewing the switch's global and VLAN spanning tree status

```
switch# show spanning-tree

 Spanning Tree Information

  STP Enabled          [No] : Yes
  Mode                      : RPVST
  Extended System ID        : Disabled
  Ignore PVID Inconsistency : Disabled
  RPVST Enabled VLANs       : 10,20


  Switch MAC Address        : 0024a8-d13a40
  Root Guard Ports          :
  Loop Guard Ports          :
  TCN Guard Ports           :
  BPDU Protected Ports      : 23-24
  BPDU Filtered Ports       : 23-24
  Auto Edge Ports           : 1-24,A1-A4
  Admin Edge Ports          :

  VLAN  Root Mac        Root       Root       Root                 Hello
  ID    Address         Priority   Path-Cost  Port                 Time(sec)
  ----- --------------- ---------- ---------- -------------------- ---------
  10    0024a8-d13a40   32,768     0          This switch is root  2
  20    0024a8-d13a40   32,768     0          This switch is root  2
```

Viewing status for a specific VLAN

Syntax:

```
show spanning-tree vlan vlan-id
```

Displays detailed spanning tree information for the VLAN and the ports belonging to the specified VLAN.

Viewing status for a specific VLAN

```
switch(config)# show spanning-tree vlan 20

 Spanning Tree Information

  STP Enabled              [No] : Yes
  Mode                               : RPVST
  Extended System ID                 : Disabled
  Ignore PVID Inconsistency          : Disabled
  Switch MAC Address                 : 0024a8-d13a40


  VLAN ID                            : 20
  RPVST Enabled                      : Enabled


  Root MAC Address                   : 0024a8-d13a40
  Root Priority                      : 32,768
  Root Path Cost                     : 0
  Root Port                          : This switch is root
  Operational Hello Time (secs)      : 2
  Topology Change Count              : 38
  Time Since Last Change             : 23 hours

                                                             Designated
  Port  Type       Cost   Priority  Role       State      Bridge
  ----- ---------- ------ --------- ---------- ---------- ---------------
  9     100/1000T  20000  128       Designated Forwarding 0024a8-d13a40
  21    100/1000T  20000  128       Designated Forwarding 0024a8-d13a40
  22    100/1000T  20000  128       Designated Forwarding 0024a8-d13a40
  23    100/1000T  200000 128       Designated Forwarding 0024a8-d13a40
  24    100/1000T  0      128                  Disabled
```

Viewing status for a specific port list

Syntax:

```
show spanning-tree port-list
```

Displays the spanning tree status for the designated port(s). You can list data for a series of ports and port trunks by specifying the first and last port or trunk of any consecutive series of ports and trunks. For example, to display data for port 20-24 and trk1, you would use this command: `show spanning-tree 20-42,trk1`

Viewing status for a specific port list

```
switch# show spanning-tree 22

 Spanning Tree Information

  STP Enabled     [No] : Yes
  Mode                 : RPVST
  RPVST Enabled VLANs : 10,20


  Switch MAC Address   : 0024a8-d13a40

  Port                     : 22
  Status                   : Up          Port Type            : 100/1000T
  BPDU Protection          : No          BPDU Filtering       : No
  Root Guard               : No          TCN Guard            : No
  Loop Guard               : No          Admin Edge Port      : No
  Admin PointToPoint MAC : Yes

  VLAN   Port       Port      Port       Designated    Hello Oper  Oper
  ID     Path-Cost  Priority  State      Bridge        Time  Edge  PtP
  ------ ---------- --------- ---------- ------------- ----- ----- ------
  20      20000       128        Forwarding 0024a8-d13a40 2      No     Yes
  25      200000      128        Forwarding 002347-587b80 2      Yes    Yes
```

Viewing status per-port per-VLAN

Syntax:

`show spanning-tree *port-list* vlan *vlan-id*`

Displays detailed information for ports in the `port-list` in the given VLAN. This command further filters the output for `show spanning-tree *port-list*`.

Viewing status per-port per-VLAN

```
switch# show spanning-tree 22 vlan 20

 Spanning Tree Information

  STP Enabled     [No] : Yes
  Mode                 : RPVST
  RPVST Enabled VLANs : 10,20


  Switch MAC Address   : 0024a8-d13a40

  Port                      : 22
  Status                    : Up           Port Type             : 100/1000T
  BPDU Protection           : No           BPDU Filtering        : No
  Root Guard                : No           TCN Guard             : No
  Loop Guard                : No           Admin Edge Port       : No
  Admin PointToPoint MAC : Yes

  VLAN    Port        Port       Port        Designated    Hello Oper   Oper
  ID      Path-Cost   Priority   State       Bridge        Time  Edge   PtP
  ------ ----------- ---------- ----------- ------------- ----- ----- ------
  20      20000       128        Forwarding 0024a8-d13a40 2      No     Yes
```

Viewing the global RPVST+ configuration

Syntax:

`show spanning-tree config`

Displays the switch’s basic and per-VLAN spanning tree configuration.

The upper part of the output shows the switch’s global spanning tree configuration. The port listing shows the spanning tree port parameter settings for the spanning tree region operation (configured by the `spanning-tree *port-list*` command).

Viewing the global RPVST+ configuration

```
switch# show spanning-tree config

 Spanning Tree Information

  STP Enabled            [No]  : Yes
  Mode                         : RPVST
  Extended System ID           : Enabled
  Ignore PVID Inconsistency : Disabled
  RPVST Enabled VLANs       : 10,20


  Switch MAC Address        : 002347-587b80

  Root Guard Ports          :
  Loop Guard Ports          :
  TCN Guard Ports           :
  BPDU Protected Ports      :
  BPDU Filtered Ports       :
  Auto Edge Ports           : 1-24
  Admin Edge Ports          :

                Max Age Forward    Hello     Admin Root
  VLAN Priority (sec)   Delay(sec) Time(sec) Bridge
  ---- --------- -------- ----------- --------- ----------------
  1    32768    20      15         2         Not Configured
  10   32768    20      15         2         Not Configured
  20   32768    20      15         2         Not Configured
```

Viewing the global RPVST+ configuration per port

Syntax:

```
show spanning-tree [ethernet]port-list config
```

Lists the spanning tree port parameter settings (global and per VLAN) for only the specified port(s) and trunk(s). You can list data for a series of ports and port trunks by specifying the first and last port or trunk of any consecutive series of ports and trunks. For example, to display data for ports 9, 11, 12, 21 and trk1, use this command: `show spanning-tree 9,11,12,21,trk1 config`

Viewing the global RPVST+ configuration per port

```
switch# show spanning-tree 9,11,12,21,22 2 trk1 config

 Spanning Tree Information

  STP Enabled     [No] : Yes
  Mode                 : RPVST
  Switch MAC Address   : 002347-587b80
  RPVST Enabled VLANs  : 10,20


        Admin Auto Admin Root Loop TCN BPDU BPDU
  Port  Edge  Edge PtP   Grd  Grd  Grd Flt  Guard
  ----- ----- ---- ----- ---- ---- --- ---- -----
  9     No    Yes  True  No   No   No  No   No
  11    No    Yes  True  No   No   No  No   No
  12    No    Yes  True  No   No   No  No   No
  21    No    Yes  True  No   No   No  No   No
  Trk1  No    Yes  True  No   No   No  No   No
```

Viewing the global RPVST+ configuration per port per VLAN

Syntax:

```
show spanning-tree<ethernet>port-list vlan vlan-id
```

Lists the spanning tree port parameter settings per port per VLAN.

Viewing the global RPVST+ configuration per port per VLAN

```
switch# show spanning-tree 9 config vlan 10

 Spanning Tree Information

  STP Enabled          [No] : Yes
  Mode                      : RPVST
  Extended System ID        : Enabled
  Ignore PVID Inconsistency : Disabled
  Switch MAC Address        : 002347-587b80


  RPVST Enabled             : Enabled
  VLAN ID                   : 10
  Switch Priority           : 32768
  Forward Delay             : 15
  Hello Time                : 2
  Max Age                   : 20
  Admin Root Bridge         : Not Configured

       Path       Port     Admin Auto Admin Root Loop TCN BPDU BPDU
 Port  Cost       Priority Edge  Edge PtP   Grd  Grd  Grd Flt  Guard
 ----- ---------- -------- ----- ---- ----- ---- ---- --- ---- -----
 9     20000      128      No    Yes  True  No   No   No  No   No
```

Viewing the global RPVST+ configuration per VLAN

Syntax:

```
show spanning-tree config vlan vlan-id
```

Lists the spanning tree port parameter settings for only the specified VLAN.

Viewing the global RPVST+ configuration per VLAN

```
switch(config)# show spanning-tree config vlan 20

 Spanning Tree Information

  STP Enabled          [No] : Yes
  Mode                      : RPVST
  Extended System ID        : Enabled
  Ignore PVID Inconsistency : Disabled
  Switch MAC Address        : 002347-587b80


  RPVST Enabled             : Enabled
  VLAN ID                   : 20
  Switch Priority           : 32768
  Forward Delay             : 15
  Hello Time                : 2
  Max Age                   : 20
  Admin Root Bridge         : Not Configured

                   Path       Port
  Port  Type       Cost       Priority
  ----- ---------- ---------- ---------
  9     100/1000T  20000      128
  20     100/1000T  200000     128
  21     100/1000T  20000      128
```

Viewing BPDU status and related information

Syntax:

```
show spanning-tree bpdu-protection port-list
```

Displays the BPDU protection state and errant BPDU count for ports in the port list.

Viewing BPDU status in show spanning tree output

```
switch# show spanning-tree 22

 Spanning Tree Information

  STP Enabled     [No] : Yes
  Mode                 : RPVST
  RPVST Enabled VLANs : 10,20


  Switch MAC Address   : 0024a8-d13a40

  Port                    : 22
  Status                  : Up            Port Type             : 100/1000T
  BPDU Protection         : No            BPDU Filtering        : No
  Root Guard              : No            TCN Guard             : No
  Loop Guard              : No            Admin Edge Port       : No
  Admin PointToPoint MAC : Yes

  VLAN    Port        Port       Port        Designated     Hello Oper   Oper
  ID      Path-Cost   Priority   State       Bridge         Time  Edge   PtP
  ------ ---------- ---------- ---------- ------------- ----- ----- ------
  20      20000       128        Forwarding 0024a8-d13a40 2     No     Yes
```

Viewing BPDU protection status on specific ports

```
switch# show spanning-tree bpdu-protection 11-12,21-24

 Status and Counters - STP BPDU Protection Information

 BPDU Protection Timeout (sec)  : 60
 BPDU Protected Ports : 23-24


  Port    Type       Protection State                Errant BPDUs
  ------ --------- ---------- ----------------- ------------
  11      100/1000T No                               0
  12      100/1000T No                               0
  21      100/1000T No                               0
  22      100/1000T No                               0
  23      100/1000T Yes                              0
  24      100/1000T Yes                              0
```

Viewing RPVST+ VLAN and vPort system limits

Each switch model supports a maximum number of active virtual ports (vPorts). New port VLAN memberships cannot be created once the vPort limit has been reached. Also, there is a maximum recommended number of active vPorts for each fixed-port switch or each module in a chassis switch. Exceeding the maximum recommended number of vPorts can result in dropped BPDUs and potential network loops. This command displays the current vPort status and maximum recommended vPort total per-switch or, for modular switches, per-module.

Syntax:

```
show spanning-tree system-limits rapid-pvst
```

Displays the RPVST+ VLAN and virtual port (vPort) status on the switch.

Viewing RPVST+ VLAN and vPort system limits

```
switch(config)# show spanning-tree system-limits rapid-pvst

 Spanning Tree Information


  STP Enabled              : Yes
  Mode                     : RPVST
  RPVST Enabled VLANs      : 20


  Switch MAC Address                   : 002347-c651c0
  Count of RPVST Enabled VLANs         : 1
  Maximum Allowed RPVST Enabled VLANs  : 400
  Count Of Total Virtual Ports         : 24
  Maximum Allowed Virtual Ports        : 424

                       Current         Operational     Recommended Maximum
  Ports                Virtual Ports   Virtual Ports   Virtual Ports
  -------------------- --------------- --------------- --------------------
  Ports 1-24           24              2               200
```

Table 24: *Virtual Port Data Fields*

vPort data field	Description
`Count of Total Virtual Ports`	The count of active vPorts (ports per VLAN) plus the count of non-active vPorts (all ports that belong to trunks).
`Maximum Allowed Virtual Ports`	The total of the system-created vPort instances plus the maximum user-assignable vPort instances. Each port on the switch belongs to at least one VLAN (VLAN-1 by default), which is a system-created vPort instance. The user-assigned VPORT instances are the system-assigned vPort instances. The `show spanning-tree system-limits rapid-pvst` command combines the system-created vPort instances and the user-assigned maximum vPort instances when calculating the maximum allowed virtual ports. **NOTE:** Each user-configured trunk on the switch increments this value by 1.

Table Continued

vPort data field	Description
`Current Virtual Ports`	The number of ports that are members of each VLAN on a per-module basis (or a per-group of ports basis).
`Operational Virtual Ports`	The number of ports belonging to each PVST-enabled VLAN on a per-module basis (or a per-group of ports basis). This value should not exceed the recommended maximum vPort limit.
`Recommended Maximum Virtual Ports`	The maximum recommended number of vPort instances that should be allowed on the switch. Exceeding this limit can potentially result in received BPDUs being dropped.

Configuring vPorts

Virtual ports on a switch are calculated as ports per-VLAN. Also, a trunk membership on one or more VLANs counts as one vPort per-VLAN, regardless of how many physical ports belong to the trunk. For example, the following configuration on a modular chassis results in 26 vPorts.

```
trunk 1,2 trk1
vlan 1
   name "DEFAULT_VLAN"
   untagged 3-24
   no untagged trk1
   exit
vlan 20
   ip address 10.243.230.75 255.255.255.248
   name "VLAN20"
   tagged trk1
   exit
vlan 30
   ip address 10.243.230.83 255.255.255.248
   name "VLAN30"
   tagged 13,14,trk1
   exit
```

	Module “A”	Module “B”	Module “C”	Total vPorts on the Switch
VLAN 1	22 (A3 - A24)	23 (B2 - B24	24 (C1 - C24)	
VLAN 20	1 (trk1: A1 - A2)	1 (trk1: B1)[1]	0	
VLAN 30	2 (A13 - A14) 1 (trk1: A1 - A2)[1]	2 (B13 - B14) 1 (trk1: B1)[1]	0	
vPorts per-module	26	27	24	77

[1] A trunk in a given VLAN counts as one vPort for each module on which it occurs.

Calculating non-active vPorts

Every port that is part of a manually configured trunk is counted as a non-active (reserved) vPort. For example, the ports in the following configuration are all non-active vPorts:

```
trunk 1, 2 trk1
trunk 3-5 trk2 lacp
trunk 17-20 trk3 dt-lacp
```

Calculating per-module vPorts on chassis switches

The switch-wide active vPort count, there is a vPort count per port module determined by the number of ports per line card that are members of each VLAN. Also, on modular switches, if a VLAN includes a trunk configured with ports on more than one module, then one vPort is counted for each module on which the trunk exists (regardless of how many ports are included in the trunk.) For example, in the following configuration, VLANs 1, 20, and 30 have a total of 74 vPorts.

```
trunk A1,A2,B1 trk1
vlan 1
    name “DEFAULT_VLAN”
    untagged A3-A24, B2-B24
    no untagged trk1
    exit
vlan 20
    ip address 10.243.230.75 255.255.255.248
    name “VLAN20”
    tagged A3-A12, B2-B12, trk1
    exit
vlan 30
    ip address 10.243.230.83 255.255/255/248
    name “VLAN30”
    tagged A13, A14, B13, B14, trk1
    exit
```

	Module “A”	Module “B”	All Modules
VLAN 1	22	23	4
VLAN 20	10 + 1	11 + 1	23
VLAN 30	2 + 1	2 + 1	6
Total vPorts	36	38	74

Troubleshooting an RPVST+ configuration

Cause

This section describes the show spanning tree commands you can use to monitor, troubleshoot, and debug the operation of a per-VLAN spanning tree configuration in your network.

NOTE: The `show spanning-tree`commands described in this section allow you to troubleshoot RPVST+ activity in your network by focusing on increasingly specific levels of operation. For example, you can display debug information for:

- All VLANs
- All ports of one VLAN
- A specific port or several ports used in one VLAN

Viewing the change history of root bridges

Syntax:

`show spanning-tree root-history vlan` *`vlan-id`*

Displays the last 10 root bridge changes on a specified VLAN configured with RPVST+. Included are the timestamp and Root Bridge ID recorded at each root bridge change.

Use the `show spanning-tree root-history` command to view the number and dates of changes in the assignment of a root bridge. Possible intrusion into your VLAN network may occur if an unauthorized external device gains access to a spanning tree by posing as the root device in a topology. To prevent a port connected to the device from being selected as the root port in a topology, use the `spanning-tree root-guard` command.

Viewing the change history of root bridges

```
switch# show spanning-tree root-history vlan 20

 Status and Counters - RPVST Root Changes History

  VLAN ID                : 20
  Root Changes Counter   : 53
  Current Root Bridge ID : 32768:0024a8-d13a40

 Root Bridge ID       Date       Time
 -------------------- ---------- --------
 32768:0024a8-d13a40 05/04/2012 21:54:11
     0:001185-c6e500 05/04/2012 21:54:07
 32768:0024a8-d13a40 05/04/2012 16:41:11
     0:001185-c6e500 05/04/2012 16:41:11
```

Enabling traps and Viewing trap configuration

Syntax:

`no spanning-tree trap`

`[[errant-bpdu | loop-guard | new-root] | [topology-change [vlan<vid-list | instance [instance-ID] {cstt}|root-guard]]`

Enables or disables SNMP traps for errant-BPDU, loop guard, new root, and root guard event notifications.

`errant-bpdu`

Enables SNMP notification when an errant BPDU is received. Designed for use with BPDU filtering.

`loop-guard`

Enables SNMP notification when a loop guard inconsistency is detected. Designed for use with the Loop Guard option.

new-root

Enables SNMP notification when a new root is elected on any VLAN configured for RPVST+ on the switch.

root-guard

Enables SNMP notifications when a root-guard inconsistency is detected.

topology-change

Enables notifications sent when a topology change occurs.

topology-change-history

Shows the spanning tree topology history changes.

Default for all of the above options: Disabled

The `no` form of the command disables traps on the switch.

Syntax:

```
show spanning-tree traps
```

Displays the current spanning tree trap configuration on the switch.

Viewing spanning tree traps in the default configuration

```
switch(config)# show spanning-tree traps

 Status and Counters - STP Traps Information

 Trap Name                | Status
 ----------------------- + --------
 errant-bpdu              | Disabled
 new-root                 | Disabled
 root-guard               | Disabled
 loop-guard               | Disabled
```

Viewing debug counters for all VLAN instances

Syntax:

```
show spanning-tree debug-counters
```

Displays the aggregate values of all RPVST+ debug counters that are maintained on a switch. These aggregate values are a summary of the information collected from all ports and from all spanning tree instances for all switch ports. Use the displayed diagnostic information to globally monitor RPVST+ operation on a per-switch basis.

Viewing debug counters for all VLANs

```
switch# show spanning-tree debug-counters

 Status and Counters - RPVST Debug Counters Information

                                Aggregated Value
  Counter Name                  Collected from VLANs
  ------------------------------ -------------------------
  Invalid BPDUs                 0
  Errant BPDUs                  0
  Looped-back BPDUs             0
  Starved BPDUs                 18
  Exceeded Max Age BPDUs         3
  Topology Changes Detected      9
  Topology Changes Tx            9
  Topology Changes Rx            4
  Topology Change ACKs Tx        0
  Topology Change ACKs Rx        6
  TCN BPDUs Tx                   4
  TCN BPDUs Rx                   0
  CFG BPDUs Tx                   0
  CFG BPDUs Rx                   0
  RST BPDUs Tx                   0
  RST BPDUs Rx                   0
  RPVST BPDUs Tx                 1881
  RPVST BPDUs Rx                 2617
```

Viewing debug counters per-VLAN

Syntax:

```
show spanning-tree debug vlan vlan-id
```

Displays the aggregate values of all RPVST+ debug counters maintained on a switch for a specified VLAN.

Viewing debug counters for a specific VLAN

```
switch(config)# show spanning-tree debug vlan 20

 Status and Counters - RPVST Debug Counters Information

  VLAN ID : 20

                                 Aggregated Value
  Counter Name                   Collected from Ports
  ------------------------------ --------------------
  Invalid BPDUs                  5
  Errant BPDUs                   10
  Looped-back BPDUs              0
  Starved BPDUs                  9
  Exceeded Max Age BPDUs         2
  Topology Changes Detected      9
  Topology Changes Tx            4
  Topology Changes Rx            181
  Topology Change ACKs Tx        0
  Topology Change ACKs Rx        0
  TCN BPDUs Tx                   0
  TCN BPDUs Rx                   0
  CFG BPDUs Tx                   0
  CFG BPDUs Rx                   0
  RST BPDUs Tx                   0
  RST BPDUs Rx                   0
  RPVST BPDUs Tx                 1531
  RPVST BPDUs Rx                 1428
```

Viewing debug counters per-port per-VLAN

Syntax:

```
show spanning-tree debug ports port-list vlan vlan-id
```

Displays the aggregate values of all RPVST+ debug counters maintained on one or more ports used by a specified VLAN.

Viewing debug counters for a specific port on a VLAN

```
Switch_A(config)# show spanning-tree debug ports 9 vlan 20

 Status and Counters - RPVST Debug Counters Information

  VLAN ID : 20
  Port : 9

  Counter Name                    Value           Last Updated
  ------------------------------ --------------- --------------------
  Invalid BPDUs                   0               04/16/2012 22:27:15
  Errant BPDUs                    0               04/16/2012 22:27:15
  Looped-back BPDUs               0               04/16/2012 22:27:15
  Starved BPDUs                   5               05/01/2012 21:48:11
  Exceeded Max Age BPDUs          0               04/16/2012 22:27:15
  Topology Changes Detected       9               05/04/2012 21:54:05
  Topology Changes Tx             5               05/05/2012 22:04:49
  Topology Changes Rx             2               05/07/2012 18:08:34
  Topology Change ACKs Tx         0               04/16/2012 22:27:15
  Topology Change ACKs Rx         0               04/16/2012 22:27:15
  TCN BPDUs Tx                    0               04/16/2012 22:27:15
  TCN BPDUs Rx                    0               04/16/2012 22:27:15
  CFG BPDUs Tx                    0               04/16/2012 22:27:15
  CFG BPDUs Rx                    0               04/16/2012 22:27:15
  RST BPDUs Tx                    0               04/16/2012 22:27:15
  RST BPDUs Rx                    0               04/16/2012 22:27:15
  RPVST BPDUs Tx                  7812            05/05/2012 22:04:49
  RPVST BPDUs Rx                  1065            05/08/2012 19:43:11
```

Field descriptions for RPVST+ debug command output

Field	Shows the number of —
`Invalid BPDUs`	Received BPDUs that failed standard RPVST+ (802.1Q-REV/D5.0 14.4) validation checks and were dropped. This counter is maintained on a per-port per-VLAN basis.
`Errant BPDUs`	Received BPDUs that were dropped on a port that is configured to not expect BPDU packets. This counter is maintained on a per-port basis and is incremented each time a BPDU is received on a port configured with the BPDU filter to ignore incoming BPDU packets (`spanning-tree bpdu-filter` command) or the BPDU protection feature to disable the port when BPDU packets are received (`spanning-tree bpdu-protection` command).
`Looped-back BPDUs`	Times that a port has received self-sent BPDU packets as the result of an external loop condition in which the BPDUs were looped back to the originating transmission port. The received BPDU is still processed by RPVST+ and the port changes to a blocked state. This counter is maintained on a per-port per-VLAN basis.

Table Continued

Field	Shows the number of —
`Starved BPDUs`	Times that no BPDUs are received within the scheduled interval (three times the Hello Time value configured with the `spanning-tree vlan hello-time` command) from a VLAN-designated peer port on the VLAN root, alternate, or backup port. As a result, the "starved" port triggers a spanning tree topology regeneration. This counter is maintained on a per-port per-VLAN basis.
`Exceeded Max Age BPDUs`	Times that a BPDU packet is received from a bridge with a Message Age value greater than the configured value of the Max Age parameter `(spanning-tree maximum age` command). This may occur if the receiving bridge is located too far from the root bridge (beyond the configured size of the spanning tree domain on the root bridge) or if a BPDU packet with invalid root information is continuously circulating between bridges in a spanning tree domain and needs to be aged out.
`Topology Changes Detected`	Times that a Topology Change event is detected by the port on a given VLAN and the port triggers a topology change propagation throughout the network. A Topology Change event occurs when a non-edge port enters forwarding state. This counter is maintained on a per-VLAN per-port basis.
`Topology Changes Tx`	Times that Topology Change information is propagated (sent out) through the port to the rest of the network. For a VLAN port running PVST (non-rapid), the counter is the number of times that a CFG or RST BPDU with the TC flag set is transmitted out of the port. This counter is maintained on a per-VLAN per-port basis.
`Topology Changes Rx`	Times that Topology Change information is received from the peer port. For a VLAN port running PVST (non-rapid), the counter is the number of times that a CFG or RST BPDU with the TC flag set is received. This counter is maintained on a per-port per-VLAN basis.
`Topology Change ACKs Tx`	Times that the Topology Change acknowledgement is transmitted through the port (number of CFG or RST BPDUs transmitted with the Topology Change Acknowledge flag set). This counter is maintained on a per-port per-VLAN basis.
`Topology Change ACKs Rx`	Times that the Topology Change acknowledgement is received on the port (number of CFG or RST BPDUs received with the Topology Change Acknowledge flag set). This counter is maintained on a per-VLAN basis.
`TCN BPDUs Tx`	Topology Change Notification BPDUs that are transmitted through the port. This counter is maintained on a per-port basis.
`TCN BPDUs Rx`	Topology Change Notification BPDUs that are received on the port. This counter is maintained on a per-port per-VLAN basis.
`CFG BPDUs Tx`	802.1D configuration BPDUs that are transmitted through the port. This counter is maintained on a per-port per-VLAN basis.
`CFG BPDUs Rx`	802.1D configuration BPDUs that are received on the port. This counter maintained on a per-port per-VLAN basis.

Table Continued

Field	Shows the number of —
`RST BPDUs Tx`	802.1w RST BPDUs that are transmitted through the port. This counter is maintained on a per-port per-VLAN basis.
`RST BPDUs Rx`	802.1w RST BPDUs that are received on the port. This counter is maintained on a per-port per-VLAN basis.

RPVST+ event log messages

Event	Log message
STP enabled/disabled on a VLAN	`Spanning tree Protocol enabled/disabled on vlan` *`vlan-id`*
Switch does not receive BPDUs from peer on a particular VLAN and port	`VLAN` *`vlan-id`* `starved for a BPDU on port` *`port number`* `from` *`bridge name`*
Switch received BPDU with inconsistent VLAN	`Blocking port-name on vlan` *`vlan-id`*`.`
Inconsistency is restored	`Unblocking port-name on vlan` *`vlan-id`* `Port consistency restored.`
Root port is changed on a VLAN	`VLAN` *`vlan-id`*`root changed from` *`bridgepriorty:mac`* `to` *`bridge priority:mac`*
Switch received a BPDU with invalid TLV	`Received SSTP BPDU with bad TLV on` *`port-number vlan-id`*
The number of `vlan-port` instances exceeds the recommended limit	`The number of vlan-port instances exceeded the recommended limit of` *`num`*
RADIUS subsystem tries to dynamically change port VLAN assignments when mode is RPVST	`RADIUS unable to assign port to VLAN` *`vlan-id`* `because spanning-tree is running in RPVST+ mode`
LLDP subsystem tries to dynamically change port VLAN assignments when mode is RPVST	`LLDP unable to assign port` *`port-number`* `to VLAN` *`vlan-id`* `because spanning-tree is running in RPVST+ mode`
VPORT counts exceed 200	`The number of vPorts on slot` *`slot-number`* `exceeds the recommended limit of` *`vport-count`*`. PVST BPDUs may be dropped.`

Using RPVST+ debug

While the Event Log records switch-level progress, status, and warning messages on the switch, the Debug/System Logging (Syslog) feature provides a way to record Event Log and debug messages on a remote device. For example, you can send messages about routing misconfigurations and other network protocol details to an external device, and later use them to debug network-level problems. The Debug/Syslog feature allows you to specify the types of Event Log and debug messages that you want to send to an external device. The two commands described next affect debug operation for RPVST+. For further information on debug operation, see the *ArubaOS-Switch Management and Configuration Guide* for your switch.

Syntax:

```
spanning-tree clear-debug-counters [ports port-list] [vlan vid-list]
```

Clears all spanning tree debug counters unless specific ports and VLANs are specified.

`ports` ***`port-list`***

Clears spanning tree debug counters on the specified ports.

`[ethernet]`***`port-list`***

Clears spanning tree debug counters on an entered list of posts or `all` for the `ports` command parameter.

`vlan`

Clears spanning tree debug counters for the VLAN.

`vlan` ***`vlan-id-list`***

One or more identifiers for the `VLAN` command parameter.

Using the `vlan` and `ports` options together clears the spanning tree debug counters on the specified ports for the specified VLANs. Counters maintained on the same ports for other VLANs are not cleared.

Syntax:

```
debug rpvst [event [filter vlan vid-list]]

no debug rpvst [event [filter vlan vid-list]]

debug rpvst [packet [filter port port-list [vlan vid-list]]]

no debug rpvst [packet [filter port port-list [vlan vid-list]]]
```

Displays RPVST+ debug messages on the destination device specified with the `debug destination logging | session | buffer` command.

`event`

Displays RPVST+ Event Log messages.

`filter vlan` ***`vid-list`***

Limits log messages to those generated on the specified VLANs.

`packet`

Displays RPVST+ packets sent and received.

`filter port` ***`port-list`*** **`vlan`** ***`vid-list`***

Limits packets displayed to those generated on the specified ports. If the `vlan` option is used, then packets displayed are further limited to the ports on the specified VLANs.

The `no` form of the command disables display of RPVST+ debug messages on the destination device.

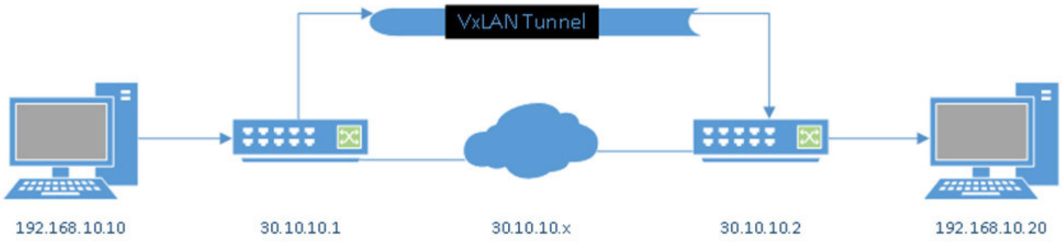
Overview of VXLAN

Virtual Extensible LAN(VXLAN) is a network virtualization technology that allows the extension of L2 networks over L3 UDP tunnels.

VXLAN tunnel

The 192.168.10.X/24 subnet is being tunneled through the 30.10.10.X/24 network.

Figure 40: *VXLAN tunnel*



L2 Forwarding in VXLAN

VXLAN learns unicast source MAC addresses similar to hosts off of physical ports. When a packet is de-encapsulated from a VXLAN tunnel, the source MAC is checked against the MAC table and bound to the tunnel interface index or if it is not present, it is learned. If the unicast source is in the MAC table but on a different port or tunnel to it, it is considered a move. The MAC table is used as the destination match to send traffic to a remote tunnel partner. Any unknown destination addresses are forwarded to all VTEP partners, configured with a common Virtual Network Identifier (VNI).

Broadcast addresses are handled similar to unknown destination addresses and flooded to all tunnels with mapped and configured VNI. Floods are copied to each unicast VTEP tunnel. Multicast addresses are not filtered, they are treated like a flood when going to multiple tunnel destinations.

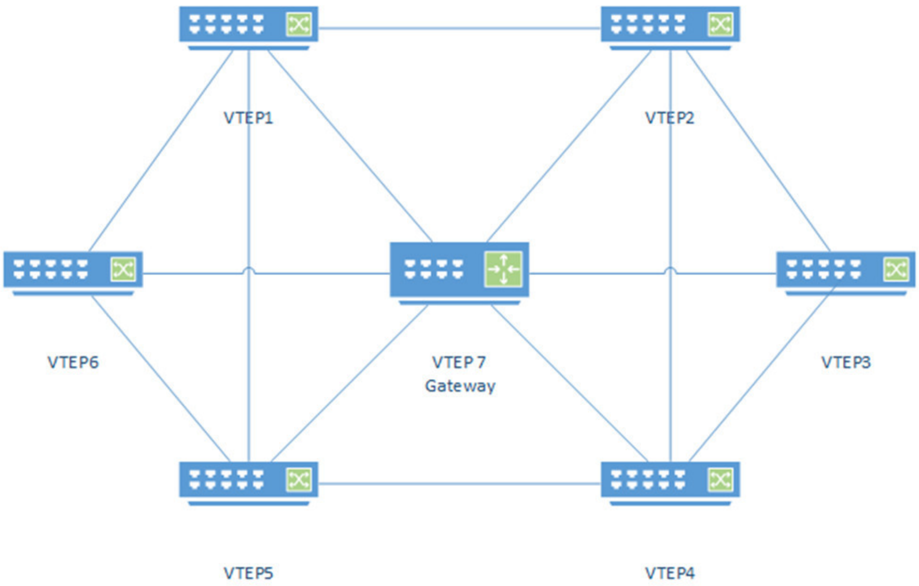
Fully Meshed Network

In a fully meshed VXLAN network, every VTEP is configured with a tunnel to every other VTEP running the same VNI. A fully meshed VXLAN network allows for any overlay network to reach any other part of the network for local L2 to L2 traffic. This is necessary when client to client traffic is needed amongst roaming clients with the same subnet.

Fully meshed network

Every VTEP needs 6 Tunnel connections to reach every other VTEP in the mesh network.

Figure 41: *Fully meshed network*



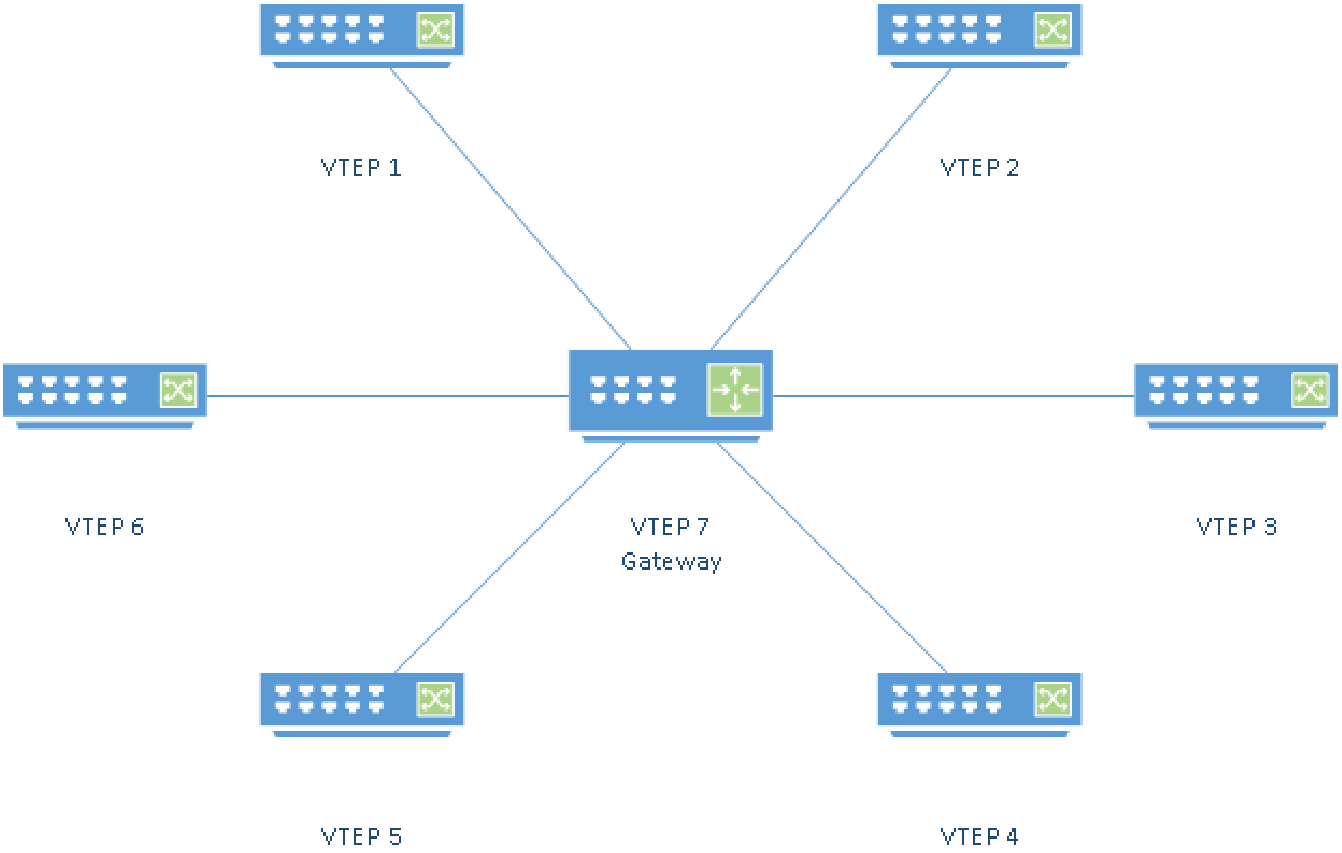
Hub Spoke Network

The configuration is simple for all VTEPs except for the gateway. Each outer VTEP needs to have a tunnel plumbed back to the VXLAN VTEP Gateway for VTEPs 1-6. The gateway however needs a tunnel to each of the VTEPs configured on the edge.

Hub spoke network

In the following Hub spoke model, traffic only flows from the edge VTEPs back to the VXLAN gateway.

Figure 42: *Hub spoke network*



Restrictions

The following is a limitation of VXLAN:

For any given L2 virtual network, only one VTEP can be present for each edge L2 LAN.

VXLAN — Feature interaction table

Feature	VXLAN impact
ARP protect	VXLAN tunnel interfaces would be (implicitly) TRUSTED interfaces with no MAC/IP verification on ARP PKTs ingressing tunnels. ARP protection will be enforced on other untrusted ports of the overlay VLAN if ARP protection is enabled for that VLAN. A VLAN that has a virtual IP subnet and a VNI configured for it is referred to as an overlay VLAN in here.
CDP	CDP only runs on physical links. VXLANs are not able to participate in CDP.
DHCP (v4/v6) relay	A gateway device is able to function as a DHCP Relay agent and support transferring DHCP PKTs between the following depending on where the DHCP server is housed: • An overlay subnet and an underlay subnet. • An overlay subnet and another overlay subnet.
DHCP snooping	VXLAN tunnel interfaces would be (implicitly) TRUSTED with no MAC/IP/Port verification on DHCP PKTs ingressing tunnels.
filter	• Connection Rate (Virus Throttling) – Cannot be enabled on a VXLAN tunnel interface. • Multicast – Tunnel interface cannot be part of the drop filter. PKTs coming in on a tunnel interface will honor drop filter if configured for the multicast address. • Protocol – Similar to Multicast filters. • Source Port - Tunnel interface cannot be part of the drop filter. A tunnel interface cannot have a source port filter configured for it.
IGMP snooping	No support for IGMP SNOOPing and Multicast pruning on VXLAN tunnel interfaces.If a device is an IGMP querier on an overlay VLAN, the IGMP queries sourced by the device will be sent over the VTEP fabric to the other nodes as well.IGMP blocked and forward configuration cannot be applied to VXLAN interfaces.
IGNORE untagged-MAC	Not applicable to VXLAN tunnel interfaces.
Link-Keepalive (UDLD)	Not supported on VXLAN tunnel interfaces.
LLDP	LLDP not supported on tunnel interfaces.

Table Continued

Feature	VXLAN impact
Lockout-MAC	Traffic ingressing a network port with source MAC that is configured as lockout-MAC will not be tunneled.Traffic ingressing a tunnel interface (i.e. payload) with a source MAC that is configured as lockout-MAC on the device will be dropped upon DECAP.
Loop Protect	Not supported on VXLAN tunnel interfaces
MAC-TRAP MAC-Count TRAP	This Mac-Trap, when configured on an interface, sends out a trap when: • MAC addresses are learned or aged-out on an interface. • The number of MAC addresses on a port exceeds a certain threshold. MAC-Count TRAPs are not supported on VXLAN tunnels.
<Interface> Monitor	Cannot monitor a VXLAN tunnel interface.
OpenFlow	• OpenFLow sees VXLAN tunnels as logical interfaces and can program outport rules for tunnels. • OpenFlow cannot program in_port match rules on VXLAN tunnel interfaces (similar to SI). • OpenFlow lookups will be bypassed for packets ingressing VXLAN tunnels.
Port Security	Not supported on VXLAN tunnel interface.
QinQ	Mutually exclusive.
QoS Source VLAN Source Port UDP TCP device protocol	**Encap** When QoS rules are configured, traffic matching QoS rules and will be prioritized and tunnelled into the underlay. The VLAN priority and DSCP is mapped to the outer header. **Decap** Traffic entering a tunnel (i.e. payload) and matching QoS rules will be prioritized upon decapsulation and forwarded out appropriately.
QOS remark policy (PCP/ DSCP)	**Encap** Traffic matching QoS remark rules will have their DSCP remarked and re-prioritized before being tunneled into the underlay. **Decap** Traffic coming in on a tunnel (i.e. payload) and matching QoS remark rules (VLAN based) will be remarked and re-prioritized upon decapsulation and forwarded out appropriately.

Table Continued

Feature	VXLAN impact
QoS rate-limit policy Interface rate-limit	**Encap** If a rate-limit policy (port/VLAN based) is applied to traffic destined to a tunnel interface, it will be enforced before the traffic is encapsulated. **Decap** If a rate-limit policy (VLAN based) is applied to (payload) traffic coming in on a tunnel interface, it will be enforced upon ingress on the loopback port (after decapsulation). Not supported on VXLAN interfaces.
SFLOW	Cannot configure SFLOW (sampling/polling) on tunnel interfaces and loopback ports. If a packet destined to a tunnel has been sampled, the outport field in the SFLOW header will be the tunnel resolved uplink port.
Smart-Link	Cannot be configured on tunnel interfaces. However Smart-Link can be used to achieve VTEP redundancy. To ensure that downlink traffic is not disrupted when a VTEP goes down and the downlink switch's Slave Port becomes Active, Smart-Link flush messages will be allowed to traverse the L3 fabric and it will wipe out MAC table entries on tunnel interfaces for the protected VLANs. It will also clear the ARP entries pointing to Tunnel entries on the VTEPs. Per-port recv-control-vlan restriction is not applicable to Flush packets received on Tunnel Interfaces as they are trusted, any filtering can always be done on external facing ports.
STP	Does not run on tunnel interfaces and loopback ports. This implies that STP cannot detect loops if any among switches that are connected via VXLAN tunnels and also directly connected among them. It is important that only one device as the VTEP for a VLAN in a network where multiple devices can function as VTEPs for the same VLAN. STP Topology Change message does not have to flush MAC addresses learned on tunnel interfaces as a TC on one segment does not impact the MAC learns of another segment. When a device that is STP disabled receives BPDUs, it floods to all ports of the VLAN. This can result in BPDUs being tunneled to the remote side as well. The loopback port of the remote end will quench the packet and not process it.
Static MAC	Not supported on VXLAN tunnel interfaces.
UFD	Not supported on VXLAN tunnel interfaces.
<Interface> IPv4/v6 ACLs	Not supported on VXLAN tunnels and loopback ports (stolen and internal).

Table Continued

Feature	VXLAN impact
<Interface> IP Source-Lockdown	Not configurable on VXLAN tunnel and loopback interfaces (stolen and internal).
<Interface> admin-status	Not configurable on VXLAN tunnel interface.
IP Source Binding	IP source binding entries cannot be configured on VXLAN tunnel interfaces and loopback ports (stolen/internal).
<VLAN> RIP, OSPF, BGP	Cannot be enabled on overlay VLANs. BGP is not supported on WC. 16.02.
<VLAN> IRDP	Cannot be enabled on overlay VLANs.
<VLAN> IGMP (Router)	Cannot be enabled on overlay VLANs.
<VLAN> IGMP Proxy	Cannot be enabled on overlay VLANs.
<VLAN> UDP Broadcast Forwarder	Supported on overlay VLANs.
<Interface> QoS	Not supported on tunnel and loopback interfaces (stolen/internal).
IPv4 unicast Routing	Supported. Can route traffic into a tunnel and route traffic coming out of a tunnel.
IPv6 unicast Routing	Not supported.
<VLAN> MLD	Not supported as overlays are IPv4 only.
<VLAN> ND<VLAN> RA	Not supported as overlays are IPv4 only.Not supported as overlays are IPv4 only.
<VLAN> IPv6	Not supported as overlays are IPv4 only
<VLAN> OSPF3	Cannot be enabled on overlay VLANs.

OpenFlow interaction

The OpenFlow features supported on VXLAN tunnel interfaces are listed in the following table. For more information about OpenFlow, see the *OpenFlow Administrators Guide*.

IS	IS-NOT
OpenFlow agent on the switch will communicate VXLAN tunnel virtual port add/remove notifications to the controller.	Cannot add multiple VXLAN tunnel interfaces as part of multi-port action.
Tunnel virtual port up/down notifications will be communicated to the controller.	Cannot group VXLAN tunnels and physical ports as part of multi-port action.

Table Continued

IS	IS-NOT
Tunnel virtual port counters (TX/RX packets) will be supported via OpenFlow multipart message.	Cannot group outport “Tunnel” action with outport “Normal” or outport “SendToController” actions.
In OpenFlow virtualized mode, only OpenFlow version1.3 instances of the Overlay VLAN will advertise VXLAN virtual ports to the controller.	Cannot group outport “Tunnel” action with Strip VLAN, Modify VLAN, Modify MAC actions.
The VNI corresponding to the Overlay VLAN needs to be associated with the VXLAN tunnel for it to be advertised as a member interface of that instance.	Cannot support re-directing PKTs to VXLAN tunnels on an OpenFlow v1.0 instance. Tunnels will only be supported on OpenFlow v1.3 instances.
In OpenFlow aggregate mode, an OpenFlow rule that has a VXLAN tunnel as an outport will be allowed only if the in_vlan is an overlay VLAN and its VNI is associated with the VXLAN tunnel.	The solution does not support OpenFlow lookup on frames coming in on a VXLAN tunnel. These PKTs will be forwarded as “normal” using the switch L2/L3 forwarding table entries.
OpenFlow meters can be attached to rules that point to tunnels as output ports and metering action will work as designed.	When tunnel interfaces are deleted by the administrator and there are OpenFlow rules that are pointing to those tunnels, the switch will not automatically remove flows associated with the tunnel ports. The controller has to delete those flows explicitly.
VXLAN tunnel interfaces will be part of an OpenFlow VLAN FLOOD action.	The send-to-tunnel outport action would be supported only on the Policy Engine (TCAM) tables and not on the OpenFlow software tables.
OpenFlow modify VLAN PCP/IP DSCP action will be supported with tunnel as output port. This action will modify the payload’s DSCP field before it’s encapsulated.	OpenFlow Port MOD requests will not be supported on tunnel ports.
PKT_OUT action will be supported on tunnel port.	Tunnel Port cannot be used as a match field (IN_PORT).
All controllers connected to an OpenFlow instance running version 1.3 will receive the tunnel notifications irrespective of their roles (master/slave/equal).	

Table Continued

IS	IS-NOT
If IP routing is enabled on the device and there is frame that the ASIC is L3 forwarding, OpenFlow rule that matches this frame with an outport of a VXLAN tunnel will encapsulate the IP routed version of the frame (MAC, VLAN and TTL fields modified). If ASIC punts this frame to software for an L3 table lookup miss, software should forward IP routed version of the frame to the tunnel after destination is resolved.	
VXLAN tunnels are HA synced to the SMM and will continue to function after a failover. OpenFlow rules that are pointing to VXLAN tunnels will continue to forward frames to tunnels even after an HA failover.	
Coexistence with SI — If an L2/L3 lookup for a frame points to a VXLAN tunnel it is possible to deflect this frame to an SI tunnel via an OpenFlow rule. Upon receiving this frame back on the SI tunnel (Sentinel validated), an L2/L3 re-lookup will result in the frame being sent out via the VXLAN tunnel interface. Note that frames coming in on tunnel interfaces (SI and VXLAN) will bypass OpenFlow lookups. OpenFlow redirect to VXLAN tunnels will not disturb the copy-CPU flags that are also set for the frame. This is so that OpenFlow/VXLAN can work in hybrid network set ups where other features like port-sec, sflow etc. are enabled along with OpenFlow.	
OpenFlow redirect to VXLAN tunnels will not override a drop action set by some other lookup in the system. If OpenFlow action conflicts with a device feature's action (OF action is FWD and Feature XYZ's action is COPY/DROP), both actions will fail.	

Configuration procedures

The following are the basic configuration procedures used to set up VXLAN traffic.

Procedure

1. Prepare the underlay for VXLAN traffic.

   VXLAN uses a UDP tunnel to send L2 traffic over an L3 network. Any transport associated with VXLAN UDP packet is known as the underlay.

2. Configure Jumbo MTU on the underlay VLANs between Virtual Tunnel Endpoints (VTEPs).

VTEPs are the termination point of UDP VXLAN tunnels.

3. When using different priorities on any overlay VLAN traffic, configure IP Differentiated Service (DSCP) QOS on all underlay VLANs between VTEPs.

   The L2 networks identified by the virtual network identifier (VNI) can also be referenced as the overlay networks.

4. Enable VXLAN at the switch.
5. Create the overlay VLAN and the Virtual IP network (if the device is an IP gateway).
6. Create a virtual network instance and associate the instance with the overlay VLAN created in step-1.
7. Repeat the same steps for any of the other access devices that are part of the overlay.
8. Create VXLAN tunnels on each access device to the other access switches that are part of the overlay.
9. Depending on which overlay VLANs are configured on each access switch, configure the VN-Tunnel mapping.

   This mapping basically instructs the device to carry the VN's traffic on that tunnel.

VXLAN configuration commands

VXLAN configuration commands include commands to enable and disable a VXLAN, configuring destination UDP port number, creating and setting the mode of a VXLAN tunnel, setting the source and destination of a VXLAN tunnel, and binding the VNI to a VLAN.

Enabling VXLAN

Syntax

```
vxlan enable
```

Description

Enable creation of VXLAN tunnels.

Disable VXLAN

Syntax

```
vxlan disable
```

Description

Disable VXLAN tunnels.

Configuring destination UDP port number

Syntax

```
vxlan udp PORT-NUM
```

Description

Configure destination UDP port for VXLAN tunnels. The `no` form of the command will set it back to its default value of 4789.

Parameters

udp

Configure destination UDP port for VXLAN tunnels.

<1-65535>

Enter the UDP port number. Default value: 4789.

Creating a VXLAN tunnel

Syntax

```
interface tunnel TUNNEL-ID
```

Description

Create or enter a tunnel context.

Parameters

<1-128>

Enter an integer number for the tunnel ID.

Set the mode of a VXLAN tunnel

Syntax

```
tunnel mode unspecified|6in4|vxlan
```

Description

Specify the tunnel mode.

Parameters

unspecified

Set an `unspecified`tunnel

6in4

Set an `IPv6-in-IPv4` tunnel.

VXLAN

Set an `MAC-in-UDP` tunnel.

Set the source of a VXLAN tunnel

Syntax

```
tunnel source IP-ADDR
```

Description

Configure the local (source) IP address for the tunnel.

Parameters

IP-ADDR

The tunnel source IP address.

Set the destination of a VXLAN tunnel

Syntax

```
tunnel destination IP-ADDR
```

Description

Configure the destination IP address for the tunnel.

Parameters

IP-ADDR

The tunnel destination IP address.

Bind the VNI to a VLAN

Syntax

```
virtual-network VN-ID VLAN-ID VN-NAME
```

Description

Configure a virtual network.

Parameters

<1-16777215>

Specify the virtual network identifier.

VLAN-ID

Specify the VLAN to be mapped to the virtual network.

VN-NAME-STR

Set the virtual network name.

Configure a virtual network

```
switch(config)# virtual-network 1 2 Virtual-Network1
```

Map overlay VLANs to VXLAN tunnel

Syntax

```
vxlan tunnel TUNNEL-ID overlay-vlan VLAN-ID-LIST
```

Description

Map the list of overlay VLANs to a VXLAN tunnel.

Parameters

<1-128>

Enter the VXLAN tunnel id.

overlay-vlan

Specify the list of overlay VLANs to be mapped to the VXLAN tunnel.

`vlan` VLAN-ID-LIST

Specify the list overlay VLANs.

Map overlay VLANs to VXLAN tunnel

```
switch(config)# vxlan tunnel 1 overlay-vlan 1
```

VXLAN show commands

VXLAN show commands include commands to display the status of a VXLAN feature, tunnels, and tunnel statistics.

Show command to display the status of VXLAN feature

Syntax

```
show VXLAN
```

Description

Show status of VXLAN tunnel feature.

Show VXLAN

```
switch# show vxlan
VXLAN Tunnel Information
----------------------------------------------------------------
Status                 : Enabled
UDP Port               : 4789


switch# show vxlan
VXLAN Tunnel Information
----------------------------------------------------------------
Status                  : Disabled
```

Show commands to display tunnels

Syntax

```
show interfaces tunnel tunnel-list|tunnel-name|brief|type
```

Description

Show tunnel configuration and status information.

Parameters

brief

Display the configuration and status of all the tunnels.

tunnel-list

Display the configuration and status of the specified tunnel ID.

tunnel-name

Display the configuration and status of the tunnel specified.

type

Specify the Tunnel type.

Show interfaces tunnel

```
switch# show interfaces tunnel

Tunnel Configuration :

Tunnel                : tunnel-1
Tunnel Name           : VxLAN_T
Tunnel Status         : Enabled
Source Address        : 2.3.4.6
Destination Address   : 2.3.4.7
Mode                  : 6in4
TOS                   : -1
TTL                   : 64
IPv6                  : Disabled
MTU                   : 1280

Current Tunnel Status :
Tunnel State               : Up
Destination Address Route  : 2.3.4.0/24
Next Hop IP                : 2.3.4.7
Next Hop Interface         : vlan-50
Next Hop IP Link Status    : Up
Source Address             : Configured on vlan-50

Tunnel Configuration :

 Tunnel                : 100664146
 Tunnel Name           : ServiceTunnel01
Tunnel Status          : Enabled
Source Address         : 2.3.4.6
Destination Address    : 1.2.4.4
Mode                   : Service Tunnel
TOS                    : 0
TTL                    : 64
IPv6                   : n/a
MTU                    : 1468


Current Tunnel Status :
Interface State             : Up
Destination Address Route   : 0.0.0.0/0
Next Hop IP                 : 120.92.82.129
Next Hop Interface          : vlan-1
Next Hop IP Link Status     : Up
Source Address              : 2.3.4.6


Tunnel Configuration :
Tunnel                 : 201327442
Tunnel Name            : VxLAN_Tunnel01
Tunnel Status          : Enabled
Source Address         : 10.0.0.1
Destination Address    : 10.0.0.2
```

```
Mode                   : VXLAN Tunnel
TOS                    : -1
TTL                    : 64
IPv6                   : n/a
MTU                    : 1460

Current Tunnel Status   :
Tunnel State            : Up
Destination Address Route : 10.0.0.0/8

Next Hop IP               : 10.0.0.2
Next Hop Interface        : vlan-20
Next Hop IP Link Status   : Up
Source Address            : 10.0.0.1

Tunnel Configuration :

Tunnel              : 201327443
Tunnel Name         : VxLAN_Tunnel02
Tunnel Status       : Enabled
Source Address      : 10.0.0.1
Destination Address : 11.0.0.2
Mode                : VXLAN Tunnel
TOS                 : -1
TTL                 : 64
IPv6                : n/a
MTU                 : 1280

Current Tunnel Status :

Tunnel State               : Down
Down Reason                : TEP Not Reachable
Destination Address Route  :
Next Hop IP                :
Next Hop Interface         :
Next Hop IP Link Status    :
Source Address             : 10.0.0.1
```

Show VXLAN tunnel statistics

Syntax

```
show interfaces tunnel type vxlan statistics tunnelifindex | tunnelname-str
```

Description

Show the statistics of the VXLAN tunnels.

Parameters

tunnelifindex

Show statistics of the specified VXLAN tunnel ID.

tunnelname-str

Show statistics of the specified VXLAN tunnel.

Show statistics of the specified VXLAN tunnel ID

```
switch# show interfaces tunnel type vxlan statistic 201327443
Tunnel Name                                  : VxLAN_Tunnel02
Rx Packets                                   : 0
```

```
Tx Packets                                   : 0
Rx 5 Minute Weighted Average Rate (Pkts/sec) : 0
Tx 5 Minute Weighted Average Rate (Pkts/sec) : 0
```

Show interfaces tunnel type VXLAN statistics tunnelname

```
switch# show interfaces tunnel type vxlan statistics VxLAN_Tunnel02
Tunnel Index                                   : 201327443
Rx Packets                                     : 0
Tx Packets                                     : 0
Rx 5 Minute Weighted Average Rate (Pkts/sec)   : 0
Tx 5 Minute Weighted Average Rate (Pkts/sec)   : 0
```

Introduction to BYOD-redirect

The BYOD (bring-your-own-device) feature lets you design, manage, and control a BYOD network when you configure the BYOD-redirect feature on your switches.

Where BYOD-redirect is enabled on a switch, the device's client credentials are sent to the BYOD server for registration. The BYOD server stores the registration information for each client's device (such as the device MAC-address), which gives that client's device access to the network.

The BYOD solution includes:

- secure user authentication
- centralized authentication process
- authorization and accounting
- unified monitoring and network management services
- ease-of-use self-registration (on-boarding) process

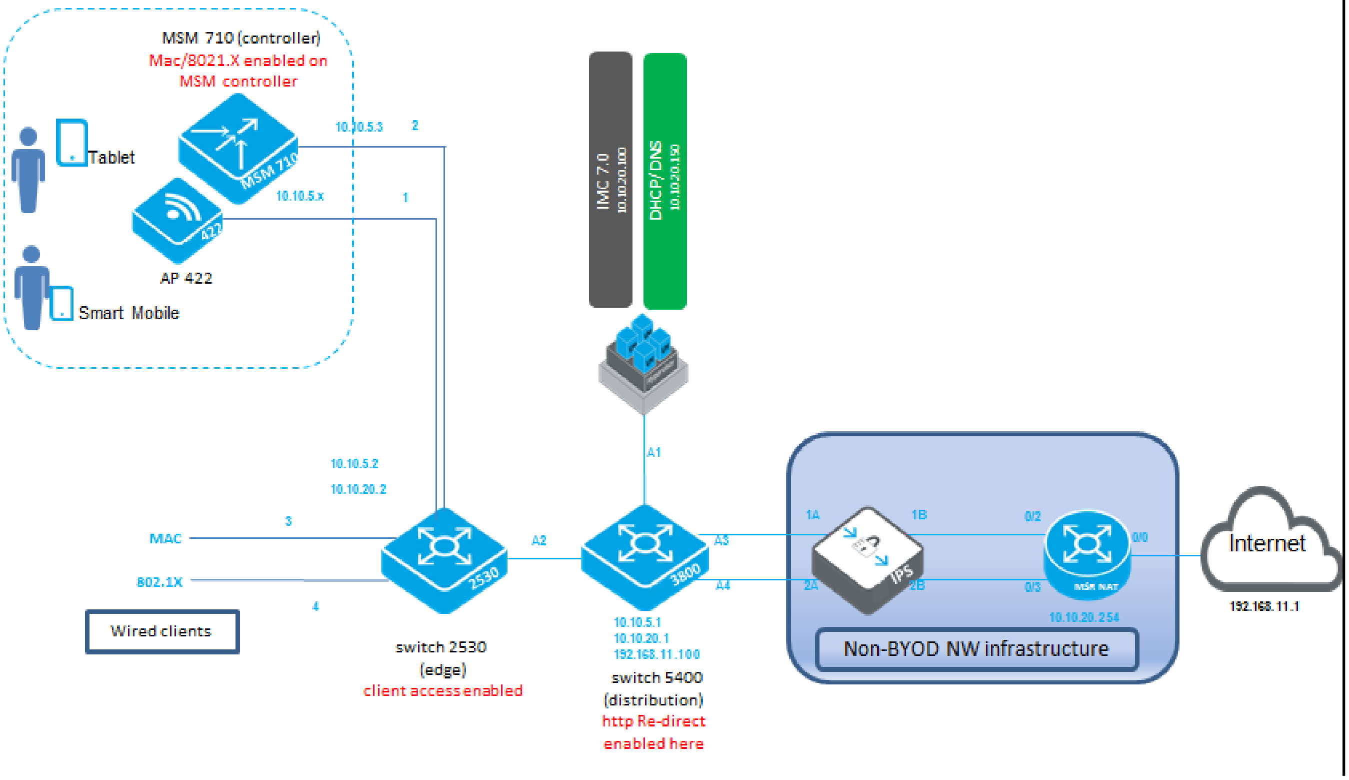
BYOD solution

The following figure illustrates a BYOD solution that includes the following:

- Access point and wireless controller: manages wireless SSIDs.
- BYOD (IMC) server: manages BYOD policy and centralized user management.

- switches: redirects user registration traffic to IMC and grants access to ports.
- BYOD Redirect feature

Figure 43: *BYOD solution*



BYOD features

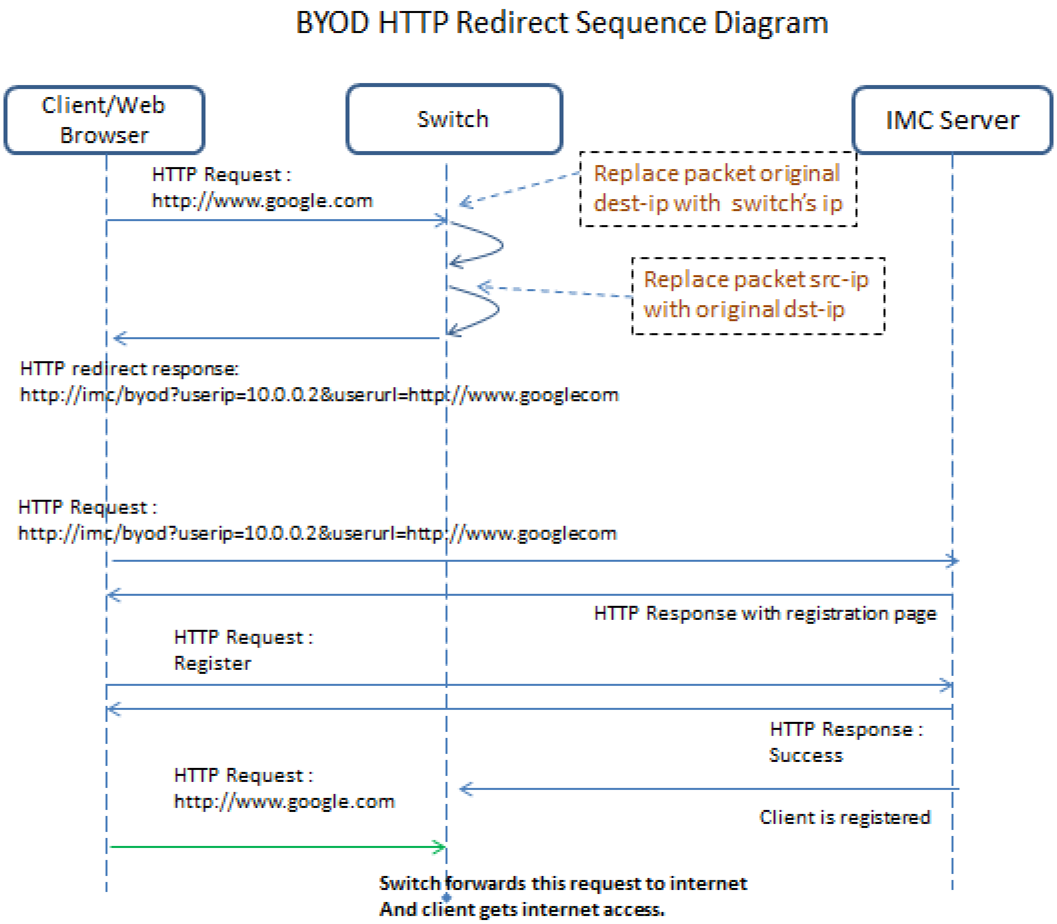
When BYOD-redirect is enabled on a VLAN, the BYOD feature intercepts HTTP traffic and blocks all other traffic for which free rules are not enabled. Most BYOD-redirect implementation is platform independent, except installing free rules to mitigate risks.

Communication between clients and the IMC server is tunneled by the edge switch:

1. A client request is read by the HTTP task.
2. The HTTP task always redirects, after embedding client IP addresses, a URL trying to access the redirected URL.

3. The redirect response includes URL parameters: **user ip address** and **url user is trying to access.**
4. The client receives a redirect response from the switch and makes an HTTP request to redirect the URL.

Figure 44: *The BYOD-redirect function*


BYOD HTTP Redirect Sequence Diagram
Client/Web Browser
Switch
IMC Server
HTTP Request : http://www.google.com
Replace packet original dest-ip with switch's ip
Replace packet src-ip with original dst-ip
HTTP redirect response: http://imc/byod?userip=10.0.0.2&userurl=http://www.googlecom
HTTP Request : http://imc/byod?userip=10.0.0.2&userurl=http://www.googlecom
HTTP Response with registration page
HTTP Request : Register
HTTP Response : Success
HTTP Request : http://www.google.com
Client is registered
Switch forwards this request to internet
And client gets internet access.

Interoperability with other switch features

The following rules can help avoid conflicts when BYOD-redirect has been deployed on a switch with other features:

1. **MAFR and BYOD-redirect are mutually exclusive** – MAFR (MAC Authentication Failure Redirect) and BYOD-redirect solve similar problems.
2. **DNS sentinel and BYOD-redirect** – When a DNS sentinel is enabled, the switch tunnels packets to the controller. Packets are re-injected to the switch only if the controller classifies DNS packets as permitted. When BYOD-redirect is enabled, the user should configure an ACL rule to pass through DNS packets to the switch. If SDN controller policy classifies a DNS packet originating from a client as drop, then BYOD-redirect does not work.
3. **IP sentinel and BYOD-redirect** – When IP sentinel is enabled for the IP flows configured by the SDN controller, the switch tunnels the IP packets to the controller. The IP packets are reinjected to the switch only if the controller classifies the IP traffic as not malicious. If the SDN controller policy classifies the client's IP traffic as malicious, then BYOD-redirect fails.

4. **OpenFlow and BYOD-redirect** – If an OpenFlow instance is enabled on a VLAN, then all traffic is given to the OpenFlow packet processing task. BYOD-redirect requires intercepting IP (HTTP) packets. If BYOD-redirect interoperates with OpenFlow, traffic should be copied to both OpenFlow and BYOD-redirect; otherwise, the switch cannot enable BYOD-redirect and OpenFlow on the same VLAN.
5. **Other TCAM rules** – If any other user has configured TCAM rules that override TCAM entries installed for BYOD-redirect, BYOD redirect does not work.

Interoperability with other vendors

Because BYOD policy integrates several logical components including MSM, UAM, and RADIUS, the redirected URL in the BYOD-redirect feature on a switch must include the `byod-server-url` and `user-ip` information to work with the IMC server.

Restrictions

BYOD-redirect has the following restrictions:

1. BYOD-redirect is a per-VLAN configuration; up to three VLANs can be enabled with BYOD-redirect.
2. BYOD-redirect supports up to three redirection servers configured on a switch. When a redirection server URL is configured, the BYOD module maintains separate data structures to store the redirected URL on the VLAN where BYOD-redirect is enabled. BYOD-redirect statistics are maintained for each server.

Configuring BYOD

Creating a BYOD server

Configure a portal redirect web-server.

Syntax

```
no portal web-server [web-server-name] url [url-string]
```

`portal`: Configure the BYOD redirect feature.

`web-server`: Configure portal redirect web-server.

`web-server-name`: Specify the BYOD web-server name in ASCII.

`url`: Configure the URL of the BYOD server.

`url-string` : A URL redirecting the client to the BYOD server must be in ASCII.

Associating a BYOD server

Associate a BYOD server with a specific VLAN to redirect clients to the assigned URL page.

Syntax

```
no vlan [vlan-id] <portal web-server [web-server-name]>
```

`vlan`: Add, delete, edit VLAN configuration, or enter a VLAN context.

`vlan-id`: VLAN identifier or VLAN name.

`portal`: Configure the BYOD redirect feature on a VLAN.

`web-server`: Specify the BYOD web-server.

web-server-name: BYOD web-server name in ASCII.

Creating a BYOD ACL rule

Configure a BYOD-free rule.

Syntax

```
no portal free-rule [rule-number] vlan [VLAN-ID] destination <<ip-address> | mask
<mask-length> | any tcp <des-tcp-port> | udp <des-udp-port> | source <ip-address>
| mask <mask-length> | any tcp <src-tcp-port> |udp <src-udp-port>>
```

Term	Meaning
portal	Configure the BYOD redirect feature.
free-rule	Configure a BYOD-free rule.
rule-number	Free rule number as an INTEGER**<1-6>**.
vlan	Free rule source VLAN ID.
VLAN-ID	VLAN identifier or VLAN name.
destination	Free rule destination.
ip-address	IP address
mask	Mask
mask-length	Mask length.
tcp	TCP protocol
udp	UDP Protocol
des-udp-port	tcp port destination
source	Free rule source.
<src/des-tcp/udp-port>	TCP or UDP port number, as an integer**<1-65534>**.
any	Free rule source any.
ip	Free rule source IP.
IP	Free rule destination IP.
any	Free rule source or destination any.

Implementing BYOD-redirect configuration

BYOD enables employees to register and access corporate resources with personally-owned devices. Though BYOD provides flexibility to employees, it can bring challenges to IT departments. BYOD-redirect is designed to help manage and control personal devices and policies at the enterprise network level.

Before implementing BYOD-redirect ensure that:

- BYOD-redirect is configured on a VLAN.
- BYOD-redirect is supported on up to three VLANs.
- BYOD-redirect is supported with Mac and 802.1X authentications.
- BYOD-redirect works with IMC 7.0 UAM module.
- The client URL and DHCP IP are included in the Redirect URL to the IMC.

NOTE:

Until the registration process has been completed, a client device cannot access the internet or the enterprise network. Any traffic from this unauthorized device is redirected to the BYOD server.

Implementing BYOD-redirect configuration examples

The following examples show how to implement BYOD-redirect for both wired and wireless solutions.

BYOD configuration on a distribution switch

To facilitate the BYOD-redirect function, complete the following tasks on the distribution switch:

1. Configure DNS and make FQDN solution successful: `ip dns server-address priority 1` <**DNS-server-IP**>.

   NOTE: The argument to the URL can be an FQDN or IP address. If you use the IP address as an argument, this step is not necessary.

2. Configure BYOD web-server URL: **portal web-server "byod" url http://imc.com:8080/byod**.
3. Enable BYOD-redirect on a VLAN: **vlan 101 portal web-server "byod."**
4. Configure BYOD-redirect free-rules on the on-boarding VLAN 101 to permit client traffic transit through DNS and DHCP servers using the following commands.To permit DNS traffic to/from a DNS server to a client through on-boarding VLAN:

   a. `portal free-rule 1 vlan 101 source any udp 0 destination any udp 53`

   b. `portal free-rule 2 vlan 101 source any udp 53 destination any udp 0`

   To permit DHCP traffic to/from DHCP server to client through on-boarding VLAN:

   a. `portal free-rule 3 vlan 101 source any udp 68 destination any udp 67`

   b. `portal free-rule 4 vlan 101 source any udp 67 destination any udp 68`

5. Register the device in IMC on the on-boarding VLAN. When registration is successful, client traffic is placed into different VLAN (guest/corporate) configurations.

Client authentication configuration on edge switch

Enable MAC authentication on edge switch port 1-2 using the following commands:

- `# enable mac authentication on ports 1-2`
- `aaa port-access mac-based 1-2`
- `# configure number of client limits on port 1 and port2`
- `aaa port-access mac-based 1 addr-limit 32`
- `aaa port-access mac-based 2 addr-limit 32`
- `radius-server host <radius ip> dyn-authorization`
- `radius-server host <radius ip> time-window 0`

Table 25: *Wired and wireless components configured in a network topology*

Access Type	Edge Switch	Distribution Switch	Configuration ProcedureNote
Wired Access	Edge switch (for example 2530)	5400 switch	**1.** Register the edge switch in HPE IMC. **2.** Create the configuration on the edge switch. **3.** Create the configuration on 5400 switch.
Wireless Access			**1.** Make the HPE MSM controller reachable by IMC. **2.** Ensure that access points (HPE 422) are managed by the MSM controller. **3.** Configure MAC or 802.1X authentication on the MSM controller. **4.** Create the configuration on the 5400 switch.

Figure 45: *Wired and wireless components configured in a network topology*

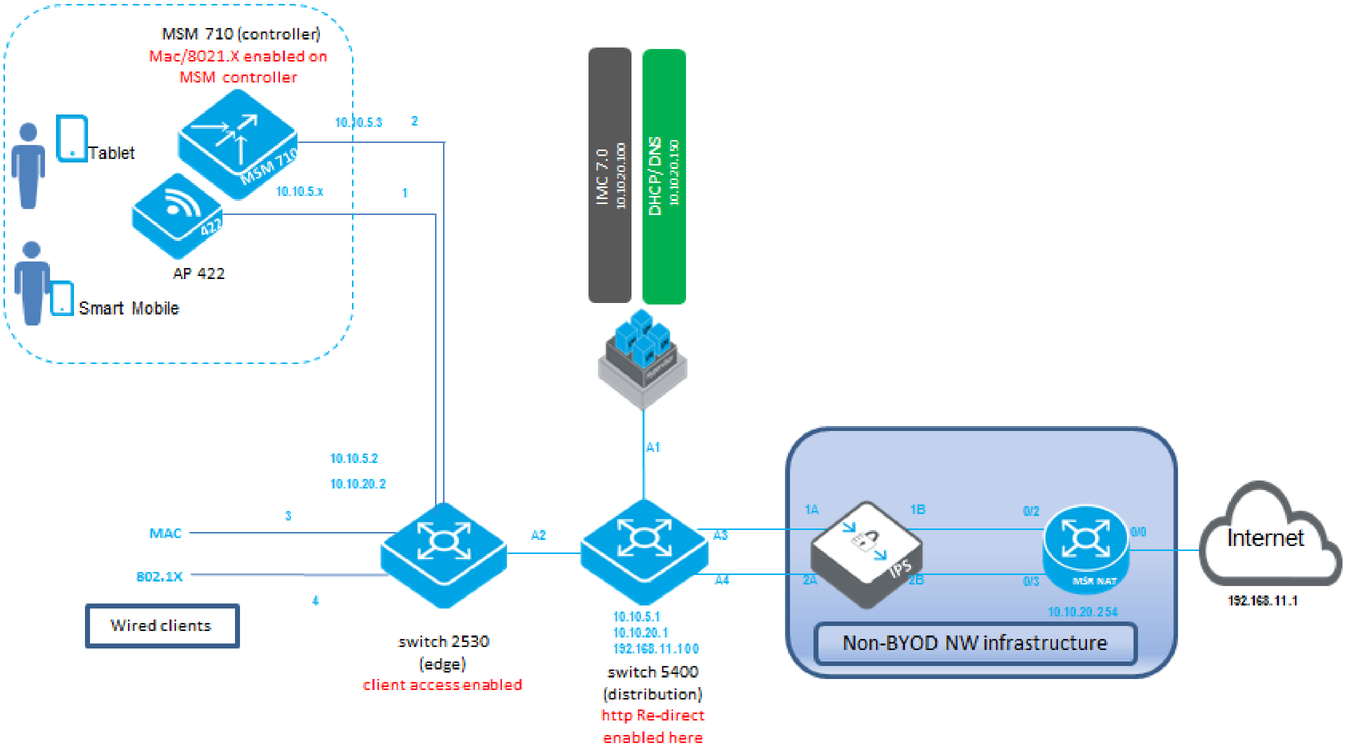


Table 26: *Wired clients solution*

Access Type	Edge Switch	Distribution Switch	Configuration Procedure
Wired Access	Edge switch (for example 2530)	Switch 3810	**1.** Register the edge switch and distribution switch in IMC. **2.** Ensure that both the edge and distribution switch can reach the DHCP and DNS server. **3.** Create the configuration on the edge switch. **4.** Create the configuration on the distribution switch.

Figure 46: *Wired clients solution*

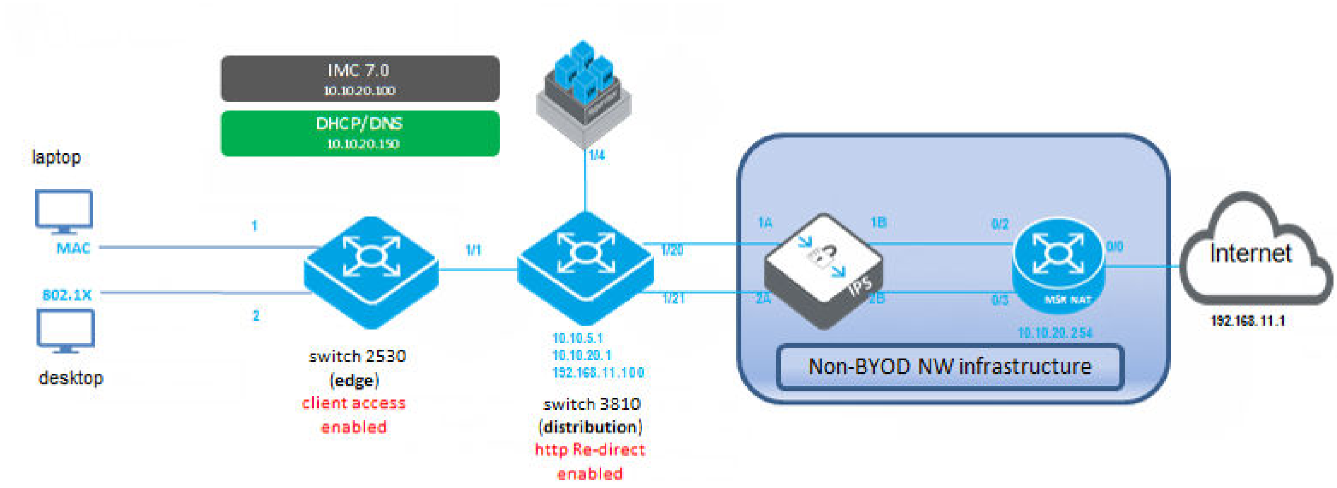
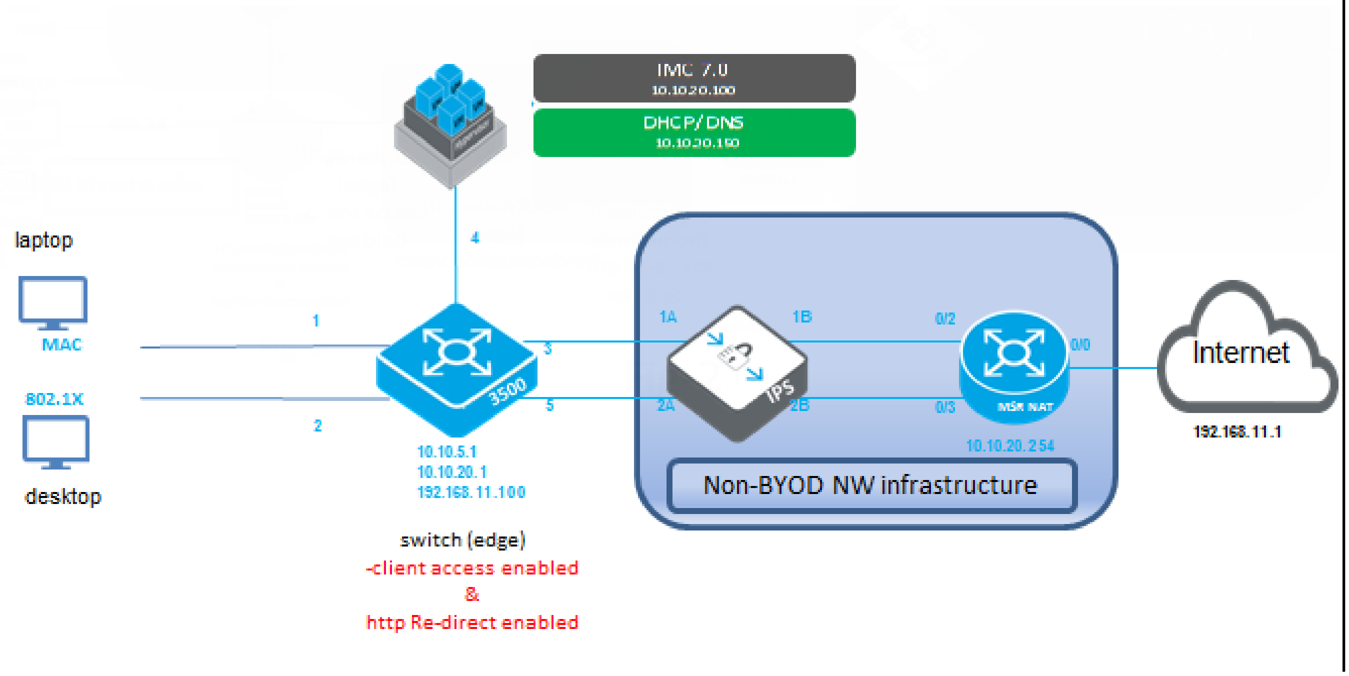


Table 27: *Configuration and access for wired clients on an edge switch*

Access Type	Edge Switch	Distribution Switch	Configuration Procedure
Wired Access	Edge switch	N/A	**1.** Register the edge switch in IMC. **2.** Ensure that the edge switch is reachable by the DHCP and DNS server. **3.** Create the configuration on the edge switch. **4.** Create the configuration on the edge switch.

Figure 47: *Configuration and access for wired clients on an edge switch*



Show commands

Show portal server

Display all BYOD servers and their attributes or specify a BYOD web-server-name to display its details.

Syntax

```
show portal web-server [web-server-name]
```

Term	Meaning
portal	Display BYOD server details..
web-server	Specify the BYOD web-server.
web-server name	Enter BYOD web-server name in ASCII.

Sample output

```
Portal Server:
1)imc:
Resolved IP       : 15.146.197.224
VPN Instance      : n/a
URL               : http://15.146.197.224:80/byod
VLAN              : 101
DNS Cache Status : 20 seconds
```

Show portal redirect statistics

Show redirect statistics of a BYOD.

Syntax

```
show portal redirect statistics
```

Term	Meaning
portal	Display BYOD server details.
redirect	Display redirect statistics
statistics	Display the statistics.

Sample output

```
show portal redirect statistics
Status and Counters - Portal Redirect Information
Total Opens               : 0
Resets Connections        : 0
Current Opens             : 0
Packets Received          : 14997
Packets Sent              : 12013
HTTP Packets Sent         : 3002
Current Connection States :
SYN_RECVD       : 0
ESTABLISHED     : 0
```

Show portal free rule

Display all BYOD free rules and their attributes; the user can specify a BYOD rule to display its free rule.

Syntax

```
show portal free-rule [free-rule-number]
```

Term	Meaning
portal	Display BYOD server details.
free-rule	Display BYOD-free rule.
free-rule-number	Free rule number as an integer <**0-50**>.

Sample output

```
Rule-Number   : 2
Vlan          : 0
Source:
Protocol   : UDP
Port       : 12345
IP         : 0.0.0.0
Mask       : 0.0.0.0
MAC        : n/a
Interface : n/a
Destination:
Protocol   : UDP
Port       : 123
IP         : 0.0.0.0
Mask       : 0.0.0.0
```

Associating with the BYOD server on a specified VLAN

Associate a BYOD server with a specific VLAN to redirect clients to the assigned URL page.

Syntax

```
no vlan <VLAN-ID > [portal web-server < web-server-name>]
```

Term	Meaning
portal	Configure the BYOD redirect feature on the VLAN.
web-server	Specify the BYOD web-server.
ASCII-STR	BYOD web server name.
vlan	Add, delete, edit VLAN configuration, or enter a VLAN context.
VLAN-ID	Enter a VLAN identifier or a VLAN name.

NOTE: All commands previously in the Summary of commands table are indexed under the entry **Command syntax**.

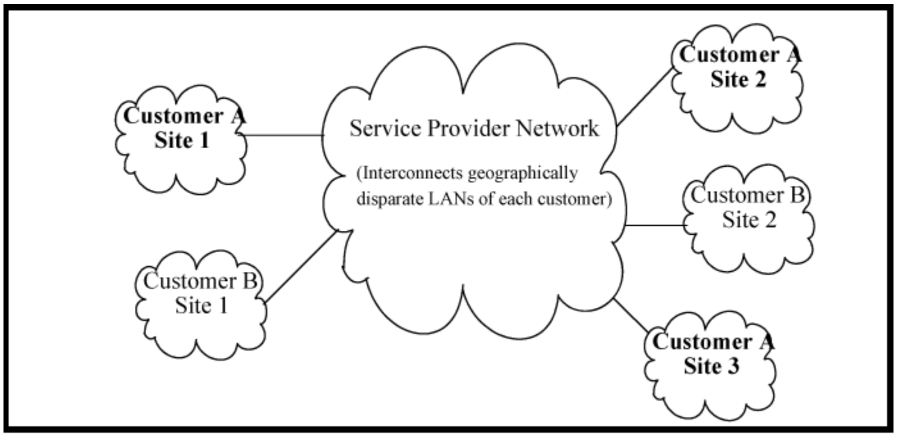
Introduction to QinQ

This chapter describes how to enable QinQ operations on the switch and how to configure provider bridge S-VLANs and port assignments.

The IEEE 802.1ad specification, commonly known as QinQ or provider bridging, extends the IEEE 802.1Q standard by providing for a second tier of VLANs in a bridged network. The general purpose of QinQ is to allow frames from multiple customers to be forwarded (or tunneled) through another topology (provider network) using service VLANs or S-VLANs. The provider bridge, which may comprise multiple devices in the service provider domain, looks like a simple bridge port to the customer's traffic and maintains the customer's VLANs.

The following diagram shows a sample QinQ topology and use model. Customer A has LANs spread across multiple site locations and may want to link them together in a single logical LAN. To do this, the customer could have a cable laid out for the entire distance interconnecting the three sites. A more cost-effective and scalable alternative, however, would be to tunnel frames through the provider's network to interconnect all the sites subscribing to the service. This solution can be delivered using QinQ.

Figure 48: *QinQ network diagram*

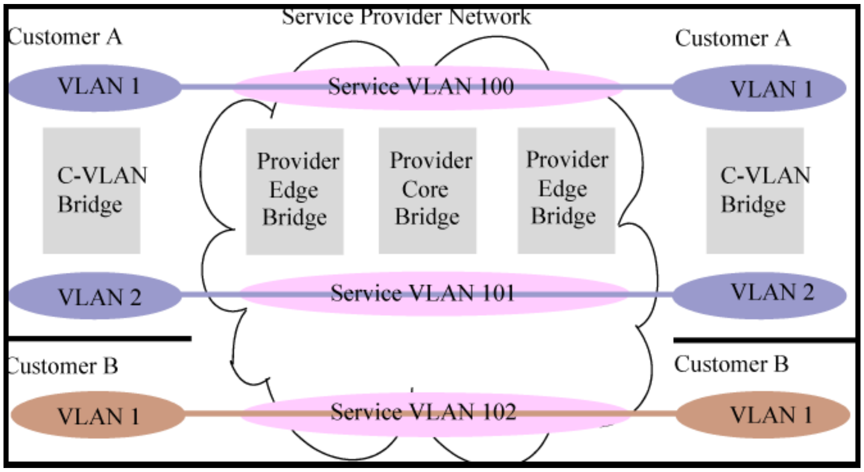


NOTE: The Service Provider and customers may belong to the same business entity, as in the case where a single enterprise uses QinQ to help segregate local networks and increase the scalability of their backbone infrastructure.

How QinQ works

Under QinQ, the provider network operates on a different VLAN space, independent of the VLANs that are used in the customer network.

Figure 49: *VLANs in a QinQ configuration*



Customer VLANs (referred to as C-VLANs by the IEEE 802.1ad specification) are not used to make any forwarding decisions inside the provider network where customer frames get assigned to service VLANs (S-VLANs). Inside the provider cloud, frames are forwarded based on the S-VLAN tag only, while the C-VLAN tag remains shielded during data transmission. The S-VLAN tag is removed when the frame exits the provider network, restoring the original customer frame.

Features and benefits

- Increases the VLAN space in a provider network or enterprise backbone.
- Reduces the number of VLANs that a provider needs to support within the provider network for the same number of customers.
- Enables customers to plan their own VLAN IDs, without running into conflicts with service provider VLAN IDs.
- Provides a simple Layer 2VPN solution for small-sized MANs (Metropolitan Area Networks) or intranets.
- Provides for customer traffic isolation at Layer 2 within a Service Provider network.

Configuring QinQ

QinQ must be configured on all the devices and ports participating in the provider bridge. Typically, customer facing ports are configured as untagged members of S-VLANs and provider facing ports are configured as tagged members of S-VLANs. Per the IEEE 802.1ad specification, there is no condition binding port types (customer or provider) to untagged or tagged S-VLAN memberships. Therefore, when configuring QinQ tunnelling on the switch, you would first configure per-port S-VLAN membership (tagged or untagged), and then configure the port type as `customer-network` or `provider-network`, depending on the device to which the switch port is connected.

NOTE: A customer-network port can receive S-VLAN tagged frames if the customer and provider agree on the S-VID association for that customer and the customer device is capable of sending S-VLAN tagged frames. Configuring QinQ with S-VLANs in a switch stack is not supported. For more information, see **Effects of QinQ on other switch features** on page 358.

To configure QinQ take the following steps on all participating provider switches:

Procedure

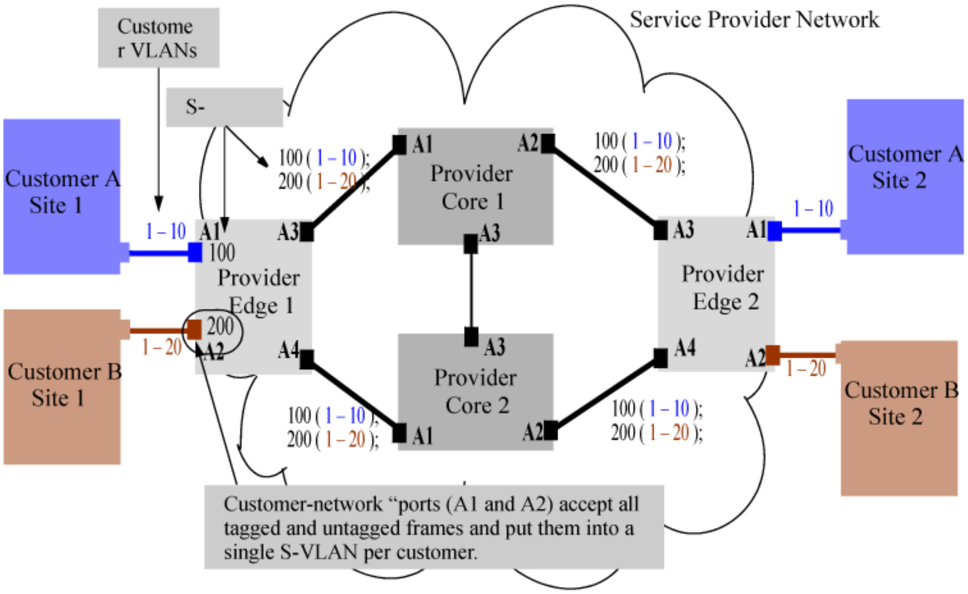
1. Enable QinQ on the device, selecting the appropriate QinQ mode (S-VLAN or mixed VLAN mode).
2. Save the configuration and reboot the switch.
3. Configure S-VLANs and assign per port VLAN membership.
4. Configure port-types for all of the switch ports that carry QinQ traffic across the network.
5. (Optional) Assign priorities to traffic passing through the provider network.

CAUTION: A reboot is required to enable/disable QinQ operations on the switch. When moving between QinQ modes (`qinq mixedvlan` to `qinq svlan` or vice versa), the switch boots up with a default configuration for the new qinq mode and the configuration parameters of the current mode will be erased.

QinQ Configuration example

This configuration example uses four switches to establish a QinQ tunnel through the provider network.

Figure 50: *QinQ configuration example*



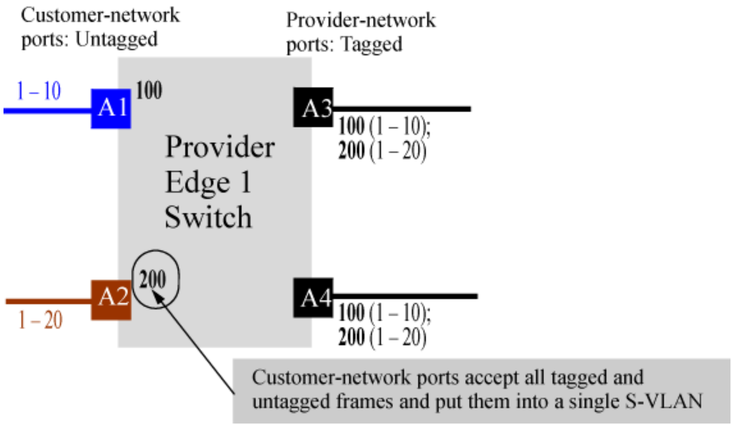
The design parameters are as follows:

- The provider edge bridge and the provider core bridge are configured in svlan mode.
- Each customer is associated with a single S-VLAN connecting two separate sites: customer A's VLANs (C-VLANs 1-10) are associated with S-VLAN 100; and customer B's VLANs (C-VLANs 1-20) are associated with S-VLAN 200.

NOTE:

- The VLANs of customers A and B can overlap: this will not result in intermixing of customer frames in the provider cloud because the S-VLANs associated with each customer are different.
- Core devices are not mandatory to establish a QinQ tunnel. For example, two edge-bridges can be connected directly to create a provider bridge network.
- The relationship between S-VLANs and C-VIDs is typically one to many. An alternative configuration might associate a single customer's C-VIDs with more than one S-VLAN. Such a configuration would most likely be used to tunnel distinct C-VIDs through various S-VLANs, but seldom be used to send the same C-VID through multiple S-VLANs.

Figure 51: *Configuration example: Edge Switch 1*


Customer-network ports: Untagged
Provider-network ports: Tagged
1 – 10
A1
100
A3
100 (1 – 10);
200 (1 – 20)
Provider Edge 1 Switch
200
A2
1 – 20
A4
100 (1 – 10);
200 (1 – 20)
Customer-network ports accept all tagged and untagged frames and put them into a single S-VLAN

At the end of the configuration, the following settings will apply:

- All customer A site traffic received on port A1 will be associated with S-VLAN 100. This is independent of the C-VLAN tag information that the customer frames may carry.
- All customer B Site 1 traffic will be associated with S-VLAN 200 and be switched out to the core (uplinks A3, A4) with the S-VLAN tag-id of 200.
- The frame size will increase by 4 since ports A3 and A4 are tagged members of S-VLAN 100 and 200.

To configure the switch, follow these steps:

Procedure

1. Enable QinQ:

```
Edgel(config)# qinq svlan tag-type 88a8
```

2. Reboot the box with the configuration saved to transfer into svlan bridge mode.

NOTE: A reboot is required for the QinQ enable command to take effect.

3. Configure S-VLANs and ports connected to the customer network.

```
Edge1(config)# svlan 100
Edge1(svlan-100)# untagged A1
Edge1(svlan-100)# exit
```

```
Edge1(config)# int A1 qinq port-type customer-network
Edge1(config)# svlan 200
Edge1(svlan-200)# untagged A2
Edge1(svlan-200)# exit
Edge1(config)# int A2 qinq port-type customer-network
```

NOTE: Customer A is assigned S-VLAN 100 and customer B is assigned S-VLAN 200. However, the same customer can be associated with more than one SVLAN. Also, interfaces A1 and A2 are configured as customer network ports because they are linked to customer bridges.

4. Configure the provider ports leading to the core of the provider network.

```
Edge1(config)# svlan 100 tagged A3, A4
Edge1(config)# svlan 200 tagged A3, A4
Edge1(config)# interface A3,A4 qinq port-type provider-network
```

NOTE: As recommended by IEEE 802.1ad specification, uplink ports should generally be configured as tagged ports for S-VLANs that are used to carry customer traffic. However, this is not a mandatory requirement on switches—S-VLANs that are used for internal provider network use (not carrying customer traffic but for management of the provider network devices) can have untagged port memberships.

QinQ Configuration example: provider Edge 2 switch

The configuration details for the Edge 2 switch mirrors the configuration for the Edge 1 switch. All customer traffic received on port A1 from customer A's site 2 will be associated with S-VLAN 100. Similarly, all customer B's site 2 traffic will be associated with S-VLAN 200.

To configure the switch, follow these steps:

Procedure

1. Enable QinQ:

```
Edge 2(config)# qinq svlan tag-type 88a8
```

2. Reboot the box with the configuration saved to transfer into S-VLAN bridge mode.
3. Configure S-VLANs and customer ports connected to the customer network.

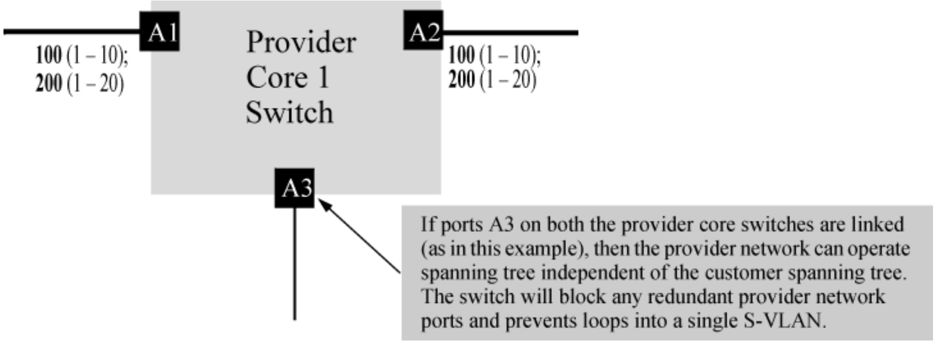
```
Edge2(config)# svlan 100
Edge2(svlan-100)# untagged A1
Edge2(svlan-100)# exit
Edge2(config)# int A1 qinq port-type customer-network
Edge2(config)# svlan 200
Edge2(svlan-200)# untagged A2
Edge2(svlan-200)# exit
Edge2(config)# int A2 qinq port-type customer-network
```

4. Configure the provider ports leading to the core of the provider network.

```
Edge1(config)# svlan 100 tagged A3, A4
Edge1(config)# svlan 200 tagged A3, A4
Edge1(config)# interface A3,A4 qinq port-type provider-network
```

Configuring example: provider core 1 switch

Figure 52: *Configuration example: Core 1 Switch*


A1
Provider
Core 1
Switch
A2
100 (1 – 10);
200 (1 – 20)
100 (1 – 10);
200 (1 – 20)
A3
If ports A3 on both the provider core switches are linked (as in this example), then the provider network can operate spanning tree independent of the customer spanning tree. The switch will block any redundant provider network ports and prevents loops into a single S-VLAN.

To configure the Core 1 switch:

Procedure

1. Enable QinQ:

```
Core 1(config)# qinq svlan tag-type 88a8
```

2. Reboot the box with the configuration saved to transfer into svlan bridge mode.
3. Configure S-VLANs and port assignments.

```
Core 1(config)# svlan 100
Core 1(svlan-100)# tagged A1, A2
Core 1(svlan-100)# exit
Core 1(config)# svlan 200
Core 1(svlan-200)# tagged A1, A2
Core 1(svlan-200)# exit
Core 1(config)#: interface A1,A2 qinq port-type provider-network
```

NOTE: The S-VLAN configuration for the core devices is based on what VLANs the edge devices (Edge 1 and 2) can send. Per the 802.1ad specification, all ports carrying customer traffic will be tagged on the VLAN that the port carries customer frames on.

To configure the Core 2 switch:

1. Enable QinQ:

```
Core 2(config)# qinq svlan tag-type 88a8
```

2. Reboot the box with the configuration saved to transfer into svlan bridge mode.
3. Configure S-VLANs and port assignments.

```
Core 2(config)# svlan 100
Core 2(svlan-100)# tagged A1, A2
Core 2(svlan-100)# exit
Core 2(config)# svlan 200
Core 2(svlan-100)# tagged A1, A2
Core 2(svlan-100)# exit
Core 2(config)# interface A1,A2 qinq port-type provider-network
```

Verifying the configuration

After the edge and core switch configurations are completed, QinQ operations can begin. To verify operations, it should be possible to assign IP-addresses to customer A or B devices in site 1 and site 2 and ping them. If everything has been configured correctly, traffic will flow through the provider network cloud and reach the other site seamlessly.

Enabling QinQ

By default, QinQ is disabled on the switch. To enable QinQ, the switch must be put into either in mixed VLAN mode or QinQ SVLAN mode by issuing one of the following commands from configuration mode on the CLI.

Syntax:

```
qinq mixedvlan tag-type [tpid]
```

From configuration mode, globally enables QinQ mixed mode, an environment that supports both S-VLAN and C-VLAN traffic on the same device. This command requires a reboot to take effect. Default: Disabled.

Syntax:

```
qinq svlan tag-type [tpid]
```

From configuration mode, globally enables QinQ SVLAN mode, an S-VLAN only environment that supports port-based or s-tagged interfaces of the standard. Requires a reboot to take effect. Default: Disabled.

Setting up S-VLANs

S-VLANs are created via the CLI using the `svlan vid` command.

Syntax:

```
svlan {vid | ascii-name-string}
```

```
no svlan vid
```

If *vid* does not exist in the switch, this command creates a port-based S-VLAN with the specified `vid` . If the command does not include options, the CLI moves to the newly created S-VLAN context. If you do not specify an optional name, the switch assigns a name in the default format: `svlan` where `n` is the `vid` assigned to the S-VLAN. If the S-VLAN already exists and you enter either the `vid` or the `ascii-name-string`, the CLI moves to the specified S-VLAN's context.

The `no` form of the command deletes the S-VLAN as follows:

If one or more ports belong only to the S-VLAN to be deleted, the CLI notifies you that these ports will be moved to the default VLAN and prompts you to continue the deletion. For member ports that also belong to another S-VLAN, there is no `move` prompt.

NOTE: When QinQ is disabled, all VLANs must be C-VLANs. When QinQ is enabled in SVLAN mode, all VLANs must be S-VLANs. When QinQ is enabled in mixed VLAN mode, VLANs can be configured as either C-VLANs or S-VLANs.

Configuring per-port S-VLAN membership

The `svlan vid` command supports tagged and untagged options to configure per-port S-VLAN memberships. Use these options from the configuration level by beginning the command with `svlan vid` , or from the context level of the specific VLAN by entering the command option.

Syntax:

`svlan` *`vid`*

`tagged` ***`port-list`***

Configures the indicated ports as `Tagged` for the specified S-VLAN. The `no` version sets the ports to either `No` or (if GVRP is enabled) to `Auto`.

`untagged` ***`port-list`***

Configures the indicated ports as `Untagged` for the specified S-VLAN. The `no` version sets the ports to either `No` or (if GVRP is enabled) to `Auto`

`forbid` ***`port-list`***

Dynamic trunks cannot be involved as a part of any static configurations like `forbid`. `Forbid` can only be applied on ports and static trunks .The `forbid` configuration can be applied to only PPORTS and Static Trunks (Trk1….Trk144), and not to Dynamic LACP Trunks (Dyn1….Dyn144.)

The `no` version sets the ports to either `no` or (if GVRP is enabled) to `Auto`.

`auto` ***`port-list`***

QinQ S-VLAN mode only. Available if GVRP is enabled on the switch. Returns the per-port settings for the specified S-VLAN to `Auto` operation. `Auto` is the default per-port setting for a static VLAN if GVRP is running on the switch.

NOTE:

Since provider-gvrp is not supported in a QinQ mixed VLAN mode environment, the `forbid` and `auto` configurations are available only in QinQ S-VLAN mode.

You can set forbid mode for MVRP enabled ports.

Set MVRP forbid mode

```
switch(eth-A1)# forbid
vlan                  Prevent this port from becoming a member of the
                      specified VLAN(s).
switch(eth-A1)# forbid vlan
[vlan]VLAN-ID-LIST    Enter a list of VLAN identifiers or one VLAN identifier.
switch(eth-A1)# forbid vlan 100
switch# show mvrp state 100

Configuration and Status - MVRP state for VLAN 100

Port     VLAN  Registrar Applicant Forbid
                State     State     Mode
-------- ----- --------- --------- ---------
A1       100    MT        AA        Yes
```

In QinQ mixed VLAN mode

An interface (port or trunk) must be explicitly GVRP-disabled before it can be assigned to the S-VLAN space. When you first attempt to configure a port as tagged for an S-VLAN, the CLI will issue a message disallowing the configuration.

```
config# svlan 200 tagged a1,a2 GVRP enabled ports cannot be members of svlans.
Disable the interface level gvrp configuration.
```

To disable GVRP at the interface, issue the following command:

```
config# interface a1,a2 unknown-vlans disable
```

When you configure the port, the CLI will issue a warning prompt:

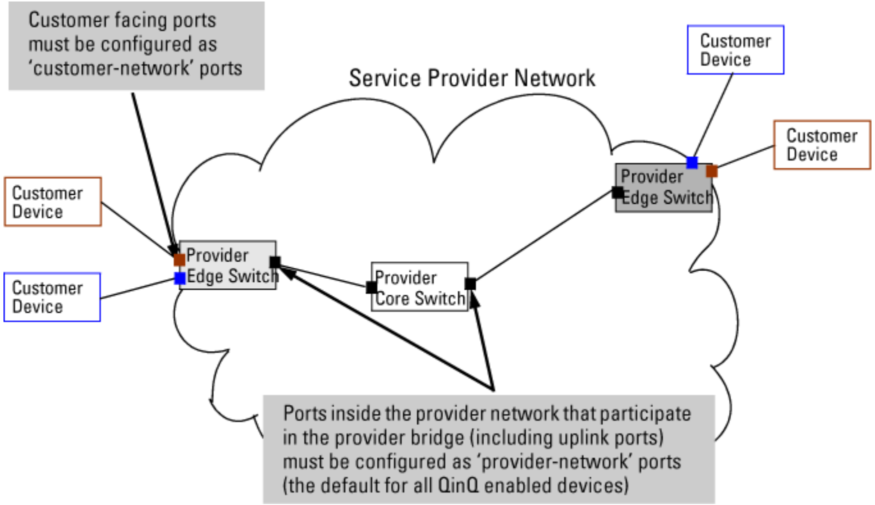
```
config# svlan 200 tagged a1,a2 Ports a1, a2 will lose their cvlan memberships if any
Do you want to continue? [y/n]
```

Press **Y** to continue and automatically configure both ports as port-type `provider-network` (the default for all S-VLAN member ports).

Configuring port-types

When QinQ is enabled on the switch all S-VLAN member ports must be categorized as either `port-type customer-network` or `provider-network` (See an example below).

Figure 53: *Customer or provider ports in the provider network*


Customer facing ports must be configured as 'customer-network' ports
Service Provider Network
Customer Device
Customer Device
Customer Device
Customer Device
Provider Edge Switch
Provider Core Switch
Provider Edge Switch
Ports inside the provider network that participate in the provider bridge (including uplink ports) must be configured as 'provider-network' ports (the default for all QinQ enabled devices)

All ports of a QinQ-enabled device default to `provider-network`. Any ports participating in the provider bridge used to connect to customer equipment must be manually configured as port-type `customer-network`. In a mixed mode device, ports that are members of C-VLANs and that do not participate in the provider-bridge cannot be configured to any port-type.

The following command allows you to configure the appropriate port-type.

Syntax:

```
no interface {[port-list] | Trkx} qinq port-type {customer-network | provider-network}
```

Configures the specified ports/trunks as a customer network port or provider network port.

Default: port-type provider (for QinQ S-VLAN mode)

Disabling QinQ

To disable QinQ once it has been enabled, issue the following commands from configuration mode on the CLI.

Syntax:

```
no qinq
```

This is the default mode when QinQ is disabled on the switch.

Moving into this configuration from another QinQ configuration requires a reboot to take effect. Upon reboot, all configuration information for the prior QinQ mode will be lost.

Default setting. Standard VLAN operations apply.

Changing VLAN port memberships (mixed VLAN mode)

On mixed VLAN mode devices, certain per-port features are not supported on S-VLANs that are supported on C-VLANs. Ports that are currently members of a regular VLAN can move to an S-VLAN only if there is no conflicting configuration.

NOTE:

To avoid a misconfiguration, HPE recommends that you use a default interface configuration when moving ports between C-VLANs and S-VLANs.

When configuring S-VLAN port memberships using the `svlan` command, the CLI issues a warning and prompt if any of the ports listed already belong to a regular VLAN. For example:

```
switch(config)# svlan 200 tagged a1,a2
Ports a1, a2 will lose their cvlan memberships if any.
Do you want to continue: y/n?
```

The warning prompt is displayed only when there is at least one port in the port list that needs to be moved out from the C-VLAN space to the S-VLAN domain. Similarly, if ports being added to the C-VLAN are already members of an S-VLAN, the CLI issues a warning that the port's membership with its existing VLANs will be removed and will prompt for a confirmation before continuing.

If all ports are just being added or removed from within the same VLAN type domain, no prompt will appear. For example, moving ports from S-VLAN 200 to S-VLAN 300, will not result in any warning as the ports are already part of the S-VLAN domain.

Moving ports between C-VLANs and S-VLANs (mixed VLAN mode)

A port (or trunk) that is a member of C-VLANs cannot be moved into the S-VLAN space with conflicting configurations for the S-VLAN mode. The following is a list of conflicting protocols/features. If a port has any of these enabled, the feature must be disabled before the port can be moved in to the S-VLAN space.

- An interface has to beGVRP-disabled to move it from the C-VLAN to the S-VLAN space. This is because S-VLANs of mixed VLAN mode do not support provider-GVRP, and also because a GVRP-enabled configuration (when the port is a C-VLAN member) is in the context of customer-GVRP which must be disabled before the port can operate in the S-VLAN space.
- Interface should not have any mirroring or monitoring sessions when moving between C-VLANs and S-VLANs. The configuration on all mirror/monitor sessions that involve the port must be removed.
- An interface that has auth-vid or unauth-vid configuration cannot move into the S-VLAN space. They have to be unset.
- Interfaces cannot have LACP enabled (active or passive modes) when moving into the S-VLAN space. They have be disabled.

Viewing QinQ configuration and status

This section outlines changes and additions to existing `show` command outputs to display QinQ configuration and status.

The `show qinq` command displays QinQ configuration information.

Syntax:

```
show qinq
```

Shows QinQ global and port configurations on the switch, including:

Bridge-mode

- cvlan bridge: QinQ is disabled, normal VLANs apply.
- mixedvlan bridge: Both S-VLANs and regular C-VLANs are available in a mixed VLAN mode environment.
- svlan: No regular VLAN commands are available. All VLANs configured on the switch are S-VLANs only.

Tag-id: Displays only if QinQ is enabled on the switch.

port-type: Displays only if QinQ is enabled on the switch. On a mixed mode device, port type is shown only for S-VLAN ports.

Viewing `show qinq` output (QinQ S-VLAN mode)

```
switch(config)# show qinq

QinQ Global Configuration:
-----------------------------------------------
Bridge-mode        : svlan bridge

QinQ Interface Configuration:
--------------------------------------------------
interface   port-type
---------   ----------
   A1       provider-network
   A2       provider-network
   Trk1     customer-network
```

Viewing a switch VLAN configuration

The following `show` commands are a subset of those listed in the chapter on Static Virtual LANs (VLANs) highlighting the changes made to show the additional QinQ VLAN types (C-VLANs and S-VLANs).

The `show vlans` command lists the VLANs currently running in the switch, including the VID, VLAN name, and VLAN status. Once QinQ is enabled in mixed VLAN mode, an additional field showing the VLAN type is added to the display output.

Syntax:

```
show vlans
```

Changes to parameters when QinQ is enabled:

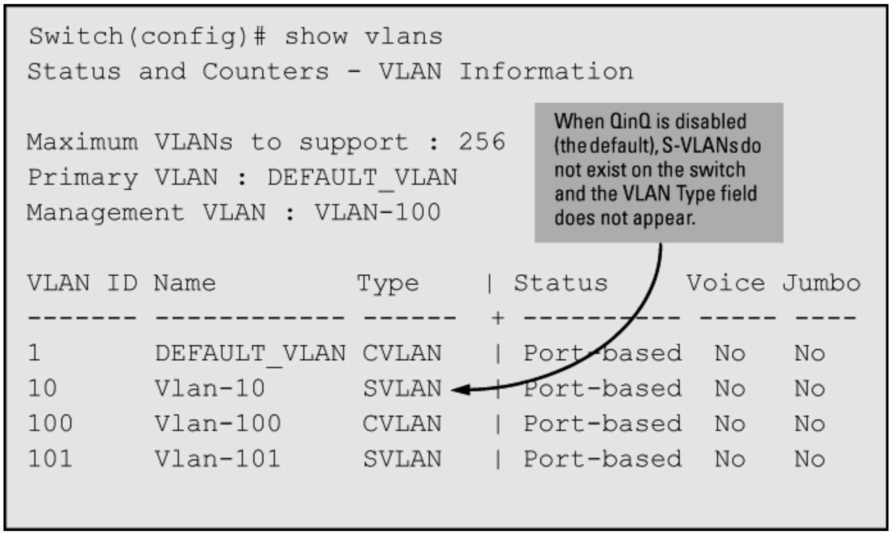
VLAN ID

Field name changes from 802.1Q VLAN ID to VLAN ID only.

Type

In a QinQ mixed mode environment, the VLAN type can be either a regular customer VLAN CVLAN, or it can be a tunnel VLAN in the provider network S-VLAN.

Figure 54: *Viewing* `show vlans` *command output with QinQ disabled*



```
Switch(config)# show vlans
Status and Counters - VLAN Information

Maximum VLANs to support : 256
Primary VLAN : DEFAULT_VLAN
Management VLAN : VLAN-100

VLAN ID Name          Type    | Status     Voice Jumbo
------- ------------ ------  + ---------- ----- ----
1       DEFAULT_VLAN CVLAN   | Port-based  No    No
10      Vlan-10      SVLAN   | Port-based  No    No
100     Vlan-100     CVLAN   | Port-based  No    No
101     Vlan-101     SVLAN   | Port-based  No    No
```

When QinQ is disabled (the default), S-VLANs do not exist on the switch and the VLAN Type field does not appear.

Viewing the configuration for a particular VLAN

This command uses the VID to identify and display the data for a specific VLAN. Once QinQ is enabled in mixed VLAN mode, an additional field showing the VLAN type is added to the display output.

Syntax:

```
show vlans vlan-id
```

Changes to parameters when QinQ is enabled:

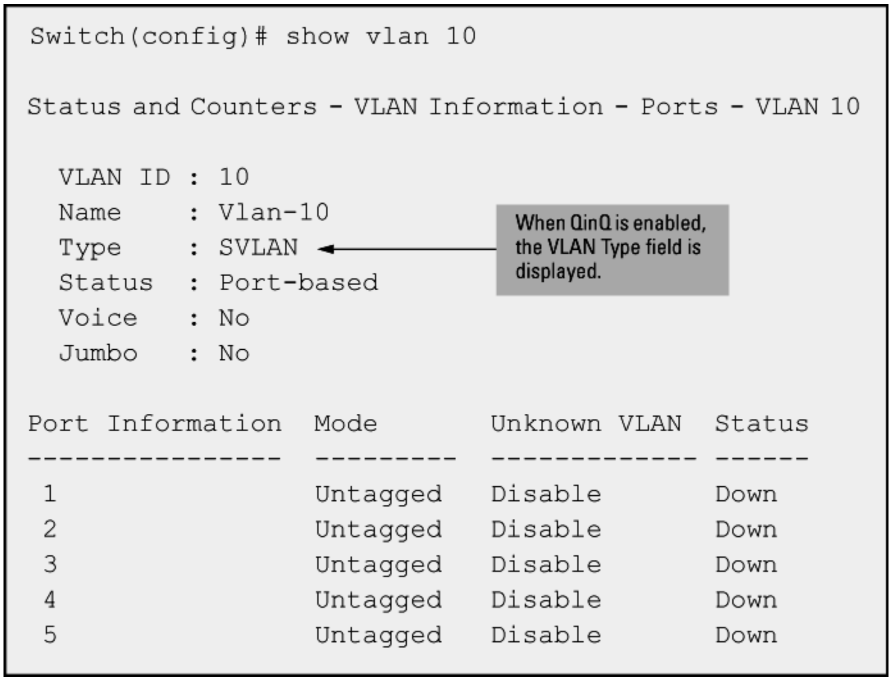
VLAN ID

Field name changes from 802.1Q VLAN ID to VLAN ID only.

Type

In a QinQ enabled environment, the VLAN type can be either a regular customer VLAN CVLAN, or it can be a tunnel VLAN in the provider network S-VLAN.

Figure 55: *Viewing `show vlan` output with QinQ enabled*



```
Switch(config)# show vlan 10

Status and Counters - VLAN Information - Ports - VLAN 10

  VLAN ID : 10
  Name    : Vlan-10
  Type    : SVLAN
  Status  : Port-based
  Voice   : No
  Jumbo   : No

Port Information  Mode       Unknown VLAN  Status
----------------  ---------  ------------- ------
 1                Untagged   Disable       Down
 2                Untagged   Disable       Down
 3                Untagged   Disable       Down
 4                Untagged   Disable       Down
 5                Untagged   Disable       Down
```

When QinQ is enabled, the VLAN Type field is displayed.

Viewing the VLAN membership of one or more ports

This command shows to which VLAN a port belongs. Once QinQ is enabled, an additional field showing the VLAN Type is added to the display output.

Syntax:

`show vlans` *`vlan-id`*

Changes to parameters when QinQ is enabled:

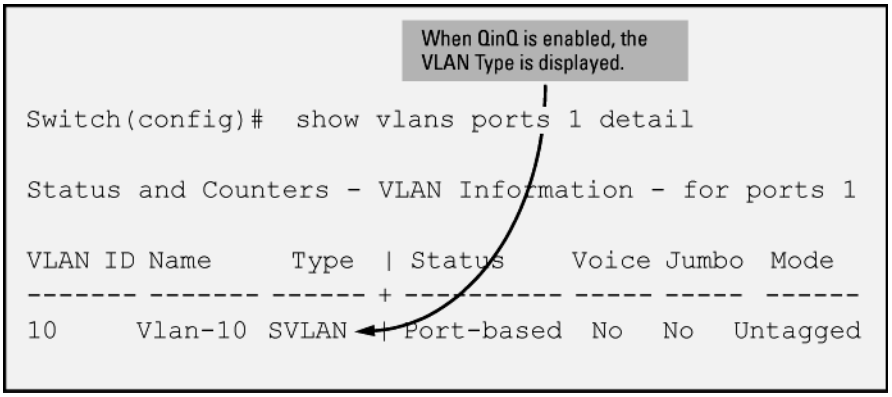
VLAN ID

Field name changes from 802.1Q VLAN ID to VLAN ID only.

Type

In a QinQ enabled environment, the VLAN type can be either a regular customer VLAN CVLAN, or it can be a tunnel VLAN in the provider network S-VLAN.

Figure 56: *Viewing VLAN membership*



When QinQ is enabled, the VLAN Type is displayed.

```
Switch(config)#  show vlans ports 1 detail

Status and Counters - VLAN Information - for ports 1

VLAN ID Name      Type  | Status     Voice Jumbo  Mode
-------- -------- ------ + ---------- ----- -----  ------
10      Vlan-10  SVLAN  | Port-based  No    No   Untagged
```

Viewing spanning tree status

In QinQ mixed mode, only ports that are members of C-VLANs will be displayed in `show spanning tree` output. This is due to the fact that ports that are members of S-VLANs do not participate in C-VLAN spanning tree and will always be in forwarding state (treated as edge ports).

About QinQ

Operating rules and guidelines

This section provides an overview of QinQ operations and restrictions on the switch.

Enabling QinQ and configuring QinQ modes

By default, QinQ is disabled. When QinQ is enabled via the CLI, an operating mode is globally configured on the switch. Two QinQ modes are supported:

qinq mixedvlan

C-VLANs and S-VLANs are both supported, with regular switching/routing based on C-VLAN tags in the C-VLAN domain, while S-VLANs are used for QinQ tunneling through the provider network.

qinq svlan

C-VLANs are not supported on the device. All configured VLANs on the switch must be S-VLANs.

The following table shows how the various QinQ modes and operations impact VLAN configuration options on the switch.

Table 28: *Relationship of QinQ operating modes to VLAN environments*

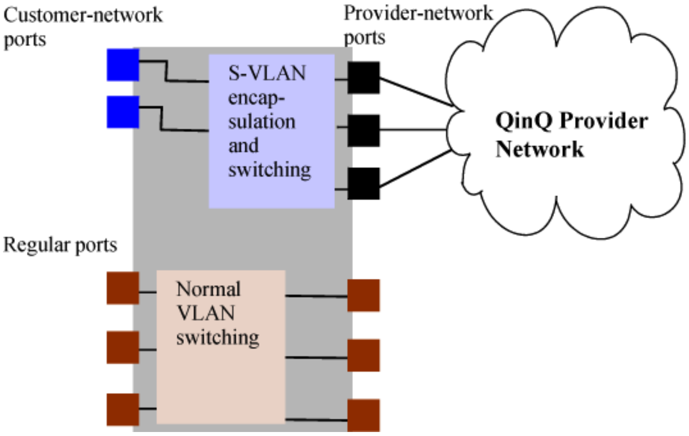
QinQ Operation	CLI Command	VLAN Options
QinQ disabled		
No QinQ support(Default)	`no qinq`	Only regular VLAN commands are available. If QinQ is disabled, S-VLAN commands are not available.
QinQ enabled		
QinQ mixed VLANmode	`qinq mixedvlan`	Both S-VLAN and regular VLAN commands (known as C-VLANs in a mixed vlan environment) are available.
QinQ S-VLAN mode	`qinq svlan`	No regular VLAN commands are available. All VLANs configured on the switch are S-VLANs only.

QinQ mixed VLAN mode

The QinQ mixed VLAN mode configuration supports both C-VLAN and S-VLAN operations on the same device. This allows the use of S-VLAN member ports for QinQ tunneling, while regular ports can still do switching or routing within the C-VLAN space. To tunnel customer frames through the provider network, you can externally connect a regular port to a customer-network port, eliminating the need for a separate S-VLAN bridge device to perform such operations. When configuring VLANs on a mixed VLAN mode device, a separate `svlan` *`vid`* command is used to distinguish the S-VLAN type from regular VLANs.

The main advantage for QinQ mixed VLAN mode is that users do not have to dedicate the entire switch as a QinQ access switch. For a high density chassis switch, customers can use regular ports for normal LAN switching, while S-VLAN member ports can be configured to access the QinQ provider network (see **Figure 50: QinQ configuration example** on page 341). There are some additional restrictions in mixed-VLAN mode.

Figure 57: *Switch in mixed-VLAN mode*



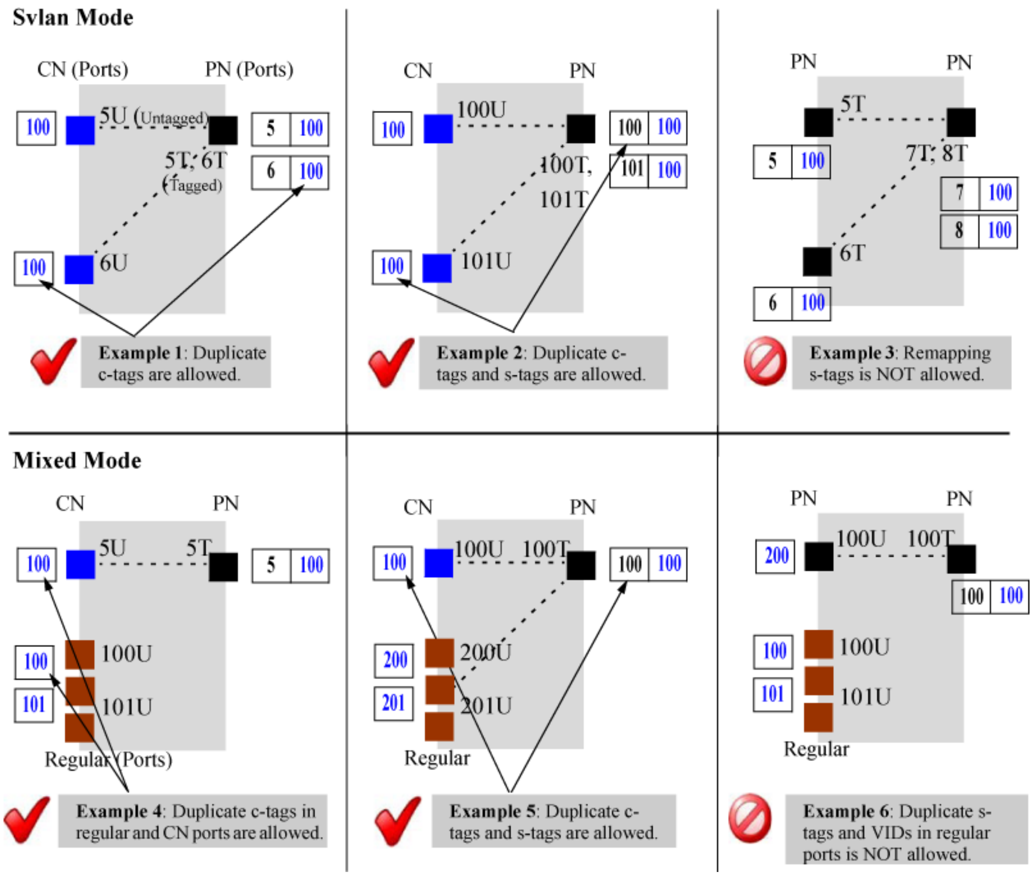
Configuring VLANs

- A VLAN created on a QinQ mixed VLAN mode device can be either a regular VLAN (C-VLAN) or a tunnel VLAN (S-VLAN). C-VLANs have no mapping/relation to the S-VLANs on the device.
- VLANs created on a QinQ S-VLAN mode device can be S-VLANs only. S-VLANs provide QinQ tunneling of customer frames and behave like a port-based/s-tagged interface.

QinQ and duplicate VIDs

Duplicate VID's for c-tagged and s-tagged VLANs (for example, C-VID=100; S-VID=100) are allowed in certain cases. Customer-network ports are essentially S-VLAN ports: they simply read the C-tags in the customer frame to insert them into the appropriate untagged S-VLAN for that port. Once this double-tagging occurs, frames are forwarded based on the S-VLAN tag only, while the C-VLAN tag remains shielded during data transmission.

Figure 58: *QinQ and duplicate VIDs: examples of allowed configurations*


Svlan Mode
CN (Ports)
PN (Ports)
100
5U (Untagged)
5 | 100
5T, 6T (Tagged)
6 | 100
100
6U
Example 1: Duplicate c-tags are allowed.
CN
PN
100
100U
100 | 100
100T, 101T
101 | 100
100
101U
Example 2: Duplicate c-tags and s-tags are allowed.
PN
PN
5T
5 | 100
7T, 8T
7 | 100
8 | 100
6T
6 | 100
Example 3: Remapping s-tags is NOT allowed.
Mixed Mode
CN
PN
100
5U
5T
5 | 100
100
100U
101
101U
Regular (Ports)
Example 4: Duplicate c-tags in regular and CN ports are allowed.
CN
PN
100
100U
100T
100 | 100
200
200U
201
201U
Regular
Example 5: Duplicate c-tags and s-tags are allowed.
PN
PN
200
100U
100T
100 | 100
100
100U
101
101U
Regular
Example 6: Duplicate s-tags and VIDs in regular ports is NOT allowed.

Assigning ports to VLANs

In mixed VLAN mode, a port can be a member of a C-VLAN or of an S-VLAN but not both.

Configuring port types

The IEEE 802.1ad standard requires that every S-VLAN member port be configured as either a provider-network or as a customer-network port. In a typical deployment scenario, customer-network ports will be configured as untagged members of S-VLANs while provider-network ports will be configured as tagged members of S-VLANs. Note the following configuration rules and guidelines:

- All ports of a device that is QinQ enabled (in S-VLAN mode or mixed VLAN mode) are provider-network ports by default—if there are any ports that connect to a customer device, they must be manually configured as customer-network ports.
- Configuring a port-type is applicable only if the device is QinQ enabled and the port is a member of an S-VLAN. In QinQ mixed mode, ports that are members of C-VLANs cannot be configured to any port-type.

NOTE:

If a device running in QinQ S-VLAN mode has one or more customer-network ports, it is considered to be a provider edge and not a provider core bridge. This may affect certain operations, such as meshing, UDLD, and stacking. This is because at the edge of the provider network such proprietary protocol are filtered out at customer network ports. This prevents the intermix of stacking meshing/ UDLD protocols in the customer and provider domains (since they use the same `dst-mac` address in either domain).

Operating notes and restrictions

Cannot run concurrently with RPVST+

QinQ cannot run concurrently with RPVST+

Changing bridge modes requires a reboot

When changing the operating mode (to/from: QinQ S-VLAN mode, QinQ mixed VLAN mode, or QinQ disabled), you will prompted to restart the system before the changes can take effect. Upon reboot, all configuration information for the prior QinQ mode will be lost. Any configurations created will be erased, and the device will boot up with a default configuration for the new QinQ mode.

Provider edge devices at Layer 2 only

QinQ does not provide Layer 3 capabilities of complete network isolation between customers. In a mixed VLAN configuration, there is no switching/routing between C-VLANs and S-VLANs. S-VLANs are essentially Layer 2 VLANs that switch packets based on S-VIDs.

IP support

Regular VLANs support IP and can be routing enabled. S-VLANs of mixed VLAN mode devices cannot be ip enabled. S-VLANs of S-VLAN mode devices can be ip-enabled, though routing related features (such as ip routing) are not supported.

Double-tagging causes frame size increases

Since there is both a provider VLAN tag and customer VLAN tag in each QinQ frame, the size of each double-tagged frame increases by 4 bytes. To accommodate the frame size increase, HPE recommends that you configure all port-based S-VLANs to accept jumbo frames.

S-VLAN configuration restrictions

S-VLAN commands are not available when QinQ is disabled on the switch.

VLAN configuration restrictions in mixed VLAN mode

- Both C-VLANs and S-VLANs can be configured on the switch. In a mixed mode device, the default VLAN is always a C-VLAN.
- VLAN types cannot be updated dynamically. A VLAN can be classified only as an S-VLAN or a C-VLAN at the time its created. Once created, the VLAN cannot be moved between being a C-VLAN and an S-VLAN. If a VID that was initially created as a regular VLAN needs to be used for an S-VLAN, the VID must be deleted and re-created as an S-VLAN.
- If a VLAN being configured as an S-VLAN already exists as a GVRP C-VLAN or a static C-VLAN on the switch, the S-VLAN creation is blocked. Similarly, a C-VLAN creation is blocked if the same VID exists as a static S-VLAN on the device.
- S-VLANs in a mixed vlan device cannot be configured as a voice-VLAN, primary-VLAN, or management-VLAN.
- S-VLANs cannot be configured with ip-layer functionality, except for ip-acls.

VLAN configuration restrictions in S-VLAN mode

- Only S-VLANs are supported—the keyword on all vlan-related command syntax changes from `vlan` to `svlan`.
- Routing related features such as ip-routing, RIP, OSPF, PIM, and VRRP are not supported in S-VLAN mode.

Port-based restrictions

- In QinQ mixed VLAN mode, a port must be explicitly GVRP-disabled before it can be assigned to the S-VLAN space.
- In QinQ mixed VLAN mode, only ports that are members of S-VLANs can be configured as customer network or provider network ports; ports that are members of C-VLANs cannot be configured to any port-type.
- QinQ mixed VLAN mode devices cannot be connected in an S-VLAN mesh topology. This is because STP cannot be run in the S-VLAN space, and so amesh topology (or the presence of any redundant links) would result in loops.
- A port can either be a member of S-VLANs or C-VLANs only, but not a combination of both.
- A port cannot be configured as a Customer-Edge as specified in Section 12.13.3 of the IEEE 802.1ad specification. In the current software release, such C-tagged interfaces are not supported—only port-based/S-tagged interfaces are supported.
- Moving ports between C-VLANs and S-VLANs may cause conflicts. For example, if a port has any mirroring/monitoring sessions set up, they will not be allowed to change VLAN domains until these sessions are re-configured.

Interoperating with other vendor devices

When enabling QinQ, you can configure a unique tpid value, such as 0x8100, to allow the device to interoperate with devices that require this value for the inner and outer VLAN-tag. If the provider tag-type is configured as 0x8100, then:

- Customer-network ports cannot be configured as tagged-S-VLAN members
- Tagged-S-VLAN members cannot be configured as customer-network ports.

Configuring QinQ with other network protocols

The networks for both the customer and provider can be complex. For information on how QinQ may impact other network protocols (such as spanning tree, LLDP, and GVRP), see the figure in **QinQ mixed VLAN mode**.

Changing QinQ modes

Changing QinQ modes (or disabling QinQ operations) will result in the current configuration being erased. See the following Caution for details.

CAUTION: Configuring the switch to operate in a different bridge mode requires a reboot to take effect. Upon reboot, all configuration information for the prior QinQ mode is lost. Any configurations created under the existing QinQ mode is erased, and the device boots up with a default configuration for the new QinQ mode.

For information on the effect of the different QinQ modes on switch protocols and operations, see the table Impacts of QinQ configurations on other switch features.

Effects of QinQ on other switch features

Per the IEEE standards, protocols such as STP and GVRP are assigned separate addresses for customer networks and provider networks, ensuring that QinQ has no impact on their operations. Bridge Protocol Data Units (BPDUs) that need to be tunneled through the provider network are treated as normal multicast frames at the provider bridge and forwarded out.

However, other protocols use common addresses for both customer and provider networks, and so are not supported when QinQ is enabled on the switch. Similarly, proprietary features such as discovery, UDLD, and loop-protect do not provide tunneling support. In such cases, where provider networks could run an instance of the same protocol as a customer could run local to their site, these frames are dropped at the customer-network ports of the provider bridge.

NOTE: The IEEE standards group is devising new addressing schemes that may support additional QinQ tunneling operations. Check the latest product release notes for implementation updates as they apply to switches.

When QinQ is not enabled (the default setting), there are no impacts to the switch's normal operations. The following table shows the impacts of QinQ on the operation of switch protocols and features based on the QinQ mode that is configured as QinQ mixed VLAN mode (C-VLANs and S-VLANs are allowed) or QinQ S-VLAN mode (S-VLANs only).

Table 29: *Impacts of QinQ configurations on other switch features*

Switch feature	Impacts of QinQ configurations and allowed operations
ACLs	In QinQ mixed VLAN or S-VLAN modes: On double-tagged frames , the VID applicable when applying ACLs will be the S-VLAN tag and not the C-VLAN tag.
aaa	In QinQ mixed VLAN mode: • auth-vid/unauth-vid configuration is not supported on S-VLAN ports; the auth-vid/unauth-vid cannot be an S-VLAN id. • If a port that is a member of C-VLANs is configured with auth-vid or unauth-vid and it needs to be added to the S-VLAN domain, the auth/unauth configuration must first be undone.
arp-protect	In QinQ mixed VLAN mode: ARP-protect is not supported on S-VLANs, nor on S-VLAN ports.
CDP	In QinQ VLAN or S-VLAN modes: CDP frames are consumed at customer network ports, if CDP is enabled on the device port, and the customer device shows up as a CDP neighbor on the customer-network port. If not, the frames are dropped.

Table Continued

Switch feature	Impacts of QinQ configurations and allowed operations
DHCP	In QinQ mixed VLAN or S-VLAN modes: • DHCP relay applies only to C-VLANs. • DHCP snooping is not supported on S-VLANs.
directed-broadcast	In QinQ S-VLAN mode: directed-broadcast is not supported on provider core devices.
GVRP	In QinQ mixed VLAN mode: • S-VLAN ports cannot be GVRP enabled. • Regular VLANs will participate in C-VLAN GVRP if enabled to do so. S-VLANs will tunnel all C-VLAN GVRP frames through. • An explicit GVRP disable on a port is a prerequisite for moving the port to an S-VLAN domain. • Port-based interfaces do not have support for provider-GVRP protocols. Provider GVRP frames received at S-VLAN interfaces will be dropped. • If a VLAN being configured as an S-VLAN is already a GVRP VLAN on the switch, this S-VLAN creation would be blocked. In QinQ S-VLAN mode: GVRP is supported on S-VLAN ports if the qinq mode is S-VLAN.
igmp-proxy	In QinQ mixed VLAN mode: IGMP-proxy cannot be configured on S-VLANs. In QinQ S-VLAN mode: IGMP-proxy is not supported.
IPv6	In QinQ mixed VLAN mode: IPv6 features are not supported on S-VLANs.

Table Continued

Switch feature	Impacts of QinQ configurations and allowed operations
ip-recv-mac	In QinQ mixed VLAN mode: `ip-recv-mac` cannot be configured on S-VLANs. In QinQ S-VLANmode: `ip-recv-mac` is not supported.
Jumbo	In QinQ mixed VLAN or S-VLAN modes: No change in operations. HPE recommends to jumbo-enable all S-VLANs used for customer data tunneling to support the addition of the extra S-tag in each frame.
LACP/ Port Trunks	In QinQ mixed VLAN mode: • Dynamic-LACP is not supported on S-VLAN ports: LACP manual trunks alone are supported. The new trunk will be a member of C-VLANs (port types are not applicable). • If two ports are added to a trunk, the resultant trunk will be a member of the default-vlan (vid-1) which is always a C-VLAN. The trunk can subsequently be manually assigned to an S-VLAN. • Port-type and VLAN configurations are not mapped. If the port-type is updated through CLI or SNMP and the port is subsequently moved from the C-VLAN space to the S-VLAN space then back again, the last configured port-type is retained through each move. In QinQ S-VLAN mode: • On S-VLAN bridges, both manual and dynamic LACP trunks are supported. HPE does not recommend that you configure dynamic trunks on customer ports because they cannot become dynamic members of S-VLANs (there is no provider-gvrp for a dynamic trunk to become a member of S-VLANs.) • A newly formed trunk will by default be of type provider-network. When the trunk is manually assigned to an S-VLAN for the first time after being created, the port-type is provider-network.

Table Continued

Switch feature	Impacts of QinQ configurations and allowed operations
Layer 3 Protocols (IP, IP+, DHCP, ARP, IGMP Layer 3, Layer 3 ACLs)	In QinQ mixed VLAN mode: • There is no IP layer functionality on S-VLANs. • No change in IP layer functionality on regular C-VLANs. • S-VLANs cannot be configured as RIP, OSPF, PIM, or VRRP interfaces. In QinQ S-VLAN mode: • S-VLANs can be ip enabled. • IP routing is not supported.
LLDP	In QinQ mixed VLAN or S-VLAN modes: • LLDP is supported on the device (in both qinq modes). However, there is no provision for tunneling customer LLDP BPDUs through the provider-network. • LLDP BPDUs received from a customer's network will be consumed at the customer-network ports of a provider device and the customer device will be displayed as an LLDP neighbor. Similarly the provider network device will show up as a neighbor on the customer's network if the customer-network ports send out LLDP advertisements.
load-sharing	In QinQ S-VLAN mode: Equal cost multi-path (ECMP) is not supported on provider core devices.
management VLAN	In QinQ mixed VLAN mode: The management VLAN cannot be an S-VLAN.
Mirroring/ Monitoring	In QinQ mixed VLAN mode: • Remote mirroring is not supported on S-VLANs. • Cannot monitor a VLAN with mirror ports in the other VLAN domain. That is, an S-VLAN or an S-VLAN port cannot be monitored using a C-VLAN port as its mirror, and vice-versa. • When a port is moved from the S-VLAN space to the C-VLAN space (or vice versa), all mirror/monitor sessions on the port must be unconfigured before the move will be allowed.
multicast-routing	In QinQ S-VLAN mode: Multicast routing is not supported on provider core devices.

Table Continued

Switch feature	Impacts of QinQ configurations and allowed operations
QoS	In QinQ mixed VLAN or S-VLAN modes: HPE does not recommend that you enable DSCP on S-VLANs used for tunneling as the customer IP-pkt will be modified in the S-VLAN space.
Routing	In QinQ S-VLAN mode: Routing is not supported on provider core devices.
source-binding	In QinQ mixed VLAN or S-VLAN modes: source-binding cannot be configured on S-VLANs.
source-route	In QinQ S-VLAN mode: source-route is not supported on provider core devices.
Spanning Tree	In QinQ mixed VLAN mode: • Customer (C-VLAN) spanning tree is supported. All C-VLAN ports will receive/transmit customer STP BPDUs and participate in regular VLAN spanning tree as usual. • When customer STP BPDUs are received at S-VLAN ports on the switch, they will be flooded out of the other ports on the S-VLAN. All such frames will be tunneled through the S-VLAN tunnel unscathed. • Provider (S-VLAN) spanning tree is not supported on the switch. If S-VLAN STP frames are received on any S-VLAN enabled ports, they will be re-forwarded out of the other ports on the S-VLAN. • STP configuration on S-VLAN ports is not supported. • If a port that is a member of C-VLANs is moved into being a member of S-VLANs, the port would, by default, tunnel customer STP BPDUs. • If a C-VLAN port has been configured with any non-default STP parameters (such as `admin-edge, auto-edge`, and `bpdu-protect`) and is then moved into an S-VLAN, the port will be put into a forwarding state regardless of the STP configurations done when the port was a member of the C-VLAN. • MSTP instances cannot include S-VLANs. In QinQ S-VLAN mode: • Provider (S-VLAN) spanning tree is supported—both provider-network ports and customer-network ports will receive/transmit provider STP BPDUs. • Customer (VLAN) spanning tree tunneling is supported on S-VLAN interfaces—customer-network or provider-network ports will tunnel customer STP BPDUs through the appropriate S-VLAN.

Table Continued

Switch feature	Impacts of QinQ configurations and allowed operations
Stacking	In QinQ mixed VLAN mode: Stacking is supported only on C-VLANs. The device does not advertise itself (using the stack discovery protocol) in the S-VLAN space. In QinQ S-VLAN mode: Configuring QinQ with S-VLANs in a switch is not supported. Stacking discovery protocol frames will not be sent out of customer-network ports; similarly, any stacking discovery protocol frames received on customer-network ports will be dropped.
UDLD	In QinQ mixed vlan or S-VLAN modes: • UDLD frames received on udld-disabled customer network ports will be dropped. However, if the customer-network port is udld-enabled, it can peer with a customer device. • UDLD frames received on udld-disabled provider network ports will be re-forwarded out of other udld-disabled provider network ports on the same VLAN. • UDLD re-forwarding in the C-VLAN space (QinQ disabled or mixed VLAN mode) will remain unaltered.
udp-bcast-forward	In QinQ S-VLAN mode: `udp-bcast-forward` is not supported on provider core devices.
unknown-vlans	In QinQ mixed VLAN mode: • GVRP (learn and disabled modes) not supported on S-VLAN ports. • A C-VLAN port that has GVRP enabled will need to disable it before it can be added to S-VLANs.
Voice VLANs	In QinQ mixed VLAN mode: S-VLANs cannot be configured as voice-VLANs.
VRRP	In QinQ mixed VLAN or S-VLAN modes: VRRP is not supported on S-VLANs.

NOTE: All commands previously in the Summary of commands table are indexed under the entry **Command syntax**.

Introduction

Classifier-based service policies are designed to work with existing globally configured switch-wide and port-wide settings by allowing you to select a subset of:

- Traffic sent to or from certain ports
- VLAN traffic

Once the traffic is selected, you can further manage it.

Classifier-based service policies take precedence over, and may override, globally configured settings. These policies provide greater control for managing network traffic. Using multiple match criteria, you can finely select and define the classes of traffic that you want to manage. You can then use policy actions to determine how the selected traffic is handled.

Classes can be based on IPv4 or IPv6 addresses (which you specify in the policy).

Configuring a traffic class

To configure a traffic class to be used in one or more policies, follow these steps:

Procedure

1. Enter the `class` command from the global configuration context.

   Context: Global configuration

   Syntax

   ```
   no class [ipv4 | ipv6 | mac] classname
   ```

   Defines a traffic class and specifies whether a policy is to be applied to IPv4 or IPv6 packets, where *classname* is a text string (64 characters maximum).

   After you enter the `class` command, you enter the class configuration context to specify match criteria. A traffic class contains a series of `match` and `ignore` commands, which specify the criteria used to classify packets.

   The `no` form of the command removes the existing class .

2. Enter one or more `match` or `ignore` commands from the traffic class configuration context to filter traffic and determine the packets on which policy actions will be performed.

   Context: Class configuration

   Syntax

   ```
   no [seq-number] [match | ignore] igmp source-address destination-address [igmp-
   type] [ip-dscp codepoint] [precedence precedence-value] [tos tos-value] [vlan
   vlan-id]
   ```

`seq-number`	(Optional) Sequentially orders the match/ignore statements that you enter in a traffic class configuration. Packets are checked by the statements in numerical order. Default: Match/ignore statements are numbered in increments of 10, starting at 10. To re-number the match/ignore statements in a class configuration, use the `resequence` command.
`match | ignore`	Defines the classifier criteria used to determine which packets belong to the traffic class. If a packet matches a `match` criterion, it becomes a member of the traffic class and is forwarded according to the actions configured with the `policy` command. If a packet matches an `ignore` criterion, no policy action is performed on the packet. You can enter one or more match/ignore statements in a traffic class. To remove a match/ignore statement from a class configuration, enter the `no` *`seq-number`* command or the complete form of a `no match` or `no ignore` command.

Table Continued

ip-protocol	Specifies an IP protocol to be matched in packet fields of IPv4 or IPv6 traffic, where *ip-protocol* is one of the values described below. When entering a match/ignore command in an IPv4 or IPv6 class, enter ? to display a list of valid *ip-protocol* entries. In an IPv4 class, you can enter any of the following IPv4 protocol match criteria: ahesp gre icmp[1] igmp[1] ipip-in-ip ipv6–in-ip ospf pim sctptcp[1] udp[1] vrrp To specify an IPv4 protocol as match criteria, you can also enter its protocol number. Valid values are from 0 to 255. For example, 8 means Exterior Gateway Protocol; 121 means Simple Message Protocol. For a list of IPv4 protocol numbers and corresponding protocol names, see the IANA "Protocol Number Assignment Services" at **www.iana.com**. In an IPv6 class, you can enter any of the following IPv6 protocol match criteria: ahesp icmp[2] ipv6 sctptcp[2] udp[2]

Table Continued

source-address *destination-address*	Defines the source IP address (SA) and destination IP address (DA) that a packet must contain to match a match/ignore statement in an IPv4 or IPv6 traffic class. Both the source and destination address parameters are required entries in a match/ignore statement. Valid values for *source-address* and *destination-address* are as follows: • `any`: Matches IPv4 or IPv6 packets from, or destined to, any SA or DA. • `host [ SA | DA ]`: Matches only packets from a specified IPv4 or IPv6 host address. Use this match criterion when you want to match IP packets from only one SA/DA. • *SAv4 mask* | *DAv4 mask*: Matches packets received from, or destined to, a subnet or a group of IP4 addresses defined by the IPv4 mask. Enter an IPv4 mask in dotted-decimal format for an IPv4 address (for example, 10.28.31.1 0.0.0.255). **NOTE:** An IPv6 address and mask are not supported as `SAv6 mask` and `DAv6 mask` match criteria. *SAv4/mask-length* | *DAv4/mask-length*: Matches packets received from, or destined to, an IPv4 subnet or a group of IPv4 addresses defined by the mask length. Enter the mask length for an IPv4 SA or DA mask in CIDR format by using the number of significant bits. (for example, 10.28.31.3/24). An IPv4 mask-length is applied to an SA or DA in a match/ignore statement to define which bits in a packet's SA/DA must exactly match the specified SA/DA and which bits need not match. For example, 10.28.31.3/24 means that the leftmost 24 bits in an IPv4 source or destination address in a packet header must match the same bit set in the specified IPv4 address (in this case, 10.28.3.3). An IPv4 mask-length is applied from right to left, starting from the rightmost bits. For example, 10.10.10.1/24 and 10.10.10.1 0.0.0.255 both match IPv4 addresses in the range 10.10.10.(1 to 255). **NOTE:** Specifying a group of non-contiguous IP source addresses may require more than one match/ignore statement. *SAv6/prefix-length* | *DAv6/prefix-length*: Matches packets received from, or destined to, an IPv6 subnet or a group of IPv6 addresses defined by the prefix length. Enter the prefix length for an IPv6 SA/DA in CIDR format by using the number of significant bits; for example: 2001:db8:2620:212::01b4/64. An IPv6 prefix-length is applied to an SA/DA in a match/ignore statement to define which bits in a packet's SA/DA must exactly match the specified SA/DA and which bits need not match. For example, 2001:db8:2620:212::01b4/64 means that the leftmost 64 bits in a 128-

Table Continued

	bit IPv6 source or destination address in a packet header must match the same bit set in the specified IPv6 address (in this case, 2001:db8:2620:212::01b4). An IPv6 prefix-length is applied from left to right, starting from the leftmost bits. For example, 2001:db8::0001: 2620:a03:e102:127/64 and 2001:db8::1: 244:17ff:feb6:d37d/64 both match IPv6 addresses with a network prefix of 2001:db8:0000:0001.

`ip-dscp codepoint`	(Optional) Matches the six-bit DSCP codepoint DSCP codepoint in IPv4 or IPv6 packets to further define match criteria. Valid values for `codepoint` are one of the following:
	• Numeric equivalent of a binary DSCP bit set from `0` (low priority) to `63` (high priority)
	• ASCII standard name for a binary DSCP bit set
	`af11 (001010) af42 (100100)`
	`af12 (001100) af43 (100110)`
	`af13 (001110) ef`
	`(101110) af21 (010010) cs1 (001000)=precedence 1`
	`af22 (010100) cs2 (010000)= precedence 2`
	`af23 (010110) cs3 (011000)= precedence 3`
	`af31 (011010) cs4 (100000)= precedence 4`
	`af32 (011100) cs5 (101000)= precedence 5`
	`af33 (011110) cs6 (110000)= precedence 6`
	`af41 (100010) cs7 (111000) = precedence 7`
	`default (000000)`
	To display a list of valid `codepoint` entries when you enter `ip-dscp` in a match/ignore statement, enter **?**.
	The DSCP codepoints are the leftmost six bits of the ToS/Traffic Class byte.

Table Continued

`precedence` *`precedence-value`*	(Optional) Matches the three-bit IP precedence value in IPv4 or IPv6 packets to further define match criteria. Valid values for precedence-value are either the numeric value (0 to 7) or corresponding name of an IP precedence bit set: 0 routine 1 priority 2 immediate 3 flash 4 flash-override 5 critical 6 internet (for internetwork control) 7 network (for network control) To display a list of valid *`precedence-value`* entries when you enter `precedence` in a match/ignore statement, enter **?**. **NOTE:** When used as a match criteria, the IP precedence value is applied to all other criteria configured in the match/ignore statement. You can enter a match/ignore statement either with or without a precedence-value. The IP precedence bits are the leftmost three bits of the ToS/Traffic Class byte. The numeric value (0 to 7) of the IP precedence bits corresponds to the hexadecimal equivalent of the three binary 0 and 1 bits in the IP precedence field. For example if the IP precedence-bit binary values are `1 1 1`, the numeric value is 7 (1+2+4). Similarly, if the IP precedence bits are `0 1 0`, the numeric value is `2` (0+2+0).

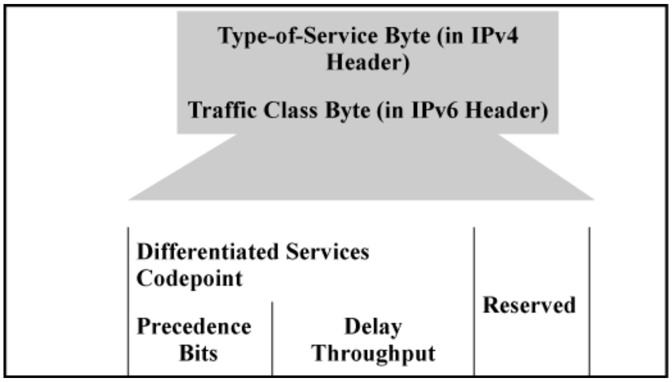
Table Continued

`tos tos-value`	(Optional) Matches the Delay Throughput Reliability (DTR) bit set in the IPv4 Type-of-Service or IPv6 Traffic Class byte to further define match criteria. Valid values are the numeric value or corresponding name of the DTR bit set. Some useful values are as follows: 0 — normal 2 — max-reliability 4 — max-throughput 8 — minimize-delay Default: 0 or `normal`. To display a list of valid `tos-value` entries when you enter `tos` in a match/ignore statement, enter **?**. **NOTE:** When used as a match criteria, the ToS/Traffic Class byte entry is applied to all other criteria configured in the match/ignore statement. You can enter a match/ignore statement either with or without a `tos-value`.
`vlan vlan-id`	(Optional) Matches the VLAN ID number in the Layer 2 header of 802.1Q VLAN packets to further define match criteria. Valid VLAN IDs are from 1 to 4094. The image below, shows a sample ToS/Traffic Class field of `10101000` to show the differences between the IP precedence (`101`), DSCP (`101010`), and ToS/Traffic Class (`10101000`) bits. The rightmost two bits are reserved as `00`.

[1] For IPv4 ICMP, IGMP, TCP, and UDP packets, you can enter additional match criteria.

[2] For IPv6 ICMP, TCP, and UDP packets, you can enter additional match criteria; see **Defining the ICMP match criteria**, **Defining the IGMP match criteria**, and **Defining TCP and UDP match criteria**

Figure 59: *A ToS class field*


Type-of-Service Byte (in IPv4 Header)
Traffic Class Byte (in IPv6 Header)
Differentiated Services Codepoint
Precedence Bits
Delay Throughput
Reserved

3. To display a class configuration, enter the following command.

```
show class [ipv4 | ipv6] [classname]
```

To edit a class configuration, re-enter the class configuration context (`class` command) and enter new match/ ignore statements as follows:

- If you do not enter a sequence number, a new statement is inserted at the end of the class configuration.
- To remove a match/ignore statement from a class configuration, enter the `no` *`sequence-number`* command or the complete form of the `no match` or `no ignore` command.
- To `resequence` the order in which match/ignore statements are listed, include the `resequence` option in the class command.
- To replace an existing match/ignore statement, enter the `no` *`sequence-number`* command to delete the entry and re-enter a complete *`sequence-number`* `match` or *`sequence-number`* `ignore` command.

When exiting the class configuration context, the changes are automatically saved and applied to existing policy configurations on the switch that use the class if the policies have not been applied to an interface. If a policy has already been applied to an interface, the editing changes are not accepted and an error message is displayed.

Class configurations

The following example shows two class configurations:

- `AdminTraffic`

  selects the administrative traffic sent to, and received from, the IPv4 address of an administrator's PC.
- `http`

  selects HTTP traffic sent to TCP ports 80, 443, and 8080, and excludes HTTP traffic sent to, and received from, TCP port 1214.

```
switch(config)# class ipv4 AdminTraffic
switch(config)# match ip 15.29.16.1/10 any
switch(config-class)# match ip any 15.29.16.1/10
switch(config-class)# exit
switch(config)# class ipv4 http
switch(config-class)# match tcp any any eq 80
switch(config-class)# match tcp any any eq 443
switch(config-class)# match tcp any any eq 8080
switch(config-class)# ignore tcp any eq 1214 any
switch(config-class)# ignore tcp any any eq 1214
switch(config-class)# exit
```

Defining the ICMP match criteria

To more precisely define the ICMP packets to match in an IPv4 or IPv6 traffic class, use the optional parameter settings below. For example, instead of matching or ignoring all ICMP traffic, you can configure a class that matches only a specific ICMP packet type by entering its numeric value.

Context: Class configuration

Syntax:

`no [`*`seq–number`*`] [match | ignore] [icmp]` *`source–address destination–address`* `[`*`icmp–type–number`* `|` *`icmpv4–type–name`* `|` *`icmpv6–type–name`*`] [ip–dscp` *`codepoint`*`] [precedence` *`precedence–value`*`] [tos` *`tos–value`*`] [`*`vlan–id`*`]`

If you enter icmp as the IP protocol type in a match/ignore statement, you can optionally specify an ICMP packet type to more precisely define match criteria for a traffic class. Enter the optional ICMP match criteria immediately after the destination address (DA) value in the command syntax; for example:

```
switch(config-class)# match icmp any any host-unknown
switch(config-class)# match icmp any any 3 7
```

icmp-type-number

Configures an ICMP packet type as match criteria in a class configuration by entering its numeric identifier. Valid values are from 0 to 255.

For information on ICMP packet-type names and numeric identifiers, go to the Internet Assigned Numbers Authority (IANA) website at **www.iana.com**, click **Protocol Number Assignment Services**, and then go to the selections under **Internet Control Message Protocol (ICMP) Parameters**.

icmpv4-type-name

Enter any of the following ICMPv4 packet-type names to configure more precise match criteria for ICMP packets in an IPv4 class configuration.

To display a list of valid icmpv4-type-name entries when entering icmp as the IP protocol type in a match/ignore statement, enter **?**. Some of the valid values are:

- administratively-prohibitednet-tos-unreachable
- alternate-addressnet-unreachable
- conversion-errornetwork-unknown
- dod-host-prohibitedno-room-for-option
- dod-net-prohibitedoption-missing
- echopacket-too-big
- echo-replyparameter-problem
- general-parameter-problemport-unreachable
- host-isolatedprecedence-unreachable
- host-precedence-unreachableprotocol-unreachable
- host-redirectreassembly-timeout
- host-tos-redirectredirect
- host-tos-unreachablerouter-advertisement
- host-unknownrouter-solicitation
- host-unreachablesource-quench
- information-replysource-route-failed
- information-requesttime-exceeded
- mask-replytimestamp-reply
- mask-requesttimestamp-request
- mobile-redirecttraceroute

- `net-redirectttl-exceeded`
- `net-tos-redirectunreachable`

icmpv6-type-name

You can also enter any of the following ICMPv6 packet-type names to configure more precise match criteria for ICMP packets in an IPv6 class configuration.

To display a list of valid `icmpv6-type-name` entries when you enter `icmp` as the IP protocol type in a match/ ignore statement, enter **?**. Some of the valid values are as follows:

- `cert-path-advertisemobile-advertise`
- `cert-path-solicitmobile-solicit`
- `destination-unreachablend-na`
- `echo-replynd-ns`
- `echo-requestnode-info`
- `home-agent-replynode-query`
- `home-agent-requestpacket-too-big`
- `inv-nd-naparameter-problem`
- `inv-nd-nsredirect`
- `mcast-router-advertiserouter-advertisement`
- `mcast-router-solicitrouter-renum`
- `mcast-router-terminate router-solicitation`
- `mld-done time-exceeded`
- `mld-query ver2-mld-report`
- `mld-report`

Defining the IGMP match criteria

To more precisely define the IGMP packets to match in an IPv4 traffic class, use the optional parameter settings described in this section. For example, instead of matching all IGMP traffic, configure a class that matches only a specific IGMP packet type.

Context: Class configuration

Syntax:

```
no [seq—number] [match | ignore] igmp source—address destination—address [igmp—
type] [ip—dscp codepoint] [precedence precedence—value] [tos tos—value] [vlan vlan—
id]
```

If you enter `igmp` as the IP protocol type in a match/ignore statement, you can optionally specify an IGMP packet type to more precisely define match criteria for a traffic class. Enter the optional IGMP match criteria immediately after the destination IP address (DA) value in the command syntax; for example:

```
switch(config-class)# match igmp any any host-query
switch(config-class)# match igmp any any 3 7
```

igmp-type

Configures an IGMP packet type as match criteria in a class configuration. Some of the valid values for IGMP packet-type names are as follows:

```
dvmrpmtrace-requesttrace
host-querymtrace-replyv2-host-leave
host-reportpimv2-host-report
v3-host-report
```

To display a list of valid `igmp-type` entries when you enter `igmp` as the IP protocol type in a match/ignore statement, enter **?**.

Defining TCP and UDP match criteria

In a class configuration, you can enter match/ignore statements that more precisely define the TCP or UDP traffic to match in an IPv4 or IPv6 traffic class. For example, enter a port number as a match criterion that specifies one or more TCP source ports, destination ports, or both.

Context: Class configuration

Syntax:

```
no [seq—number] [match | ignore] {tcp | udp} source—address [operator tcp—src—port
| udp—src—port] destination—address [operator tcp—dest—port [established] [tcp—
flag tcp—flag ...] udp—dest—port] [ip—dscp codepoint] [precedence precedence—
value] [tos tos—value]
 [vlan vlan—id]
```

If you use TCP or UDP as the IP protocol type in a match/ignore statement, you can optionally configure TCP or UDP source and destination port numbers or ranges of numbers to more precisely define match criteria for a traffic class. Enter the optional TCP/UDP match criteria immediately after the source and destination address in the command syntax; for example:

```
switch(config-class)# match tcp host 10.20.10.17 eq 23 host 10.20.10.155  established
switch(config-class)# match tcp host 10.10.10.100 host 10.20.10.17  eq telnet
switch(config-class)# ignore udp 10.30.10.1/24 host 10.20.10.17  range 161 162
```

{operator | {*tcp-src-port* | *udp-src-port*}}

To specify a TCP or UDP source port number as a match criteria, enter a comparison operator from the following list with a TCP/UDP port number or well-known port name immediately after the source-address value in the command.

Comparison Operators:

- `eq` *`tcp/udp-port-number`*

  Equal To matches a packet with the same TCP or UDP source port number as *`tcp/udp-port-number`* .

- `gt` *`tcp/udp-port-number`*

  Greater Than matches any packet with a TCP or UDP source port number greater than *`tcp/udp-port-number`*.

- `lt` *`tcp/udp-port-number`*

Less Than matches any packet with a TCP or UDP source port number less than *tcp/udp-port-number* .

- neq *tcp/udp-port-number*

  Not Equal matches any packet with a TCP or UDP source port number that is not equal to *tcp/udp-port-number*.

- range *start-port-number end-port-number*

  Matches any packet with a TCP or UDP source port number in the range*start-port-number* to*end-port-number*.

TCP/UDP well-known source-port names and numbers

Enter a comparison operator with the source TCP or UDP port number used by the applications you want to match. Valid port numbers are from 0 to 255. You can also enter well-known TCP or UDP port names as an alternative to the corresponding port number; for example:

- TCP: bgp, dns, ftp, http, imap4, ldap, nntp, pop2, pop3, smtp, ssl, telnet
- UDP: bootpc, bootps, dns, ntp, radius, radius-old, rip, snmp, snmp-trap, tftp

To display a list of valid TCP/UDP source ports, enter **?** after you enter an operator.

operator tcp-dest-port established {[tcp-flag *tcp-flag* ...] | *udp-destport*}

To specify a TCP or UDP destination port number as a match criteria, enter acomparison operator with a TCP/UDP port number or well-known port name immediately after the destination-address value in the command.

NOTE: The optional established and *tcp-flag* values apply only to TCP destination-port criteria.

TCP/UDP well-known destination-port names and numbers

The same operators, port numbers, and well-known names are supported for TCP/UDP destination-port match criteria as for TCP/UDP source-port criteria. To display a list of valid TCP/UDP destination ports, enter **?** after you enter an operator.

established

(Optional) Applies only to TCP destination-port match criteria and matches only on the TCP Acknowledge (ACK) or Reset (RST) flags. The established keyword ignores the synchronizing packet associated with the establishment of a TCP connection in one direction on a port or VLAN, and matches all other IP traffic in the opposite direction.

For example, a Telnet connection requires TCP traffic to move both ways between a host and the target device. If you configure a match statement for inbound Telnet traffic, policy actions are normally applied to Telnet traffic in both directions because responses to outbound requests are also matched. However, if you enter the established option, inbound Telnet traffic arriving in response to outbound Telnet requests is matched, but inbound Telnet traffic trying to establish a connection is not matched.

tcp-flag *tcp-flag* ...

(Optional) Applies only to TCP bit settings in packets destined to a TCP destination port configured as match criteria (with the *tcp-dest-port* parameter) and can be one or more of the following values:

ack

Acknowledge matches TCP packets with the ACK flag.

fin

Finish matches TCP packets with the FIN flag.

rst

Reset matches TCP packets with the RST bit set.

syn

Synchronized matches TCP packets with the SYN flag.

Net-destination and Net-services for classifiers

Syntax

```
no match|ignore {alias-src <NAME-STR>} {alias-dst <NAME-STR>}
alias-srvc <NAME-STR>
```

Description

This command provides options to support net-destination and net-services for classifiers.

Parameters

alias-src

Specifies net-destination to control incoming packets.

alias-dst

Specifies destination IP address alias.

alias-srvc

Specifies service alias

Example net-service tcp-service tcp 100 for classifiers

```
  netdestination “src-ip”
   host 10.120.0.1
   host 10.91.1.1
   host 10.0.100.12
netdestination “destn-ip”
   host 16.90.51.12
   host 10.93.24.1
netservice “tcp-service” tcp 100
class ipv4 “abc”
   match alias-src “any” alias-dst “destn-ip” alias-srvc "tcp-service”
```

How IPv4 mask bit settings define a match (Example)

The following configuration exists:

- A match statement in a class configuration uses an IPv4 source-address/mask-length of 10.38.31.125/21. The mask-length of 21 results in an IPv4 mask of 0.0.7.255. In the second octet of the mask, 7 means that the rightmost three bits are on or 1.
- The second octet of the corresponding source address is 31, which means that the rightmost five bits are on or 1.

A match occurs when the second octet of the SA in a packet being classified has a value in the range of 24 (binary 00011000) to 31 (binary 00001111), as shown in the last row in the following table.

Table 30: *How IPv4 mask defines a match*

Location of octet	Bit position in the octet							
	128	64	32	16	8	4	2	1
SA in match statement	0	0	0	1	1	1	1	1
Mask for SA	0	0	0	0	0	1	1	1
Bits in the corresponding octetof a packet's SA that must exactly match	0	0	0	1	1	0/1	0/1	0/1

The shaded area indicates the bits in the packet that must exactly match the bits in the source IPv4 address in the match/ignore statement.

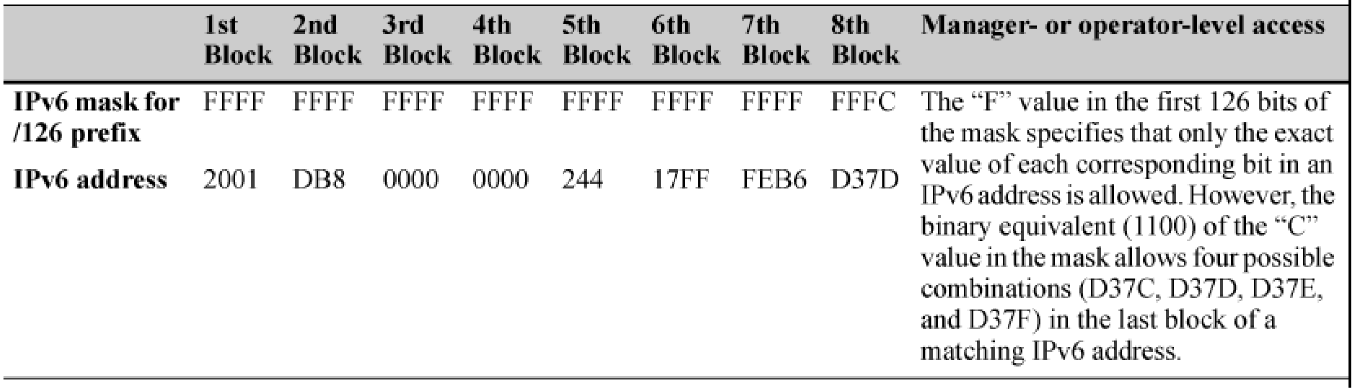
- If a mask bit is 1 (wildcard value), the corresponding bits in a source/destination address in an IPv4 packet header can be any value.
- If a mask bit is 0, the corresponding bits in a source/destination address must be the same value as in the IPv4 address in the match/ignore statement.

NOTE: Only one octet in an IPv4 address is used as a match criterion. The mask in a match/ ignore statement may apply a packet filter to all four octets of a source/destination address in IPv4 packet headers.

How IPv6 mask bit settings define a match

For an example in which an IPv6 prefix-length of 126 is used to select four IPv6 addresses in a match statement, see the following figure. The specified source IPv6 address is: `2001:DB8:0000:0000:244:17FF:FEB6:D37D`. The IPv6 prefix-length (/126) results in the IPv6 mask: `FFFF:FFFF:FFFF:FFFF:FFFF:FFFF:FFFF:FFFC`.

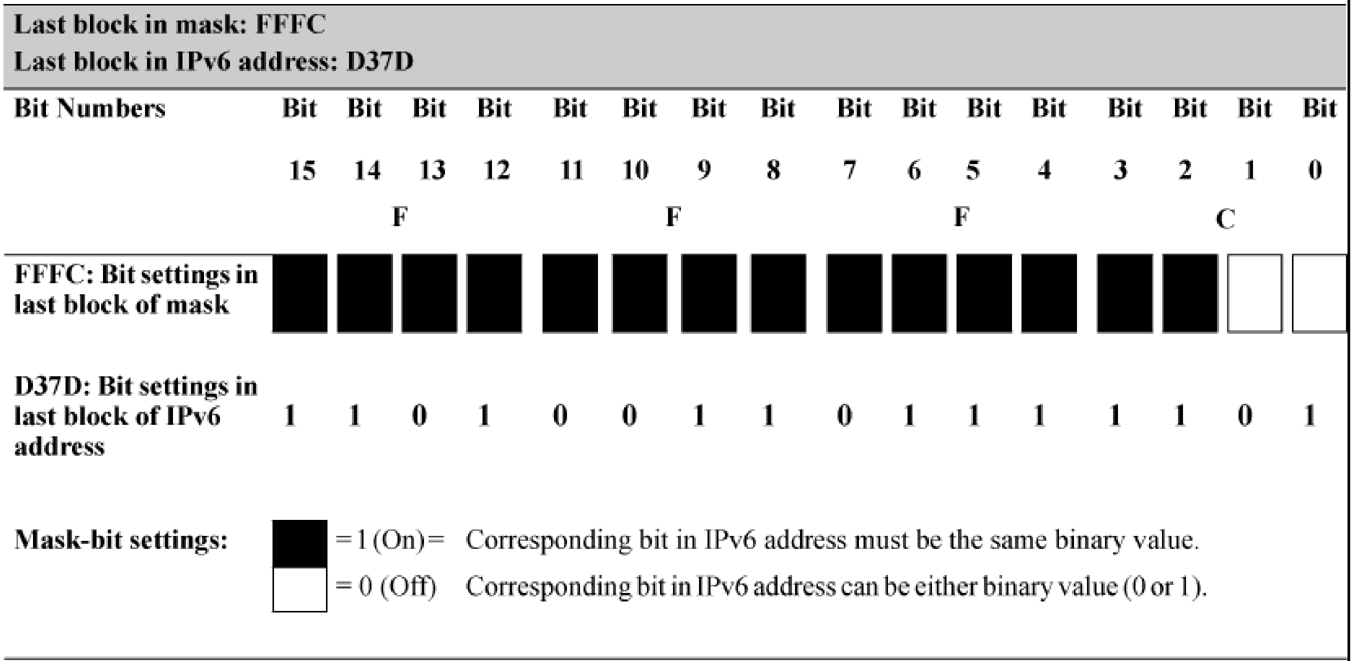
Figure 60: *Mask for matching four IPv6 devices*



	1st Block	2nd Block	3rd Block	4th Block	5th Block	6th Block	7th Block	8th Block	Manager- or operator-level access
IPv6 mask for /126 prefix	FFFF	FFFF	FFFF	FFFF	FFFF	FFFF	FFFF	FFFC	The "F" value in the first 126 bits of the mask specifies that only the exact value of each corresponding bit in an IPv6 address is allowed. However, the binary equivalent (1100) of the "C" value in the mask allows four possible combinations (D37C, D37D, D37E, and D37F) in the last block of a matching IPv6 address.
IPv6 address	2001	DB8	0000	0000	244	17FF	FEB6	D37D	

To see the on and off settings in the last block of the resulting IPv6 mask that determine the matching IPv6 addresses, see the preceding figure. In this mask, all bits except the last two are set to 1 (on) and must be the same in an IPv6 address. The binary equivalent of hexadecimal `C` is 1100, which allows the last two bits to differ.

Figure 61: *How a mask determines four authorized IPv6 manager addresses*



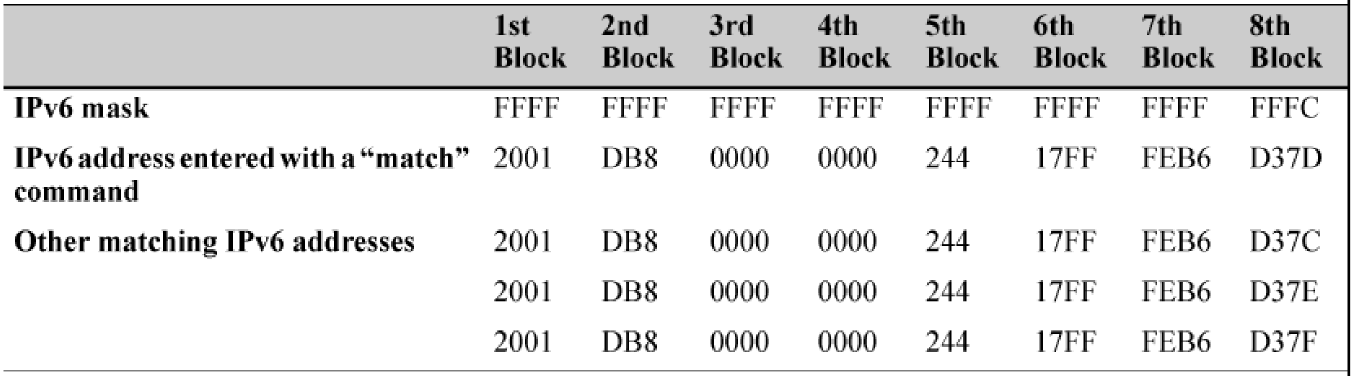
Last block in mask: FFFC
Last block in IPv6 address: D37D

Bit Numbers	Bit 15	Bit 14	Bit 13	Bit 12	Bit 11	Bit 10	Bit 9	Bit 8	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
		F				F				F				C		
FFFC: Bit settings in last block of mask	■	■	■	■	■	■	■	■	■	■	■	■	■	■	□	□
D37D: Bit settings in last block of IPv6 address	1	1	0	1	0	0	1	1	0	1	1	1	1	1	0	1

Mask-bit settings:
■ = 1 (On) = Corresponding bit in IPv6 address must be the same binary value.
□ = 0 (Off) Corresponding bit in IPv6 address can be either binary value (0 or 1).

To see how the binary equivalent (1100) of the C value in the last block of the resulting IPv6 mask supports four possible combinations (D37C, D37D, D37E, and D37F) in the last block of a matching IPv6 address, see the following figure. Therefore, the IPv6 mask that results from a /126 prefix-length matches inbound traffic from four IPv6-based devices.

Figure 62: *How hexadecimal C in an IPv6 mask matches four IPv6 addresses*



	1st Block	2nd Block	3rd Block	4th Block	5th Block	6th Block	7th Block	8th Block
IPv6 mask	FFFF	FFFF	FFFF	FFFF	FFFF	FFFF	FFFF	FFFC
IPv6 address entered with a "match" command	2001	DB8	0000	0000	244	17FF	FEB6	D37D
Other matching IPv6 addresses	2001	DB8	0000	0000	244	17FF	FEB6	D37C
	2001	DB8	0000	0000	244	17FF	FEB6	D37E
	2001	DB8	0000	0000	244	17FF	FEB6	D37F

For more detailed information on how to use CIDR notation to specify masks in match criteria, see the *ArubaOS-Switch Access Security Guide*.

Resequencing match/ignore statements

Use the `class` command with the `resequence` option to reconfigure the number at which the first match/ignore statement in the class starts, and reset the interval used to number other match/ignore statements.

Resequencing match/ignore statements is useful when you want to insert a new match/ignore statement between two numbered entries.

Context: Global configuration

Syntax:

```
class resequence [ipv4 | ipv6] name seq-number interval
```

`resequence`

Resets the sequence numbers for all match/ignore statements in the class.

`name`

Specifies the name of the class that contains the match/ignore statements that you want to resequence.

`seq-number`

Specifies the sequence number of the first match/ignore statement in the class. Default: 10.

`interval`

Specifies the interval between sequence numbers of match/ignore statements in the class to allow additional match/ignore statements to be inserted. Default: 10.

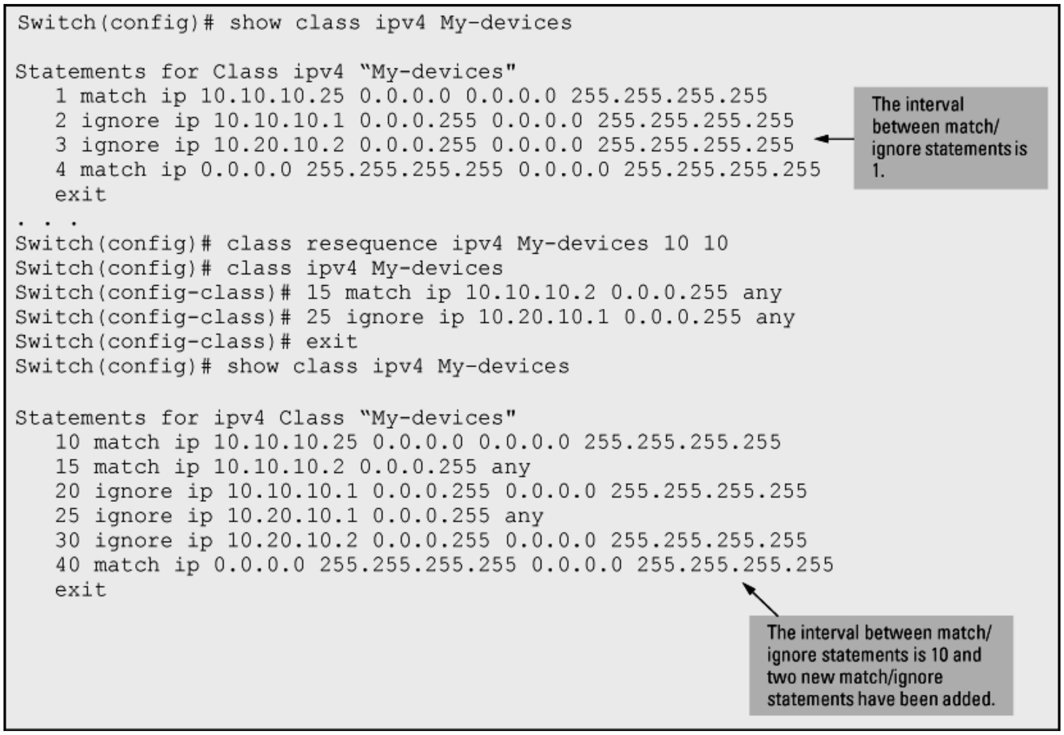
To view the current sequence numbering in a particular class, enter the following command:

```
show class [ipv4 | ipv6] classname
```

Resequencing a class configuration

The following example shows how to resequence a class configuration so that you can insert new match/ignore statements between sequentially numbered statements. The resequenced class contains two additional match/ignore statements and renumbers the criteria with an interval of 10.

Figure 63: *Resequencing a class configuration*



```
 Switch(config)# show class ipv4 My-devices

Statements for Class ipv4 “My-devices"
   1 match ip 10.10.10.25 0.0.0.0 0.0.0.0 255.255.255.255
   2 ignore ip 10.10.10.1 0.0.0.255 0.0.0.0 255.255.255.255
   3 ignore ip 10.20.10.2 0.0.0.255 0.0.0.0 255.255.255.255
   4 match ip 0.0.0.0 255.255.255.255 0.0.0.0 255.255.255.255
   exit
. . .
Switch(config)# class resequence ipv4 My-devices 10 10
Switch(config)# class ipv4 My-devices
Switch(config-class)# 15 match ip 10.10.10.2 0.0.0.255 any
Switch(config-class)# 25 ignore ip 10.20.10.1 0.0.0.255 any
Switch(config-class)# exit
Switch(config)# show class ipv4 My-devices

Statements for ipv4 Class “My-devices"
   10 match ip 10.10.10.25 0.0.0.0 0.0.0.0 255.255.255.255
   15 match ip 10.10.10.2 0.0.0.255 any
   20 ignore ip 10.10.10.1 0.0.0.255 0.0.0.0 255.255.255.255
   25 ignore ip 10.20.10.1 0.0.0.255 any
   30 ignore ip 10.20.10.2 0.0.0.255 0.0.0.0 255.255.255.255
   40 match ip 0.0.0.0 255.255.255.255 0.0.0.0 255.255.255.255
   exit
```

The interval between match/ignore statements is 1.

The interval between match/ignore statements is 10 and two new match/ignore statements have been added.

Creating a service policy

In the classifier-based configuration model, the service policy you create for one or more traffic classes is always relative to a software feature, such as QoS, port and VLAN mirroring, or PBR. The software feature must support class and policy configuration. Each feature supports different actions for managing selected packets.

NOTE: Policy Based Routing (PBR) is available on switches running v2 or higher modules.

For example, QoS policies support QoS-specific actions, such as rate limiting, 802.1p-priority, IP-precedence, and DSCP-codepoint assignment. Port and VLAN mirroring policies support mirror-destination assignment for matching packets. PBR policies support specifying the IP next-hop and IP default next-hop, tunnel ID, or null for matching packets.

Procedure

1. To create a service policy that performs feature-specific actions on selected packets, enter the `policy` *`feature-name`* command from the global configuration context.

   Context: Global configuration

   Syntax

   ```
   policy [qos | mirror | pbr] [policy-name]
   ```

   ```
   no policy [qos | mirror | pbr] [policy-name]
   ```

   Defines the name of a service policy and enters the policy configuration context, where *policy-name* is a text string (64 characters maximum).

   A traffic policy consists of one or more actions that are configured for each class of traffic. The configured actions are executed on packets that match a `match` statement in a class. No policy action is performed on packets that match an `ignore` statement. You can configure multiple classes in a policy.

2. To configure the actions that you want to execute on packets that match the `match` criteria in a specified class, enter one or more `class action` commands from the policy configuration context.

   Context: Policy configuration

   ```
   [seq-number] class [ipv4 | ipv6 classname action action-name] [action action-name ...]
   ```

   ```
   no [seq-number] class [ipv4 | ipv6 classname action action-name] [action action-name ...]
   ```

   Defines the actions to be applied on a pre-configured IPv4 or IPv6 traffic class when a packet matches the `match` criteria in the class.

   You can enter multiple class-action statements for the same class. The actions supported for a class command differ according to the feature-specific policy (for example, QoS or mirroring) configured with the `policy` command in Step 1.

   `seq-number`

   (Optional) Sequentially orders the class-action statements in a policy configuration. Actions are executed on matching packets in numerical order.

   Default: Class-action statements are numbered in increments of 10, starting at 10.

   `class ipv4|ipv6` *`classname`*

Defines the preconfigured class on which the actions in a class-action statement are executed, and specifies whether the class consists of IPv4 or IPv6 traffic. The class name is a text string (64 characters maximum).

NOTE: You can configure multiple class-action statements to include different classes in a policy. The execution of actions is performed in the order in which the class-actions are numerically listed.

```
action action-name [action action-name ...]
```

The `action` keyword configures the action specified by the `action-name` parameter. The action is executed on any packet that matches the `match` criteria in the class. The action is not executed on packets that match `ignore` criteria. You can configure more than one action for a class. The complete `no` form of the `class action` command or the `no` *`seq-number`* command removes an action from the policy configuration.

Be sure to enter a class and its associated actions in the precise order in which you want packets to be checked and handled by `class action` commands.

3. (Optional) To configure a default class, enter the `default-class` command and specify one or more actions to be executed on packets that are not matched and not ignored.

   Context: Policy configuration

   ```
   default-class action action-name [action action-name ...]
   ```

   ```
   no default-class action action-name [action action-name ...]
   ```

   Configures a default class to be used to execute one or more actions on packets that are not matched nor ignored in any of the class configurations in a policy. The `default-class action` command supports only the feature-specific commands supported in the `class action`command.

   The default class manages packets that do not match the `match` or `ignore` criteria in all classes in a policy, and otherwise would have no actions performed on them.

   The default class differs from other classes because it contains no match/ignore statements and uses implicit `match ipv4 any any` and `match ipv6 any any` statements to manage all unmatched packets. If you do not configure a default class, unmatched and ignored packets are transmitted without an action performed on them.

4. Enter the `exit` command to exit the policy configuration context.

To display a policy configuration, enter the `show policy` *`policy-name feature-name`* command where *feature-name* is a software feature (such as `qos`, `mirror`, or `pbr`) that supports classifier-based configuration.

To edit a policy configuration, re-enter the policy context (`policy` command) and modify class-action statements.

To resequence the order in which class-action statements are listed, enter the `resequence` command.

In the following QoS policy configuration, matching HTTP packets are rate limited to 10000 kbps. All unmatched packets are managed by the default class, which assigns a slightly higher 802.1p priority (4) and a new DSCP codepoint (5).

```
switch(config)# class ipv4 http
switch(config-class)# match tcp any any eq 80
switch(config-class)# match tcp any any eq 8080
switch(config-class)# exit
switch(config)# policy qos RateLimitPrioritizeSuspectTraffic
switch(policy-qos)# class ipv4 http action rate-limit kbps 10000
```

```
switch(policy-qos)# default-class action priority 4 action dscp 5
switch(policy-qos)# exit
```

A policy configuration requires a feature-specific `policy` command to identify the software feature used to manage one or more traffic classes:

- To configure a QoS policy, use the `policyqos` command as described in the "Quality of Service" chapter in the *Advanced Traffic Management Guide*.
- To configure a mirroring policy, use the `policy mirror` command as described in the *Management and Configuration Guide* for your switch.

Creating a PBR policy

PBR enables you to manipulate a packet's path based on attributes of the packet. Traffic with the same destination can be routed over different paths, so that different types of traffic, such as VOIP or traffic with special security requirements, can be better managed.

NOTE: Policy Based Routing (PBR) is available on switches running v2 or higher modules.

The supported actions for PBR are:

- Setting the next hop for routing the packet (`[ipv4 | ipv6] next-hop [ip-addr]`).
- Setting the next hop for routing the packet if there is no explicit route for this destination (`[ipv4 | ipv6] ip default-next-hop [ip-addr]`).
- Setting the outbound tunnel interface for the packet (`interface tunnel [tunnel-ID]`). See the *IPv6 Configuration Guide* for your switch.
- Setting `interface null`, which specifies that the packets are dropped if no other actions have occurred.

Operating notes for PBR

- Multiple actions can be configured for a class, up to 8 actions per class.
- If you configure an action of interface null, no more actions for that class may be configured.
- Only one of the 8 possible actions can be active at one time.
- The precedence of actions is indicated by the order in which they are added to the policy.
- Actions can only be added to a class, and they are added to the end of the action list for the class.
- To remove actions from a class, the entire class must be removed from the policy.
- When an action becomes inactive, for example, if the configured address becomes unreachable (for `next-hop` and `default-next-hop)` or the interface goes down (for a tunnel), the policy is configured with the next action for that class, if possible. If that action is not active, the next action is tried, and so on, until an interface null or the end of the list of configured actions is encountered. If the end of the list is reached, the policy action for that class behaves as if no PBR policy is applied.
- The maximum combined number of unique IP next-hops and default-next-hops supported is 16.

TCP and UDP traffic routing

The following example shows TCP and UDP traffic routed on different network paths. First, the traffic classes are created, then the PBR policy is created, and lastly the PBR policy is applied to an interface.

```
switch(config)# class ipv4 TCP
switch(config-class)# match tcp 10.0.8.1/24 15.29.16.104/24 eq 80
switch(config-class)# match tcp 10.0.8.1/24 15.29.16.104/24 eq 22
switch(config-class)# match tcp 10.0.8.1/24 15.29.16.104/24 eq 23
switch(config-class)# exit
switch(config)# class ipv4 UDP
switch(config-class)# match udp 10.0.8.1/24 15.29.16.104/24 eq 80
switch(config-class)# match udp 10.0.8.1/24 15.29.16.104/24 eq 22
switch(config-class)# match upd 10.0.8.1/24 15.29.16.104/24 eq 23
switch(config-class)# exit
switch(config)# class ipv6 TCP
switch(config-class)# match tcp 2001::1/64 3001::1/64 eq 80
switch(config-class)# match tcp 2001::1/64 3001::1/64 eq 22
switch(config-class)# match tcp 2001::1/64 3001::1/64 eq 23
switch(config-class)# exit
switch(config)# class ipv6 UDP
switch(config-class)# match udp 2001::1/64 3001::1/64 eq 80
switch(config-class)# match udp 2001::1/64 3001::1/64 eq 22
switch(config-class)# match udp 2001::1/64 3001::1/64 eq 23
switch(config-class)# exit
switch(config)# policy pbr TCP_UDP
switch(policy-pbr)# class ipv4 TCP
switch(policy-pbr-class)# action ip next-hop 20.0.0.1
switch(policy-pbr-class)# action interface null
switch(policy-pbr-class)# exit
switch(policy-pbr)# class ipv4 UDP
switch(policy-pbr-class)# action ip default-next-hop 30.0.0.1
switch(policy-pbr-class)# action interface tunnel 3
switch(policy-pbr-class)# exit
switch(policy-pbr)# class ipv6 TCP
switch(policy-pbr-class)# action ip next-hop 20.0.0.1
switch(policy-pbr-class)# exit
switch(policy-pbr)# class ipv6 UDP
switch(policy-pbr-class)# action ip next-hop 30.0.0.1
switch(policy-pbr-class)# exit
switch(policy-pbr)# exit
switch(config)# vlan 10
switch(vlan-10)# service-policy TCP_UDP in
```

To enable debug logging for PBR, enter the debug **`ip pbr`** command. A message is logged when a PBR policy is applied, when the action in a class becomes inactive, and when an action in a class becomes active. See the *Management and Configuration Guide* for your switch.

NOTE: Policy Based Routing (PBR) is available on switches running v2 or higher modules.

Troubleshooting PBR

Cause

Use the `show statistics policy` command to display information about which PBR action for an applied policy is active. Hit counts for each entry in the class and policy with the active action are displayed.

```
switch(vlan-111)# show statistics policy TCP_UDP vlan 111 in
HitCounts for Policy TCP_UDP
Total
100 class ipv4 TCP action
( 5 ) 10 match tcp 10.0.8.1 0.0.0.255 15.29.16.104 0.0.0.255 eq 80
( 2 ) 20 match tcp 10.0.8.1 0.0.0.255 15.29.16.104 0.0.0.255 eq 22
( 2 ) 30 match tcp 10.0.8.1 0.0.0.255 15.29.16.104 0.0.0.255 eq 23
110 class ipv4 voice action
( 4 ) 10 match tcp 10.0.8.1 0.0.0.255 15.29.16.104 0.0.0.255 eq 80
```

To enable debug logging for PBR, enter the `debug ip pbr` command. A message will be logged when a PBR policy is applied, when the action in a class becomes inactive, and when an action in a class becomes active. See the *Management and Configuration Guide* for your switch.

NOTE: Policy Based Routing (PBR) is available on the 5400 series switch which all have v2 or higher modules. Any v1 modules will prevent PBR from functioning.

Modifying classes in a policy

You can modify the classes and class-action statements in a policy configuration without removing them from the policy:

- To modify the match/ignore statements in a class, enter the class-configuration context with the command, and make the necessary changes by removing or replacing existing statements. To display a class configuration, enter the following command as shown in **Figure 63: Resequencing a class configuration** on page 379:

  ```
  show class [ipv4 | ipv6] classname
  ```

  When you exit class configuration context, the changes are automatically saved and applied to existing policy configurations on the switch that use the class if the policies have not been applied to an interface. If a policy has already been applied to an interface, the editing changes are not accepted and an error message is displayed.

- To modify the class-action statements in a policy, enter the policy-configuration context with the `policy` *`feature-name policy-name`* command. To display a policy configuration, enter the following command as shown in **Resequencing a policy configuration** on page 385:

  ```
  show policy feature-name policy-name
  ```

  Then do one of the following:

  - You can enter a new class-action statement. If you do not enter a sequence number, the new class-action statement is inserted at the end of the policy configuration.
  - To remove a class-action statement from a policy configuration, enter the `no` *`sequence-number`* command or the complete form of the `no class ... action` command.
  - To resequence the order in which class-action statements are listed, enter the `resequence`command.
  - To replace an existing class-action statement, enter the `no` *`sequence-number`* command to delete the entry, and re-enter the following complete command:

    ```
    class [ipv4 | ipv6] classname action action-name or default-class action action-name
    ```

When exiting the policy-configuration context, the changes are automatically saved and applied to the policy configuration if the policy has not been applied to an interface. If the policy has already been applied to an interface, the editing changes are not accepted and an error message is displayed.

Resequencing classes in a policy

You can use the `policy` command with the `resequence` option to reconfigure the number at which the first class-action statement starts, and reset the interval used to number other class-actions.

Resequencing class-actions is useful when you want to insert a new class-action between two numbered entries.

Context: Global configuration

Syntax:

```
policy resequence name seq-number interval
```

resequence

Resets the sequence numbers for all class-action statements in the policy.

name

Specifies the name of the policy that contains the class-action statements that you want to resequence.

seq-number

Specifies the sequence number of the first class-action-statement in the policy. Default: 10.

interval

Specifies the interval between sequence numbers of class-action statements in the policy to allow additional statements to be inserted. Default: 10.

NOTE:

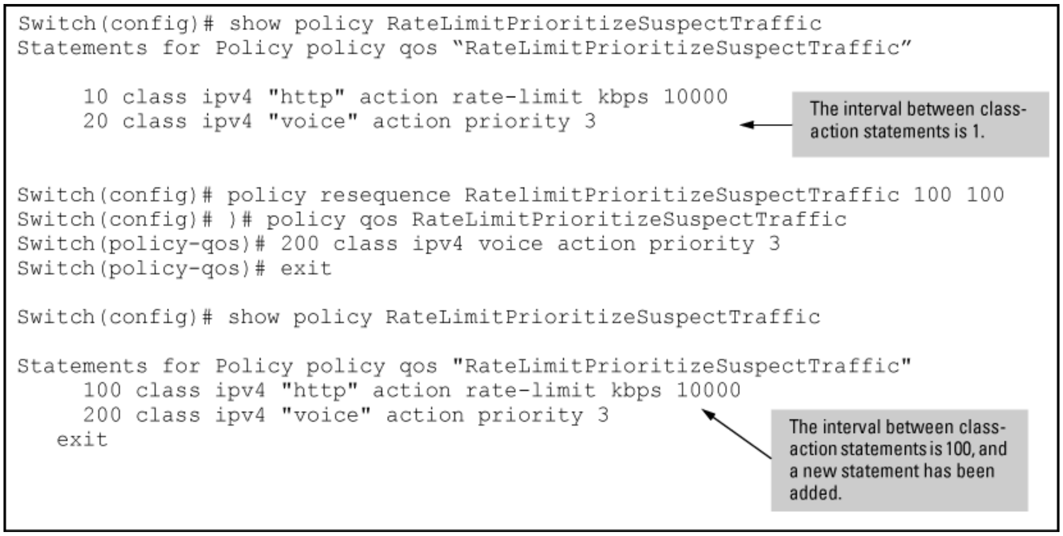
When resequencing class-action statements in a policy, the default class-action-statement always remains as the last class-action statement.

To view the current class-action numbering in a policy, enter the following command:

```
show policy feature-name policy-name
```

Resequencing a policy configuration

The following example shows how to resequence a policy configuration after Viewing its contents. The resequenced policy allows you to add a new class-action statement between entries 100 and 200.



```
Switch(config)# show policy RateLimitPrioritizeSuspectTraffic
Statements for Policy policy qos “RateLimitPrioritizeSuspectTraffic”

     10 class ipv4 "http" action rate-limit kbps 10000
     20 class ipv4 "voice" action priority 3


Switch(config)# policy resequence RatelimitPrioritizeSuspectTraffic 100 100
Switch(config)# )# policy qos RateLimitPrioritizeSuspectTraffic
Switch(policy-qos)# 200 class ipv4 voice action priority 3
Switch(policy-qos)# exit

Switch(config)# show policy RateLimitPrioritizeSuspectTraffic

Statements for Policy policy qos "RateLimitPrioritizeSuspectTraffic"
     100 class ipv4 "http" action rate-limit kbps 10000
     200 class ipv4 "voice" action priority 3
   exit
```

The interval between class-action statements is 1.

The interval between class-action statements is 100, and a new statement has been added.

Applying a service policy to an interface

To apply feature-specific service policies to inbound port or VLAN interfaces, use the `interface service-policy in` or `vlan service-policy in` command.

The following service-policy restrictions apply to all software features:

- A service policy is supported only on inbound traffic.
- Only one feature-specific policy (for example, QoS or mirroring) is supported on a port or VLAN interface.
- PBR is only supported within a `vlan [vlan-id] service-policy [policy-name] in` command or within a VLAN context. PBR is not applicable a port - specific interface.
- If you apply a policy to a port or VLAN interface on which a policy of the same type (for example, QoS) is already configured, an error message is displayed. The new policy does not overwrite the existing one.

  Before you can apply a new policy, you must first remove the existing policy with the `no interface service-policy in` or `no vlan service-policy in` command.

Because only one policy of each type is supported on a port or VLAN interface, ensure that the policy you want to apply contains all the required classes and actions for your configuration.

NOTE: If ICMP rate limiting is already configured on a port, a service policy cannot be applied to the port until you disable the ICMP rate limiting configuration.

To apply a service policy to the port, maintain ICMP rate limiting by configuring a QoS policy in which you add the necessary `match` statements for ICMP packets to a class configuration and configure a `rate-limit` action for the class in the policy configuration.

For information on globally configured ICMP, see the *ArubaOS-Switch Management Configuration Guide* for your switch.

To apply a service policy on a port or VLAN interface, enter one of the following commands from the global configuration context.

Context: Global configuration

Syntax:

```
interface port-list service-policy policy-name in | out
```

Configures the specified ports with a policy that is applied to inbound traffic on each interface. Separate individual port numbers in a series with a comma; for example, `a1, b4, d3`. Enter a range of ports by using a dash; for example, `a1-a5`.

The policy name you enter must be the same as the policy name you configured with the `policy` command.

Context: Global configuration

Syntax:

```
vlan vlan-id service-policy policy-name in | out
```

Configures a policy on the specified VLAN that is applied to inbound traffic on the VLAN interface. Valid VLAN ID numbers range from `1` to `4094`.

The policy name you enter must be the same as the policy name you configured with the `policy` command.

Applying a QoS policy to a port range and a VLAN interface

The following example shows how to apply a QoS policy to a port range and a VLAN interface:

```
switch(config)# interface a4 service-policy RateLimitPrioritizeSuspectTraffic in
switch(config)# vlan 10 service-policy RateLimitPrioritizeSuspectTraffic in
```

Checking resource usage

Syntax:

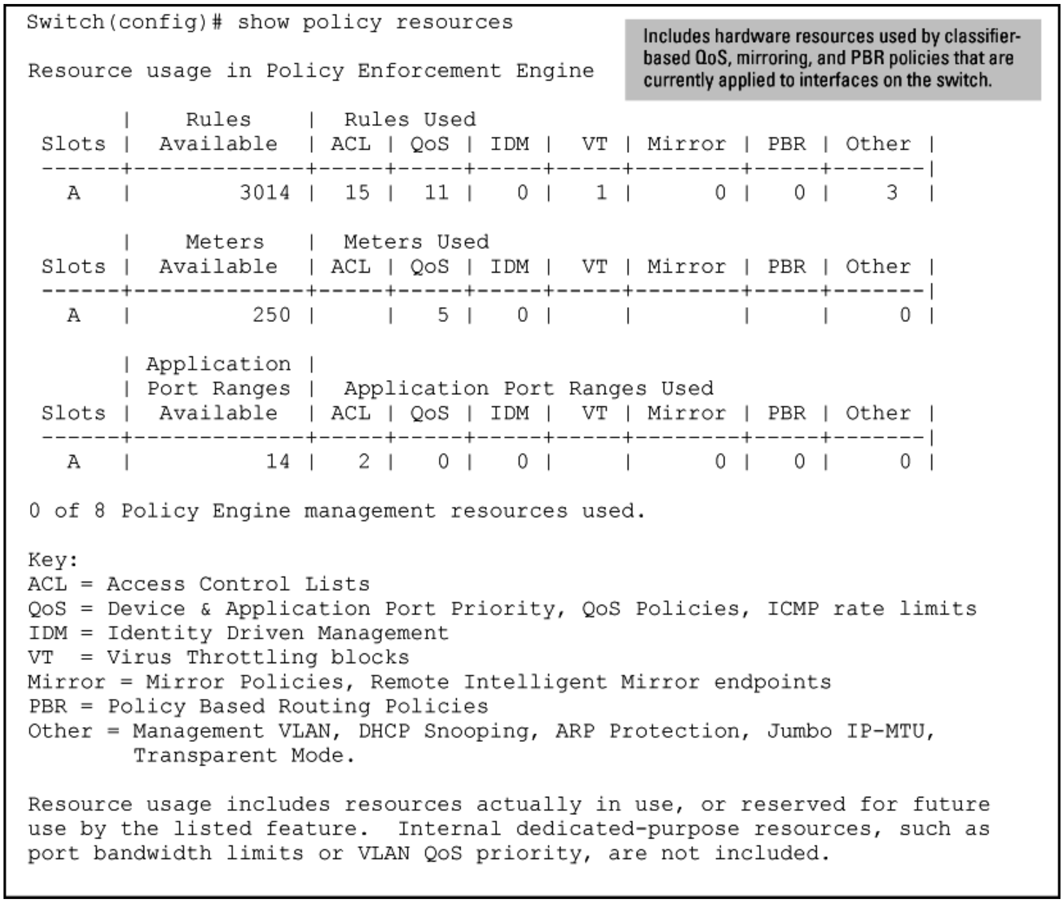
```
show policy resources
```

After applying service policies to an interface, use the `show policy resources` command to verify the amount of additional resources used and the amount of resources that are still available on the switch. Classifier-based service policies (such as QoS or mirroring) share the same hardware resources with other software features, such as ACLs, virus throttling, management VLAN, globally configured QoS policies, MAC-based mirroring policies, and so on.

Use the displayed information to decide whether to re-prioritize current resource usage by reconfiguring or disabling software features to free the resources reserved for less important features. For a detailed explanation of the information displayed with the `show policy resources` command, see the *ArubaOS-Switch Management and Configuration Guide*.

Viewing policy resources

The `show policy resources` command output displays the number of hardware resources (rules, meters, and application port ranges) used by classifier-based QoS and mirroring policies that are currently applied to interfaces on the switch and other software features.



```
Switch(config)# show policy resources

Resource usage in Policy Enforcement Engine

       |    Rules     |  Rules Used
 Slots |  Available   | ACL | QoS | IDM |  VT | Mirror | PBR | Other |
 ------+--------------+-----+-----+-----+-----+--------+-----+-------|
   A   |         3014 |  15 |  11 |   0 |   1 |      0 |   0 |    3  |

       |    Meters    |  Meters Used
 Slots |  Available   | ACL | QoS | IDM |  VT | Mirror | PBR | Other |
 ------+--------------+-----+-----+-----+-----+--------+-----+-------|
   A   |          250 |     |   5 |   0 |     |        |     |     0 |

       | Application  |
       | Port Ranges  |  Application Port Ranges Used
 Slots |  Available   | ACL | QoS | IDM |  VT | Mirror | PBR | Other |
 ------+--------------+-----+-----+-----+-----+--------+-----+-------|
   A   |           14 |   2 |   0 |   0 |     |      0 |   0 |     0 |

0 of 8 Policy Engine management resources used.

Key:
ACL = Access Control Lists
QoS = Device & Application Port Priority, QoS Policies, ICMP rate limits
IDM = Identity Driven Management
VT  = Virus Throttling blocks
Mirror = Mirror Policies, Remote Intelligent Mirror endpoints
PBR = Policy Based Routing Policies
Other = Management VLAN, DHCP Snooping, ARP Protection, Jumbo IP-MTU,
        Transparent Mode.

Resource usage includes resources actually in use, or reserved for future
use by the listed feature.  Internal dedicated-purpose resources, such as
port bandwidth limits or VLAN QoS priority, are not included.
```

Includes hardware resources used by classifier-based QoS, mirroring, and PBR policies that are currently applied to interfaces on the switch.

Viewing statistics for a policy

Only the active redirects (matches, ignores, etc.) are displayed when executing the **`show statistics`** command.

Statistical output for a policy with active redirects

```
switch(vlan-111)# show statistics policy TCP_UDP vlan 111 in
HitCounts for Policy TCP_UDP
Total
100 class ipv4 TCP action
( 0 ) 10 match tcp 10.0.8.1 0.0.0.255 15.29.16.104 0.0.0.255 eq 80
( 0 ) 20 match tcp 10.0.8.1 0.0.0.255 15.29.16.104 0.0.0.255 eq 22
( 0 ) 30 match tcp 10.0.8.1 0.0.0.255 15.29.16.104 0.0.0.255 eq 23
110 class ipv4 voice action
( 0 ) 10 match tcp 10.0.8.1 0.0.0.255 15.29.16.104 0.0.0.255 eq 80
```

Viewing statistics for a policy

Only the active redirects (matches, ignores, etc.) are displayed when executing the `show statistics` command.

Statistical output for a policy with active redirects

```
switch(vlan-111)# show statistics policy TCP_UDP vlan 111 in

HitCounts for Policy TCP_UDP

  Total

 100 class ipv4 TCP action
( 0 )     10 match tcp 10.0.8.1 0.0.0.255 15.29.16.104 0.0.0.255 eq 80
( 0 )     20 match tcp 10.0.8.1 0.0.0.255 15.29.16.104 0.0.0.255 eq 22
( 0 )     30 match tcp 10.0.8.1 0.0.0.255 15.29.16.104 0.0.0.255 eq 23

  110 class ipv4 voice action
( 0 )     10 match tcp 10.0.8.1 0.0.0.255 15.29.16.104 0.0.0.255 eq 80
```

About Classifier-based configuration

Traffic classes and software releases

The Classifier feature introduces:

- A finer granularity than globally configured features for placing network traffic (IPv4 or IPv6) into classes that can be used in cross-feature software configurations
- Additional policy actions, such as rate limiting and IP precedence marking, to manage selected traffic
- The configuration of service policies for classified traffic with the following software features:
  - Quality of Service (QoS)
  - Traffic mirroring
  - Policy Based Routing (PBR)
- The application of service policies to specific inbound traffic flows on individual port and VLAN interfaces (rather than only on switch-wide or port-wide traffic).

Using CIDR notation for IPv4/IPv6 addresses

You can use CIDR (Classless Inter-Domain Routing) notation to enter an IPv4 mask-length or an IPv6 prefix-length with a source and destination address that are used as match criteria in a match/ignore statement. The switch interprets the IP address with CIDR notation to compute the range of corresponding IP source or destination addresses in packet headers that are considered to be a match for the traffic class.

When the switch uses a match/ignore statement to compare an IP address and corresponding mask/prefix length to the IP source/destination address carried in a packet, the IPv4 mask-bit settings and IPv6 prefix-bit settings select packets in different ways.

- An IPv4 mask length creates a mask in which:
  - A mask-bit setting set to 0 (off) requires the corresponding bit in a packet's IPv4 source/destination address to be the same binary value as the mask-bit in the matching IPv4 source/destination address.
  - A mask-bit setting set to 1 (on) is used as a wildcard and allows the corresponding bit in a packet's IPv4 source/destination address to be either binary value (0 or 1).

Table 31: *How CIDR notation is used with IPv4 SA/DA match criteria*

IPv4 Source/ Destination address used with CIDR notation in a Match/ Ignore statement	Resulting mask	Range of IPv4 addresses selected by the match criteria
10.38.240.125/15	0.1.255.255	The leftmost 15 bits must match; the remaining bits are wildcards.
10.38.240.125/20	0.0.15.255	The leftmost 20 bits must match; the remaining bits are wildcards.
10.38.240.125/21	0.0.7.255	The leftmost 21 bits must match; the remaining bits are wildcards.
10.38.240.125/24	0.0.0.255	The leftmost 24 bits must match; the remaining bits are wildcards.
18.38.240.125/32	0.0.0.0	All bits must match.

- An IPv6 prefix-length creates a mask in which:
  - A mask-bit setting set to 1 (on) requires the corresponding bit in a packet's IPv6 source/destination address to be the same binary value as the mask-bit in the matching IPv6 source/destination address.
  - A mask-bit setting set to 0 (off) is used as a wildcard and allows the corresponding bit in a packet's IPv6 source/destination address to be either binary value (0 or 1).

Table 32: *How CIDR notation is used with IPv6 SA/DA match criteria*

IPv6 source/ destination address used with CIDR notation in a Match/Ignore statement	Resulting mask	Range of IPv6 addresses selected by the match criteria
2001:db8:0:7::5/64	FFFF:FFFF:FFFF:FFFF::	The leftmost 64 bits must match; the remaining bits are wildcards.
2001:db8:0:7::5/72	FFFF:FFFF:FFFF:FFFF:FF00::	The leftmost 72 bits must match; the remaining bits are wildcards.
2001:db8::244:17ff:feb6:d37d/126	FFFF:FFFF:FFFF:FFFF:FFFF:FFFF:FFFF:FFFC	The first 126 bits mst match; the C value in the mask allows four possible combinations (D37C, D37D, D37E, and D37F) in the last block of a matching IPv6 address.
2001:db8:0:7:af:e2:c1:5/128	FFFF:FFFF:FFFF:FFFF:FFFF:FFFF:FFFF:FFFF	All bits must match.

NOTE: Although IPv4 and IPv6 masks are applied in opposite directions:

- An IPv4 mask-length is applied from right to left, starting from the rightmost bits.
- An IPv6 prefix-length is applied from left to right, starting from the leftmost bits.

The behavior of IPv4 and IPv6 masks as match criteria and wildcards is the same.

Where to go from here

Classifier-based service policies are designed to work with your existing globally-configured software settings. While existing software features allow you to globally manage all network traffic on a switch or port, classifier-based service policies allow you to zoom in on subsets of network traffic to further manage it on a per-port or per-VLAN basis.

You can use the match criteria described in this chapter across software features to configure classes of traffic for use in feature-specific service policies.

After you decide on the IPv4 and IPv6 network traffic you want to manage, see the *ArubaOS-Switch Management and Configuration Guide* for more information about how to configure and use classifier-based quality-of-service and mirroring policies.

Traffic class-based configuration model

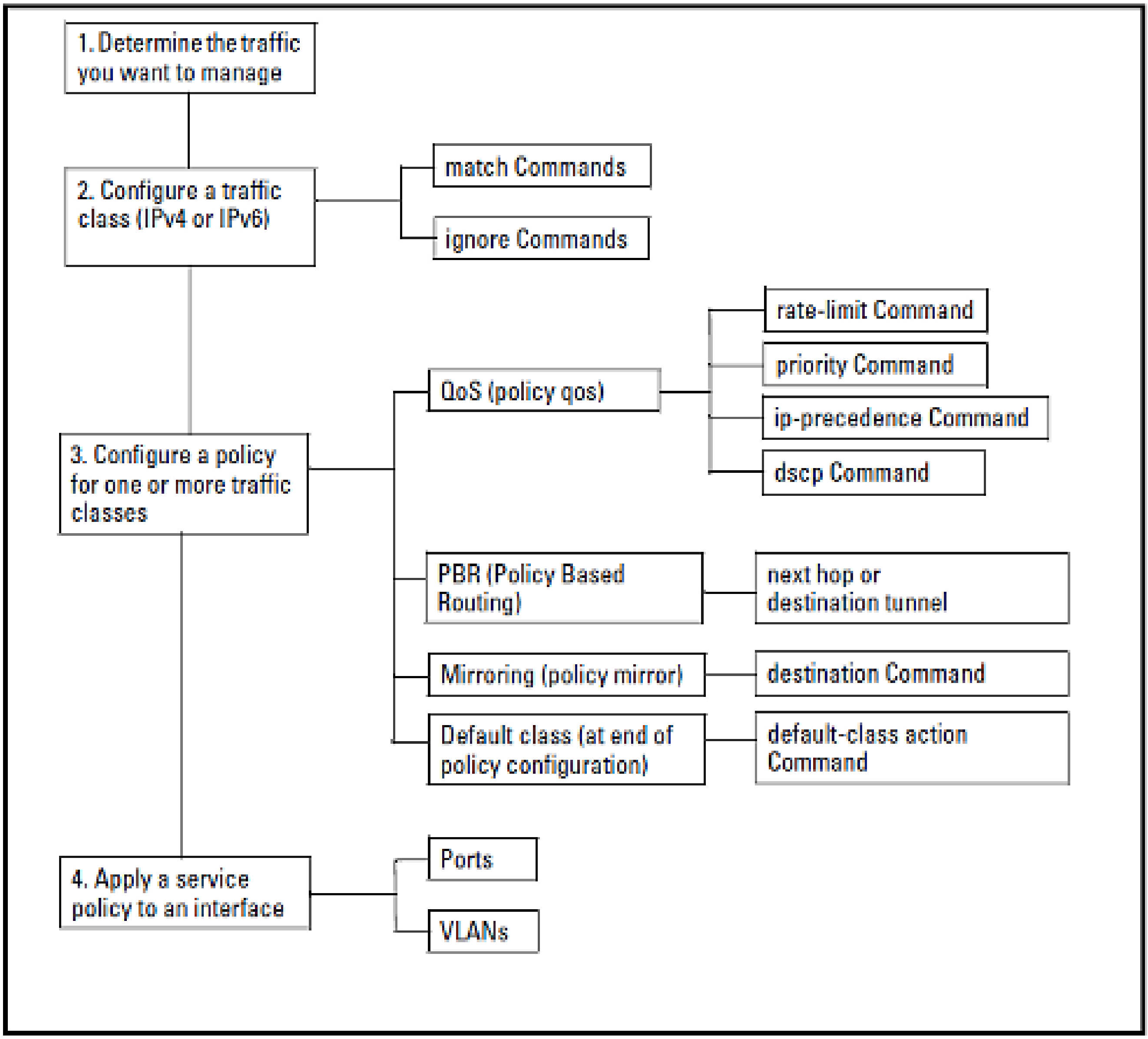
Traffic class-based software configuration consists of the following general steps:

Procedure

1. Determine the inbound traffic you want to manage and how you want to manage it. For example, you may want to rate limit certain traffic, prioritize it, mirror it, and so on.
2. Classify the traffic that you want to manage by configuring a class, using `match` and `ignore` commands. A traffic class is configured separately from service policies and can be used in various policies.
3. Configure a service policy for one or more address classes, including an optional, default class. A policy consists of configuration commands executed on specified traffic classes for one of the following software features:
   - Quality of Service (`policy qos` command)
   - Port and VLAN mirroring (`policy mirror` command)
   - Policy Based Routing (`policy pbr` command)
4. Assign the policy to an inbound port or VLAN interface using the `interface service-policy in` or `vlan service-policy in` command.

The following figure shows an overview of traffic class-based software configuration:

Figure 64: *Traffic class-based configuration model*


1. Determine the traffic you want to manage
2. Configure a traffic class (IPv4 or IPv6)
match Commands
ignore Commands
rate-limit Command
priority Command
QoS (policy qos)
ip-precedence Command
3. Configure a policy for one or more traffic classes
dscp Command
PBR (Policy Based Routing)
next hop or destination tunnel
Mirroring (policy mirror)
destination Command
Default class (at end of policy configuration)
default-class action Command
Ports
4. Apply a service policy to an interface
VLANs

Creating a traffic class

In the traffic class-based configuration model, you use match criteria to create a class of IPv4 or IPv6 traffic and select the packets you want to manage. In a traffic class configuration, match criteria consist of `match` and `ignore` commands. These commands determine the packets that belong to a class. (Match/ignore criteria are modelled on the permit/deny criteria used in ACLs.)

The traffic classes you configure can be used later in the service policies you create for different software features, such as QoS and port mirroring. The match criteria used in match/ignore statements are the same across software features.

Using match criteria

To identify the packets that belong to a traffic class for further processing by policy actions, use `match` and `ignore` commands in a class configuration:

`match` commands

Define the values that header fields must contain for a packet to belong to the class and be managed by policy actions.

`ignore` commands

Define the values which, if contained in header fields, exclude a packet from the policy actions configured for the class. An ignored packet is transmitted without having a policy action performed on it.

The switch compares match/ignore statements to the values in packet fields. It compares the specified criteria in the sequential order in which the statements are entered in the class, until a match is found. Be sure to enter match/ignore statements in the precise order in which you want their criteria to be used to check packets.

- As soon as a field in a packet header matches the criteria in a `match` statement, the sequential comparison of match criteria in the class stops, and the policy actions configured for the class are executed on the packet.
- If a packet matches the criteria in an `ignore` statement, the sequential comparison of match criteria in the class stops, and no policy action is performed on the packet.

If a packet does not match the criteria in any match/ignore statement in a traffic class configuration, one of the following actions is taken:

- The packet is transmitted without a policy action performed on it.
- If a default class is configured in the policy, the actions specified in the `default-class` command are performed on packets that do not match the criteria in preceding classes in the policy.

The following match criteria are supported in match/ignore statements for inbound IPv4/IPv6 traffic:

- IP source address (IPv4 and IPv6)
- IP destination address (IPv4 and IPv6)
- Layer 2 802.1Q VLAN ID
- Layer 3 IP protocol
- Layer 3 IP precedence bits
- Layer 3 DSCP bits
- Layer 4 TCP/UDP application port (including TCP flags)
- VLAN ID

Control Plane Policing

CoPP prioritizes traffic handled by the CPU and also serves to protect the device from Denial-of-Service (DoS) attacks.

Aruba OS now supports policing the different CPU traffic classes based on a user configurable set of rate limits. The feature is disabled by default and can be enabled using a set of CoPP CLI commands, which can do either of the following:

- Apply a default rate-limit profile to all traffic classes via a single command
- Apply a rate limit on a per-traffic class

copp traffic-class

Syntax

```
copp traffic-class {<traffic-class-type> limit {default | <VALUE>}}

no copp traffic-class {<traffic-class-type> limit {default | <VALUE>}}
```

Description

Enables control plane policing for the specified traffic class. The `no` form of this command disables control plane policing.

Command context

```
config
```

Parameters

<traffic-class-type>

Specifies the traffic class type.

all

Specifies control plane policing for all traffic classes.

bgp

Specifies control plane policing for BGP control packets.

broadcast

Specifies control plane policing for broadcast packets.

exception-notification

Specifies control plane policing for MAC security, IP security flow and host.

icmp-redirect

Specifies control plane policing for ICMP redirect packets.

ip-gateway-control

Specifies control plane policing for VRRP traffic.

layer2-control-others

Specifies control plane policing for LLDP, LACP, and 802.1x.

loop-protect

Specifies control plane policing for loop-protect control packets.

mac-notification

Specifies control plane policing for MAC learn and move notifications.

multicast-route-control

Specifies control plane policing for IGMP, MLD, and PIM control packets.

multicast-sw-forward

Specifies control plane policing for IGMP data driven and multicast routing unknown flows.

mvrp

Specifies control plane policing for MVRP control packets.

ospf

Specifies control plane policing for OSPF control packets.

pvst

Specifies control plane policing for PVST packets.

rip

Specifies control plane policing for RIP control packets.

sampling

Specifies control plane policing for SFLOW.

smart-links

Specifies control plane policing for smart link control packets.

station-arp

Specifies control plane for the traffic class that applies to all ARP and ND packets reaching the switch CPU.

station-icmp

Specifies control plane policing for the traffic class that applies to all ICMP and ICMPv6 packets reaching the switch CPU.

station-ip

Specifies control plane policing for the traffic-class that applies to all ip and ipv6 (for example, telnet, snmp, http, ftp and other services enabled on the switch) traffic reaching the switch CPU.

udld-control

Specifies control plane policing for UDLD and DLDP control packets.

unclassified

Specifies control plane policing for various other protocol packets copied to the switch.

unicast-sw-forward

Specifies control plane policing for unknown IP destination packets to be routed.

limit {default | <*VALUE*>}

Specifies the rate limit value in packets per second. See **Traffic class limits**.

Example

Enable control plane policing for icmp traffic:

```
switch(config)# copp traffic-class station-icmp limit default
```

Example

Limit mac-learns or move notifications copied to the CPU to 128 packets per second on each slot (slave cards or interface modules):

```
switch(config)# copp traffic-class mac-notificaiton limit 128
```

copp user-def

Syntax

```
copp user-def <copp-id> {ipv4 | ipv6}{any <protocol> <dest-port> | <dest-ipaddress> [<protocol> <dest-port>]}
     {limit <limit-val> | none | drop}
     [tcp | udp]}

no copp user-def <copp-id> {ipv4 | ipv6}{any <protocol> <dest-port> | <dest-ipaddress> [<protocol> <dest-port>]}
     {limit <limit-val> | none | drop}
     [tcp | udp]}
```

Description

Enables control plane policing for a user-defined traffic class. Matches on a particular ip address and layer 4 port for CoPP.

The `no` form of this command disables control plane policing.

Command context

`config`

Parameters

user-def <*copp-id*>

Specifies up to 8 user defined copp configurations.

`{ipv4|ipv6}`

Specifies control plane policing for IPv4 or IPv6.

`any` ***`<protocol> <dest-port>`***

Specify the port number between 0-65535.

`<dest-ipaddress>`

Specifies the IP address of User Defined Control Plane Policing.

`<protocol>`

Specifies control plane policing for transport layer protocol (TCP, UDP), the destination port for the given TCP or UDP protocol .

`limit` ***`<limit-val>`***

Specifies the rate limit value in packets per second.

`none`

Allows all traffic matching the user-defined traffic class with no limit.

`drop`

Blocks user-defined traffic from reaching the switch CPU.

`tcp`

Telnet protocol for user-defined control plane policing.

`udp`

Protocol for user-defined control plane policing.

Example

Enable control plane policing only for SNMP traffic reaching the switch at up to 80 packets per second on each slot:

```
switch(config)# copp user-def ipv4 any udp 161 limit 80
```

Example

Block all Telnet traffic from reaching the switch:

```
switch(config)# copp user-def 1 ipv4 any tcp 23 drop
```

```
switch(config)# show copp user-def config

CoPP-ID  L3    Protocol  Port   Threshold
-------  --    --------  ----   ---------
1        IPv4  udp       161    drop
2        IPv4  udp        69    none
```

Traffic class limits

Table 33: *Traffic class limits*

Traffic class	Default limit	Range
station-arp	512	8 to 1024
station-icmp	128	8 to 1024
station-ip	512	8 to 1024
ip-gateway-control	128	8 to 512
ospf	512	8 to 1024
bgp	512	8 to 1024
rip	512	8 to 1024
multicast-route-control	256	8 to 1024
loop-ctrl-mstp	256	8 to 512
loop-ctrl-pvst	256	8 to 512
loop-ctrl-loop-protect	256	8 to 512
loop-ctrl-smart-links	256	8 to 512
layer2-control-others	512	8 to 1024
udld-control	256	8 to 256
sampling	256	8 to 512

Table Continued

icmp-redirect	64	8 to 128
unicast-sw-forward	512	8 to 1024
multicast-sw-forward	512	8 to 1024
mac-notification	512	8 to 1024
exception-notification	256	8 to 512
broadcast	512	8 to 512
unclassified	64	8 to 512

show copp

Syntax

`show copp {config | status | user-def}`

Description

Show CoPP configuration information.

Command context

`config`

Parameters

`config`

Show CoPP configuration details for all traffic classes.

`status`

Show CoPP status information for all traffic classes.

`user-def`

Show the user-defined CoPP configuration.

Example

```
slot(config)# show copp config

 Traffic-Class                  CoPP Status    Rate-Limit
 ------------------------------ -------------- ------------
 station-arp                    Disabled       0
 station-icmp                   Enabled        128
 station-ip                     Disabled       0
 ip-gateway-control             Disabled       0
 ospf                           Disabled       0
 bgp                            Disabled       0
 rip                            Disabled       0
 multicast-route-control        Disabled       0
 loop-ctrl-mstp                 Disabled       0
 loop-ctrl-pvst                 Disabled       0
 loop-ctrl-loop-protect         Disabled       0
 loop-ctrl-smart-links          Disabled       0
 layer2-control-others          Disabled       0
 udld-control                   Disabled       0
 sampling                       Disabled       0
 icmp-redirect                  Disabled       0
 unicast-sw-forward             Disabled       0
 multicast-sw-forward           Disabled       0
 mac-notification               Enabled        128
 exception-notification         Disabled       0
 broadcast                      Disabled       0
 unclassified                   Disabled       0
```

Example

```
slot(config)# show copp status

 Traffic-Class                  CoPP Status Threshold Ex Rx Violate Pkts
 ------------------------------ ----------- ------------ ----------------------
 station-arp                    Disabled    No
 station-icmp                   Enabled     No
 station-ip                     Disabled    No
 ip-gateway-control             Disabled    No
 ospf                           Disabled    No
 bgp                            Disabled    No
 rip                            Disabled    No
 multicast-route-control        Disabled    No
 loop-ctrl-mstp                 Disabled    No
 loop-ctrl-pvst                 Disabled    No
 loop-ctrl-loop-protect         Disabled    No
 loop-ctrl-smart-links          Disabled    No
 layer2-control-others          Disabled    No
 udld-control                   Disabled    No
 sampling                       Disabled    No
 icmp-redirect                  Disabled    No
 unicast-sw-forward             Disabled    No
 multicast-sw-forward           Disabled    No
 mac-notification               Enabled     No
 exception-notification         Disabled    No
 broadcast                      Disabled    No
 unclassified                   Disabled    No
```

Example

```
slot(config)# show copp user-def  config

 CoPP-ID L3   Protocol Port   Threshold
 ------- ---- --------- ------ --------------
 1       ipv4 udp      161    80
```

Example

```
slot(config)# show copp user-def  status

 CoPP-ID L3   Protocol Port   Threshold Ex   Dropped Pkts
 ------- ---- --------- ------ -------------- ----------------------
 1       ipv4 udp      161    No
```

Overview of MAC classes

MAC classes provide a new functionality to the existing Classifier policy feature. MAC classes allow for the matching on Ethernet header information: source MAC address, destination MAC address, EtherType, CoS, or VLAN. MAC classes open up the ability to match traffic that the existing IPv4 and IPv6 classes were unable to match. After the class is configured the class can be added into a policy and be associated with an action (that is, remark, rate limits, or mirroring). MAC classes can be included in QoS and Mirror policies and be applied to port, trunk, and VLAN interfaces.

There are some limitations to using MAC classes with the existing IPv4 and IPv6 classes; they cannot both be used in the same policy. However, it is possible to have a policy that contains MAC classes applied to a port interface and also have a policy that contains IPv4 and IPv6 classes applied to a VLAN. The user interface to configure MAC classes works similarly to the way classes are defined for IP-based traffic.

To use a MAC class, this feature requires the class to be configured, the policy to be configured, and the policy to be applied to an interface. This section describes how to configure each of these items and provides examples showing the use of this feature.

MAC Class configuration commands

MAC classes creation syntax

Syntax

```
no class ipv4|ipv6|mac|zoneCLASS_NAME
```

This command creates a new MAC class with a given name. The name is used when configuring the policy to associate a class with a given action. Upon configuring a class, the user is placed into the class context to configure the rules.

ipv4

Create a traffic class for IPv4 packets.

ipv6

Create a traffic class for IPv6 packets.

mac

Create a traffic class for MAC packets.

zone

Enter the zone name.

CLASS_NAME

Enter an ASCII string.

Create a new MAC class

```
(config)# class mac “mac-class-1”
(config-class)#
```

NOTE: The command `class ipv4 NAME` is the IPv4 equivalent command.

MAC class resequence

Syntax

```
class resequence ipv4|ipv6|mac start_increment
```

Resequencing a class renumbers the class from a specified starting point. Renumbering allows for additional space to be placed between the class entries while maintaining the proper order of the entries.

ipv4

Classify traffic based on IPv4 information.

ipv6

Classify traffic based on IPv6 information.

mac

Classify traffic based on Ethernet header information.

<1-2147483647>

The starting sequence number.

<1-2147483646>

The increment for each entry.

Resequence a class starting at sequence 10 and incrementing by 5

```
(config)# class resequence mac 10 5
```

MAC configuring class entries

Syntax

```
no SEQ_NUM match|ignore any|host SRC-MAC|SRC-MAC SRC-MAC-MASK any|host SRC-MAC|SRC-MAC SRC-MAC-MASK any|ETHERTYPE cos COS vlan vlan_id
```

A class is composed of entries that describe the traffic to be matched and ignored. Later, when the class is added to a policy, an action can be applied to the matched traffic. Traffic that is ignored excludes the traffic from the rest of the policy and takes no action. A class can be composed of many entries, which are processed sequentially.

Behavior

match

Create a rule to match specified packets.

ignore

Create a rule to ignore specified packets.

SOURCE MAC

any

Match packets with any source MAC address.

host

Match a specified source MAC address.

SRC-MAC

Match a specified source MAC address Range.

SRC-MAC-MASK

The source MAC address mask.

DESTINATION MAC

any

Match packets with any source MAC address.

host

Match a specified source MAC address.

SRC-MAC

Match a specified source MAC address Range.

SRC-MAC-MASK

The source MAC address mask.

ETHERTYPE

600-FFFF

Match a specific EtherType.

any

Match any EtherType.

aarp

AppleTalk Address Resolution Protocol.

appletalk

AppleTalk/EtherTalk.

arp

Address Resolution Protocol.

fcoe

Fibre Channel over Ethernet.

fcoe-init

Fibre Channel over Ethernet Initialization.

lldp

Link Layer Discovery Protocol.

ip

Internet Protocol Version 4.

ipv6

Internet Protocol Version 6.

ipx-arpa

IPX Advanced Research Projects Agency.

ipx-non-arpa

IPX non-ARPA

is-is

Intermediate System to Intermediate System

mpls-unicast

MPLS Unicast

mpls-multicast

MPLS Multicast

q-in-q

IEEE 802.1ad encapsulation

rbridge

RBridge Channel Protocol

trill

IETF TRILL protocol

wake-on-lan

Wake on LAN

Parameters

cos

Match packets with a specified 802.1Q Priority Code Point value.

vlan

Match packets for a configured VLAN.

0-7

Match packets with a specified 802.1Q Priority Code Point value.

vlan

Match packets for a configured VLAN.

VLAN-ID

Enter VLAN identifier or name.

Configuring a basic rule to match ARP traffic

```
(config)# class mac “mac-class-1”
(config-class)# match any any arp
```

Configuring a basic rule to match range of source MAC addresses

```
(config)# class mac “mac-class-2”
(config-class)# match AABB.CCDD.0000 0000.0000.FFFF any any
```

Configuring a basic rule to match a specific destination MAC addresses

```
(config)# class mac “mac-class-2”
(config-class)# match any host AABB.CCDD.EEFF any
```

Creating policy

A policy associates a class with an action. The policy is not active until it is applied to an interface. A policy that contains no entries has no effect on the traffic. The available actions of a policy differ depending on the type of policy configured. Mac classes are valid only with QoS and Mirror policies. After creating a policy, the user is placed in that policy's context in order to add, remove, or modify entries in the policy. A policy can contain many entries that provide the same or different actions for a class.

Syntax

```
no policy qos|pbr|mirror|zone POLICY-NAME
```

mirror

Create or modify a policy that has mirror actions.

pbr

Create or modify a policy that has PBR actions.

qos

Create or modify a policy that has QOS actions.

POLICY-NAME

Enter an ASCII string.

Create a new qos policy

```
(config)# policy qos “qos-policy-1”
(policy-qos)#
```

Create a new mirror policy

```
(config)# policy mirror “mirror-policy-1”
(policy-mirror)#
```

Mirror policy context

Syntax

```
no SEQ_NUM class ipv4|ipv6|mac CLASS_NAME action mirror SESSION mirror SESSION
```

The mirror policy context associates classes with mirror actions. After creating a policy, the user is placed in the mirror policy context. Traffic that matches the rules inside the class is mirrored using the associated mirror instance. Traffic that matches an ignore rule is not mirrored. Traffic that does not match any entries has the default-mac-class action applied. MAC classes cannot be configured in the same policy with IPv4 and IPv6 classes. The default-class is to be used with policies that contain IPv4 and IPv6 classes, whereas the default-mac-class is to be used with MAC classes.

class_name

Enter an ASCII string

<1-4>

Specify a mirror session as the action.

Create a new mirror policy that uses a MAC class

```
(config)# class mac mac-class-1

(config-class)# match any any arp

(config)# policy mirror "mirror-policy-1"

(policy-mirror)# class mac mac-class-1 action mirror 1
```

NOTE: The mirror action can be matched multiple times for the number of mirror sessions available. If there is a maximum of four mirror sessions available, you may have four mirror actions associated with a single class.

Syntax

```
no default-mac-class action mirror SESSION
```

The default-mac-class applies an action to the packets that are neither matched nor ignored by any class associated with the mirror policy. The default MAC class is used in a policy that contains only MAC classes.

Create a Mirror policy with a default MAC class

```
(config)# policy mirror "mirror-policy-1"
(policy-mirror)# default-mac-class action mirror 1
```

NOTE: The default-mac-class is to be used only in a policy that contains MAC classes.

Adding a remark to the policy

Syntax

```
no SEQ_NUM remark REMARK_STR
```

REMARK_STR

Add a comment to the policy.

The remark command inserts a comment into the policy at the specified sequence number. If no sequence number is given, the remark is added to the end of the list. Remarks consume the sequence number they are given and remain in order if the policy is resequenced.

Add a remark to a mirror policy

```
(config)# policy mirror "mirror-policy-1"
(policy-mirror)# 5 remark "This rule was added to fix problems related to ticket 234223"
```

QoS policy context

Syntax

```
no SEQ_NUM class ipv4|ipv6|mac CLASS_NAME action dscp|ip_precedence| priority |rate_limit
```

The QoS policy context associates classes with policy actions. One is placed in this context after creating a QoS policy. Traffic that matches the rules in the class will have the specified QoS action applied. Traffic that matches an ignore statement in the class does not have an action applied. Traffic that does not match any entries has the default-mac-class action applied. MAC classes cannot be configured in the same policy with IPv4 and IPv6 classes. The default-class is used with policies that contain IPv4 and IPv6 classes, whereas the default-mac-class is to be used with MAC classes.

dscp

Specify an IP DSCP.

ip-precedence

Specify the IP precedence.

priority

Specify the priority.

rate-limit

Configure rate limiting for all traffic.

DSCP OPTIONS

<0-63>
af11 Match DSCP AF11 (001010)
af12 Match DSCP AF12 (001100)
af13 Match DSCP AF13 (001110)
af21 Match DSCP AF21 (010010)
af22 Match DSCP AF22 (010100)
af23 Match DSCP AF23 (010110)
af31 Match DSCP AF31 (011010)
af32 Match DSCP AF32 (011100)
af33 Match DSCP AF33 (011110)
af41 Match DSCP AF41 (100010)
af42 Match DSCP AF42 (100100)
af43 Match DSCP AF43 (100110)
cs1 Match DSCP CS1 (001000)
cs2 Match DSCP CS2 (010000)
cs3 Match DSCP CS3 (011000)
cs4 Match DSCP CS4 (100000)
cs5 Match DSCP CS4 (101000)
cs6 Match DSCP CS6 (110000)
cs7 Match DSCP CS7 (111000)
default Match DSCP default (000000)
ef Match DSCP EF (101110)

IP-PRECEDENCE OPTIONS

<0-7>
routine
priority
immediate

flash
flash-override
critical
internet
network

PRIORITY Options

<0-7> Enter an integer number.

RATE-LIMIT Options

kbps Specify the rate limit in kilobits per second.

Create a QoS policy that rate remarks a packet

```
(config)# policy qos “qos-policy-1”
(policy-qos)# class mac mac-class-1 action priority 3
```

Create a QoS policy that rate limits traffic

```
(config)# policy qos “qos-policy-1”
(policy-qos)# class mac mac-class-1 action rate-limit kbps 1000
```

NOTE: The dscp and ip-precedence actions set the same set of bits in the IP header.

Default MAC Class

Syntax

```
no default-mac-class action mirror SESSION
```

The default-mac-class applies an action to the packets that are neither matched nor ignored by any class associated with the mirror policy. The default MAC class is used in a policy that contains only MAC classes.

Create a QoS policy with a default MAC class

```
(config)# policy qos “qos-policy-1”
(policy-qos)# default-mac-class action dscp af11
```

NOTE: The default-mac-class is to be used only in a policy that contains MAC classes.

Inserting a remark into a policy

Syntax

```
no SEQ_NUM remark REMARK_STR
```

REMARK_STR

Add a comment to the policy.

The **remark** command inserts a comment into the policy at the specified sequence number. If no sequence number is given, the remark is added to the end of the list. Remarks consume the sequence number that they are given and remain in order if the policy is resequenced.

Add a remark to a mirror policy

```
(config)# policy qos “qos-policy-1”
(policy-qos)# 100 remark “Add user specific rules above this point”
```

NOTE: A policy does not perform any action until it is applied to an interface. A policy that contains MAC classes can be applied to a port, trunk, or VLAN. The interface and direction of the application determine where in the flow of traffic through the switch the traffic will be compared with the policy's entries.

Applying the Service-policy

Syntax

```
no service-policy POLICY_NAME in
```

A policy does not perform any action until it is applied to an interface. A policy that contains MAC classes can be applied to a port, trunk, or VLAN. The interface and direction of the application determine where in the flow of traffic through the switch the traffic will be compared with the policy's entries.

in

Apply policy on inbound packets.

policy-name

Enter an ASCII string.

Apply a QoS policy to the inbound direction of a port

```
(config)# interface a1
(eth-A1)# service-policy qos-policy-1 in
```

Apply a QoS policy to the inbound direction of a VLAN

```
(config)# vlan 10
(vlan-10)# service-policy qos-policy-2 in
```

Apply a QoS policy to the inbound direction of a trunk

```
(config)# interface trk1
(eth-trk1)# service-policy qos-policy-3 in
```

Creating class assigning to mirror policy and applying to a port

```
(config)# class mac mac-class-1
(config-class)# match any any any
```

```
(config)# policy mirror “mirror-policy-1”
(policy-mirror)# class mac mac-class-1 action mirror 1
(config)# interface a1
(eth-a1)# service-policy mirror-policy-1 in
```

Show MAC class by name

Syntax

```
show class mac class_name
```

Displays information about a specific class.

ASCII-STR

Enter an ASCII string.

Show class mac

```
switch(config)# show class mac macClass Statements for class mac "macClass"
class mac "macClass" 10 match 1111.2222.3333 ffff.ffff.0000 4444.5555.6666 ffff.ffff.0000 aarp
exit
```

Show class ports

Syntax

```
show class ports port_list
```

Displays the classes that are applied on the specified port.

[ethernet] PORT-LIST

Enter a port number, a list of ports, or **`all`** for all ports

Show class ports

```
switch(config)# show class ports A1
Classes for port A1
 Name    : test
 Type    : MAC
```

show class vlan

Syntax

```
show class vlan vlan-id
```

Displays the classes that are applied on the specified VLAN.

vlan-id

Enter a VLAN identifier or a VLAN name.

Show class vlan

```
switch(config)# show class vlan 2
Classes for vlan 2
 Name    : test
 Type    : MAC
```

Show policy

Syntax

```
show policy policy-name
```

Shows a specific policy.

POLICY-NAME

Enter an ASCII string.

Show policy by name

```
switch(config)# show policy qos-policy-1
Statements for policy "qos-policy-1"
policy qos "qos-policy-1"
10 class mac "macClass" action ip-precedence 3 action priority 0
exit
(
```

show policy ports

Syntax

```
show policy ports port-list
```

Shows the applied policies on a specified port.

[ethernet] PORT-LIST

Enter a port number, a list of ports or 'all' for all ports.

Show policy ports

```
switch(config)# show policy ports A1
Policies for port A1
Name    : macClass
Type    : QOS
```

show policy vlan

Syntax

```
show policy vlan vlan-id
```

Shows policies that are applied on a specified VLAN.

all

Show policies applied to all VLANs.

VLAN-ID

Show policies applied to the specified VLAN.

Create a new mirror policy

```
show policy vlan 2
Policies for VLAN 2
```

```
Name     : macClass
Type       : QOS
```

show statistics policy port

Syntax

`show statistics policy POLICY-NAME-STR port PORT-NUM`

Displays hit count statistics for a given policy on a port.

PORT-NUM

Enter a port name.

POLICY-NAME-STR

The policy to show statistics for.

Show statistics for policy by port

```
switch(config)# show statistics policy temp port a1
HitCounts for Policy qos-policy-1 since the last 2277 seconds
Total
10 class mac "macClass" action ip-precedence 3 action priority 0
( 69171 ) 10 match 1111.2222.3333 ffff.ffff.0000 4444.5555.6666 ffff.ffff.0000
aarp
```

Show statistics policy VLAN

Syntax

`show statistics policy POLICY-NAME vlan VLAN-ID`

Displays hit counts for a specified policy on a VLAN.

VLAN-ID

The VLAN ID or VLAN name.

POLICY-NAME

The policy to show statistics for.

show statistics policy

```
switch(config)# show statistics policy temp vlan 1
HitCounts for Policy temp occurring in the last 851 seconds
Total
10 class mac "macClass" action ip-precedence 3 action priority 0
( 0 ) 10 match 1111.2222.3333 ffff.ffff.0000 4444.5555.6666 ffff.ffff.0000
aarp
```

clear statistics

Syntax

`clear statistics policy|aclv6|aclv4|mac`

Clears hit counts.

policy

QoS/Mirror/PBR policy.

aclv6

IPv6 ACL.

aclv4

IPv4 ACL.

mac

MAC ACL.

Clear statistics for a policy on a port

```
(config)# clear statistics policy policy-name port a1
```

Clear statistics for a policy on a vlan

```
(config)# clear statistics policy policy-name vlan 1
```

Chapter 16 Websites

Networking Websites

Hewlett Packard Enterprise Networking Information Library

www.hpe.com/networking/resourcefinder

Hewlett Packard Enterprise Networking Software

www.hpe.com/networking/software

Hewlett Packard Enterprise Networking website

www.hpe.com/info/networking

Hewlett Packard Enterprise My Networking website

www.hpe.com/networking/support

Hewlett Packard Enterprise My Networking Portal

www.hpe.com/networking/mynetworking

Hewlett Packard Enterprise Networking Warranty

www.hpe.com/networking/warranty

General websites

Hewlett Packard Enterprise Information Library

www.hpe.com/info/EIL

For additional websites, see **Support and other resources**.

Accessing Hewlett Packard Enterprise Support

- For live assistance, go to the Contact Hewlett Packard Enterprise Worldwide website:

  http://www.hpe.com/info/assistance
- To access documentation and support services, go to the Hewlett Packard Enterprise Support Center website:

  http://www.hpe.com/support/hpesc

Information to collect

- Technical support registration number (if applicable)
- Product name, model or version, and serial number
- Operating system name and version
- Firmware version
- Error messages
- Product-specific reports and logs
- Add-on products or components
- Third-party products or components

Accessing updates

- Some software products provide a mechanism for accessing software updates through the product interface. Review your product documentation to identify the recommended software update method.
- To download product updates:

  Hewlett Packard Enterprise Support Center
    www.hpe.com/support/hpesc
  Hewlett Packard Enterprise Support Center: Software downloads
    www.hpe.com/support/downloads
  Software Depot
    www.hpe.com/support/softwaredepot
- To subscribe to eNewsletters and alerts:

  www.hpe.com/support/e-updates
- To view and update your entitlements, and to link your contracts and warranties with your profile, go to the Hewlett Packard Enterprise Support Center **More Information on Access to Support Materials** page:

  www.hpe.com/support/AccessToSupportMaterials

IMPORTANT: Access to some updates might require product entitlement when accessed through the Hewlett Packard Enterprise Support Center. You must have an HPE Passport set up with relevant entitlements.

Customer self repair

Hewlett Packard Enterprise customer self repair (CSR) programs allow you to repair your product. If a CSR part needs to be replaced, it will be shipped directly to you so that you can install it at your convenience. Some parts do not qualify for CSR. Your Hewlett Packard Enterprise authorized service provider will determine whether a repair can be accomplished by CSR.

For more information about CSR, contact your local service provider or go to the CSR website:

http://www.hpe.com/support/selfrepair

Remote support

Remote support is available with supported devices as part of your warranty or contractual support agreement. It provides intelligent event diagnosis, and automatic, secure submission of hardware event notifications to Hewlett Packard Enterprise, which will initiate a fast and accurate resolution based on your product's service level. Hewlett Packard Enterprise strongly recommends that you register your device for remote support.

If your product includes additional remote support details, use search to locate that information.

Remote support and Proactive Care information

HPE Get Connected

www.hpe.com/services/getconnected

HPE Proactive Care services

www.hpe.com/services/proactivecare

HPE Proactive Care service: Supported products list

www.hpe.com/services/proactivecaresupportedproducts

HPE Proactive Care advanced service: Supported products list

www.hpe.com/services/proactivecareadvancedsupportedproducts

Proactive Care customer information

Proactive Care central

www.hpe.com/services/proactivecarecentral

Proactive Care service activation

www.hpe.com/services/proactivecarecentralgetstarted

Warranty information

To view the warranty information for your product, see the links provided below:

HPE ProLiant and IA-32 Servers and Options

www.hpe.com/support/ProLiantServers-Warranties

HPE Enterprise and Cloudline Servers

www.hpe.com/support/EnterpriseServers-Warranties

HPE Storage Products

www.hpe.com/support/Storage-Warranties

HPE Networking Products

www.hpe.com/support/Networking-Warranties

Regulatory information

To view the regulatory information for your product, view the *Safety and Compliance Information for Server, Storage, Power, Networking, and Rack Products*, available at the Hewlett Packard Enterprise Support Center:

www.hpe.com/support/Safety-Compliance-EnterpriseProducts

Additional regulatory information

Hewlett Packard Enterprise is committed to providing our customers with information about the chemical substances in our products as needed to comply with legal requirements such as REACH (Regulation EC No 1907/2006 of the European Parliament and the Council). A chemical information report for this product can be found at:

www.hpe.com/info/reach

For Hewlett Packard Enterprise product environmental and safety information and compliance data, including RoHS and REACH, see:

www.hpe.com/info/ecodata

For Hewlett Packard Enterprise environmental information, including company programs, product recycling, and energy efficiency, see:

www.hpe.com/info/environment

Documentation feedback

Hewlett Packard Enterprise is committed to providing documentation that meets your needs. To help us improve the documentation, send any errors, suggestions, or comments to Documentation Feedback (**docsfeedback@hpe.com**). When submitting your feedback, include the document title, part number, edition, and publication date located on the front cover of the document. For online help content, include the product name, product version, help edition, and publication date located on the legal notices page.