

AOS-CX 10.10 Layer-2 Bridging Guide

8400 Switch Series



a Hewlett Packard
Enterprise company

Copyright Information

© Copyright 2023 Hewlett Packard Enterprise Development LP.

Open Source Code

This product includes code licensed under the GNU General Public License, the GNU Lesser General Public License, and/or certain other open source licenses. A complete machine-readable copy of the source code corresponding to such code is available upon request. This offer is valid to anyone in receipt of this information and shall expire three years following the date of the final distribution of this product version by Hewlett Packard Enterprise Company. To obtain such source code, send a check or money order in the amount of US \$10.00 to:

Hewlett Packard Enterprise Company
6280 America Center Drive
San Jose, CA 95002
USA

Notices

The information contained herein is subject to change without notice. The only warranties for Hewlett Packard Enterprise products and services are set forth in the express warranty statements accompanying such products and services. Nothing herein should be construed as constituting an additional warranty. Hewlett Packard Enterprise shall not be liable for technical or editorial errors or omissions contained herein.

Confidential computer software. Valid license from Hewlett Packard Enterprise 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.

Links to third-party websites take you outside the Hewlett Packard Enterprise website. Hewlett Packard Enterprise has no control over and is not responsible for information outside the Hewlett Packard Enterprise website.

Acknowledgment

Intel®, Itanium®, Optane™, Pentium®, Xeon®, Intel Inside®, and the Intel Inside logo are trademarks of Intel Corporation in the U.S. and other countries.

Microsoft® and Windows® are either registered trademarks or trademarks of Microsoft Corporation in the United States and/or other countries.

Adobe® and Acrobat® are trademarks of Adobe Systems Incorporated.

Java® and Oracle® are registered trademarks of Oracle and/or its affiliates.

UNIX® is a registered trademark of The Open Group.

All third-party marks are property of their respective owners.

Contents	3
About this document	8
Applicable products	8
Latest version available online	8
Command syntax notation conventions	8
About the examples	9
Identifying switch ports and interfaces	9
Identifying modular switch components	10
Introduction	11
MAC address table	12
MAC address table commands	13
clear mac-address	13
mac-address-table age-time	14
show mac-address-table	15
show mac-address-table address	16
show mac-address-table count	17
show mac-address-table dynamic	18
show mac-address-table interface	19
show mac-address-table lockout	20
show mac-address-table port	21
show mac-address-table static	22
show mac-address-table vlan	22
static-mac	23
VLANs	25
VLAN interfaces	25
Access interface	25
Trunk interface	26
Traffic handling summary	27
Comparing VLAN commands on PVOS, Comware, and AOS-CX	28
Protocol-mapped VLANs	29
VLAN translation	29
Assigning a VLAN to an interface	30
Assigning a VLAN ID to an access interface	30
Assigning a VLAN ID to a trunk interface	31
Assigning a native VLAN ID to a trunk interface	32
VLAN numbering	33
Configuring VLANs	33
Creating and enabling a VLAN	33
Disabling a VLAN	33
Viewing VLAN configuration information	34
VLAN scenario	36
VLAN commands	42
description	42
vlan name	42

show capacities-status vlan-count	43
show capacities svi-count	44
show capacities vlan-count	44
show vlan	45
show vlan port	46
show vlan summary	47
show vlan translation	48
show vlan translation pending	49
show vlan voice	50
shutdown	51
trunk-dynamic-vlan-include	51
vlan	52
vlan access	53
vlan protocol	55
vlan translate	56
vlan trunk allowed	57
vlan trunk native	58
vlan trunk native tag	60
voice	61
Loop protection	62
Interaction with other protocols	62
Configuring loop protection	63
Loop protect commands	64
loop-protect	64
loop-protect action	65
loop-protect re-enable-timer	66
loop-protect transmit-interval	66
loop-protect trap loop-detected	67
loop-protect vlan	68
show loop-protect	68
MVRP	71
MVRP functionality and limitations	71
MRP messages	72
Join message	72
New message	72
Leave message	73
LeaveAll message	73
Configuring MVRP	73
MVRP scenario 1	74
MVRP scenario 2	78
MVRP commands	85
clear mvrp statistics	85
mvrp	86
mvrp registration	87
mvrp timer	88
show mvrp config	89
show mvrp state	89
show mvrp statistics	90
Spanning tree protocols (STP)	93
Protocols and feature details	93
STP	93
Root bridge	93
Root port	93

Designated bridge and designated port	93
Path cost	94
STP timers	94
BPDUs forwarding mechanism	95
STP protocol packets	95
Comparing spanning tree options	96
Preparing for spanning tree configuration	96
STP cost calculation	97
Simplified calculation overview	97
Calculation example	98
STP supported platforms and standards	103
Scale	103
RPVST+ protocol and feature details	104
RPVST+ vPorts	107
RPVST+ configuration tasks	108
Viewing RPVST+ information	110
RPVST+ Considerations and best practices	111
RPVST+ use cases	113
RPVST+ use case: Deterministic root bridges	113
Switch A configuration	114
Switch B configuration	114
Switch C and D configuration	115
Checking the configuration	115
Observe port behavior and state	116
RPVST+ use case: BPDUs protection	118
RPVST+ use case: Root protection	121
RPVST+ use case: Spanning tree on edge ports	123
RPVST+ use case: Preventing loops	125
RPVST+ commands	128
clear spanning-tree statistics	128
show capacities rpvst	128
show capacities-status rpvst	129
show spanning-tree	129
show spanning-tree detail	131
show spanning-tree inconsistent-ports	132
show spanning-tree summary port	133
show spanning-tree summary root	134
show spanning-tree vlan	135
show spanning-tree vlan detail	136
spanning-tree bpdus-guard timeout	138
spanning-tree extend-system-id	139
spanning-tree ignore-pvid-inconsistency	139
spanning-tree link-type	141
spanning-tree mode	141
spanning-tree pathcost-type	142
spanning-tree rpvst-mstp interconnect vlan	143
spanning-tree tcu-guard	144
spanning-tree vlan	145
spanning-tree vlan cost	146
spanning-tree vlan port-priority	147
spanning-tree trap	147
MSTP protocol and feature details	150
MSTP key concepts	151
MSTP configuration tasks	155
MSTP use cases	156
MSTP use case: Eliminating a loop with MSTP	156

MSTP commands	159
clear spanning-tree statistics	159
show spanning-tree	159
show spanning-tree detail	160
show spanning-tree inconsistent-ports	162
show spanning-tree mst	163
show spanning-tree mst-config	165
show spanning-tree mst detail	166
show spanning-tree mst <INSTANCE-ID>	170
show spanning-tree mst <INSTANCE-ID> detail	171
show spanning-tree mst interface	172
show spanning-tree summary port	173
show spanning-tree summary root	174
spanning-tree	175
spanning-tree bpdu-filter	175
spanning-tree bpdu-guard	176
spanning-tree bpdu-guard timeout	177
spanning-tree config-name	178
spanning-tree config-revision	179
spanning-tree cost	180
spanning-tree forward-delay	181
spanning-tree hello-time	181
spanning-tree instance cost	182
spanning-tree instance port-priority	183
spanning-tree instance priority	184
spanning-tree instance vlan	185
spanning-tree link-type	186
spanning-tree loop-guard	187
spanning-tree max-age	188
spanning-tree max-hops	188
spanning-tree mode	189
spanning-tree port-priority	190
spanning-tree port-type	191
spanning-tree priority	192
spanning-tree root-guard	193
spanning-tree rpvst-filter	194
spanning-tree rpvst-guard	195
spanning-tree tcu-guard	195
spanning-tree transmit-hold-count	196
spanning-tree trap	197
UDLD	200
Configuring UDLD	201
UDLD scenario	202
UDLD commands	203
clear udd statistics	203
show udd	204
udd	206
udd interval	207
udd mode	209
udd retries	210
Support and Other Resources	212
Accessing Aruba Support	212
Accessing Updates	213
Aruba Support Portal	213

My Networking	213
Warranty Information	213
Regulatory Information	213
Documentation Feedback	214
spanning-tree rpvt-mstp interconnect vlan	214

This document describes features of the AOS-CX network operating system. It is intended for administrators responsible for installing, configuring, and managing Aruba switches on a network.

Applicable products

This document applies to the following products:

- Aruba 8400 Switch Series (JL375A, JL376A)

Latest version available online

Updates to this document can occur after initial publication. For the latest versions of product documentation, see the links provided in [Support and Other Resources](#).

Command syntax notation conventions

Convention	Usage
<code>example-text</code>	Identifies commands and their options and operands, code examples, filenames, pathnames, and output displayed in a command window. Items that appear like the example text in the previous column are to be entered exactly as shown and are required unless enclosed in brackets ([]).
example-text	In code and screen examples, indicates text entered by a user.
Any of the following: <ul style="list-style-type: none">■ <code><example-text></code>■ <i><example-text></i>■ <code>example-text</code>■ <i>example-text</i>	Identifies a placeholder—such as a parameter or a variable—that you must substitute with an actual value in a command or in code: <ul style="list-style-type: none">■ For output formats where italic text cannot be displayed, variables are enclosed in angle brackets (< >). Substitute the text—including the enclosing angle brackets—with an actual value.■ For output formats where italic text can be displayed, variables might or might not be enclosed in angle brackets. Substitute the text including the enclosing angle brackets, if any, with an actual value.
	Vertical bar. A logical OR that separates multiple items from which you can choose only one. Any spaces that are on either side of the vertical bar are included for readability and are not a required part of the command syntax.
{ }	Braces. Indicates that at least one of the enclosed items is required.
[]	Brackets. Indicates that the enclosed item or items are optional.

Convention	Usage
... or ...	Ellipsis: <ul style="list-style-type: none"> ■ In code and screen examples, a vertical or horizontal ellipsis indicates an omission of information. ■ In syntax using brackets and braces, an ellipsis indicates items that can be repeated. When an item followed by ellipses is enclosed in brackets, zero or more items can be specified.

About the examples

Examples in this document are representative and might not match your particular switch or environment.

The slot and port numbers in this document are for illustration only and might be unavailable on your switch.

Understanding the CLI prompts

When illustrating the prompts in the command line interface (CLI), this document uses the generic term `switch`, instead of the host name of the switch. For example:

```
switch>
```

The CLI prompt indicates the current command context. For example:

```
switch>
```

Indicates the operator command context.

```
switch#
```

Indicates the manager command context.

```
switch(CONTEXT-NAME)#
```

Indicates the configuration context for a feature. For example:

```
switch(config-if)#
```

Identifies the interface context.

Variable information in CLI prompts

In certain configuration contexts, the prompt may include variable information. For example, when in the VLAN configuration context, a VLAN number appears in the prompt:

```
switch(config-vlan-100)#
```

When referring to this context, this document uses the syntax:

```
switch(config-vlan-<VLAN-ID>)#
```

Where `<VLAN-ID>` is a variable representing the VLAN number.

Identifying switch ports and interfaces

Physical ports on the switch and their corresponding logical software interfaces are identified using the format:

```
member/slot/port
```

On the 8400 Switch Series

- *member*: Always 1. VSF is not supported on this switch.
- *slot*: Specifies physical location of a module in the switch chassis.
 - Management modules are on the front of the switch in slots 1/5 and 1/6.
 - Line modules are on the front of the switch in slots 1/1 through 1/4, and 1/7 through 1/10.
- *port*: Physical number of a port on a line module

For example, the logical interface 1/1/4 in software is associated with physical port 4 in slot 1 on member 1.

Identifying modular switch components

- Power supplies are on the front of the switch behind the bezel above the management modules. Power supplies are labeled in software in the format: *member/power supply*:
 - *member*: 1.
 - *power supply*: 1 to 4.
- Fans are on the rear of the switch and are labeled in software as: *member/tray/fan*:
 - *member*: 1.
 - *tray*: 1 to 4.
 - *fan*: 1 to 4.
- Fabric modules are not labeled on the switch but are labeled in software in the format: *member/module*:
 - *member*: 1.
 - *member*: 1 or 2.
- The display module on the rear of the switch is not labeled with a member or slot number.

Switches use network bridging to facilitate the interconnection of local area networks (LANs) so that traffic can be exchanged between devices. Bridging occurs at layer 2 of the OSI model.

When creating network bridges on HPE switches, network administrators can configure MAC addressing, VLANs, and various loop prevention protocols.

Devices on a network are identified by their MAC address. The switch maintains a MAC address table where it stores information about the other Ethernet interfaces to which a switch is connected. The table enables the switch to send outgoing data (Ethernet frames) on the specific port required to reach its destination, instead of broadcasting the data on all ports (flooding).

VLANs are primarily used to provide network segmentation at layer 2. VLANs enable the grouping of users by logical function instead of physical location. Layer 2 VLANs can be associated with a single physical port, or multiple aggregated ports (referred to as LAG, short form for Link Aggregation). Link Aggregation enables a logical grouping of individual interfaces to function as a single, higher-speed link, providing dramatically increased bandwidth. This mechanism provides network resiliency when individual link failures occur. Aruba switches include advanced network resiliency through MCLAG (Multi Chassis Link Aggregation) which offers network resiliency on individual device failure as well.

When multiple individual links are connected to one another, there is a possibility that multiple paths (loops) will exist between devices. Loops reduce network operational efficiency. AOS-CX provides several features to detect and avoid loops, including:

- **MSTP:** Multiple-Instance spanning tree protocol (MSTP) 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.
- **RPVST+:** Rapid Per VLAN Spanning Tree+ (RPVST+) is an updated implementation of STP (Spanning Tree Protocol). It enables the creation of a separate spanning tree for each VLAN on a switch, and ensures that only one active, loop-free path exists between any two nodes on a given VLAN.
- **Loop Protection:** In cases where spanning tree protocols cannot be used to prevent loops at the edge of the network, loop protection may provide a suitable alternative. Loop protection can find loops in untagged layer 2 links, as well as on tagged VLANs.

AOS-CX also supports the MVRP (Multiple VLAN Registration Protocol), 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, reducing the configuration workload.

Additionally, AOS-CX supports the Unidirectional Link Detection (UDLD) protocol. UDLD monitors the link between two network devices, and if the link fails, blocks the ports on both ends of the link. UDLD is useful for detecting failures in fiber links and trunks.

The MAC address table is where the switch stores information about the other Ethernet interfaces to which it is connected on a network. The table enables the switch to send outgoing data (Ethernet frames) on the specific port required to reach its destination, instead of broadcasting the data on all ports (flooding).

The MAC address table can contain two types of entries:

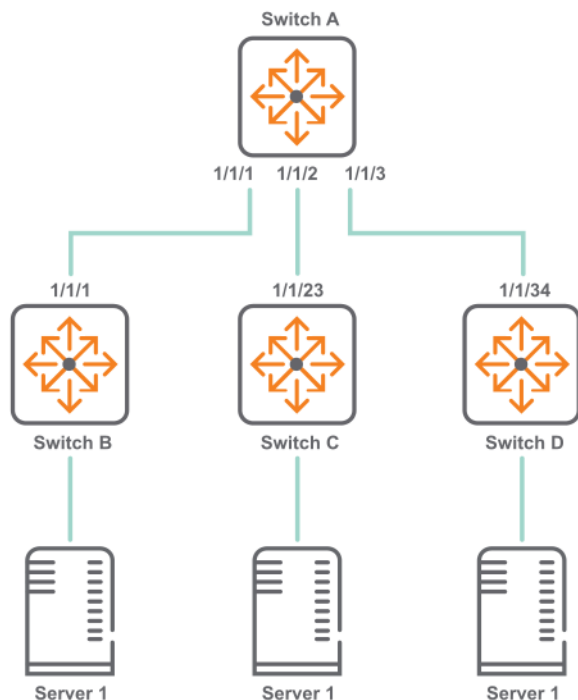
- **Static:** Static entries are manually added to the table by a switch administrator. Static entries have higher priority than dynamic entries. Static entries remain active until they are removed by the switch administrator.
- **Dynamic:** Dynamic entries are automatically added to the table through a process called MAC learning, in which the switch retrieves the source MAC address (and VLAN ID, if present) of each Ethernet frame received on a port. If the retrieved address does not exist in the table, it is added. Dynamic entries remain in the table for a predetermined amount of time (defined with the command `mac-address-table age-time`), after which they are automatically deleted.

Dynamic MAC address learning does not distinguish between illegitimate and legitimate frames, which can invite security hazards. When Host A is connected to port A, a MAC address entry will be learned for the MAC address of Host A (for example, MAC A). When an illegal user sends frames with MAC A as the source MAC address to port B, the device performs the following operations:

1. Learns a new MAC address entry with port B as the outgoing interface and overwrites the old entry for MAC A.
2. Forwards frames destined for MAC A out of port B to the illegal user.

As a result, the illegal user obtains the data of Host A. To improve the security for Host A, manually configure a static entry to bind Host A to port A. Then, the frames destined for Host A are always sent out of port A. Other hosts using the forged MAC address of Host A cannot obtain the frames destined for Host A.

For example, in the following topology, switch A learns the MAC addresses of ports on switch B, C, and D. This way, traffic between any two switches is not broadcast to the other switches. For example, if server 1 sends traffic to server 3, it does not get broadcast onto the link to switch C, only on the link to switch D.



MAC address table commands

clear mac-address

```
clear mac-address {interface <INTERFACE> | port <PORT-NUM> [vlan <VLAN-ID>] | vlan <VLAN-ID> [port <PORT-NUM>] | <MAC-ADDR> [vlan <VLAN-ID>] [force]}
```

Description

Clears the dynamic learned MAC addresses on the specified interface, combination of interface and VLAN, port, VLAN, combination of port and VLAN, MAC address, or combination of MAC address and VLAN. The command does not clear any port-security learned MAC addresses.

Port-security MAC addresses are cleared when the port on which the MAC addresses were learned are shut down or the port-access-security feature is disabled on the port or the switch.

Parameter	Description
<INTERFACE>	Specifies the list of interfaces, for example, 1/1/1 or 1/1/1-1/1/3 or lag1 or vxlan1.
<PORT-NUM>	Specifies a physical port on the switch. Format: member/slot/port.
<VLAN-ID>	Specifies the number of a VLAN.
<MAC-ADDR>	Specifies the MAC address.
force	Clears the specified MAC address even if the MAC address is internally programmed by MAC management.

Examples

Clearing the learned MAC addresses on a port:

```
switch# clear mac-address port 1/1/1
```

Clearing the learned MAC addresses on a combination of a VLAN and a port:

```
switch# clear mac-address port 1/1/1 vlan 20
```

```
switch# clear mac-address vlan 2 port 1/1/3
```

Clearing the learned MAC addresses on a combination of a VLAN and an interface or a list of interfaces:

```
switch# clear mac-address interface 1/1/1 vlan 10
```

```
switch# clear mac-address vlan 1 interface 1/1/1-1/1/3
```

Clearing the specified MAC addresses entry on the VLAN:

```
switch# clear mac-address 14:FA:01:F1:8B:8F vlan 1
```

Clearing the specified MAC addresses entry by force:

```
switch# clear mac-address 14:FA:01:F1:8B:8F force
```

Command History

Release	Modification
10.09	Added parameters for interface and MAC address.
10.07 or earlier	--

Command Information

Platforms	Command context	Authority
All platforms	Manager (#)	Administrators or local user group members with execution rights for this command.

mac-address-table age-time

```
mac-address-table age-time <SECONDS>  
no mac-address-table age-time [<SECONDS>]
```

Description

Sets the maximum amount of time a MAC address remains in the MAC address table. When this time expires, the MAC address is removed.

The `no` form of this command resets the MAC aging timer to the default value (300 seconds).

Parameter	Description
age-time <SECONDS>	Specifies the MAC address aging time in seconds. Range: 60 to 3600. Default: 300.

Example

```
switch(config)# mac-address-table age-time 120
```

Command History

Release	Modification
10.07 or earlier	--

Command Information

Platforms	Command context	Authority
All platforms	config	Administrators or local user group members with execution rights for this command.

show mac-address-table

```
show mac-address-table [hsc] [vsx-peer]
```

Description

Shows MAC address table information. If HSC is enabled, MAC addresses discovered by the HSC manager are also displayed.

Parameter	Description
[hsc]	Displays only MAC address discovered by the HSC manager on the remote controller.
vsx-peer	Shows the output from the VSX peer switch. If the switches do not have the VSX configuration or the ISL is down, the output from the VSX peer switch is not displayed. This parameter is available on switches that support VSX.

Examples

Showing output when table entries exist:

```
switch# show mac-address-table
MAC age-time           : 300 seconds
Number of MAC addresses : 5
```

MAC Address	VLAN	Type	Port
00:00:00:00:00:05	1	dynamic	1/1/2
00:00:00:00:00:06	2	dynamic	1/1/1

```
00:00:00:00:00:08    3      hsc      vxlan1(10.1.1.1)
00:00:00:00:00:12    3      hsc      vxlan1(10.1.1.3)
00:00:00:00:00:34    3      hsc      vxlan1(10.1.1.4)
```

Showing output when there are no MAC table entries:

```
switch# show mac-address-table
No MAC entries found.
```

Showing only MAC address discovered by the HSC manager:

```
switch# show mac-address-table hsc
Number of MAC addresses : 3
MAC Address          VLAN    Type      Port
-----
00:00:00:00:00:08    3      hsc      vxlan1(10.1.1.1)
00:00:00:00:00:12    3      hsc      vxlan1(10.1.1.3)
00:00:00:00:00:34    3      hsc      vxlan1(10.1.1.4)
```

Command History

Release	Modification
10.07 or earlier	--

Command Information

Platforms	Command context	Authority
All platforms	Operator (>) or Manager (#)	Operators or Administrators or local user group members with execution rights for this command. Operators can execute this command from the operator context (>) only.

show mac-address-table address

```
show mac-address-table address <MAC-ADDR> [vsx-peer]
```

Description

Shows MAC address table information for a specific MAC address.

Parameter	Description
<MAC-ADDR>	Specifies the MAC address.
vsx-peer	Shows the output from the VSX peer switch. If the switches do not have the VSX configuration or the ISL is down, the output from the VSX peer switch is not displayed. This parameter is available on switches that support VSX.

Example


```
switch# show mac-address-table address 00:00:00:00:00:01
MAC age-time           : 300 seconds
Number of MAC addresses : 2
```

MAC Address	VLAN	Type	Port
00:00:00:00:00:01	2	dynamic	1/1/1
00:00:00:00:00:01	1	dynamic	1/1/1

Command History

Release	Modification
10.07 or earlier	--

Command Information

Platforms	Command context	Authority
All platforms	Operator (>) or Manager (#)	Operators or Administrators or local user group members with execution rights for this command. Operators can execute this command from the operator context (>) only.

show mac-address-table count

```
show mac-address-table count
    [dynamic | port <PORT-NUM> | vlan <VLAN-ID>] [vsx-peer]
```

Description

Displays the number of MAC addresses.

Parameter	Description
dynamic	Show the count of dynamically learned MAC addresses.
<PORT-NUM>	Specifies a physical port on the switch. Format: member/slot/port.
vlan <VLAN-ID>	Specifies the number of a VLAN.
vsx-peer	Shows the output from the VSX peer switch. If the switches do not have the VSX configuration or the ISL is down, the output from the VSX peer switch is not displayed. This parameter is available on switches that support VSX.

Examples

Showing the number of MAC addresses:

```
switch# show mac-address-table count
Number of MAC addresses : 8
```

Showing the number of dynamically learned MAC addresses:

```
switch# show mac-address-table count dynamic
Number of MAC addresses : 8
```

Showing the number of MAC addresses per physical port on the switch:

```
switch# show mac-address-table count port 1/1/1
Number of MAC addresses : 2
```

Showing the number of MAC addresses per VLAN:

```
switch# show mac-address-table count vlan 100
Number of MAC addresses : 5
```

Showing the number of MAC addresses on the VSX primary and secondary (peer) switch:

```
vsx-primary# show mac-address-table count
Number of MAC addresses : 26114
vsx-primary# show mac-address-table count vsx-peer
Number of MAC addresses : 26113
```

Command History

Release	Modification
10.07 or earlier	--

Command Information

Platforms	Command context	Authority
All platforms	Operator (>) or Manager (#)	Operators or Administrators or local user group members with execution rights for this command. Operators can execute this command from the operator context (>) only.

show mac-address-table dynamic

```
show mac-address-table dynamic [port <PORT-NUM> | vlan <VLAN-ID>] [vsx-peer]
```

Description

Shows MAC address table information about dynamically learned MAC addresses.

Parameter	Description
<PORT-NUM>	Specifies a physical port on the switch. Format: member/slot/port.
<VLAN-ID>	Specifies the number of a VLAN.
vsx-peer	Shows the output from the VSX peer switch. If the switches do not have the VSX configuration or the ISL is down, the output from the

Parameter	Description
	VSX peer switch is not displayed. This parameter is available on switches that support VSX.

Examples

Showing all dynamic MAC address table entries:

```
switch# show mac-address-table dynamic
MAC age-time           : 300 seconds
Number of MAC addresses : 2

MAC Address      VLAN    Type      Port
-----
00:00:00:00:00:05  1      dynamic   1/1/2
00:00:00:00:00:06  2      dynamic   1/1/1
```

Showing dynamic MAC address table entries for VLAN 1:

```
switch# show mac-address-table dynamic vlan 1
MAC age-time           : 300 seconds
Number of MAC addresses : 1

MAC Address      VLAN    Type      Port
-----
00:00:00:00:00:05  1      dynamic   1/1/2
```

Showing dynamic MAC address table entries for port 1/1/1:

```
switch# show mac-address-table dynamic port 1/1/1
MAC age-time           : 300 seconds
Number of MAC addresses : 1

MAC Address      VLAN    Type      Port
-----
00:00:00:00:00:06  2      dynamic   1/1/1
```

Command History

Release	Modification
10.07 or earlier	--

Command Information

Platforms	Command context	Authority
All platforms	Operator (>) or Manager (#)	Operators or Administrators or local user group members with execution rights for this command. Operators can execute this command from the operator context (>) only.

show mac-address-table interface

show mac-address-table interface <INTERFACE>

Description

Shows the MAC address table entries for the specified interface.

Parameter	Description
<INTERFACE>	Specifies an interface or a list of interfaces on the switch.

Examples

Showing the MAC address table entries for interface **1/1/1**:

```
switch# show mac-address-table interface 1/1/1
MAC age-time           : 300 seconds
Number of MAC addresses : 1

MAC Address      VLAN    Type      Interface
-----
00:00:00:00:00:01  2      dynamic   1/1/1
```

Command History

Release	Modification
10.09	Command introduced

Command Information

Platforms	Command context	Authority
All platforms	Operator (>) or Manager (#)	Operators or Administrators or local user group members with execution rights for this command. Operators can execute this command from the operator context (>) only.

show mac-address-table lockout

show mac-address-table lockout [vsx-peer]

Description

Shows MAC lockout table information.

Parameter	Description
vsx-peer	Shows the output from the VSX peer switch. If the switches do not have the VSX configuration or the ISL is down, the output from the VSX peer switch is not displayed. This parameter is available on switches that support VSX.

Examples

```
switch# show mac-address-table lockout
Number of MAC lockout addresses :
```

2MAC Address	Type
00:00:00:00:01:10	static
00:00:00:00:10:03	static

Command History

Release	Modification
10.07 or earlier	--

Command Information

Platforms	Command context	Authority
All platforms	Operator (>) or Manager (#)	Operators or Administrators or local user group members with execution rights for this command. Operators can execute this command from the operator context (>) only.

show mac-address-table port

show mac-address-table port <PORT-NUM> [vsx-peer]

Description

Shows the MAC address table entries for the specified port.

Parameter	Description
<PORT-NUM>	Specifies a physical port on the switch. Format: member/slot/port.
vsx-peer	Shows the output from the VSX peer switch. If the switches do not have the VSX configuration or the ISL is down, the output from the VSX peer switch is not displayed. This parameter is available on switches that support VSX.

Examples

Showing the MAC address table entries for port **1/1/1**:

```
switch# show mac-address-table port 1/1/1
MAC age-time           : 300 seconds
Number of MAC addresses : 1

MAC Address      VLAN    Type      Port
-----
00:00:00:00:00:01  2      dynamic   1/1/1
```

Command History

Release	Modification
10.07 or earlier	--

Command Information

Platforms	Command context	Authority
All platforms	Operator (>) or Manager (#)	Operators or Administrators or local user group members with execution rights for this command. Operators can execute this command from the operator context (>) only.

show mac-address-table static

show mac-address-table static

Description

Shows all statically configured MAC addresses.

Examples

```
switch# show mac-address-table static
Number of MAC addresses : 2

MAC Address          VLAN    Port
-----
00:00:00:00:10:02    1       1/1/1
00:00:00:00:10:03    1       1/1/1
```

Command History

Release	Modification
10.07 or earlier	--

Command Information

Platforms	Command context	Authority
All platforms	Operator (>) or Manager (#)	Operators or Administrators or local user group members with execution rights for this command. Operators can execute this command from the operator context (>) only.

show mac-address-table vlan

show mac-address-table vlan <VLAN-ID> [vsx-peer]

Description

Shows MAC addresses learned by or configured on the specified VLAN.

Parameter	Description
vlan <VLAN-ID>	Specifies the VLAN ID.
vsx-peer	Shows the output from the VSX peer switch. If the switches do not have the VSX configuration or the ISL is down, the output from the VSX peer switch is not displayed. This parameter is available on switches that support VSX.

Examples

```
switch# show mac-address-table vlan 1
MAC age-time       : 300 seconds
Number of MAC addresses : 1
```

MAC Address	VLAN	Type	Port
00:00:00:00:00:01	1	dynamic	1/1/1

Command History

Release	Modification
10.07 or earlier	--

Command Information

Platforms	Command context	Authority
All platforms	Operator (>) or Manager (#)	Operators or Administrators or local user group members with execution rights for this command. Operators can execute this command from the operator context (>) only.

static-mac

```
static-mac <MAC-ADDR> vlan <VLAN-ID> port <PORT-NUM>
no static-mac <MAC-ADDR> vlan <VLAN-ID> port <PORT-NUM>
```

Description

Adds a static MAC address to the MAC address table and associates it with a port or existing VLAN. Static MAC addresses can only be assigned to layer 2 (non-routed) interfaces. Static MAC addresses are not affected by the MAC address aging time.

The `no` form of this command deletes a static MAC address.

Parameter	Description
<MAC-ADDR>	Specifies a MAC address (xx:xx:xx:xx:xx:xx), where x is a hexadecimal number from 0 to F.
vlan <VLAN-ID>	Specifies number of an existing VLAN.
port <PORT-NUM>	Specifies a physical port on the switch. Format: member/slot/port.

Examples

```
switch(config)# static-mac 00:00:00:00:00:01 vlan 1 port 1/1/1
switch(config)# no static-mac 00:00:00:00:00:01 vlan 1 port 1/1/1

switch(config)# static-mac 00:00:00:00:00:01 vlan 1 port 1/1/2
1/1/2 is not an L2 port
```

```
switch(config)# static-mac 00:00:00:00:00:01 vlan 2 port 1/1/1
VLAN 2 not found
```

Command History

Release	Modification
10.07 or earlier	--

Command Information

Platforms	Command context	Authority
All platforms	config	Administrators or local user group members with execution rights for this command.

VLANs are primarily used to provide network segmentation at layer 2. VLANs enable the grouping of users by logical function instead of physical location. They make managing bandwidth usage within networks possible by:

- Allowing grouping of 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

VLANs are generally assigned on an organizational basis rather than on a physical basis. For example, a network administrator could assign all workstations and servers used by a particular workgroup to the same VLAN, regardless of their physical locations.

Hosts in the same VLAN can directly communicate with one another. A router or a Layer 3 switch is required for hosts in different VLANs to communicate with one another.

VLANs help reduce bandwidth waste, improve LAN security, and enable network administrators to address issues such as scalability and network management.

Maximum VLANs allowed

Aruba Switch Series	Maximum VLANs Allowed
8400	4094

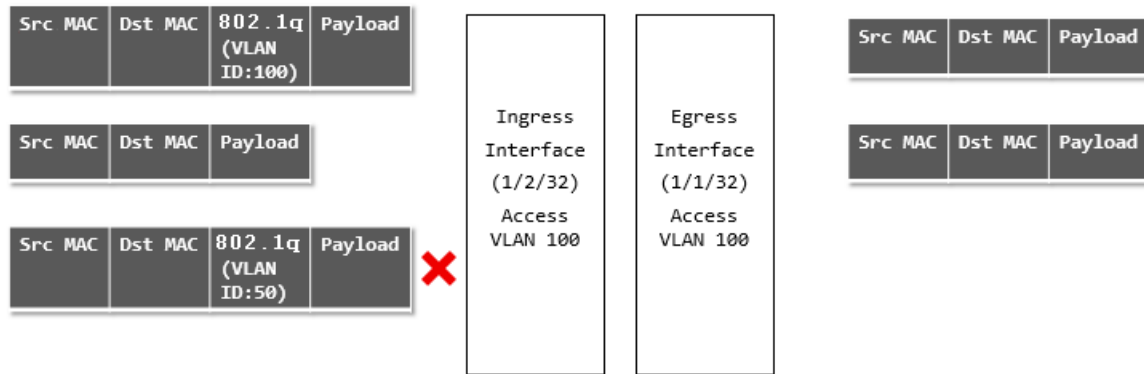
VLAN interfaces

Access interface

An access interface carries traffic for a single VLAN ID. Access interfaces are generally used to connect end devices that do not support VLANs to the network. The devices connected to an access interface are not aware of the VLAN. Access interface can carry traffic on only one VLAN, either tagged or untagged.

Example

This example shows ingress and egress traffic behavior for an access interface.



- An ingress tagged frame with VLAN ID of 100 arrives on interface 1/2/32. The switch accepts this frame and sends it to its target address on interface 1/1/32, where it egresses untagged.
- An ingress untagged frame arrives on interface 1/2/32. The switch accepts this frame and sends it to its target address on interface 1/1/32, where it egresses untagged.
- An ingress tagged frame with VLAN ID of 50 arrives on interface 1/2/32. The switch drops this frame as VLAN ID 50 is not configured on the interface.

Trunk interface

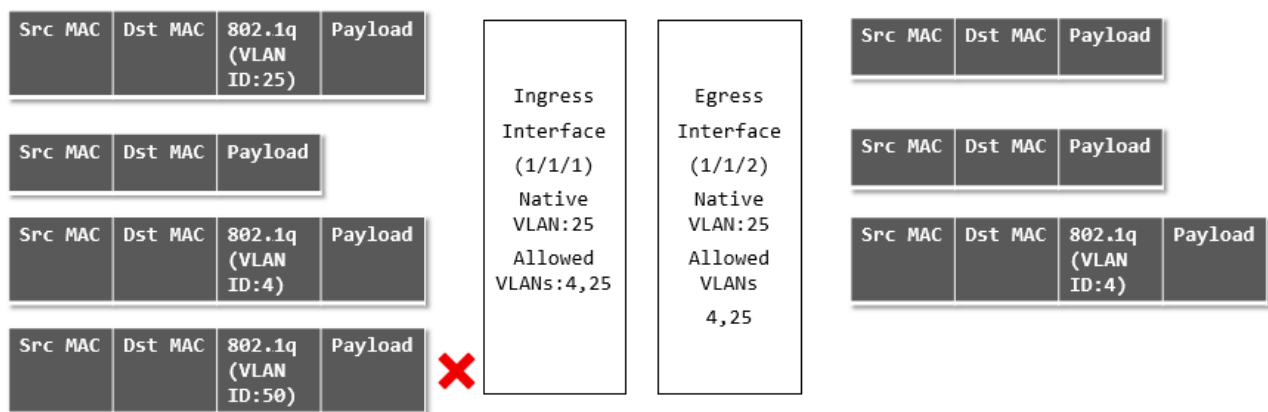
A trunk interface can carry traffic for one or more VLAN IDs. In most cases, a trunk interface is used to transport data to other switches or routers.

A trunk interface has two important settings:

- Native VLAN: This is the VLAN to which incoming untagged traffic is assigned. Only one VLAN can be assigned as the native VLAN. By default, VLAN 1 is assigned as the native VLAN for all trunk interfaces.
- Allowed VLANs: This is the list of VLANs that can be transported by the trunk. If the native VLAN is not included in the allowed list, all untagged frames that ingress on the trunk interface are dropped.

Example 1: Native untagged VLAN

This example shows ingress and egress traffic behavior when a trunk interface has a native untagged VLAN.



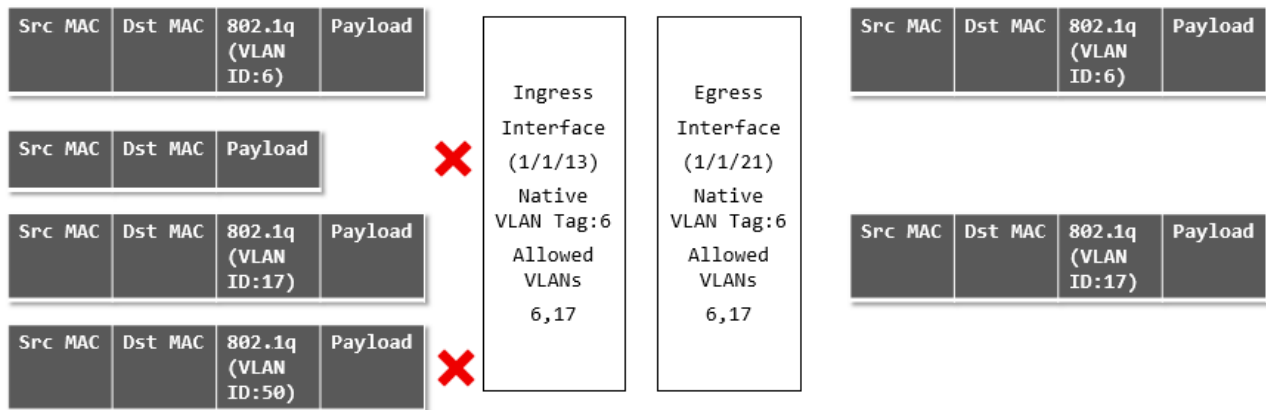
- An ingress tagged frame with VLAN ID of 25 arrives on interface 1/1/1. The switch accepts this frame and sends it to its target address on interface 1/1/2, where it egresses with a VLAN ID of 25 untagged

since port 1/1/2 is configured with a native VLAN ID of 25.

- An ingress untagged frame arrives on interface 1/1/1. The switch accepts this frame and sends it to its target address on interface 1/1/2, where it egresses with a VLAN ID of 25 untagged since port 1/1/2 is configured with a native VLAN ID of 25.
- An ingress tagged frame with VLAN ID of 4 arrives on interface 1/1/1. The switch accepts this frame and sends it to its target address on interface 1/1/2, where it egresses with a VLAN ID of 4 tagged since port 1/1/2 is configured to allow traffic with a VLAN ID of 4.
- An ingress tagged frame with VLAN ID of 50 arrives on interface 1/1/1. The switch drops this frame as VLAN ID 50 is not in the allowed list for interface 1/1/1.

Example 2: Native tagged VLAN

This example shows ingress and egress traffic behavior when a trunk interface has a native tagged VLAN.



- An ingress tagged frame with VLAN ID of 6 arrives on interface 1/1/13. The switch accepts this frame and sends it to its target address on interface 1/1/21, where it egresses with a VLAN ID of 6 tagged since port 1/1/2 is configured with a native VLAN ID of 6.
- An ingress untagged frame arrives on interface 1/1/13. The switch drops this frame since the interface is configured as native tagged (all untagged frames are dropped in such a configuration).
- An ingress tagged frame with VLAN ID of 17 arrives on interface 1/1/13. The switch accepts this frame and sends it to its target address on interface 1/1/21, where it egresses with a VLAN ID of 17 tagged since port 1/1/2 is configured to allow traffic with a VLAN ID of 17.
- An ingress tagged frame with VLAN ID of 50 arrives on interface 1/1/13. The switch drops this frame as VLAN ID 50 is not in the allowed list for interface 1/1/13.

Traffic handling summary

VLAN configuration	Ingress traffic	Egress traffic
Access interface with: VLAN ID = X	<ol style="list-style-type: none"> Untagged Tagged with VLAN ID = X Tagged with any other VLAN ID 	<ol style="list-style-type: none"> Untagged on VLAN X Untagged on VLAN X Dropped at ingress port itself
Trunk interface with: <ul style="list-style-type: none"> Untagged Native VLAN ID = 	<ol style="list-style-type: none"> Untagged Tagged with VLAN ID = X 	<ol style="list-style-type: none"> Untagged on VLAN X Untagged on VLAN X

VLAN configuration	Ingress traffic	Egress traffic
X <ul style="list-style-type: none"> Allowed VLAN IDs = X, Y, Z 	3. Tagged with VLAN ID = Y 4. Tagged with VLAN ID = Z 5. Tagged with any other VLAN ID	3. Tagged on VLAN Y 4. Tagged on VLAN Z 5. Dropped at ingress port itself
Trunk interface with: <ul style="list-style-type: none"> Untagged Native VLAN ID = X Allowed VLAN IDs = ALL 	1. Untagged 2. Tagged with VLAN ID = X 3. Tagged with a VLAN ID defined on the switch 4. Tagged with a VLAN ID not defined on the switch	1. Untagged on VLAN X 2. Untagged on VLAN X 3. Tagged on the matching VLAN 4. Dropped at ingress port itself
Trunk interface with: <ul style="list-style-type: none"> Tagged Native VLAN ID = X Allowed VLAN IDs = X, Y, Z 	1. Untagged 2. Tagged with VLAN ID = X 3. Tagged with VLAN ID = Y 4. Tagged with VLAN ID = Z 5. Tagged with any other VLAN ID	1. Dropped at ingress port itself 2. Tagged on VLAN X 3. Tagged on VLAN Y 4. Tagged on VLAN Z 5. Dropped at ingress port itself
Trunk interface with: <ul style="list-style-type: none"> Tagged Native VLAN ID = X Allowed VLAN IDs = ALL 	1. Untagged 2. Tagged with VLAN ID = X 3. Tagged with a VLAN ID defined on the switch 4. Tagged with a VLAN ID not defined on the switch	1. Dropped at ingress port itself 2. Tagged on VLAN X 3. Tagged on the matching VLAN 4. Dropped at ingress port itself
Trunk interface with: <ul style="list-style-type: none"> Untagged Native VLAN ID = A Allowed VLAN IDs = X, Y, Z 	1. Untagged 2. Tagged with VLAN ID = X 3. Tagged with VLAN ID = Y 4. Tagged with VLAN ID = Z 5. Tagged with any other VLAN ID	1. Dropped at ingress port itself 2. Tagged on VLAN X 3. Tagged on VLAN Y 4. Tagged on VLAN Z 5. Dropped at ingress port itself

Comparing VLAN commands on PVOS, Comware, and AOS-CX

The following examples compare the commands needed to implement typical VLAN configurations on different HPE products.

AOS-CX <pre>interface 1/1/1 no routing vlan trunk native 1 vlan trunk allowed 10,30,50</pre> A native VLAN must be defined on the switch. By default, this is VLAN 1. Since only VLANs 10, 30, and 50 are allowed on the trunk, all untagged traffic is dropped.	PVOS <pre>interface A1 tagged vlan 10,30,50 no untagged vlan 1</pre>	Comware <pre>Interface G1/0/1 port link type trunk port trunk permit vlan 10,30,50 port trunk pvid vlan 1</pre> PVID 1 is the default setting.
--	--	--

AOS-CX <pre>interface 1/1/1 no routing vlan trunk native 10 vlan trunk allowed 10,30,50</pre> Same as scenario 1, but allows untagged traffic on VLAN 10 as well.	PVOS Not directly supported in PVOS. Scenario 1 is a workaround if there is no need to support untagged traffic.	Comware Not directly supported in Comware. A possible workaround is: <pre>interface g1/0/1 port link-mode bridge port link-type hybrid port hybrid protocol-vlan vlan 10 port hybrid vlan 10 tagged port hybrid vlan 30 tagged port hybrid vlan 50 tagged</pre>
AOS-CX <pre>interface 1/1/1 no routing vlan trunk native 5 vlan trunk allowed 5, 10,30,50</pre> VLAN 5 must be allowed on the trunk so that untagged traffic is not dropped.	PVOS <pre>interface A1 untagged vlan 5 tagged vlan 10,30,50</pre>	Comware <pre>interface G1/0/1 Port link-mode bridge port link-type trunk port trunk pvid vlan 5 port trunk permit vlan 5,10,30,50</pre> link-mode is only needed on later Comware 7 devices. 5930 is port link-mode route by default. 5900 is bridge by default.
AOS-CX <pre>interface 1/1/1 no routing vlan access 5</pre>	PVOS <pre>interface A1 untagged vlan 5</pre>	Comware <pre>interface G1/0/0 port link-mode bridge port access vlan 5</pre>

Protocol-mapped VLANs

Protocol-mapped VLANs process traffic based on the specified protocol. An access port can be a part of multiple VLANs with only one VLAN being port-based and others being protocol-mapped VLANs.

- When protocol-mapped VLANs are configured, untagged packets that are ingressing are checked for the protocol type and switched according to the protocol-mapped VLAN configuration for that protocol on the interface.
- If there are no protocol-mapped VLANs configured, all untagged packets are switched as part of the port-based VLAN that is configured. Packets egressing on an access port have no 802.1Q header. Any packet with an 802.1Q header with a non-zero VLAN ID that ingresses on an access port is dropped, except when the VLAN specified in its 802.1Q header matches the VLAN configured on the access port.

VLAN translation

VLAN translation is used to configure a set of VLAN translation rules on an interface. Once these rules are applied, VLAN-IDs in the incoming and outgoing packets of that interface are mapped to the appropriate VLAN-IDs from the translation rules. This configuration can be used in cases where the VLAN identifiers on the frames need to be modified at the interface.

VLAN translation allows you to configure bidirectional VLAN identifier translation. This allows you to use unique VLAN identifiers internally and maintain legacy VLAN identifiers on logical interfaces. When this configuration is applied on an interface, the ingress traffic for that interface is translated from VLAN1-ID to VLAN2-ID, and the egress traffic for that interface is translated from VLAN2-ID to VLAN1-ID.

On the Aruba 8400, Switch Series, If VLAN translations is configured, VLAN trunk allowed VLANs are respected and traffic is allowed.

Maximum VLAN translation rules supported

Aruba Switch Series	Maximum VLAN Translation Rules Supported
8400	1024

Assigning a VLAN to an interface

To use a VLAN, it must be assigned to an interface on the switch. VLANs can only be assigned to non-routed (Layer 2) interfaces. All interfaces are non-routed (Layer 2) by default when created. Use `routing` and `no routing` commands to move ports between Layer 3 and Layer 2 interfaces; this makes the port an access port in VLAN 1 by default.

Assigning a VLAN ID to an access interface

Prerequisites

At least one defined VLAN.

Procedure

1. Switch to configuration context with the command `config`.
2. Switch to the interface that you want to define as an access interface with the command `interface`.
3. Disable routing with the command `no routing`.
4. Configure the access interface and assign a VLAN ID with the command `vlan access`.

Examples

This example configures interface **1/1/2** as an access interface with VLAN ID set to **20**.



The port must be an L2 port; it can be configured as an L2 port using the command `no routing`.

```
switch# config
switch(config)# vlan 20
switch(config-vlan-20)# exit
switch(config)# interface 1/1/2
switch(config-lag-if)# no routing
switch(config-if)# vlan access 20
```

This example a range of interfaces (**1/1/4-1/1/9**) as an access interface with VLAN ID set to **20**.

```
switch# config
switch(config)# vlan 20
switch(config-vlan-20)# exit
switch(config)# interface 1/1/4-1/1/9
switch(config-lag-if)# no routing
switch(config)# int 1/1/4-1/1/9
switch(config-if-<1/1/4-1/1/9>)# vlan access 20
```

This example configures LAG **1** as an access interface with VLAN ID set to **30**.

```
switch# config
switch(config)# vlan 30
switch(config-vlan-30)# exit
switch(config)# interface lag 1
switch(config-lag-if)# no shutdown
switch(config-lag-if)# no routing
switch(config-lag-if)# vlan access 30
```

Assigning a VLAN ID to a trunk interface

Prerequisites

At least one defined VLAN.

Procedure

1. Switch to configuration context with the command `config`.
2. Switch to the interface that you want to define as a trunk interface with the command `interface`.
3. Disable routing with the command `no routing`.
4. Configure the trunk interface and assign a VLAN ID with the command `vlan trunk allowed`.

Examples

This example configures interface **1/1/2** as a trunk interface allowing traffic with VLAN ID set to **20**.

```
switch# config
switch(config)# vlan 20
switch(config-vlan-20)# exit
switch(config)# interface 1/1/2
switch(config-lag-if)# no routing
switch(config-if)# vlan trunk allowed 20
```

This example configures a range of interfaces (**1/1/4-1/1/9**) as a trunk interface with VLAN ID set to **20**.

```
switch# config
switch(config)# vlan 20
switch(config-vlan-20)# exit
switch(config)# interface 1/1/4-1/1/9
switch(config-lag-if)# no routing
switch(config)# int 1/1/4-1/1/9
switch(config-if-<1/1/4-1/1/9>)# vlan trunk allowed 20
```

This example configures interface **1/1/2** as a trunk interface allowing traffic with VLAN IDs **2, 3, and 4**.

```
switch# config
switch(config)# vlan 2,3,4
switch(config)# interface 1/1/2
switch(config-lag-if)# no routing
switch(config-if)# vlan trunk allowed 2,3,4
```

This example configures interface **1/1/2** as a trunk interface allowing traffic with VLAN IDs **2 to 8**.

```
switch# config
```

```
switch(config)# vlan 2-8
switch(config)# interface 1/1/2
switch(config-lag-if)# no routing
switch(config-if)# vlan trunk allowed 2-8
```

This example configures interface **1/1/2** as a trunk interface allowing traffic with VLAN IDs **2 to 8** and **10**.

```
switch# config
switch(config)# vlan 2-8,10
switch(config)# interface 1/1/2
switch(config-lag-if)# no routing
switch(config-if)# vlan trunk allowed 2-8,10
```

This example configures interface **1/1/2** as a trunk interface allowing traffic on all configured VLAN IDs (20-100).

```
switch# config
switch(config)# vlan 20-100
switch(config)# interface 1/1/2
switch(config-lag-if)# no routing
switch(config-if)# vlan trunk allowed all
```



With trunk configuration, when native membership is not specified, the port automatically becomes a native member of VLAN 1.

Assigning a native VLAN ID to a trunk interface

Prerequisites

At least one defined VLAN.

Procedure

1. Switch to configuration context with the command `config`.
2. Switch to the trunk interface to which you want to assign the native VLAN ID with the command `interface`.
3. Disable routing with the command `no routing`.
4. Assign the native VLAN ID with the command `vlan trunk native`. If tagging is required for the native VLAN, use the command `vlan trunk native tag`.
5. Allow traffic tagged with the native VLAN ID to be transported by the trunk using the command `vlan trunk allowed`.

Example

This example assigns native VLAN ID **20** to trunk interface **1/1/2**.

```
switch# config
switch(config)# vlan 20
switch(config-vlan-20)# exit
switch(config)# interface 1/1/2
switch(config-if)# no routing
```



```
switch(config-if)# vlan trunk native 20
```

This example assigns native VLAN ID **40** to trunk interface **1/1/5**, enables tagging, and allows traffic with VLAN ID 40 to be transported by the trunk.

```
switch# config  
switch(config)# vlan 40  
switch(config-vlan-40)# exit  
switch(config)# interface 1/1/5  
switch(config-if)# no routing  
switch(config-if)# vlan trunk native 40 tag  
switch(config-if)# vlan trunk allow 40
```

VLAN numbering

VLANs are numbered in the range 1 to 4094.

By default, VLAN 1 (the default VLAN) is associated with all interfaces on the switch. VLAN 1 cannot be removed from the switch.

Configuring VLANs

Creating and enabling a VLAN

Procedure

1. Switch to the configuration context with the command `config`.
2. Create a new VLAN with the command `vlan`.

Example

This example creates **VLAN 10**. The VLAN is enabled by default.

```
switch# config  
switch(config)# vlan 10  
switch(config-vlan-10)#
```

Disabling a VLAN

Procedure

1. Switch to configuration context with the command `config`.
2. Switch to configuration context for the VLAN you want to disable with the command `vlan`.
3. Disable the VLAN with the command `shutdown`.

Example

This example disables **VLAN 10**.

```
switch(config)# config
switch(config)# vlan 10
switch(config-vlan-10)# shutdown
```

Viewing VLAN configuration information

Prerequisites

At least one defined VLAN.

Procedure

1. View a summary of VLAN configuration information with the command `show vlan summary`.
2. View VLAN configuration settings with the command `show vlan`.
3. View VLANs configured for a specific layer 2 interface with the command `show vlan port`.
4. View the commands used to configure VLAN settings with the command `show running-config interface`.

Example

This example displays a summary of all VLANs.

```
switch# show vlan summary
Number of existing VLANs      : 2
Number of static VLANs       : 2
Number of dynamic VLANs      : 0
Number of port-access VLANs  : 0
```

This example displays configuration information for all defined VLANs.

```
switch# show vlan
-----
-
VLAN  Name                Status Reason          Type      Interfaces
-----
-
1      DEFAULT_VLAN_1         up      ok              static    1/1/3-1/1/4
2      UserVLAN1              up      ok              static    1/1/1,1/1/3,1/1/5
3      UserVLAN2              up      ok              static    1/1/2-1/1/3,1/1/5-1/1/6
5      UserVLAN3              up      ok              static    1/1/3
10     TestNetwork            up      ok              static    1/1/3,1/1/5
11     VLAN11                 up      ok              static    1/1/3
12     VLAN12                 up      ok              static    1/1/3,1/1/6,lag1-lag2
13     VLAN13                 up      ok              static    1/1/3,1/1/6
14     VLAN14                 up      ok              static    1/1/3,1/1/6
20     ManagementVLAN         down    admin_down      static    1/1/3,1/1/10
```

This example displays configuration information for **VLAN 2**.

```
switch# show vlan 2
-----
-
VLAN  Name                Status Reason          Type      Interfaces
-----
-----
```

-					
2	UserVLAN1	up	ok	static	1/1/1,1/1/3,1/1/5

This example displays the VLANs configured on interface **1/1/3**.

```
switch# show vlan port 1/1/3
```

VLAN	Name	Mode	Mapping
1	DEFAULT_VLAN_1	native-untagged	port
2	UserVLAN1	trunk	port
3	UserVLAN2	trunk	port
5	UserVLAN3	trunk	port
10	TestNetwork	trunk	port
11	VLAN11	trunk	port
12	VLAN12	trunk	port
13	VLAN13	trunk	port
14	VLAN14	trunk	port
20	ManagementVLAN	trunk	port
30	VLAN30	trunk	port
40	VLAN40	trunk	port
50	VLAN50	trunk	port
100	VLAN100	trunk	port
200	VLAN200	trunk	port

This example displays VLAN configuration commands for interface **1/1/16**.

```
switch# show running-config interface 1/1/16
interface 1/1/16
  no routing
  vlan trunk native 108
  vlan trunk allowed all
  exit
```

This example displays VLAN configuration commands for a range of VLANs: **20-30**.

```
Switch(config)# vlan 20-30
```

```
Switch(config-vlan-<20-30>)# show vlan
```

VLAN	Name	Status	Reason	Type	Interfaces
1	DEFAULT_VLAN_1	down	no_member_forwarding	default	1/3/1-1/3/28,1/5/1-1/5/12, 1/6/1-1/6/12
10	VLAN10	down	no_member_port	static	
20	VLAN20	down	no_member_port	static	
21	VLAN21	down	no_member_port	static	
22	VLAN22	down	no_member_port	static	
23	VLAN23	down	no_member_port	static	
24	VLAN24	down	no_member_port	static	
25	VLAN25	down	no_member_port	static	
26	VLAN26	down	no_member_port	static	
27	VLAN27	down	no_member_port	static	
28	VLAN28	down	no_member_port	static	
29	VLAN29	down	no_member_port	static	
30	VLAN30	down	no_member_port	static	

```
6405(config-vlan-<20-30>)#
```

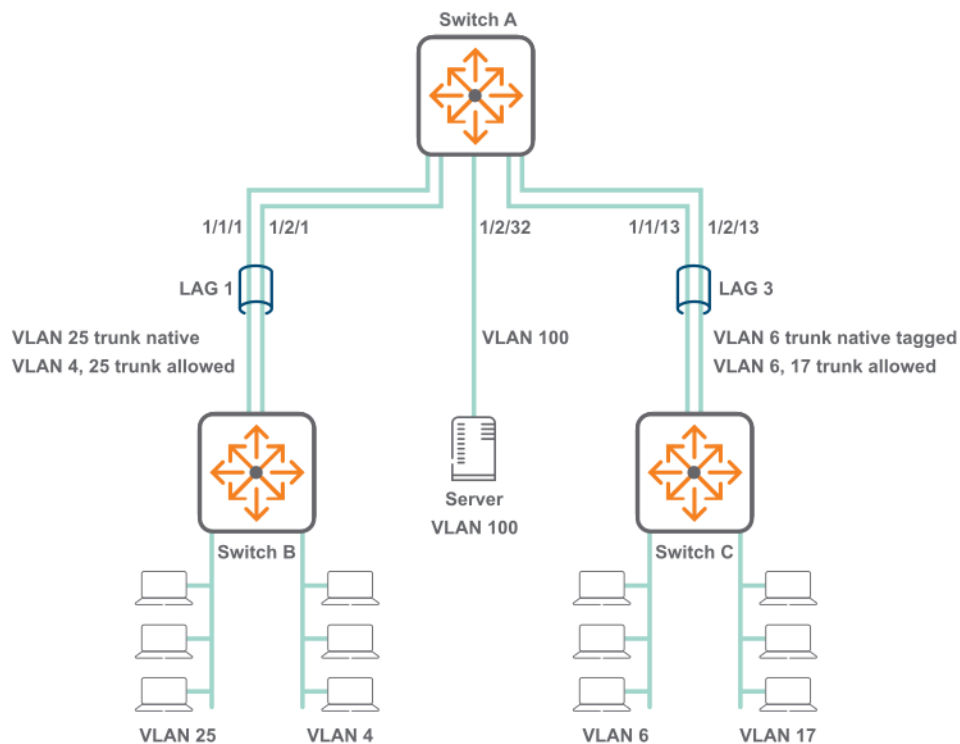
This example displays VLAN configuration commands for VLANs **15,20,25**.

```
Switch(config)# vlan 15,20,25
Switch(config-vlan-<15,20,25>)# show vlan
```

```
-----
VLAN  Name                Status  Reason                        Type         Interfaces
-----
1      DEFAULT_VLAN_1        down    no_member_forwarding         default      1/3/1-1/3/28,1/5/1-1/5/12,
                                                1/6/1-1/6/12
15     VLAN15                 down    no_member_port                static
20     VLAN20                 down    no_member_port                static
25     VLAN25                 down    no_member_port                static
switch(config-vlan-<15,20,25>)#
```

VLAN scenario

This scenario shows how to assign VLAN IDs to access and trunk interfaces for the following deployment:



In this scenario, VLANs are used to isolate the traffic from different devices.

- VLAN 25 carries tagged and untagged traffic from computers connected to switch B.
- VLAN 4 carries tagged traffic from computers connected to switch B.
- VLAN 6 carries tagged and untagged traffic from computers connected to switch C.

- VLAN 17 carries tagged traffic from computers connected to switch C.
- VLAN 100 carries tagged/untagged traffic from the server and only untagged traffic to the server.

Procedure

1. Execute the following commands on switch A and B.
 - a. Create VLANs 4 and 25.

```
switch# config  
switch(config)# vlan 4,25
```

- b. Define LAG 1 and assign the VLANs to it.

```
switch(config)# interface lag 1  
switch(config-lag-if)# no shutdown  
switch(config-lag-if)# no routing  
switch(config-lag-if)# vlan trunk native 25  
switch(config-lag-if)# vlan trunk allowed 4,25
```



`no routing` is not a mandatory command to add the interface in LAG1; when adding the interface to LAG1, the trunk or routing type is automatically assigned.

- c. Add ports **1/1/1** and **1/2/1** to LAG 1.

```
switch(config-lag-if)# interface 1/1/1  
switch(config-if)# no shutdown  
switch(config-lag-if)# no routing  
switch(config-if)# lag 1  
switch(config-if)# interface 1/2/1  
switch(config-if)# no shutdown  
switch(config-lag-if)# no routing  
switch(config-if)# lag 1
```

2. Execute the following commands on switch A and C.
 - a. Create VLANs 6 and 17.

```
switch# config  
switch(config)# vlan 6,17
```

- b. Define LAG 3 and assign the VLANs to it.

```
switch(config)# interface lag 3  
switch(config-lag-if)# no shutdown  
switch(config-lag-if)# no routing  
switch(config-lag-if)# vlan trunk native 6 tag  
switch(config-lag-if)# vlan trunk allowed 6,17
```

- c. Add ports **1/1/13** and **1/2/13** to LAG 3.

```
switch(config-lag-if)# interface 1/1/13
switch(config-if)# no shutdown
switch(config-lag-if)# no routing
switch(config-if)# lag 3
switch(config-if)# interface 1/2/13
switch(config-if)# no shutdown
switch(config-if)# no routing
switch(config-if)# lag 3
```

3. Execute the following commands on switch A to configure the connection to the server. Configure interface **1/2/13** as an access interface with VLAN ID set to 100.

```
switch# config
switch (config)# vlan 100
switch(config-vlan-100)# interface 1/2/32
switch(config-if)# no shutdown
switch(config-lag-if)# no routing
switch(config-if)# vlan access 100
switch(config-if)# exit
```

4. Verify VLAN configuration by running the command `show vlan`. For example:

```
switch# show vlan
```

VLAN	Name	Status	Reason	Type	Interfaces
1	DEFAULT_VLAN_1	down	no_member_port	default	
4	VLAN4	up	ok	static	lag1
6	VLAN6	up	ok	static	lag3
17	VLAN17	up	ok	static	lag3
25	VLAN25	up	ok	static	lag1
100	VLAN100	up	ok	static	1/2/32

5. Verify that the connection to the DHCP server is sending/receiving data with the command `show interface`. Check that the **Rx** and **Tx** fields are incrementing. For example:

```
switch# show interface 1/2/32
Interface 1/2/32 is up
Admin state is up
Description:
Hardware: Ethernet, MAC Address: 70:72:cf:3a:8a:0b
MTU 1500
Type SFP+LR
qos trust none
Speed 10000 Mb/s
Auto-Negotiation is off
Input flow-control is off, output flow-control is off
VLAN Mode: access
Access VLAN: 100

Rx
```

	20 input packets	1280 bytes
	0 input error	0 dropped
	0 CRC/FCS	
Tx	9 output packets	1054 bytes
	0 input error	0 dropped
	0 collision	

6. Verify LAG interface configuration with the command `show interface`. Check the fields admin state, MAC address, Aggregated-interfaces, VLAN Mode, Native VLAN, Allowed VLAN, Rx count, and Tx count. For example:

```
switch# show interface lag1
Aggregate-name lag1
Description :
Admin state      : up
MAC Address      : 94:f1:28:21:63:00
Aggregated-interfaces : 1/1/1 1/2/1
Aggregation-key  : 1
Speed           : 1000 Mb/s
L3 Counters: Rx Disabled, Tx Disabled
qos trust none
VLAN Mode: native-untagged
Native VLAN: 25
Allowed VLAN List: 4,25
Rx
    10 input packets      1280 bytes
    0 input error         0 dropped
    0 CRC/FCS
Tx
    8 output packets      980 bytes
    0 input error         0 dropped
    0 collision
```

```
switch# show interface lag3
Aggregate-name lag3
Description :
Admin state      : up
MAC Address      : 94:f1:28:21:63:00
Aggregated-interfaces : 1/1/13 1/2/13
Aggregation-key  : 3
Speed 1000 Mb/s
L3 Counters: Rx Disabled, Tx Disabled
qos trust none
VLAN Mode: native-tagged
Native VLAN: 6
Allowed VLAN List: 6,17
Rx
    19 input packets      1280 bytes
    0 input error         0 dropped
    0 CRC/FCS
Tx
    15 output packets     1000 bytes
    0 input error         0 dropped
0 Collision
```

- a. To check just the LAG interface statistics, you can use the `show interface lag 1 statistics` command:



The following output has been truncated for display purposes and appears differently on the switch.

```
switch(config-if)# sho interface lag1 statistics
-----
Interface                RX      RX      RX
                        Bytes  Packets  Drops
-----
1/1/40  - lag1  1663368814276  3249823417  0
lag1      1663368814276  3249823417  0
-----

Interface                TX      TX      TX
                        Bytes  Packets  Drops
-----
1/1/40  - lag1  2134926620343  4506158466  50555880
lag1      2134926620343  4506158466  50555880
-----

Interface                RX      RX      TX      TX      RX      TX
                        Broadcast Multicast Broadcast Multicast Pause
-----
1/1/40  - lag1      12823      629874  204989954  185789535  0
lag1      12823      629874  204989954  185789535
-----
```

7. Verify the physical interfaces (1/1/1, 1/2/1, 1/1/13, 1/2/13) with the command `show interface`. Check that the **Rx** and **Tx** fields are incrementing. For example:

```
switch# show interface 1/1/1
Interface 1/1/1 is up
Admin state is up
Description:
Hardware: Ethernet, MAC Address: 94:f1:28:21:73:ff
MTU 1500
Type SFP+LR
qos trust none
Speed 1000 Mb/s
Auto-Negotiation is off
Input flow-control is off, output flow-control is off
Rx
    6 input packets          620 bytes
    0 input error            0 dropped
    0 CRC/FCS
Tx
    4 output packets         422 bytes
    0 input error            0 dropped
0 collision
```


8. Verify the lag 1 interface with the command `show running-config`. For example:

```
switch# show running-config interface lag 1
...
vlan 1
vlan 2
    name UserVLAN1
vlan 3
    name UserVLAN2
vlan 5
    name UserVLAN3
vlan 10
    name TestNetwork
    voice
    description This is a test only VLAN
vlan 11-14
vlan 20
    name ManagementVLAN
    shutdown
vlan 30,40,50,100,200
trunk-dynamic-vlan-include
interface lag 1
    no shutdown
    no routing
    vlan access 12
interface lag 2
    no shutdown
    no routing
    vlan access 12
interface 1/1/1
    no shutdown
    no routing
    vlan protocol arp 3
    vlan protocol ipv4 3
    vlan protocol ipv6 5
    vlan access 2
interface 1/1/2
    no shutdown
    no routing
    vlan access 3
interface 1/1/3
    no shutdown
    no routing
    vlan trunk native 1
    vlan trunk allowed all
interface 1/1/4
    no shutdown
    no routing
    vlan access 1
interface 1/1/5
    no shutdown
    no routing
    vlan trunk native 1
    vlan trunk allowed 2-3,10,20,30,40,50,100
    vlan translate 10 20
    vlan translate 30 40
    vlan translate 50 100
interface 1/1/6
    no shutdown
    no routing
    vlan trunk native 1 tag
    vlan trunk allowed 3,12-14,100,200
```

```

    vlan translate 100 200
interface 1/1/10
    no shutdown
    no routing
    vlan access 20
interface 1/1/11
    no shutdown
    lag 1
interface 1/1/12
    no shutdown
    lag 2
...

```

VLAN commands

description

description <DESCRIPTION>

Description

Specifies a descriptive for a VLAN.

Parameter	Description
<DESCRIPTION>	Specifies a description for the VLAN.

Examples

Assigning a description to VLAN 20:

```

switch(config)# vlan 20
switch(config-vlan-20)# description primary

```

Command History

Release	Modification
10.07 or earlier	--

Command Information

Platforms	Command context	Authority
All platforms	config-vlan-<VLAN-ID>	Administrators or local user group members with execution rights for this command.

vlan name

name <VLAN-NAME>

Description

Associates a name with a VLAN.

Parameter	Description
<VLAN-NAME>	Specifies a name for a VLAN. Length: 1 to 32 alphanumeric characters, including underscore (_) and hyphen (-).

Usage

- Each named VLAN must have a unique name; there cannot be duplicate names for VLANs.
- By default, VLANs are created with the default name: VLAN <VLAN-ID>

Examples

Assigning the name **backup** to VLAN 20:

```
switch(config)# vlan 20
switch(config-vlan-20)# name backup
```

Command History

Release	Modification
10.07 or earlier	--

Command Information

Platforms	Command context	Authority
All platforms	config-vlan-<VLAN-ID>	Administrators or local user group members with execution rights for this command.

show capacities-status vlan-count

```
show capacities-status vlan-count
```

Description

Shows the number of VLANs present on the switch and the maximum number of VLANs allowed on the switch.

Example

Showing switch VLAN capacity status:

```
show capswitch# show capacities-status vlan-count
System Capacities: Filter VLAN count
Capacities Name                                     Value    Maximum
-----
Maximum number of VLANs currently configured         1         xxxx
```

Command History

Release	Modification
10.07 or earlier	--

Command Information

Platforms	Command context	Authority
All platforms	Manager (#)	Administrators or local user group members with execution rights for this command.

show capacities svi-count

show capacities svi-count

Description

Shows the maximum number of SVIs supported by the switch.

Examples

Showing switch SVI capacity:

```
switch# show capacities svi-count
System Capacities: Filter SVI count
Capacities Name                                     Value
-----
Maximum number of SVIs supported in the system      128
```

Showing switch SVI capacity:

```
switch# show capacities svi-count
System Capacities: Filter SVI count
Capacities Name                                     Value
-----
Maximum number of SVIs supported in the system      494
```

Command History

Release	Modification
10.07 or earlier	--

Command Information

Platforms	Command context	Authority
All platforms	Manager (#)	Administrators or local user group members with execution rights for this command.

show capacities vlan-count

show capacities vlan-count

Description

Shows the maximum number of VLANs allowed on the switch.

Example

Showing switch VLAN capacity:

```
show capswitch# show capacities vlan-count
System Capacities: Filter VLAN count
Capacities Name                                     Value
-----
Maximum number of VLANs supported in the system      4094
```

Command History

Release	Modification
10.07 or earlier	--

Command Information

Platforms	Command context	Authority
All platforms	Manager (#)	Administrators or local user group members with execution rights for this command.

show vlan

```
show vlan [<VLAN-ID>] [vsx-peer]
```

Description

Displays configuration information for all VLANs or a specific VLAN.

Parameter	Description
<VLAN-ID>	Specifies a VLAN ID.
vsx-peer	Shows the output from the VSX peer switch. If the switches do not have the VSX configuration or the ISL is down, the output from the VSX peer switch is not displayed. This parameter is available on switches that support VSX.

Examples

Displaying configuration information for VLAN 2:

```
switch# show vlan 2
-----
VLAN  Name                Status Reason                Type    Interfaces
-----
2      UserVLAN1              up    ok                    static  1/1/1,1/1/3,1/1/5
```

Displaying configuration information for all defined VLANs:

```
switch# show vlan
```

VLAN	Name	Status	Reason	Type	Interfaces
-					
1	DEFAULT_VLAN_1	up	ok	static	1/1/3-1/1/4
2	UserVLAN1	up	ok	static	1/1/1,1/1/3,1/1/5
3	UserVLAN2	up	ok	static	1/1/2-1/1/3,1/1/5-1/1/6
5	UserVLAN3	up	ok	static	1/1/3
10	TestNetwork	up	ok	static	1/1/3,1/1/5
11	VLAN11	up	ok	static	1/1/3
12	VLAN12	up	ok	static	1/1/3,1/1/6,lag1-lag2
13	VLAN13	up	ok	static	1/1/3,1/1/6
14	VLAN14	up	ok	static	1/1/3,1/1/6
20	ManagementVLAN	down	admin_down	static	1/1/3,1/1/10

Command History

Release	Modification
10.07 or earlier	--

Command Information

Platforms	Command context	Authority
All platforms	Manager (#)	Operators or Administrators or local user group members with execution rights for this command. Operators can execute this command from the operator context (>) only.

show vlan port

```
show vlan port <INTERFACE-ID>
```

Description

Displays the VLANs configured for a specific layer 2 interface.

Parameter	Description
<INTERFACE-ID>	Specifies an interface ID. Format: member/slot/port.
vsx-peer	Shows the output from the VSX peer switch. If the switches do not have the VSX configuration or the ISL is down, the output from the VSX peer switch is not displayed. This parameter is available on switches that support VSX.

Examples

Displaying the VLANs configured on interface **1/1/1**:

```
switch# show vlan port 1/1/1
```

VLAN	Name	Mode	Mapping
------	------	------	---------

2	UserVLAN1	access	port
3	UserVLAN2	access	arp, ipv4
5	UserVLAN5	access	ipv6

Displaying the VLANs configured on interface **1/1/3**:

```
switch# show vlan port 1/1/3
```

VLAN	Name	Mode	Mapping
1	DEFAULT_VLAN_1	native-untagged	port
2	UserVLAN1	trunk	port
3	UserVLAN2	trunk	port
5	UserVLAN3	trunk	port
10	TestNetwork	trunk	port
11	VLAN11	trunk	port
12	VLAN12	trunk	port
13	VLAN13	trunk	port
14	VLAN14	trunk	port
20	ManagementVLAN	trunk	port
30	VLAN30	trunk	port
40	VLAN40	trunk	port
50	VLAN50	trunk	port
100	VLAN100	trunk	port
200	VLAN200	trunk	port

Command History

Release	Modification
10.07 or earlier	--

Command Information

Platforms	Command context	Authority
All platforms	Manager (#)	Operators or Administrators or local user group members with execution rights for this command. Operators can execute this command from the operator context (>) only.

show vlan summary

```
show vlan summary [vsx-peer]
```

Description

Displays a summary of the VLAN configuration on the switch.

Parameter	Description
vsx-peer	Shows the output from the VSX peer switch. If the switches do not have the VSX configuration or the ISL is down, the output from the VSX peer switch is not displayed. This parameter is available on switches that support VSX.

Examples

Displaying a summary of the VLAN configuration on the switch:

```
switch# show vlan summary
Number of existing VLANs: 11
Number of static VLANs: 11
Number of dynamic VLANs: 0
```

Command History

Release	Modification
10.07 or earlier	--

Command Information

Platforms	Command context	Authority
All platforms	Manager (#)	Operators or Administrators or local user group members with execution rights for this command. Operators can execute this command from the operator context (>) only.

show vlan translation

```
show vlan translation [interface <INTERFACE-NAME>] [vsx-peer]
```

Description

Shows a summary of all VLAN translations rules defined on the switch, or the rules defined for a specific interface.

Parameter	Description
interface <INTERFACE-NAME>	Specifies the name of a layer 2 interface. Format: member/slot/port.
vsx-peer	Shows the output from the VSX peer switch. If the switches do not have the VSX configuration or the ISL is down, the output from the VSX peer switch is not displayed. This parameter is available on switches that support VSX.

Examples

Displaying a summary of all VLAN translations rules defined on the switch:

```
switch# show vlan translation
-----
Interface  VLAN-1      VLAN-2
-----
1/1/5      10          20
1/1/5      30          40
1/1/5      50          100
1/1/6      100         200

Total number of translation rules : 4
```


Displaying a summary of all VLAN translations rules defined on interface **1/1/5**:

```
switch# show vlan translation interface 1/1/5
-----
Interface  VLAN-1      VLAN-2
-----
1/1/5      10         20
1/1/5      30         40
1/1/5      50         100
```

Displaying VLAN translation information when VSX peer is configured:

```
switch(config-if)# show vlan translation vsx-peer
-----
Interface  VLAN-1  VLAN-2
-----
1/3/1      10      20
Total number of translation rules : 1
```

Displaying VLAN translation information when VSX peer is not configured:

```
switch(config-if)# show vlan translation vsx-peer
VSX is not configured
```

Command History

Release	Modification
10.07 or earlier	--

Command Information

Platforms	Command context	Authority
8400	Manager (#)	Operators or Administrators or local user group members with execution rights for this command. Operators can execute this command from the operator context (>) only.

show vlan translation pending

show vlan translation pending

Description

Shows a list of pending VLAN translation rules.

Examples

Displaying a list of VLAN translations rules pending on the switch:

```
switch# show vlan translation pending
-----
Interface  VLAN-1      VLAN-2
```

```

-----
1/1/5      10      20
1/1/5      30      40
1/1/5      50     100
1/1/6     100     200

```

Total number of VLAN translation rules that are pending: 4

Displaying the output when there are no VLAN translation rules in the pending list:

```

switch# show vlan translation interface 1/1/5
No pending VLAN translation rules

```

Command History

Release	Modification
10.08 or earlier	--

Command Information

Platforms	Command context	Authority
8400	Manager (#)	Operators or Administrators or local user group members with execution rights for this command. Operators can execute this command from the operator context (>) only.

show vlan voice

show vlan voice

Description

Displays the voice VLAN list showing the VLAN ID, name, operational state of the VLAN, and the interfaces associated with the VLAN.

Example

Displaying the voice VLANs list :

```

switch# show vlan voice
-----
VLAN  Name                               Status      Type      Interfaces
-----
10    TestNetwork                           up          static
1/1/3,1/1/5

```

Displaying the information when voice VLANs are not configured:

```

switch# show vlan voice
Voice VLAN not configured

```

Command History

Release	Modification
10.07 or earlier	--

Command Information

Platforms	Command context	Authority
All platforms	Manager (#)	Operators or Administrators or local user group members with execution rights for this command. Operators can execute this command from the operator context (>) only.

shutdown

shutdown
no shutdown

Description

Disables a VLAN. (By default, a VLAN is automatically enabled when it is created with the `vlan` command.)

The `no` form of this command enables a VLAN.

Examples

Enabling VLAN 20:

```
switch(config)# vlan 20  
switch(config-vlan-20)# no shutdown
```

Disabling VLAN 20:

```
switch(config)# vlan 20  
switch(config-vlan-20)# shutdown
```

Command History

Release	Modification
10.07 or earlier	--

Command Information

Platforms	Command context	Authority
All platforms	<code>config-vlan-<VLAN-ID></code>	Administrators or local user group members with execution rights for this command.

trunk-dynamic-vlan-include

trunk-dynamic-vlan-include

```
no trunk-dynamic-vlan-include
```

Description

Indicates if dynamically learned VLANs from MVRP and port-access should be included or excluded on ports configured with `vlan trunk allowed all`. By default, dynamic VLANs are not included in the trunk allowed list. This command is used at the system-level.

The `no` form of this command disables the inclusion of dynamic VLANs in the VLANs table. This is the default.

Examples

Including the dynamic VLANs in the VLAN table:

```
switch(config)# trunk-dynamic-vlan-include
```

Disabling the inclusion of dynamic VLANs in the VLAN table (default):

```
switch(config)# no trunk-dynamic-vlan-include
```

Command History

Release	Modification
10.08	Command introduced

Command Information

Platforms	Command context	Authority
All platforms	<code>config</code>	Administrators or local user group members with execution rights for this command.

vlan

```
vlan <VLAN-LIST>  
no vlan <VLAN-LIST>
```

Description

Creates a VLAN and changes to the `config-vlan-id` context for the VLAN. By default, the VLAN is enabled. To disable a VLAN, use the `shutdown` command.

If the specified VLAN exists, this command changes to the `config-vlan-id` context for the VLAN. If a range of VLANs is specified, the context does not change.



VLANs used for internal purposes using the command `system internal vlan range` cannot be used for any other (L2) purposes.

The `no` form of this command removes a VLAN. VLAN 1 is the default VLAN and cannot be deleted.

Parameter	Description
<VLAN-LIST>	Specifies a single ID, or a series of IDs separated by commas (2, 3, 4), dashes (2-4), or both (2-4,6). Range: 1 to 4094.

Examples

Creating VLAN **20**:

```
switch(config)# vlan 20
switch(config-vlan-20)#
```

Removing VLAN **20**:

```
switch(config)# no vlan 20
```

Creating VLANs **2 to 8** and **10**:

```
switch(config)# vlan 2-8,10
```

Removing VLANs **2 to 8** and **10**:

```
switch(config)# no vlan 2-8,10
```

Creating a VLAN which is already configured as an internal VLAN:

```
switch(config)# vlan 3001
Ignoring the operation on internal VLAN(s) 3001.
```

Deleting an unconfigured VLAN which is already configured as internal VLAN:

```
switch(config)# no vlan 300
Ignoring the operation for non-configured VLAN(s) 300.
```

Command History

Release	Modification
10.07 or earlier	--

Command Information

Platforms	Command context	Authority
All platforms	config	Administrators or local user group members with execution rights for this command.

vlan access

```
vlan access <VLAN-ID>
no vlan access [<VLAN-ID>]
```

Description

Creates an access interface and assigns an VLAN ID to it. Only one VLAN ID can be assigned to each access interface.

VLANs can only be assigned to non-routed (Layer 2) interfaces. All interfaces are non-routed (Layer 2) by default when created. Use `routing` and `no routing` commands to move ports between Layer 3 and Layer 2 interfaces.

The `no` form of this command removes an access VLAN from the interface in the current context and sets it to the default VLAN ID of 1.

Command context

Parameter	Description
<VLAN-ID>	Specifies a single ID, or a series of IDs separated by commas (2, 3, 4), dashes (2-4), or both (2-4,6). Range: 1 to 4094.

Examples

Configuring interface **1/1/2** as an access interface with VLAN ID set to **20**:

```
switch(config)# interface 1/1/2
switch(config-if)# no routing
switch(config-if)# vlan access 20
```

Removing VLAN ID **20** from interface **1/1/2**:

```
switch(config)# interface 1/1/2
switch(config-if)# no vlan access 20
```

or:

```
switch(config)# interface 1/1/2
switch(config-if)# no vlan access
```

Command History

Release	Modification
10.07 or earlier	--

Command Information

Platforms	Command context	Authority
All platforms	config-if	Administrators or local user group members with execution rights for this command.

vlan protocol

```
vlan protocol <PROTOCOL_NAME> <VLAN-ID>  
no vlan protocol <PROTOCOL_NAME> <VLAN-ID>
```

Description

Adds protocol mapping to a VLAN on an interface.

The `no` form of this command removes protocol mapping from the VLAN on an interface.

Parameter	Description
<VLAN-ID>	Specifies a VLAN ID. Range: 2 to 4094.
<PROTOCOL_NAME>	Specifies the protocol that the VLAN is bound to for a given interface. Options are: AppleTalk, ARP, IPv4, IPv6, IPX, NetBEUI, and SNA.

Usage

- This command is only applicable to access ports.
- Protocol VLAN should be different from access VLANs.
- VLAN should be configured on the switch.
- Routing must be disabled on the interface.
- Interface must be a physical or LAG interface.
- The same protocol-mapped VLAN is recommended for ARP and IPv4 protocols to avoid IPv4 traffic loss.

Examples

Assigning a protocol mapping to a VLAN on an interface:

```
switch(config)# interface 1/1/2  
switch(config-if)# vlan protocol ipv4 10
```

Assigning a protocol mapping to a VLAN on a LAG interface:

```
switch(config)# interface lag 2  
switch(config-lag-if)# vlan protocol ipv6 10
```

Removing a protocol mapping from a VLAN on an interface:

```
switch(config)# interface 1/1/2  
switch(config-if)# no vlan protocol ipv6 10
```

Removing a protocol mapping from a VLAN on a LAG interface:

```
switch(config)# interface lag 2  
switch(config-lag-if)# no vlan protocol ipv6 10
```

Command History

Release	Modification
10.07 or earlier	--

Command Information

Platforms	Command context	Authority
8400	config-if config-lag-if	Administrators or local user group members with execution rights for this command.

vlan translate

```
vlan translate <VLAN-1> <VLAN-2>
no vlan translate <VLAN-1> <VLAN-2>
```

Description

Defines a bidirectional VLAN translation rule that maps an original VLAN ID (VLAN-1) to a translated internal VLAN ID (VLAN-2) on a LAG or layer 2 interface. Applies to both incoming and outgoing traffic. The `no` form of this command removes an existing VLAN translation rule on the current interface.

VLAN translation and MVRP cannot be enabled on the same interface.

A translated VLAN must be present on the switch before the rule is created; the original VLAN need not be present.



Parameter	Description
<VLAN-1>	Specifies the number of an origin VLAN. Range: 1 - 4094.
<VLAN-2>	Specifies the number of a translated VLAN. Range: 1 - 4094.

Usage

- This configuration can be applied only on layer 2 trunk ports.
- Routing must be disabled on the interface.
- Interface must be a layer 2 physical or LAG interface.
- Maximum unique VLAN translation rules supported on the Aruba 8400 Switch Series—1024

Examples

Translates origin VLAN **200** to translated VLAN **20** on interface **1/1/2**.

```
switch# config
switch(config)# vlan 20
switch(config-vlan-20)# exit
switch(config)# interface 1/1/2
switch(config-if)# no routing
switch(config-if)# vlan trunk allowed 20
switch(config-if)# vlan translate 200 20
```


Translates origin VLANs **100** and **300** to translated VLANs **10** and **20** on interface **1/1/2**.

```
switch# config
switch(config)# vlan 10,30
switch(config-vlan-20)# exit
switch(config)# interface 1/1/2
switch(config-if)# no routing
switch(config-if)# vlan trunk allowed 10,30
switch(config-if)# vlan translate 100 10
switch(config-if)# vlan translate 300 30
```

Command History

Release	Modification
10.07 or earlier	--

Command Information

Platforms	Command context	Authority
8400	config-if	Administrators or local user group members with execution rights for this command.

vlan trunk allowed

```
vlan trunk allowed [<VLAN-LIST> | all]
no vlan trunk allowed [<VLAN-LIST>]
```

Description

Assigns a VLAN ID to an trunk interface. Multiple VLAN IDs can be assigned to a trunk interface. These VLAN IDs define which VLAN traffic is allowed across the trunk interface.

VLANs can only be assigned to non-routed (Layer 2) interfaces. All interfaces are non-routed (Layer 2) by default when created. Use `routing` and `no routing` commands to move ports between Layer 3 and Layer 2 interfaces.

The `no` form of this command removes one or more VLAN IDs from a trunk interface. When the last VLAN is removed from a trunk interface, the interface continues to operate in trunk mode, and will trunk all the VLANs currently defined on the switch, and any new VLANs defined in the future. To disable the trunk interface, use the command `shutdown`.

Parameter	Description
<VLAN-LIST>	Specifies a single ID, or a series of IDs separated by commas (2, 3, 4), dashes (2-4), or both (2-4,6). Range: 1 to 4094.
all	Configures the trunk interface to allow all the VLANs currently configured on the switch and any new VLANs that are configured in the future.

Examples

Assigning VLANs **2**, **3**, and **4** to trunk interface **1/1/2**:

```
switch(config)# interface 1/1/2
switch(config-if)# no routing
switch(config-if)# vlan trunk allowed 2,3,4
```

Assigning VLAN IDs **2** to **8** to trunk interface **1/1/2**:

```
switch(config)# interface 1/1/2
switch(config-if)# no routing
switch(config-if)# vlan trunk allowed 2-8
```

Assigning VLAN IDs **2** to **8** and **10** to trunk interface **1/1/2**:

```
switch(config)# interface 1/1/2
switch(config-if)# no routing
switch(config-if)# vlan trunk allowed 2-8,10
```

Removing VLAN IDs **2**, **3**, and **4** from trunk interface **1/1/2**:

```
switch(config)# interface 1/1/2
switch(config-if)# no vlan trunk allowed 2,3,4
```

Removing all VLANs assigned to trunk interface **1/1/2**:

```
switch(config)# interface 1/1/2
switch(config-if)# no vlan trunk allowed 2
```

Command History

Release	Modification
10.07 or earlier	--

Command Information

Platforms	Command context	Authority
All platforms	config-if	Administrators or local user group members with execution rights for this command.

vlan trunk native

```
vlan trunk native <VLAN-ID>
no vlan trunk native [<VLAN-ID>]
```

Description

Assigns a native VLAN ID to a trunk interface. By default, VLAN ID 1 is assigned as the native VLAN ID for all trunk interfaces. VLANs can only be assigned to a non-routed (layer 2) interface or LAG interface. Only one VLAN ID can be assigned as the native VLAN.



When a native VLAN is defined, the switch automatically executes the `vlan trunk allowed all` command to ensure that the default VLAN is allowed on the trunk. To only allow specific VLANs on the trunk, issue the `vlan trunk allowed` command specifying only specific VLANs.

The `no` form of this command removes a native VLAN from a trunk interface and assigns VLAN ID 1 as its native VLAN.

Parameter	Description
<VLAN-ID>	Specifies a VLAN ID. Range: 1 to 4094.

Examples

Assigning native VLAN ID **20** to trunk interface **1/1/2**:

```
switch(config)# interface 1/1/2
switch(config-if)# no routing
switch(config-if)# vlan trunk native 20
```

Removing native VLAN **20** from trunk interface **1/1/2** and returning to the default VLAN 1 as the native VLAN.

```
switch(config)# interface 1/1/2
switch(config-if)# no vlan trunk native 20
```

or:

```
switch(config)# interface 1/1/2
switch(config-if)# no vlan trunk native
```

Assigning native VLAN ID **20** to trunk interface **1/1/2** and then removing it from the list of allowed VLANs. (Only allow VLAN 10 on the trunk.)

```
switch(config)# interface 1/1/2
switch(config-if)# no routing
switch(config-if)# vlan trunk native 20
switch(config-if)# vlan trunk allowed 10
```

Command History

Release	Modification
10.07 or earlier	--

Command Information

Platforms	Command context	Authority
All platforms	config-if	Administrators or local user group members with execution rights for this command.

vlan trunk native tag

vlan trunk native <VLAN-ID> tag
no vlan trunk native <VLAN-ID> tag

Description

Enables tagging on a native VLAN. Only incoming packets that are tagged with the matching VLAN ID are accepted. Incoming packets that are untagged are dropped except for BPDUs. Egress packets are tagged.

The **no** form of this command removes tagging on a native VLAN.

Parameter	Description
<VLAN-ID>	Specifies the number of a VLAN. Range: 1 to 4094.

Examples

Enabling tagging on native VLAN **20** on trunk interface **1/1/2**:

```
switch(config)# interface 1/1/2
switch(config-if)# no routing
switch(config-if)# vlan trunk native 20
switch(config-if)# vlan trunk native 20 tag
```

Removing tagging on native VLAN **20** assigned to trunk interface **1/1/2**:

```
switch(config)# interface 1/1/2
switch(config-if)# no vlan trunk native 20 tag
```

Enabling tagging on native VLAN **20** assigned to LAG trunk interface **2**:

```
switch(config)# interface lag 2
switch(config-if)# no routing
switch(config-lag-if)# vlan trunk native 20
switch(config-lag-if)# vlan trunk native 20 tag
```

Command History

Release	Modification
10.07 or earlier	--

Command Information

Platforms	Command context	Authority
All platforms	config-if	Administrators or local user group members with execution rights for this command.

voice

voice
no voice

Description

Configures a VLAN as a voice VLAN.

The `no` form of this command removes voice configuration from a VLAN.

Examples

Configuring VLAN 10 as a voice VLAN:

```
switch(config)# vlan 10
switch(config-vlan-10)# voice
```

Removing voice from VLAN 10:

```
switch(config-vlan-10)# no voice
```

Command History

Release	Modification
10.07 or earlier	--

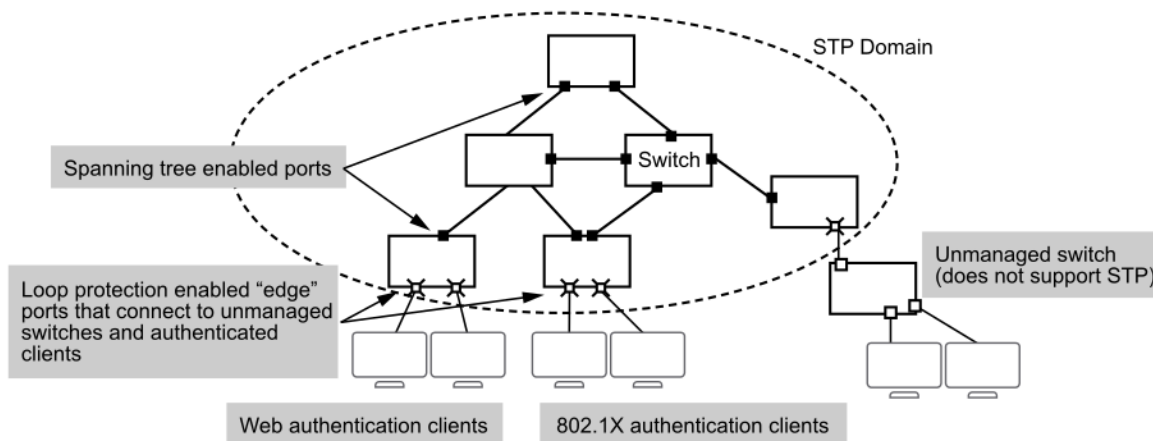
Command Information

Platforms	Command context	Authority
All platforms	config-vlan- <i><VLAN-ID></i>	Administrators or local user group members with execution rights for this command.

In cases where spanning tree protocols cannot be used to prevent loops at the edge of the network, loop protection may provide a suitable alternative. Loop protection can find loops in untagged layer 2 links, as well as on tagged VLANs.

The cases where loop protection might be chosen ahead of spanning tree to detect and prevent loops are:

- On ports with client authentication: When spanning tree is enabled on a switch that uses 802.1X, web authentication, or 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 finds loops by sending loop protection packets on each port, LAG, or VLAN on which loop protection is enabled. If a loop protection packet is received by the same switch that sent it, it indicates a loop exists and one of the following actions is taken:

- Discovery of the loop is logged but port states are not changed.
- The sending port is disabled.
- The sending and receiving ports are both disabled.

Interaction with other protocols

- When loop protection is enabled before STP, and if there is an L2 loop, then the loop will be detected and the port will be disabled.

- When STP is enabled before loop protection, and if there is a L2 loop, then the port will be moved to the **blocked** state by STP. When a port is blocked, the loop protection packet will not reach the sending switch, and the loop will not be detected by loop protection. When multiple instances of STP are configured and different spanning trees are formed for different instances, the PSPO state will be **forwarding**. In this case, loop- protection will consider those ports as normal forwarding ports and will override the STP states.
- STP is mutually exclusive with loop protection. If STP and loop protection are both enabled on the same VLAN, STP takes precedence. This means that loop protection does not take any action on a port blocked by STP.
- MVRP and the loop protection interoperate with each other. However, dynamic VLANs cannot be tagged to a port through user configuration. Therefore, it is not possible to configure a dynamic VLAN as a loop protection enabled VLAN.
- If MCLAG has marked a port as transmit disable (`mclag_pdu_tx_disable` is set to true), then loop-protect will not transmit packets on the port. Similarly, if the `loop_detect_source` column is set to `mclag` then loop protection will not re-enable the port when the re-enable timer expires on that port.

Configuring loop protection

Procedure

1. Enable loop protection on each layer 2 interface (port, LAG, or VLAN) for which loop protection is needed, with the commands `loop-protect` and `loop-protect vlan`.
2. Define the action to be taken when a loop is detected with the command `loop-protect action`. The default action is `tx-disable`, which means that the port that transmitted the loop detection packet is disabled. When this action is enabled, environments with N loops must have loop protection configured on at least N-1 ports to ensure a loop free topology.



When the default action (`tx-disable`) is used, it is optional to enable loop protect in all interfaces. By enabling loop protect in a single interface, the loop is detected and the default action is executed. So when the packet from a loop protect-enabled port is received back on an interface where loop protect is not enabled, the loop protect receiver action corresponding to the receiving interface is executed. Please note that all the L2 ports will have a default receiver action of `tx-disable` even when loop protect is not enabled.

3. If required, change the interval at which loop protection messages are sent with the command `loop-protect transmit-interval`.
4. If required, change the length of time the switch waits before re-enabling an interface with the command `loop-protect re-enable-timer`.
5. Review loop protection configuration settings with the command `show loop-protect`.

Example

This example creates the following configuration:

- Enables loop protection on data port **1/1/1** and sets the loop detection action to disable the transmit port.
- Enables loop protection on LAG **25** and sets the loop detection action to disable both transmit and receive ports.
- Enables loop protection on VLANs **100-125** and **200**.

- Sets the re-enable timer to **10** seconds.
- Sets the transmit-interval to **30** seconds.

```
switch(config)# interface 1/1/1
switch(config-if)# no routing
switch(config-if)# loop-protect
switch(config-if)# loop-protect action tx-disable
switch(config-if)# exit
switch(config)# interface lag 25
switch(config-lag-if)# loop-protect
switch(config-if)# loop-protect action tx-rx-disable
switch(config-if)# loop-protect vlan 100-125,200
switch(config-if)# exit
switch(config)# loop-protect re-enable-timer 30
switch(config)# exit
switch# show loop-protect
Status and Counters - Loop Protection Information
Transmit Interval          : 30 (sec)
Port Re-enable Timer      : 10 (sec)
Interface 1/1/1
  Loop-protect enabled     : Yes
  Loop-Protect enabled VLANs :
  Action on loop detection : TX disable
  Loop detected count      : 0
  Loop detected            : No
  Interface status         : up
Interface lag 25
  Loop-protect enabled     : Yes
  Loop-Protect enabled VLANs : 100-125,200
  Action on loop detection : TX-RX disable
  Loop detected count      : 0
  Loop detected            : No
  Interface status         : up
```

Loop protect commands

loop-protect

```
loop-protect
no loop-protect
```

Description

Enables loop protection on a layer 2 interface or LAG. Loop protection packets are sent/received on the LAG and not the interface which are members of the LAG. Loop protection only works on layer 2 interfaces. If a layer 2 interface is changed to a layer 3 interface, all loop protection configuration settings are lost for that interface.

The `no` form of this command disables loop protection on a layer 2 interface or LAG.

Examples

Enabling loop protection on interface **1/1/1**:

```
switch# config
switch(config)# interface 1/1/1
switch(config-if)# no routing
```



```
switch(config-if)# loop-protect
```

Enabling loop protection on LAG 25:

```
switch# config
switch(config)# interface lag 25
switch(config-if)# no routing
switch(config-lag-if)# loop-protect
```

Command History

Release	Modification
10.07 or earlier	--

Command Information

Platforms	Command context	Authority
All platforms	config-if config-lag-if	Administrators or local user group members with execution rights for this command.

loop-protect action

```
loop-protect action {do-not-disable | tx-disable | tx-rx-disable}
no loop-protect action {do-not-disable | tx-disable | tx-rx-disable}
```

Description

Sets the action to be taken when a loop protection packet is received on a port.

If an action is configured after a loop is detected, then the new action only takes effect after the re-enable timer expires. To have the action take effect immediately, disable and then re-enable loop protect.

The `no` form of this command resets the action to the default (`tx-disable`).

Parameter	Description
do-not-disable	No ports are disabled. On every transmit interval, the loop will be detected and the detection will be reported via an SNMP trap and an event log message.
tx-disable	The port that transmitted the loop detection packet is disabled. When this setting is enabled, environments with N loops, must have loop protection be configured on at least N-1 ports to have a loop free topology. Default.
tx-rx-disable	The ports that transmitted and received the loop detection packet are disabled.

Example

```
switch(config-if)# loop-protect action do-not-disable
switch(config-if)# no loop-protect action do-not-disable
```

Command History

Release	Modification
10.07 or earlier	--

Command Information

Platforms	Command context	Authority
All platforms	config-if	Administrators or local user group members with execution rights for this command.

loop-protect re-enable-timer

```
loop-protect re-enable-timer <TIME>
no loop-protect re-enable-timer <TIME>
```

Description

Configures the time interval after which an interface disabled by loop protection is re-enabled. The loop protection timer is disabled by default.

The `no` form of this command disables the loop protect timer.

Parameter	Description
<TIME>	Specify the number of seconds after which a disabled interface is re-enabled. Range: 15 to 604800.

Example

```
switch# config
switch(config)# loop-protect re-enable-timer 60
```

Command History

Release	Modification
10.07 or earlier	--

Command Information

Platforms	Command context	Authority
All platforms	config	Administrators or local user group members with execution rights for this command.

loop-protect transmit-interval

```
loop-protect transmit-interval <TIME>
no loop-protect transmit-interval [<TIME>]
```

Description

Configures the time interval between successive loop protect packets sent on an interface. The `no` form of this command sets the time interval to the default value of 5 seconds.

Parameter	Description
<TIME>	Configures the transmit interval in seconds. Range: 5 to 10. Default: 5.

Examples

```
switch(config)# loop-protect transmit-interval 10
switch(config)# no loop-protect transmit-interval
```

Command History

Release	Modification
10.07 or earlier	--

Command Information

Platforms	Command context	Authority
All platforms	config	Administrators or local user group members with execution rights for this command.

loop-protect trap loop-detected

```
loop-protect trap loop-detected
no loop-protect trap loop-detected
```

Description

Enables sending SNMP traps for loop-protect related events. The `no` form of this command disables sending SNMP traps for loop-protect related events.

Examples

Enabling the sending of SNMP traps:

```
switch# loop-protect trap loop-detected
```

Disabling the sending of SNMP traps:

```
switch# no loop-protect trap loop-detected
```

Command History

Release	Modification
10.07 or earlier	--

Command Information

Platforms	Command context	Authority
All platforms	config	Administrators or local user group members with execution rights for this command.

loop-protect vlan

```
loop-protect vlan <VLAN-LIST>
no loop-protect vlan
```

Description

Specifies the trunk allowed VLANs on which loop protection packets are sent. By default, loop protection packets are only sent on access VLANs and native VLANs on a port. To send loop protection packets on trunk allowed VLANs, the VLANs must be explicitly added using this command.

Loop protection can be configured on a maximum of VLANs across all interfaces.

The `no` form of this command removes loop protection from all VLANs on the interface.

Parameter	Description
<VLAN-LIST>	Specifies the number of a single VLAN, or a series of numbers for a range of VLANs, separated by commas (1, 2, 3, 4), dashes (1-4), or both (1-4, 6).

Example

```
switch(config-if) # loop-protect vlan 2-6,10,15-20
```

Command History

Release	Modification
10.07 or earlier	--

Command Information

Platforms	Command context	Authority
All platforms	config-if	Administrators or local user group members with execution rights for this command.

show loop-protect

```
show loop-protect [<INTERFACE-NAME>] [vsx-peer]
```

Description

This command shows the following global configurations.

- Transmit interval.
- Re-enable timer.
- Per-port configurations.
- Loop-protect enable or disable status.
- Loop detection.
- Loop detected count.
- Timestamp of latest loop detection.
- Loop is detected on VLAN.
- Interface status.
- List of configured VLAN's for that port.

Specify the interface name on display for the filter. When rebooting the switch or switchover, The loop-detected count on the loop detected port is reset to zero.

Parameter	Description
<code><INTERFACE-NAME></code>	Specifies the name of a logical interface on the switch. This can be one of the following: <ul style="list-style-type: none">■ An Ethernet interface associated with a physical port. Format: member/slot/port.■ A LAG (link aggregation group). Specify the ID of LAG . For example: lag100.
<code>vsx-peer</code>	Shows the output from the VSX peer switch. If the switches do not have the VSX configuration or the ISL is down, the output from the VSX peer switch is not displayed. This parameter is available on switches that support VSX.

Examples

```
switch# show loop-protect

Transmit Interval (sec)           : 5
Port Re-enable Timer (sec)       : Disabled
Loop Detected Trap                : Enabled

Interface 1/1/1
  Loop-protect enabled           : Yes
  Loop-Protect enabled VLANs     :
  Action on loop detection       : TX disable
  Loop detected count            : 0
  Loop detected                  : No
  Interface status               : up

Interface 1/1/2
  Loop-protect enabled           : Yes
  Loop-Protect enabled VLANs     :
  Action on loop detection       : TX disable
  Loop detected count            : 0
  Loop detected                  : No
  Interface status               : up
```

```
switch# show loop-protect 1/1/3
```

Status and Counters - Loop Protection Information

```
Transmit Interval (sec)      : 5
Port Re-enable Timer (sec)   : 0
Loop Detected Trap           : Disabled

Interface 1
  Loop-protect enabled       : Yes
  Loop-Protect enabled VLANs :
  Action on loop detection   : TX disable
  Loop detected count        : 0
  Loop detected              : No
  Interface status           : up
```

Command History

Release	Modification
10.07 or earlier	--

Command Information

Platforms	Command context	Authority
All platforms	Operator (>) or Manager (#)	Operators or Administrators or local user group members with execution rights for this command. Operators can execute this command from the operator context (>) only.

MVRP provides a mechanism to dynamically share VLAN configuration information across layer 2 switches on a network. MVRP eliminates the need to manually configure VLANs on each switch, enabling the network to dynamically maintain VLANs based on the current network configuration. MVRP propagates local VLAN information to other devices, receives VLAN information from other devices, and dynamically updates local VLAN information. When the network topology changes, MVRP propagates and learns VLAN information again according to the new topology.

MVRP is defined in the IEEE 802.1ak standard. It performs the same functions as Generic Attribute Registration Protocol (GARP), while overcoming GARP limitations, such as bandwidth usage and convergence time in networks with a large number of VLANs.

MVRP makes use of the Multiple Registration Protocol (MRP). MRP provides the mechanism for switches on the same layer 2 network to transmit attribute values on a per MSTI (Multiple Spanning Tree Instance) basis. (An MSTI is a group or set of VLANs, all of which are part of the same spanning tree.)

Each MRP-enabled interface is called an MRP participant, and each MVRP-enabled interface is called an MVRP participant. When the VLAN configuration on an MVRP participant changes, it sends a Protocol Data Unit (PDU) to notify other MVRP participants to register and deregister the changed VLAN. MRP rapidly propagates the configuration information of an MRP participant throughout the layer 2 network. MRP registers and deregisters VLAN attributes as follows:

- When an interface receives a declaration for a VLAN, the interface registers the VLAN and joins the VLAN.
- When an interface receives a withdrawal for a VLAN, the interface deregisters the VLAN and leaves the VLAN.

MVRP only applies to trunk interfaces.

MVRP functionality and limitations

MIB support

The MVRP feature supports objects in the following standard MIBs:

- IEEE8021-Q-BRIDGE-MIB (Version 200810150000Z)
- IEEE8021-BRIDGE-MIB (Version 200810150000Z)

It also supports MVRP objects in the HPE proprietary MIB:

HPE-MVRP-MIB (`hpeMvrp.mib`)

MVRP limitations

- MVRP is only supported on L2 trunk ports.
- MVRP and VLAN translation cannot be enabled on the same interface.
- MVRP will propagate only the first 1024 VLANs. This number includes existing static VLANs locally. For example, if a peer device already has 100 static VLANs, then it can only learn 924 VLANs.

- MVRP and PVST cannot be enabled at the same time.
- For security purposes, MVRP is disabled by default. MVRP packets are blocked on MVRP disabled ports, but can be enabled on ports that are security enabled.
- MVRP supports 1024 VLANs and 512 logical ports.
- If MVRP is enabled globally, MVRP is automatically enabled on LAG interfaces and cannot be disabled.

MRP messages

MRP messages include the following types:

- Declaration: Includes Join and New messages.
- Withdrawal: Includes Leave and LeaveAll messages.

Join message

An MRP participant sends a Join message to request the peer participant to register attributes in the Join message.

When receiving a Join message from the peer participant, an MRP participant performs the following tasks:

- Registers the attributes in the Join message.
- Propagates the Join message to all other participants on the device.

After receiving the Join message, other participants send the Join message to their respective peer participants.

Join messages sent from a local participant to its peer participant include the following types:

- JoinEmpty: Declares an unregistered attribute. For example, when an MRP participant joins an unregistered static VLAN, it sends a JoinEmpty message. VLANs created manually and locally are called static VLANs. VLANs learned through MRP are called dynamic VLANs.
- JoinIn: Declares a registered attribute. A JoinIn message is used in one of the following situations:
 - An MRP participant joins an existing static VLAN and sends a JoinIn message after registering the VLAN.
 - The MRP participant receives a Join message propagated by another participant on the device and sends a JoinIn message after registering the VLAN.

New message

Similar to a Join message, a New message enables MRP participants to register attributes.

When the MSTP topology changes, an MRP participant sends a New message to the peer participant to declare the topology change.

Upon receiving a New message from the peer participant, an MRP participant performs the following tasks:

- Registers the attributes in the message.
- Propagates the New message to all other participants on the device.

After receiving the New message, other participants send the New message to their respective peer participants.

Leave message

An MRP participant sends a Leave message to the peer participant when it wants the peer participant to deregister attributes that it has deregistered.

When the peer participant receives the Leave message, it performs the following tasks:

- Deregisters the attribute in the Leave message.
- Propagates the Leave message to all other participants on the device.

After a participant on the device receives the Leave message, it determines whether to send the Leave message to its peer participant depending on the attribute status on the device.

- If the VLAN in the Leave message is a dynamic VLAN not registered by any participants on the device, both of the following events occur:
 - The VLAN is deleted on the device.
 - The participant sends the Leave message to its peer participant.
- If the VLAN in a Leave message is a static VLAN, the participant will not send the Leave message to its peer participant.

LeaveAll message

Each MRP participant starts its LeaveAll timer when starting up. When the timer expires, the MRP participant sends LeaveAll messages to the peer participant.

Upon sending or receiving a LeaveAll message, the local participant starts the Leave timer. The local participant determines whether to send a Join message depending on its the attribute status. A participant can re-register the attributes in the received Join message before the Leave timer expires.

When the Leave timer expires, a participant deregisters all attributes that have not been re-registered to periodically clear useless attributes in the network.

Configuring MVRP

Prerequisites

MVRP must be enabled globally to facilitate dynamic VLAN learning.

Procedure

1. Enable MVRP globally on all interfaces or only for specific interfaces with the command `mvrp`. (For Dynamic LAGs, MVRP is enabled by default).
2. By default, MVRP supports dynamic registration and deregistration of VLANs on all interfaces. If required, customize the behavior for each interface with the command `mvrp registration`.
3. If required, adjust the MVRP timers from their default values with the command `mvrp timer`. To avoid frequent registrations and deregistrations, use the same MVRP timer values throughout the network.
4. Review your MVRP configuration settings with the commands `show mvrp config`, `show mvrp state`, and `show mvrp statistics`.

Example

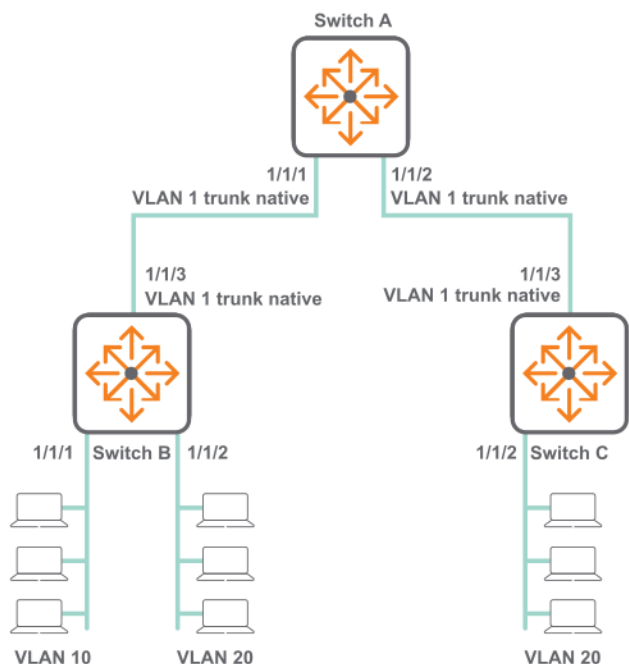
This example creates the following configuration:

- Enables MVRP on all interfaces.
- Sets interface 1/1/1 to ignore VLAN 100.

```
switch(config)# mvrp
switch(config)# interface 1/1/1
switch(config-if)# mvrp registration forbidden 100
switch(config-if)# mvrp
switch(config-if)# quit
switch# show mvrp config
Configuration and Status - MVRP
Global MVRP status : Disabled
Port      Status      Registration Type      Join Timer  Leave Timer  LeaveAll Timer  Periodic Timer
-----
1/1/1     Disabled    Normal                20       300        1000         100
switch# show mvrp state 1/1/1
Configuration and Status - MVRP state for VLAN 1
Port      VLAN Registrar Applicant
State      State
-----
1/1/1     1       MT       QA
```

MVRP scenario 1

This scenario illustrates the configuration of a simple MVRP deployment.



Procedure

1. On switch A, enable MVRP globally, define VLANs on interface **1/1/1** and **1/1/2**, and enable MVRP on each interface.

```
switch# config
switch(config)# mvrp
```

```

switch(config)# interface 1/1/1
switch(config-if)# no shutdown
switch(config-if)# no routing
switch(config-if)# vlan trunk native 1
switch(config-if)# mvrp
switch(config-if)# exit
switch(config)# interface 1/1/2
switch(config-if)# no shutdown
switch(config-if)# no routing
switch(config-if)# vlan trunk native 1
switch(config-if)# mvrp

```

2. On switch B, enable MVRP globally, define VLANs 10 and 20, assign a trunk native VLAN to interface **1/1/3**, and enable MVRP on this interface.

```

switch# config
switch(config)# mvrp
switch(config)# vlan 10
switch(config)# vlan 20
switch(config)# interface 1/1/3
switch(config-if)# no shutdown
switch(config-if)# no routing
switch(config-if)# vlan trunk native 1
switch(config-if)# mvrp

```

3. On switch C, enable MVRP globally, define VLAN 20, assign a trunk native VLAN to interface **1/1/3**, and enable MVRP on this interface.

```

switch# config
switch(config)# mvrp
switch(config)# vlan 20
switch(config)# interface 1/1/3
switch(config-if)# no shutdown
switch(config-if)# no routing
switch(config-if)# vlan trunk native 1
switch(config-if)# mvrp

```

4. Verify VLAN configuration by running the command `show vlan`. It should show that VLAN 10 and 20 are learned by switch A, and VLAN 10 should be learned by switch C. For example:

On switch A:

```

switch# show vlan
-----
-
VLAN  Name                               Status Reason                               Type    Interfaces
-----
-
1      DEFAULT_VLAN_1                         up    ok                                    default 1/1/1-
1/1/2
10     VLAN10                                up    ok                                    dynamic 1/1/2
20     VLAN20                                up    ok                                    dynamic 1/1/1-
1/1/2
switch#

```

```

switch# show mvrp config
Configuration and Status - MVRP
Global MVRP status : Enabled
Port      Status      Registration Join   Leave   LeaveAll Periodic
          Type        Timer   Timer   Timer   Timer
-----
1/1/1 Enabled    normal    20     300    1000    100
1/1/2 Enabled    normal    20     300    1000    100
switch# show mvrp state
Configuration and Status - MVRP state
Port      VLAN Registrar Applicant Forbid
          State      State      Mode
-----
1/1/1 1    IN      QA      No
1/1/1 10   MT      QA      No
1/1/1 20   IN      QA      No
1/1/2 1    IN      QA      No
1/1/2 10   IN      VO      No
1/1/2 20   IN      QA      No
switch# show mvrp statistics
Status and Counters - MVRP
MVRP statistics for port : 1/1/1
-----
Failed registration : 0
Last PDU origin    : e0:07:1b:cb:01:ab
Total PDU Transmitted : 313
Total PDU Received  : 377
Frames Discarded    : 0
Message type      Transmitted   Received
-----
New                0             0
Empty              179105        2264
In                 0             346
Join Empty         366           62
Join In            342           692
Leave               0             0
Leaveall            43            32

Status and Counters - MVRP
MVRP statistics for port : 1/1/2
-----
Failed registration : 0
Last PDU origin    : e0:07:1b:cb:22:54
Total PDU Transmitted : 450
Total PDU Received  : 84
Frames Discarded    : 0
Message type      Transmitted   Received
-----
New                0             0
Empty              173629        382
In                 328           0
Join Empty         83            93
Join In            711           65
Leave               0             0
Leaveall            41            33

```

On switch B:

```

switch# show vlan
-----
--
VLAN  Name                      Status Reason                      Type
Interfaces
-----
--
1      DEFAULT_VLAN_1              up    ok                          default  1/1/3
10     VLAN10                        up    ok                          static   1/1/3
20     VLAN20                        up    ok                          static   1/1/3
SW1-8320#

SW1-8320# show mvrp config
Configuration and Status - MVRP
Global MVRP status : Enabled
Port      Status      Registration Join    Leave    LeaveAll Periodic
          Type          Type      Timer    Timer    Timer    Timer
-----
1/1/3    Enabled    normal    20      300     1000     100
SW1-8320# show mvrp state
Configuration and Status - MVRP state
Port      VLAN Registrar Applicant Forbid
          State      State      Mode
-----
1/1/3    1      IN        QA        No
1/1/3    10     MT        QA        No
1/1/3    20     IN        QA        No
SW1-8320# show mvrp statistics
Status and Counters - MVRP
MVRP statistics for port : 1/1/3
-----
Failed registration : 0
Last PDU origin    : 48:0f:cf:af:f2:fa
Total PDU Transmitted : 77
Total PDU Received  : 303
Frames Discarded    : 0
Message type      Transmitted    Received
-----
New                0              0
Empty              115067         1754
In                 0              268
Join Empty         100            1
Join In            53             581
Leave               0              0
Leaveall            28             27

```

On switch C:

```

switch# show vlan
-----
--
VLAN  Name                      Status Reason                      Type
Interfaces
-----
--
1      DEFAULT_VLAN_1              up    ok                          default  1/1/3
10     VLAN10                        up    ok                          dynamic  1/1/3
20     VLAN20                        up    ok                          static   1/1/3
switch#

```

```

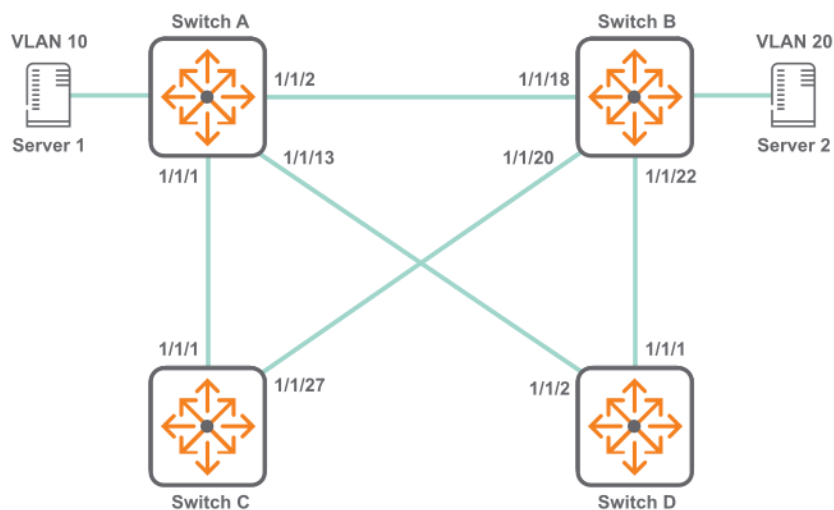
switch# show mvrp config
Configuration and Status - MVRP
Global MVRP status : Enabled
Port      Status      Registration      Join      Leave      LeaveAll      Periodic
-----  -
Type      Timer      Timer      Timer      Timer
-----  -
1/1/3    Enabled    normal          20        300        1000         100
switch# show mvrp state
Configuration and Status - MVRP state
Port      VLAN Registrar Applicant Forbid
-----  -
State      State      Mode
-----  -
1/1/3     1          IN          QA        No
1/1/3     10         IN          VO        No
1/1/3     20         IN          QA        No
switch#

switch# show mvrp statistics
Status and Counters - MVRP
MVRP statistics for port : 1/1/3
-----
Failed registration      : 0
Last PDU origin          : 48:0f:cf:af:f2:fb
Total PDU Transmitted    : 203
Total PDU Received       : 95
Frames Discarded          : 0
Message type      Transmitted      Received
-----
New                0                0
Empty              72915           586
In                 183             0
Join Empty         40             101
Join In            366            176
Leave               0                0
Leaveall            17             16

```

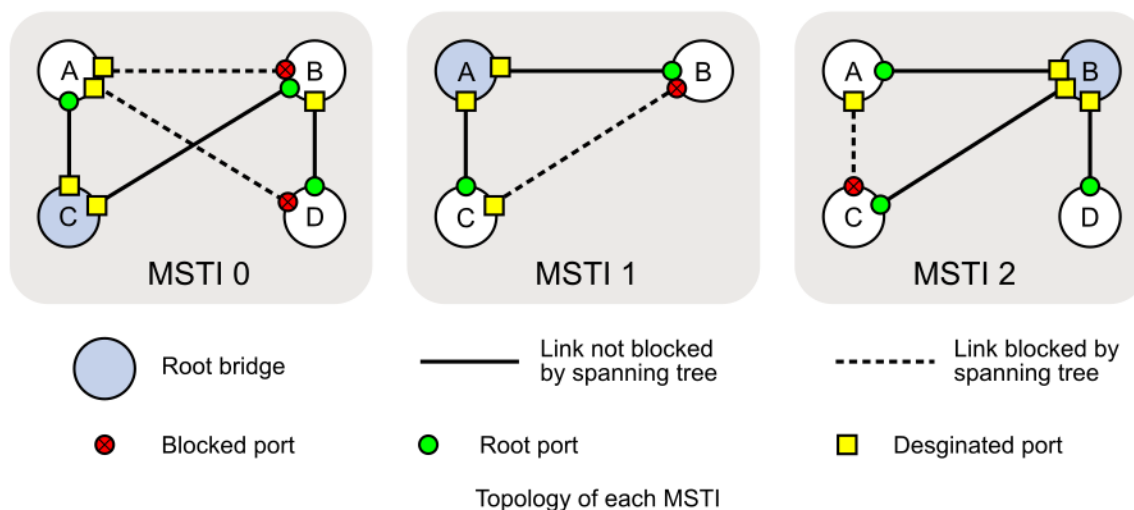
MVRP scenario 2

This scenario illustrates the configuration of an MVRP deployment with two MSTIs.



Two MSTIs are defined for this scenario:

- VLAN 10 assigned to MSTI 1
- VLAN 20 assigned to MSTI 2
- All other VLANs assigned to the default MSTI 0



Procedure

1. On switch A:

```
switch# config
switch(config)# mvrp
switch(config)# vlan 10
switch(config)# spanning-tree
switch(config)# spanning-tree priority 1
switch(config)# spanning-tree config-name sp1
switch(config)# spanning-tree config-revision 1
switch(config)# spanning-tree instance 1 vlan 10
switch(config)# spanning-tree instance 2 vlan 20
switch(config)# interface 1/1/1
switch(config-if)# no shutdown
switch(config-if)# no routing
switch(config-if)# vlan trunk native 1
switch(config-if)# mvrp
switch(config-if)# exit
switch(config)# interface 1/1/2
switch(config-if)# no shutdown
switch(config-if)# no routing
switch(config-if)# vlan trunk native 1
switch(config-if)# mvrp
switch(config-if)# exit
switch(config)# interface 1/1/3
switch(config-if)# no shutdown
switch(config-if)# no routing
switch(config-if)# vlan trunk native 1
switch(config-if)# mvrp
switch(config-if)# exit
switch# show mvrp config
Configuration and Status - MVRP
Global MVRP status : Enabled
Port      Status   Registration Join   Leave   LeaveAll Periodic
```

		Type	Timer	Timer	Timer	Timer
1/1/1	Enabled	normal	20	300	1000	100
1/1/3	Enabled	normal	20	300	1000	100
1/1/2	Enabled	normal	20	300	1000	100

```

switch# show mvrp state
Configuration and Status - MVRP state
Port    VLAN Registrar Applicant Forbid
      State      State      Mode
----
1/1/1   1      IN      QA      No
1/1/1   20     MT      QA      No
1/1/3   1      IN      QA      No
1/1/3   20     IN      VO      No
1/1/2   1      MT      QA      No
1/1/2   20     MT      QA      No

switch# show spanning-tree mst
#### MST0
Vlans mapped: 1-9,11-19,21-4094
Bridge      Address:48:0f:cf:af:f1:82      priority:4096
Operational Hello time(in seconds): 2    Forward delay(in seconds):15  Max-
age(in seconds):20  txHoldCount(i6
Configured  Hello time(in seconds): 2    Forward delay(in seconds):15  Max-
age(in seconds):20  txHoldCount(i6
Root        Address:48:0f:cf:af:14:0a    Priority:4096
Port:1/1/3  Path cost:0
Regional Root Address:48:0f:cf:af:14:0a    Priority:4096
Internal cost:20000  Rem Hops:19

Port      Role      State      Cost      Priority  Type
-----
1/1/1     Designated Forwarding  20000     128       point_to_
point
1/1/2     Designated Forwarding  20000     128       point_to_
point
1/1/3     Root      Forwarding  20000     128       point_to_
point

#### MST1
Vlans mapped: 10
Bridge      Address:48:0f:cf:af:f1:82      Priority:32768
Root        Address:48:0f:cf:af:14:0a    Priority:32768
Port:1/1/3, Cost:20000, Rem Hops:19

Port      Role      State      Cost      Priority  Type
-----
1/1/1     Designated Forwarding  20000     128       point_to_point
1/1/2     Designated Forwarding  20000     128       point_to_point
1/1/3     Root      Forwarding  20000     128       point_to_point

#### MST2
Vlans mapped: 20
Bridge      Address:48:0f:cf:af:f1:82      Priority:32768
Root        Address:48:0f:cf:af:14:0a    Priority:32768
Port:1/1/3, Cost:20000, Rem Hops:19

Port      Role      State      Cost      Priority  Type
-----

```


1/1/1	Designated	Forwarding	20000	128	point_to_point
1/1/2	Designated	Forwarding	20000	128	point_to_point
1/1/3	Root	Forwarding	20000	128	point_to_point

2. On switch B:

```
switch# config
switch(config)# mvrp
switch(config)# vlan 20
switch(config)# spanning-tree
switch(config)# spanning-tree priority 1
switch(config)# spanning-tree config-name sp1
switch(config)# spanning-tree config-revision 1
switch(config)# spanning-tree instance 1 vlan 10
switch(config)# spanning-tree instance 2 vlan 20
switch(config)# interface 1/1/18
switch(config-if)# no shutdown
switch(config-if)# no routing
switch(config-if)# vlan trunk native 1
switch(config-if)# vlan trunk allowed all
switch(config-if)# mvrp
switch(config-if)# exit
switch(config)# interface 1/1/20
switch(config-if)# no shutdown
switch(config-if)# no routing
switch(config-if)# vlan trunk native 1
switch(config-if)# vlan trunk allowed all
switch(config-if)# mvrp
switch(config-if)# exit
switch(config)# interface 1/1/22
switch(config-if)# no shutdown
switch(config-if)# no routing
switch(config-if)# vlan trunk native 1
switch(config-if)# vlan trunk allowed all
switch(config-if)# mvrp
switch(config-if)# exit
switch# show mvrp config
Configuration and Status - MVRP
Global MVRP status : Enabled
Port      Status      Registration Type      Join Timer      Leave Timer      LeaveAll Timer      Periodic Timer
-----
1/1/18    Enabled      normal                20              300              1000              100
1/1/20    Enabled      normal                20              300              1000              100
1/1/22    Enabled      normal                20              300              1000              100
switch# show mvrp state
Configuration and Status - MVRP state
Port      VLAN  Registrar Applicant Forbid
-----
1/1/20 1    MT      AA      No
1/1/20 10   MT      AA      No
1/1/20 20   MT      AA      No
1/1/22 1    IN      AP      No
1/1/22 10   IN      VO      No
1/1/22 20   MT      VP      No

switch# show spanning-tree mst
```

```

#### MST0
Vlans mapped: 1-9,11-19,21-4094
Bridge Address:e0:07:1b:cb:22:1c priority:4096
Operational Hello time(in seconds): 2 Forward delay(in seconds):15 Max-
age(in seconds):20 txHoldCount(in pp6
Configured Hello time(in seconds): 2 Forward delay(in seconds):15 Max-
age(in seconds):20 Max-Hops:20
Root Address:48:0f:cf:af:14:0a Priority:4096
Port:1/1/22 Path cost:0
Regional Root Address:48:0f:cf:af:14:0a Priority:4096
Internal cost:20000 Rem Hops:19

Port Role State Cost Priority Type
-----
1/1/18 Alternate Blocking 20000 128 point_to_
point
1/1/20 Designated Forwarding 20000 128 point_to_
point
1/1/22 Root Forwarding 20000 128 point_to_
point

#### MST1
Vlans mapped: 10
Bridge Address:e0:07:1b:cb:22:1c Priority:32768
Root Address:48:0f:cf:af:14:0a Priority:32768
Port:1/1/22, Cost:20000, Rem Hops:19

Port Role State Cost Priority Type
-----
1/1/18 Alternate Blocking 20000 128 point_to_point
1/1/20 Designated Forwarding 20000 128 point_to_point
1/1/22 Root Forwarding 20000 128 point_to_point

#### MST2
Vlans mapped: 20
Bridge Address:e0:07:1b:cb:22:1c Priority:32768
Root Address:48:0f:cf:af:14:0a Priority:32768
Port:1/1/22, Cost:20000, Rem Hops:19

Port Role State Cost Priority Type
-----
1/1/18 Alternate Blocking 20000 128 point_to_point
1/1/20 Designated Forwarding 20000 128 point_to_point
1/1/22 Root Forwarding 20000 128 point_to_point

```

3. On switch C:

```

switch# config
switch(config)# mvrp
switch(config)# vlan 1,20
switch(config)# spanning-tree
switch(config)# spanning-tree priority 1
switch(config)# spanning-tree config-name spl
switch(config)# spanning-tree config-revision 1
switch(config)# spanning-tree instance 1 vlan 10
switch(config)# spanning-tree instance 2 vlan 20

```

```

switch(config)# interface 1/1/25
switch(config-if)# no shutdown
switch(config-if)# no routing
switch(config-if)# vlan trunk native 1
switch(config-if)# vlan trunk allowed all
switch(config-if)# mvrp
switch(config-if)# exit
switch(config)# interface 1/1/27
switch(config-if)# no shutdown
switch(config-if)# no routing
switch(config-if)# vlan trunk native 1
switch(config-if)# vlan trunk allowed all
switch(config-if)# mvrp
switch(config-if)# exit
switch# show mvrp config
Configuration and Status - MVRP
Global MVRP status : Enabled

```

Port	Status	Registration Type	Join Timer	Leave Timer	LeaveAll Timer	Periodic Timer
1/1/25	Enabled	normal	20	300	1000	100
1/1/27	Enabled	normal	20	300	1000	100

```

switch# show mvrp state
Configuration and Status - MVRP state

```

Port	VLAN	Registrar State	Applicant State	Forbid Mode
1/1/25	1	IN	QA	No
1/1/25	10	IN	VO	No
1/1/25	20	IN	VO	No

```

switch# show spanning-tree mst
#### MST0
Vlans mapped: 1-9,11-19,21-4094
Bridge Address:e0:07:1b:cb:01:7a priority:4096
Operational Hello time(in seconds): 2 Forward delay(in seconds):15 Max-age(in seconds):20 txHoldCount(6)
Configured Hello time(in seconds): 2 Forward delay(in seconds):15 Max-age(in seconds):20 Max-Hops:20
Root Address:48:0f:cf:af:14:0a Priority:4096
Port:1/1/25 Path cost:0
Regional Root Address:48:0f:cf:af:14:0a Priority:4096
Internal cost:40000 Rem Hops:18

```

Port	Role	State	Cost	Priority	Type
1/1/25	Root	Forwarding	20000	128	point_to_point
1/1/27	Alternate	Blocking	20000	128	point_to_point

```

#### MST1
Vlans mapped: 10
Bridge Address:e0:07:1b:cb:01:7a Priority:32768
Root Address:48:0f:cf:af:14:0a Priority:32768
Port:1/1/25, Cost:40000, Rem Hops:18

```

Port	Role	State	Cost	Priority	Type
1/1/25	Root	Forwarding	20000	128	point_to_point

```

1/1/27          Alternate      Blocking      20000      128          point_to_point

#### MST2
Vlans mapped:  20
Bridge         Address:e0:07:1b:cb:01:7a      Priority:32768
Root           Address:48:0f:cf:af:14:0a      Priority:32768
               Port:1/1/25, Cost:40000, Rem Hops:18

Port           Role           State          Cost          Priority       Type
-----
1/1/25         Root           Forwarding     20000         128           point_to_point
1/1/27         Alternate      Blocking       20000         128           point_to_point

```

4. On switch D:

```

switch# config
switch(config)# mvrp
switch(config)# vlan 1
switch(config)# spanning-tree
switch(config)# spanning-tree priority 1
switch(config)# spanning-tree config-name spl
switch(config)# spanning-tree config-revision 1
switch(config)# spanning-tree instance 1 vlan 10
switch(config)# spanning-tree instance 2 vlan 20
switch(config)# interface 1/1/1
switch(config-if)# no shutdown
switch(config-if)# no routing
switch(config-if)# vlan trunk native 1
switch(config-if)# vlan trunk allowed all
switch(config-if)# mvrp
switch(config-if)# exit
switch(config)# interface 1/1/2
switch(config-if)# no shutdown
switch(config-if)# no routing
switch(config-if)# vlan trunk native 1
switch(config-if)# vlan trunk allowed all
switch(config-if)# mvrp
switch(config-if)# exit
switch# show mvrp config
Configuration and Status - MVRP
Global MVRP status : Enabled
Port      Status      Registration Type      Join Timer      Leave Timer      LeaveAll Timer      Periodic Timer
-----
1/1/1     Enabled     normal      20          300          1000          100
1/1/2     Enabled     normal      20          300          1000          100
switch# show mvrp state
Configuration and Status - MVRP state
Port      VLAN  Registrar Applicant Forbid
      State      State      Mode
-----
1/1/1     1      IN        QA        No
1/1/1     10     MT        QA        No
1/1/1     20     IN        VO        No
1/1/2     1      IN        AA        No
1/1/2     10     IN        VO        No
1/1/2     20     MT        AA        No

switch# show spanning-tree mst

```

```

#### MST0
Vlans mapped: 1-9,11-19,21-4094
Bridge Address:48:0f:cf:af:14:0a priority:4096
Root
Regional Root
Operational Hello time(in seconds): 2 Forward delay(in seconds):15 Max-
age(in seconds):20 txHoldC6
Configured Hello time(in seconds): 2 Forward delay(in seconds):15 Max-
age(in seconds):20 Max-Hop0
Root Address:48:0f:cf:af:14:0a Priority:4096
Port:0 Path cost:0
Regional Root Address:48:0f:cf:af:14:0a Priority:4096
Internal cost:0 Rem Hops:20

Port Role State Cost Priority Type
-----
1/1/1 Designated Forwarding 20000 128 point_to_
point
1/1/2 Designated Forwarding 20000 128 point_to_
point

#### MST1
Vlans mapped: 10
Bridge Address:48:0f:cf:af:14:0a Priority:32768
Root Address:48:0f:cf:af:14:0a Priority:32768
Port:0, Cost:0, Rem Hops:20

Port Role State Cost Priority Type
-----
1/1/1 Designated Forwarding 20000 128 point_to_point
1/1/2 Designated Forwarding 20000 128 point_to_point

#### MST2
Vlans mapped: 20
Bridge Address:48:0f:cf:af:14:0a Priority:32768
Root Address:48:0f:cf:af:14:0a Priority:32768
Port:0, Cost:0, Rem Hops:20

Port Role State Cost Priority Type
-----
1/1/1 Designated Forwarding 20000 128 point_to_point
1/1/2 Designated Forwarding 20000 128 point_to_point

```

MVRP commands

clear mvrp statistics

clear mvrp statistics [<PORT-NUM> | <PORT-LIST> | LAG <LAG-NUM>]

Description

Resets the MVRP statistic counters globally or for the specified ports or LAG.

Parameter	Description
<PORT-NUM>	Specifies a port number.

Parameter	Description
<PORT-LIST>	Specifies a list of ports.
LAG <LAG-NUM>	Specifies a Link Aggregation number. Range: 1 to 128.

Examples

```
switch# clear mvrp statistics 1/1/1
```

Command History

Release	Modification
10.07 or earlier	--

Command Information

Platforms	Command context	Authority
All platforms	Manager (#)	Administrators or local user group members with execution rights for this command.

mvrp

mvrp
no mvrp

Description

Enables the MVRP feature globally or on a specific interface. By default, MVRP is disabled. The `no` form of this command disables MVRP.



MVRP and VLAN translation cannot be enabled on the same interface.

Examples

Enabling MVRP globally:

```
switch(config)# mvrp
```

Enabling MVRP on an interface:

```
switch(config)# interface 1/1/1
switch(config-if)# mvrp
```

Command History

Release	Modification
10.07 or earlier	--

Command Information

Platforms	Command context	Authority
All platforms	config config-if	Administrators or local user group members with execution rights for this command.

mvrp registration

```
mvrp registration {normal | fixed | forbidden [<VLAN-LIST>]}
no mvrp registration forbidden {<VLAN-LIST>}
```

Description

Configures the MVRP registrar state which determines how an MVRP participant responds to MRP messages. The default registration mode is normal.

The `no` command removes the specified VLANs from the forbidden list.

Parameter	Description
normal	Enables dynamic registration and deregistration of VLANs on the interface, and propagates VLAN information to other switches on the network. Default.
fixed	Disables dynamic deregistration of VLANs and drops received MVRP frames. The interface does not deregister dynamic VLANs or register new dynamic VLANs.
forbidden	Disables dynamic registration of VLANs and drops received MVRP frames. The MVRP participant does not register new dynamic VLANs or re-register a deregistered dynamic VLAN.
<VLAN-LIST>	Disables dynamic registration of VLANs and drops received MVRP frames for specific VLANs only. Normal behavior applies to all other VLANs. Specify the number of a single VLAN, or a series of numbers for a range of VLANs, separated by commas (1, 2, 3, 4), dashes (1-4), or both (1-4,6).

Examples

```
switch(config)# switch(config-if)# mvrp registration forbidden 10
```

```
switch(config-if)# mvrp registration fixed
```

```
switch(config-if)# mvrp registration forbidden 1,2,10-20
```

Command History

Release	Modification
10.07 or earlier	--

Command Information

Platforms	Command context	Authority
All platforms	config-if	Administrators or local user group members with execution rights for this command.

mvrp timer

```
mvrp timer {join | leave | leaveall | periodic} <TIME>
no mvrp timer {join | leave | leaveall | periodic}
```

Description

Sets an MVRP timer.

The `no` form of this command sets the specified timer to its default value.

Parameter	Description
<code>join <TIME></code>	Sets the join timer. You can use the timer to space MVRP join messages. To ensure that join messages are transmitted to other participants, an MRP participant waits for the specified period of the join timer before sending a join message. The Join timer must be less than half of the Leave Timer. Range: 20 to 100 in centiseconds. Default: 20.
<code>leave <TIME></code>	Sets the leave timer for the port, specifying the time that the registrar state machine waits in the LV state before transiting to the MT state. The leave timer must be at least twice the join timer and must be less than the leave all timer. Range: 40 - 1000000 centiseconds. Default: 300 centiseconds.
<code>leaveall <TIME></code>	Sets the leave all timer for the port, specifying the frequency with which the leave all state machine generates leave all PDUs. Range: 500 to 1000000 centiseconds. Default: 1000.
<code>periodic <TIME></code>	Sets the periodic timer for the port, specifying the frequency with which the periodic transmission state machine generates periodic events. The periodic timer is set to 1 second when it is started. Range: 100 to 1000000 centiseconds. Default: 100.

Examples

```
switch(config-if)# mvrp timer join 22
```

Command History

Release	Modification
10.07 or earlier	--

Command Information

Platforms	Command context	Authority
All platforms	config-if	Administrators or local user group members with execution rights for this command.

show mvrp config

```
show mvrp config [<PORT-NUM> | <PORT-LIST> | LAG <LAG-NUM>] [vsx-peer]
```

Description

Displays the MVRP configuration for all L2 ports or optionally for the ports specified.

Parameter	Description
<PORT-NUM>	Specifies displaying information for a particular port number.
<PORT-LIST>	Specifies displaying information for a list of ports.
LAG <LAG-NUM>	Specifies displaying information by LAG. Range: 1 to 128.
vsx-peer	Shows the output from the VSX peer switch. If the switches do not have the VSX configuration or the ISL is down, the output from the VSX peer switch is not displayed. This parameter is available on switches that support VSX.

Examples

```
switch# show mvrp config
Configuration and Status - MVRP
Global MVRP status : Disabled
Port      Status   Registration   Join   Leave   LeaveAll   Periodic
-----  -
Type
-----  -
1/1/1    Disabled  Normal        20    300    1000      100
1/1/2    Disabled  Normal        20    300    1000      100
1/1/3    Disabled  Normal        20    300    1000      100
```

Command History

Release	Modification
10.07 or earlier	--

Command Information

Platforms	Command context	Authority
All platforms	Operator (>) or Manager (#)	Operators or Administrators or local user group members with execution rights for this command. Operators can execute this command from the operator context (>) only.

show mvrp state

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

Description

Displays the MVRP Registrar and Applicant state machine information for all ports on which MVRP is enabled, or for specific ports.

Parameter	Description
<VLAN-ID>	Specifies the number of a VLAN. Range: 1 - 4094.
<PORT-NUM>	Specifies a physical port on the switch. Format: member/slot/port.
vsx-peer	Shows the output from the VSX peer switch. If the switches do not have the VSX configuration or the ISL is down, the output from the VSX peer switch is not displayed. This parameter is available on switches that support VSX.

Examples

```
switch# show mvrp state 1
Configuration and Status - MVRP state for VLAN 1
Port   VLAN Registrar Applicant
      State      State
----
1/1/1  1      MT          QA
```

```
switch# show mvrp state 10 1/1/1
Configuration and Status - MVRP state for VLAN 10
Port   VLAN Registrar Applicant Forbid
      State      State      Mode
----
1/1/1  10      MT          LO          Yes
switch#
```

Command History

Release	Modification
10.07 or earlier	--

Command Information

Platforms	Command context	Authority
All platforms	Operator (>) or Manager (#)	Operators or Administrators or local user group members with execution rights for this command. Operators can execute this command from the operator context (>) only.

show mvrp statistics

```
show mvrp statistics [<PORT-LIST>] [vsx-peer]
```

Description

Displays MVRP statistics for all ports or on the ports specified in the list.

Parameter	Description
<PORT-LIST>	Specifies a list of ports. When specifying a list of ports, the ports for which there are no statistics will be listed in the output.
vsx-peer	Shows the output from the VSX peer switch. If the switches do not have the VSX configuration or the ISL is down, the output from the VSX peer switch is not displayed. This parameter is available on switches that support VSX.

Examples

```
switch# show mvrp statistics
Status and Counters - MVRP
MVRP statistics for port : 1/1/1
-----
Failed registration      : 0
Last PDU origin         : 48:0f:cf:af:b1:76
Total PDU Transmitted   : 13127
Total PDU Received      : 327
Frames Discarded        : 0
Message type      Transmitted      Received
-----
New                0                0
Empty              50029394         1264
In                 0                4
Join Empty         1425             48
Join In            563             555
Leave               0                0
Leaveall            12218            25
```

```
switch# show mvrp statistics 1/1/1

Status and Counters - MVRP
MVRP statistics for port : 1/1/1
-----
Failed registration      : 0
Last PDU origin         : 48:0f:cf:af:b1:76
Total PDU Transmitted   : 14874
Total PDU Received      : 327
Frames Discarded        : 0
Message type      Transmitted      Received
-----
New                0                0
Empty              57181612         1264
In                 0                4
Join Empty         1425             48
Join In            563             555
Leave               0                0
Leaveall            13965            25
```

Command History

Release	Modification
10.07 or earlier	--

Command Information

Platforms	Command context	Authority
All platforms	Operator (>) or Manager (#)	Operators or Administrators or local user group members with execution rights for this command. Operators can execute this command from the operator context (>) only.

Protocols and feature details

Spanning tree protocols eliminate loops in a physical link-redundant network by selectively blocking redundant links and putting them in a standby state.

Recent versions of STP include the Rapid Per-VLAN Spanning Tree Protocol (RPVST+) and the Multiple Spanning Tree Protocol (MSTP).

STP

Spanning tree protocol (STP) was developed based on the 802.1d standard of IEEE to eliminate loops at the data link layer in a LAN. Networks often have redundant links as backups in case of failures, but loops are a very serious problem. Devices running STP detect loops in the network by exchanging information with one another. They eliminate loops by selectively blocking certain ports to prune the loop structure into a loop-free tree structure. This avoids proliferation and infinite cycling of packets that would occur in a loop network.

In a narrow sense, STP refers to IEEE 802.1d STP. In a broad sense, STP refers to the IEEE 802.1d STP and various enhanced spanning tree protocols derived from that protocol, such as RPVST and MSTP.

Root bridge

A tree network must have a root bridge. The entire network contains only one root bridge, and all the other bridges in the network are called leaf nodes. The root bridge is not permanent and can change when there are changes in the network topology.

Upon initialization of a network, each device generates and periodically sends configuration BPDUs, with itself as the root bridge. After network convergence, only the root bridge generates and periodically sends configuration BPDUs. The other devices only forward the BPDUs.

Root port

On a non-root bridge, the port which has the least cost to reach the root bridge is the root port.

The root port communicates with the root bridge. Each non-root bridge has only one root port. The root bridge has no root port.

Designated bridge and designated port

A designated bridge is a bridge on each LAN that provides the minimum root path cost. The designated bridge of a LAN is the only bridge allowed to forward frames to and from the LAN.

The designated bridge:

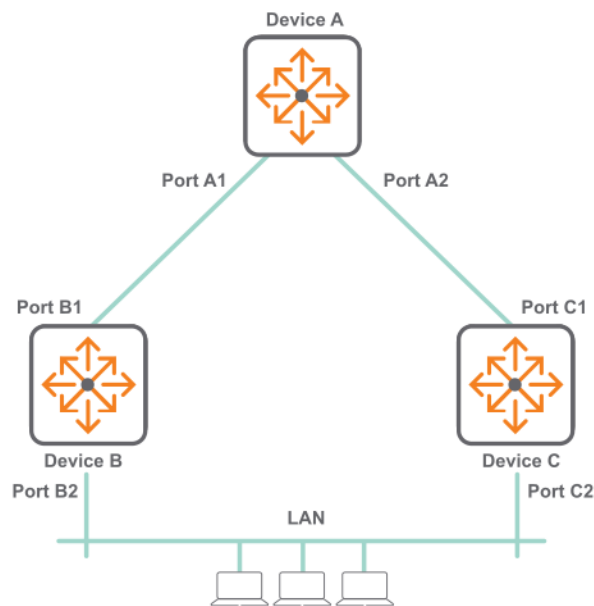
- For a device: Device directly connected with the local device and responsible for forwarding BPDUs to the local device.
- For a LAN: Device responsible for forwarding BPDUs to this LAN segment.

The designated port:

- For a device: Port through which the designated bridge forwards BPDUs to this device.
- For a LAN: Port through which the designated bridge forwards BPDUs to this LAN segment.

In the following topology, Device B and Device C are directly connected to a LAN.

Figure 1 *Designated bridge and designated port*



If Device A forwards BPDUs to Device B through port A1, the designated bridge and designated port are as follows:

- The designated bridge for Device B is Device A.
- The designated port of Device B is port A1 on Device A.

If Device B forwards BPDUs to the LAN, the designated bridge and designated port are as follows:

- The designated bridge for the LAN is Device B.
- The designated port for the LAN is port B2 on Device B.

Path cost

Path cost is a reference value used for link selection in STP. To prune the network into a loop-free tree, STP calculates path costs to select the most robust links and block redundant links that are less robust.

STP timers

The most important timing parameters in STP calculation are forward delay, hello time, and max age.

- Forward delay: Forward delay is the delay time for port state transition. A path failure can cause spanning tree recalculation to adapt the spanning tree structure to the change. However, the resulting new configuration BPDU cannot propagate throughout the network immediately. If the

newly elected root ports and designated ports start to forward data immediately, a temporary loop will likely occur. The newly elected root ports or designated ports require twice the forward delay time before they transit to the forwarding state. This allows the new configuration BPDU to propagate throughout the network.

- Hello time: The device sends hello packets at the hello time interval to the neighboring devices to make sure the paths are fault-free.
- Max age: The device uses the max age to determine whether a stored configuration BPDU has expired and discards it if the max age is exceeded.

BPDU forwarding mechanism

The configuration BPDUs of STP are forwarded according to these guidelines:

- Upon network initiation, every device regards itself as the root bridge and generates configuration BPDUs with itself as the root. Then it sends the configuration BPDUs at a regular hello interval.
- If the root port received a configuration BPDU superior to the configuration BPDU of the port, the device performs the following tasks:
 - Increases the message age carried in the configuration BPDU.
 - Starts a timer to time the configuration BPDU.
 - Sends this configuration BPDU through the designated port.
- If a designated port receives a configuration BPDU with a lower priority than its configuration BPDU, the port immediately responds with its configuration BPDU.
- If a path fails, the root port on this path no longer receives new configuration BPDUs and the old configuration BPDUs will be discarded due to timeout. The device generates a configuration BPDU with itself as the root and sends the BPDUs and TCN BPDUs. This triggers a new spanning tree calculation process to establish a new path to restore the network connectivity.

However, the newly calculated configuration BPDU cannot be propagated throughout the network immediately. As a result, the old root ports and designated ports that have not detected the topology change continue forwarding data along the old path. If the new root ports and designated ports begin to forward data as soon as they are elected, a temporary loop might occur.

STP protocol packets

STP uses bridge protocol data units (BPDUs), also known as configuration messages, as its protocol packets. STP-enabled network devices exchange BPDUs to establish a spanning tree.

STP uses the following types of BPDUs:

- Configuration BPDUs: Used by the network devices to calculate a spanning tree and maintain the spanning tree topology.
- Topology change notification (TCN) BPDUs: Use to notify network devices of network topology changes.

Configuration BPDUs contain sufficient information for network devices to complete spanning tree calculation. Important fields in a configuration BPDU include the following:

- Root bridge ID: Priority and MAC address of the root bridge.
- Root path cost: Cost of the path to the root bridge indicated by the root identifier from the transmitting bridge.
- Designated bridge ID: Priority and MAC address of the designated bridge.

- Designated port ID: Priority and global port number of the designated port.
- Message age: Age of the configuration BPDU while it propagates in the network.
- Max age: Maximum age of the configuration BPDU stored on the switch.
- Hello time: Configuration BPDU transmission interval.
- Forward delay: Delay that STP bridges use to transit port state.

Comparing spanning tree options

Without spanning tree, having more than one active path between a pair of network devices causes loops in the network that 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 onto a given physical topology, resulting 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. This significantly improves network link utilization and the speed of reconvergence after a failure in the network's physical topology. RPVST+ is a proprietary Cisco protocol, whereas MSTP is an open standard protocol based on IEEE 802.1s. So, in multi-vendor environments, MSTP is the preferred option because of interoperability.

In RPVST+, the number of spanning tree instances is equal to the number of VLANs. RPVST+ may become quite resource intensive as a result. The number of spanning tree instances in MSTP can theoretically be the same as the number of VLANs, but in practice, the number of spanning tree instances is limited to a number of physical topologies that is much fewer than the number of VLANs in the network.

RPVST+ is a Cisco-proprietary enhancement to PVST+, which is itself a Cisco-proprietary enhancement to 802.1D STP. PVST+ enables you to create one instance of spanning-tree per VLAN. Similar to PVST+, RPVST+ also enables you to create one spanning-tree instance per VLAN. The difference is network convergence is faster with RPVST+ than PVST+.

With RPVST+, 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.

Preparing for spanning tree configuration

Before configuring a spanning tree:

- Determine the spanning tree protocol to be used: RPVST+ or MSTP. RPVST+ is ideal in networks having fewer VLANs. In networks having more VLANs, MSTP is the recommended spanning tree choice due to the increased load on the switch CPU. Even if you have more VLANs, MSTP supports 64 instances, which is more than enough to disperse the load. The switch can distribute the VLANs in use among instances as evenly as feasible, allowing one instance to block redundant links while allowing another instance to forward traffic over the same links for non-redundant use.
- Plan the device roles (the root bridge or leaf node) by adjusting instance priority.

When you configure spanning tree protocols, follow these guidelines:

- If MSTP is enabled on the switch, MSTP takes all MSTI information along with the packet. To advertise a specific VLAN within the network through MSTP, make sure that the VLAN is mapped to an MSTI when you configure the VLAN-to-instance table.
- Configuring instances is not mandatory. It is optional. Simply enable spanning tree (with command `spanning-tree`) and then MSTP works with CIST on all switches (CIST is the common instance for all VLANs in the switch).
- STP is mutually exclusive with loop protection. If STP and loop protection are both enabled on the same VLAN, STP takes precedence. This means that loop protection does not take any action on a port blocked by STP.
- RPVST+ uses IEEE BPDU on the native VLAN and VLAN 1, to converge with MSTP. However RPVST+ uses proprietary PVST MAC address 01:00:0c:cc:cc:cd to converge with other RPVST VLANs. For example, if we enable 'spanning-tree vlan 2' on two switches, these switches converge by exchanging PVST proprietary MAC and not IEEE MAC. In this case, 'spanning-tree vlan 1' sends 1 IEEE MAC to converge with the MSTP network and it also sends 1 PVST MAC to converge with RPVST network.
- 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.
- The following features cannot run concurrently with RPVST+:
 - Multiple VLAN Registration Protocol (MVRP).
 - Filter multicast in RPVST+ mode (The multicast MAC address value cannot be set to the PVST MAC address 01:00:0c:cc:cc:cd.)
- After you enable a spanning tree protocol on a layer 2 aggregate interface, the system performs spanning tree calculation on the layer 2 aggregate interface. It does not perform the spanning tree calculation on the aggregation member ports. The spanning tree protocol enable state and forwarding state of each selected member port is consistent with the state of the corresponding layer 2 aggregate interface.
- Before using AAA and RPVST IOP, you must configure RADIUS-based and MAC-based VLANs statically and also enable RPVST on those VLANs.

STP cost calculation

Simplified calculation overview

A tree-shape topology forms once the root bridge, root ports, and designated ports are selected.

1. Upon initialization of a device, each port generates a BPDU with the following contents:
 - The port as the designated port.
 - The device as the root bridge.
 - 0 as the root path cost.
 - The device ID as the designated bridge ID.
2. Initially, each STP-enabled device on the network assumes itself to be the root bridge, with its own device ID as the root bridge ID. By exchanging configuration BPDUs, the devices compare their root bridge IDs to elect the device with the smallest root bridge ID as the root bridge.
3. Root port and designated ports selection on the non-root bridges.
 - A non-root-bridge device regards the port on which it received the optimum configuration BPDU as the root port.
 - Upon receiving a configuration BPDU on a port, the device compares the priority of the received configuration BPDU with that of the configuration BPDU generated by the port. If

the former priority is lower, the device discards the received configuration BPDU and keeps the configuration BPDU the port generated. If the former priority is higher, the device replaces the content of the configuration BPDU generated by the port with the content of the received configuration BPDU.

- The device compares the configuration BPDUs of all the ports and chooses the optimum configuration BPDU.
- Based on the configuration BPDU and the path cost of the root port, the device calculates a designated port configuration BPDU for each of the other ports.
 - The root bridge ID is replaced with that of the configuration BPDU of the root port.
 - The root path cost is replaced with that of the configuration BPDU of the root port plus the path cost of the root port.
 - The designated bridge ID is replaced with the ID of this device.
 - The designated port ID is replaced with the ID of this port.
- The device compares the calculated configuration BPDU with the configuration BPDU on the port whose port role will be determined, and acts depending on the result of the comparison:
 - If the calculated configuration BPDU is superior, the device performs the following tasks:
 - Considers this port as the designated port.
 - Replaces the configuration BPDU on the port with the calculated configuration BPDU.
 - Periodically sends the calculated configuration BPDU.
 - If the configuration BPDU on the port is superior, the device blocks this port without updating its configuration BPDU. The blocked port can receive BPDUs, but cannot send BPDUs or forward data traffic.

When the network topology is stable, only the root port and designated ports forward user traffic. Other ports are all in the blocked state to receive BPDUs but not to forward BPDUs or user traffic.

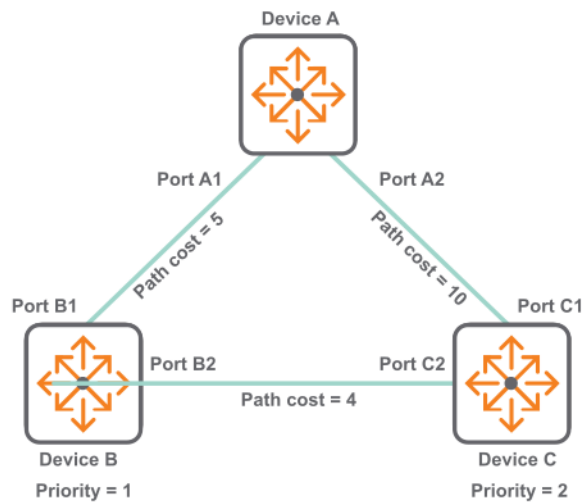
4. The principles of configuration BPDU comparison:
 - The configuration BPDU with the lowest root bridge ID has the highest priority.
 - If configuration BPDUs have the same root bridge ID, their root path costs are compared. For example, the root path cost in a configuration BPDU plus the path cost of a receiving port is S. The configuration BPDU with the smallest S value has the highest priority.
 - If all configuration BPDUs have the same root bridge ID and S value, the following attributes are compared in sequence: Designated bridge IDs, Designated port IDs, and IDs of the receiving ports.

The configuration BPDU that contains a smaller designated bridge ID, designated port ID, or receiving port ID is selected.

Calculation example

The following topology is used to illustrate an STP calculation. The priority values of Device A, Device B, and Device C are 0, 1, and 2, respectively. The path costs of links among the three devices are 5, 10, and 4.

Figure 2 STP calculation



Each configuration BPDU contains the following fields: root bridge ID, root path cost, designated bridge ID, and designated port ID.

1. The initial state of the BPDUs on each device is:

Device	Port name	Configuration BPDU on the port
Device A	Port A1	{0, 0, 0, Port A1}
	Port A2	{0, 0, 0, Port A2}
Device B	Port B1	{1, 0, 1, Port B1}
	Port B2	{1, 0, 1, Port B2}
Device C	Port C1	{2, 0, 2, Port C1}
	Port C2	{2, 0, 2, Port C2}

2. BPDU comparison on each device occurs as follows:

Device	Comparison process	Configuration BPDU on ports after comparison
Device A	Port A1 performs the following tasks: <ol style="list-style-type: none"> 1. Receives the configuration BPDU of Port B1 {1, 0, 1, Port B1}. 2. Determines that its existing configuration BPDU {0, 0, 0, Port A1} is superior to the received configuration BPDU. 3. Discards the received one. Port A2 performs the following tasks: <ol style="list-style-type: none"> 1. Receives the configuration BPDU of Port C1 {2, 0, 2, Port C1}. 	<ul style="list-style-type: none"> ■ Port A1: {0, 0, 0, Port A1} ■ Port A2: {0, 0, 0, Port A2}

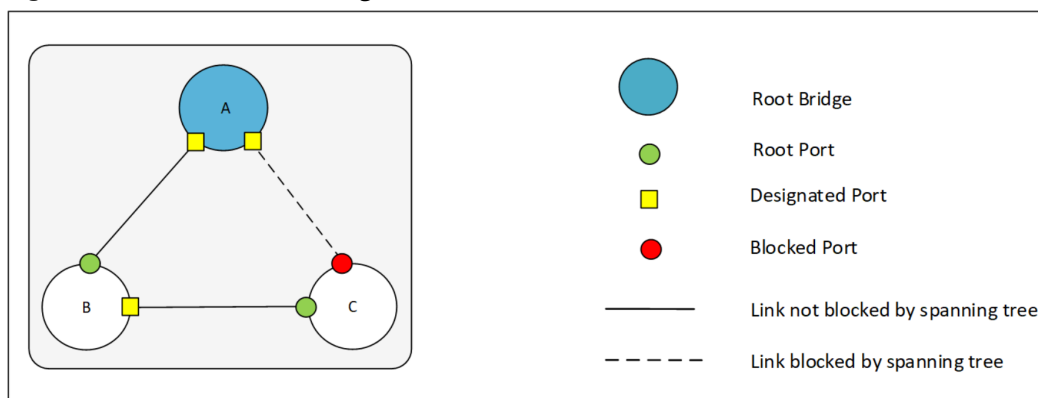
Device	Comparison process	Configuration BPDU on ports after comparison
	<ol style="list-style-type: none"> Determines that its existing configuration BPDU {0, 0, 0, Port A2} is superior to the received configuration BPDU. Discards the received one. <p>Device A determines that it is both the root bridge and designated bridge in the configuration BPDUs of all its ports. It considers itself as the root bridge. It does not change the configuration BPDU of any port and starts to periodically send configuration BPDUs.</p>	
Device B	<p>Port B1 performs the following tasks:</p> <ol style="list-style-type: none"> Receives the configuration BPDU of Port A1 {0, 0, 0, Port A1}. Determines that the received configuration BPDU is superior to its existing configuration BPDU {1, 0, 1, Port B1}. Updates its configuration BPDU. <p>Port B2 performs the following tasks:</p> <ol style="list-style-type: none"> Receives the configuration BPDU of Port C2 {2, 0, 2, Port C2}. Determines that its existing configuration BPDU {1, 0, 1, Port B2} is superior to the received configuration BPDU. Discards the received BPDU. 	<ul style="list-style-type: none"> Port B1: {0, 0, 0, Port A1} Port B2: {1, 0, 1, Port B2}
	<p>Device B performs the following tasks:</p> <ol style="list-style-type: none"> Compares the configuration BPDUs of all its ports. Decides that the configuration BPDU of Port B1 is the optimum. Selects Port B1 as the root port with the configuration BPDU unchanged. <p>Based on the configuration BPDU and path cost of the root port, Device B calculates a designated port configuration BPDU for Port B2 {0, 5, 1, Port B2}. Device B compares it with the existing configuration BPDU of Port B2 {1, 0, 1, Port B2}. Device B determines that the calculated one is superior, and determines that Port B2 is the designated port. It replaces the configuration BPDU on Port B2 with the calculated one, and periodically sends the calculated configuration BPDU.</p>	<ul style="list-style-type: none"> Root port (Port B1): {0, 0, 0, Port A1} Designated port (Port B2): {0, 5, 1, Port B2}
Device C	<p>Port C1 performs the following tasks:</p> <ol style="list-style-type: none"> Receives the configuration BPDU of Port A2 {0, 	<ul style="list-style-type: none"> Port C1: {0, 0, 0, Port A2} Port C2: {1, 0, 1, Port B2}

Device	Comparison process	Configuration BPDU on ports after comparison
	<p>0, 0, Port A2}.</p> <ol style="list-style-type: none"> Determines that the received configuration BPDU is superior to its existing configuration BPDU {2, 0, 2, Port C1}. Updates its configuration BPDU. <p>Port C2 performs the following tasks:</p> <ol style="list-style-type: none"> Receives the original configuration BPDU of Port B2 {1, 0, 1, Port B2}. Determines that the received configuration BPDU is superior to the existing configuration BPDU {2, 0, 2, Port C2}. Updates its configuration BPDU. 	
	<p>Device C performs the following tasks:</p> <ol style="list-style-type: none"> Compares the configuration BPDUs of all its ports. Decides that the configuration BPDU of Port C1 is the optimum. Selects Port C1 as the root port with the configuration BPDU unchanged. <p>Based on the configuration BPDU and path cost of the root port, Device C calculates the configuration BPDU of Port C2 {0, 10, 2, Port C2}. Device C compares it with the existing configuration BPDU of Port C2 {1, 0, 1, Port B2}.</p> <p>Device C determines that the calculated configuration BPDU is superior to the existing one, selects Port C2 as the designated port, and replaces the configuration BPDU of Port C2 with the calculated one.</p>	<ul style="list-style-type: none"> Root port (Port C1): {0, 0, 0, Port A2} Designated port (Port C2): {0, 10, 2, Port C2}
	<p>Port C2 performs the following tasks:</p> <ol style="list-style-type: none"> Receives the configuration BPDU of Port B2 {0, 5, 1, Port B2}. Determines that the received configuration BPDU is superior to its existing configuration BPDU {0, 10, 2, Port C2}. Updates its configuration BPDU. <p>Port C1 performs the following tasks:</p> <ol style="list-style-type: none"> Receives a periodic configuration BPDU {0, 0, 0, Port A2} from Port A2. Determines that it is the same as the existing configuration BPDU. Discards the received BPDU. 	<ul style="list-style-type: none"> Port C1: {0, 0, 0, Port A2} Port C2: {0, 5, 1, Port B2}

Device	Comparison process	Configuration BPDU on ports after comparison
	<p>Device C determines that the root path cost of Port C1 (10) (root path cost of the received configuration BPDU (0) plus path cost of Port C1 (10)) is larger than that of Port C2 (9) (root path cost of the received configuration BPDU (5) plus path cost of Port C2 (4)). Device C determines that the configuration BPDU of Port C2 is the optimum, and selects Port C2 as the root port with the configuration BPDU unchanged.</p> <p>Based on the configuration BPDU and path cost of the root port, Device C performs the following tasks:</p> <ol style="list-style-type: none"> 1. Calculates a designated port configuration BPDU for Port C1 {0, 9, 2, Port C1}. 2. Compares it with the existing configuration BPDU of Port C1 {0, 0, 0, Port A2}. 3. Determines that the existing configuration BPDU is superior to the calculated one and blocks Port C1 with the configuration BPDU unchanged. <p>Port C1 does not forward data until a new event triggers a spanning tree calculation process: for example, the link between Device B and Device C is down.</p>	<ul style="list-style-type: none"> ■ Blocked port (Port C1): {0, 0, 0, Port A2} ■ Root port (Port C2): {0, 5, 1, Port B2}

3. After the comparison, a spanning tree with Device A as the root bridge is established as shown:

Figure 3 Device A as root bridge



STP supported platforms and standards

STP is supported on all AOS-CX switches.

Scale

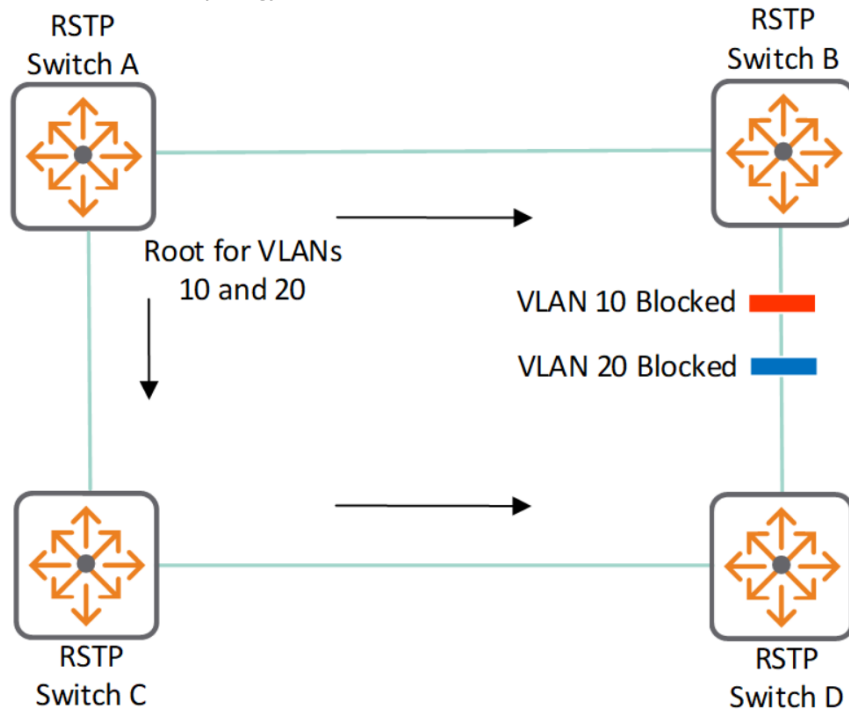
Platform	RPVST+ VLANs	RPVST+ vPorts	MSTP instances
8400	254	2048	64

RPVST+ protocol and feature details

Rapid Per VLAN Spanning Tree+ (RPVST+) is an updated implementation of STP (Spanning Tree Protocol). It enables the creation of a separate spanning tree for each VLAN on a switch, and ensures that only one active, loop-free path exists between any two nodes on a given VLAN.

Spanning tree protocols are used to prevent loops from occurring when multiple paths exist between the devices on a network. They are also used to provide redundancy, enabling data to use an alternative path when one link to a device fails. For example, in the following topology several paths exist between each switch.

Figure 4 RSTP topology with VLANs 10 and 20 blocked

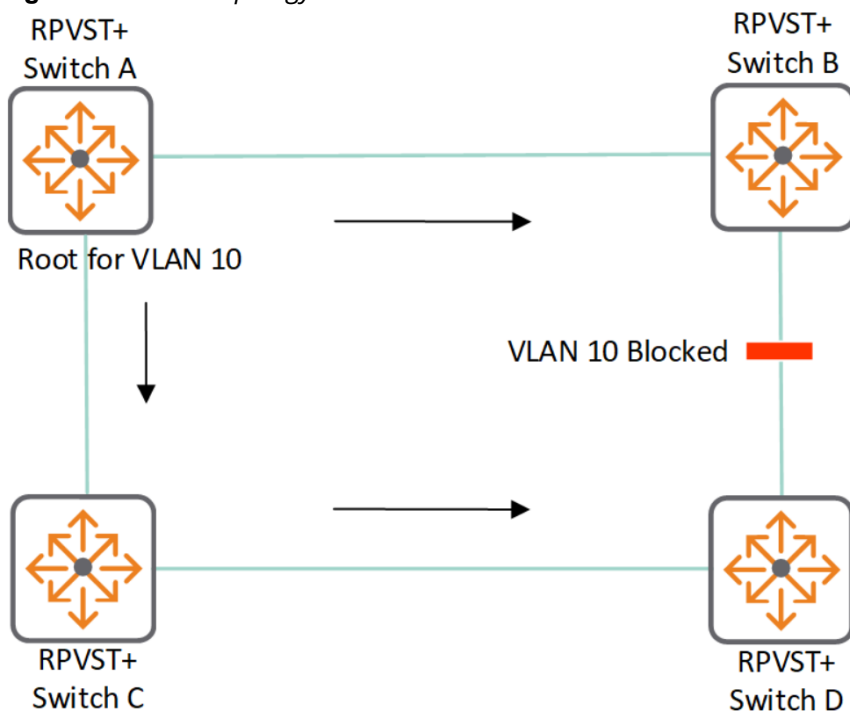


The above topology has four switches running RSTP. Switch "A" is the root switch. To prevent a loop, RSTP blocks the link between switch "B" and switch "D". There are two VLANs in this network (VLAN 10 and VLAN 20). Since RSTP does not have VLAN intelligence, it forces all VLANs in a layer 2 domain to follow the same spanning tree.

In the following topologies, there will not be any traffic through the link between switch "B" and switch "D" and therefore the link bandwidth is wasted. On the other hand, RPVST+ runs different spanning trees for different VLANs.

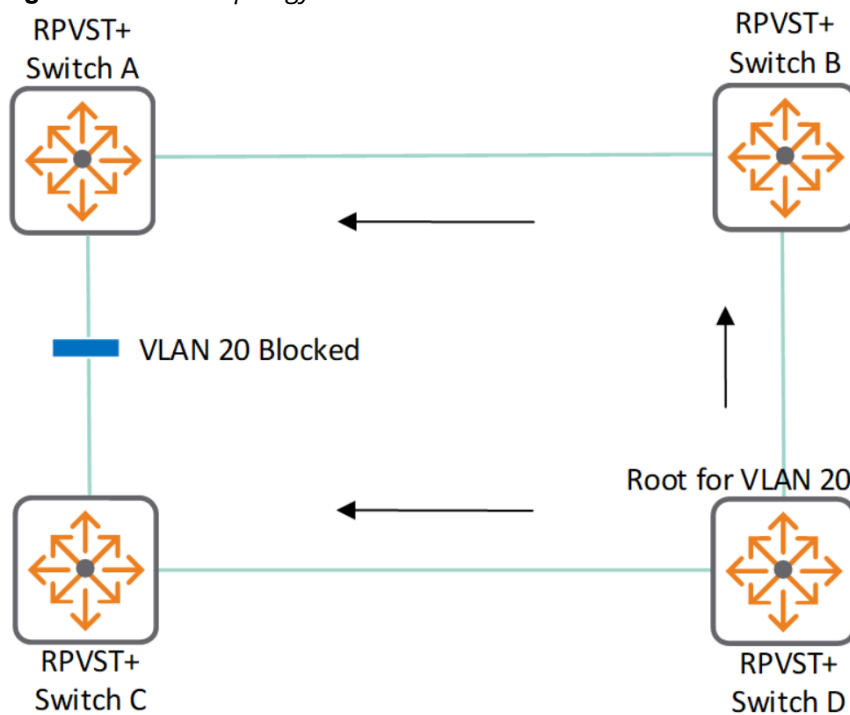
RPVST+ creates a spanning tree for VLAN 10.

Figure 5 *RPVST+ topology with VLAN 10 blocked*



RPVST+ creates another spanning tree for VLAN 20.

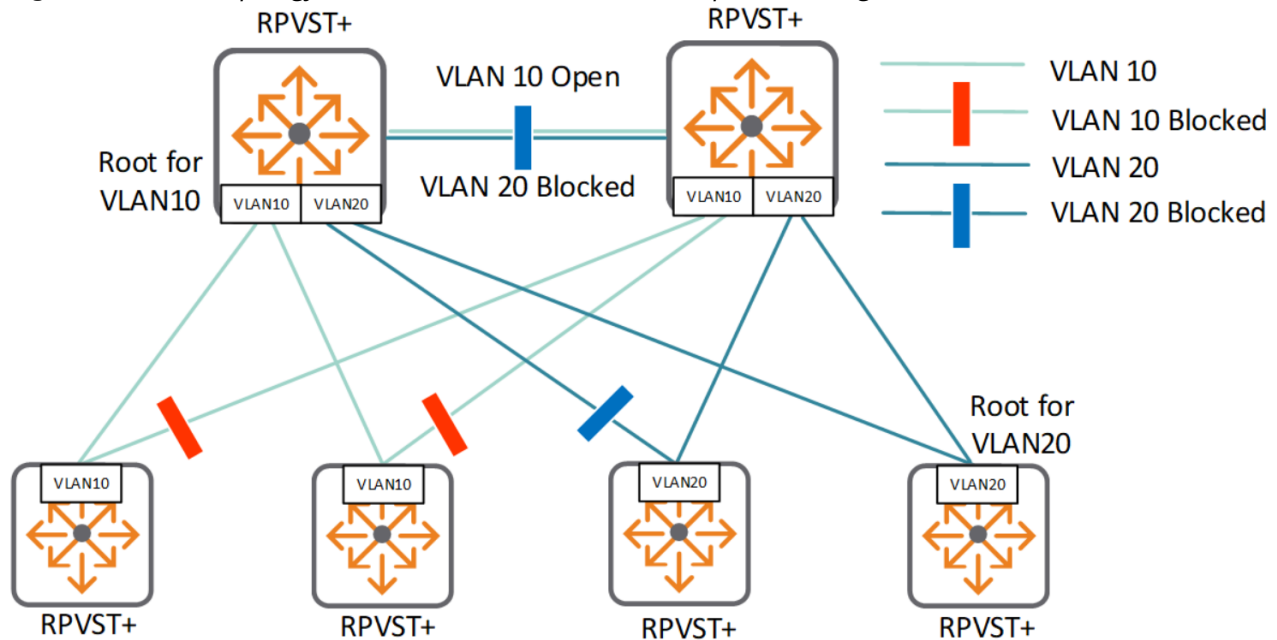
Figure 6 *RPVST+ topology with VLAN 20 blocked*



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.

The following topology shows a further example of shared links and redundant path-blocking in a network running RPVST+.

Figure 7 *RPVST+ topology with shared links and redundant path blocking*



In the factory default configuration, spanning tree operation is disabled. For STP, MSTP, and RSTP, configuration with default settings is automatic, and in many cases does not require any adjustments. Configuring spanning tree mode as RPVST+ and then enabling spanning tree, requires that spanning tree VLAN instances be enabled manually for the intended VLANs.

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.

RPVST+ interoperates with devices that run legacy IEEE 802.1D STP and MSTP-IEEE 802.1s.

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.



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.

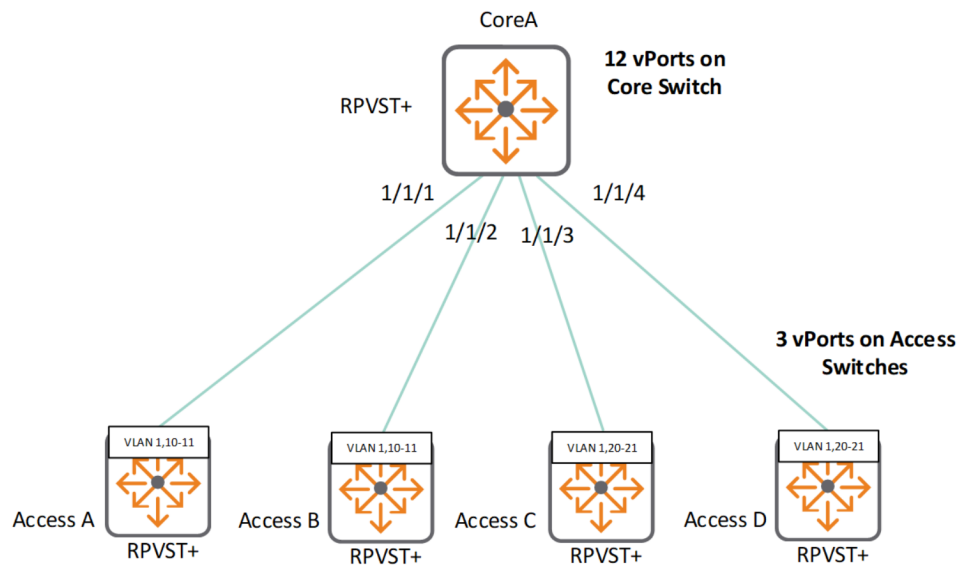
The switch supports 254 RPVST instances.

RPVST+ vPorts

When considering vPorts these are defined as active spanning tree VLANs that are declared on an interface. As vPorts increase the load on the CPU and switch resources increase as more BPDUs are processed.

In the following example core switch **CoreA** has four links with three VLANs on each link, for a total of 12 vPorts (4 links x 3 VLANs), allocated on the switch. The four access switches (**Access A** through **Access D**) use three vPorts each.

Figure 8 *RPVST+ topology with vPorts*



Configuration of switch **CoreA** seen in the above topology:

```
Hostname CoreA
vlan 1,10-11,20-21
spanning-tree mode rpvt
spanning-tree
spanning-tree vlan 1,10,11,20,21
interface 1/1/1
  no shutdown
  no routing
  vlan trunk native 1
  vlan trunk allowed 10-11
interface 1/1/2
  no shutdown
  no routing
  vlan trunk native 1
  vlan trunk allowed 10-11
interface 1/1/3
  no shutdown
  no routing
  vlan trunk native 1
  vlan trunk allowed 20-21
interface 1/1/4
  no shutdown
  no routing
  vlan trunk native 1
  vlan trunk allowed 20-21
```

RPVST+ configuration tasks

Procedure

1. Set RPVST+ as the spanning tree mode with the command `spanning-tree mode rpvst`.
2. Enable spanning tree with the command `spanning-tree`.
3. Configure the list of VLANs that are part of the spanning tree with the command `spanning-tree vlan`.
4. Set the priority for each VLAN with the command `spanning-tree vlan port-priority`. If you do not use this, STP will use the default priority.
5. Set the port cost and priority for each VLAN with the commands `spanning-tree vlan cost` and `spanning-tree vlan port-priority`. If you do not do this, STP will internally calculate port cost based on the link speed and set port priority to its default value.
6. For most deployments, the default values for the following settings do not need to be changed. If your deployment requires different settings, change the default values with the indicated commands:

RPVST+ setting	Default value	Command to change it
Include VLAN ID in spanning tree packets.	Enabled.	<code>spanning-tree extend-system-id</code>
Block links when VLAN mismatch is detected.	Disabled.	<code>spanning-tree ignore-pvid-inconsistency</code>
STP link type.	Point-to-point.	<code>spanning-tree link-type</code>
Support extended range of paths costs for high-speed links.	Enabled.	<code>spanning-tree pathcost-type</code>
Propagate topology changes to other ports.	Disabled.	<code>spanning-tree tcn-guard</code>

7. Review RPVST+ configuration settings with the command `show spanning tree`.

Example

This example creates the following configuration:

- Sets the spanning tree mode to **rpvst**.
- Enables spanning tree.
- Defines spanning tree support for VLANs **2-5**.
- Sets the priority for each VLAN.

```
switch# config
switch(config)# spanning-tree mode rpvst
switch(config)# spanning-tree
switch(config)# spanning-tree vlan 2-5
switch(config)# spanning-tree vlan 2 priority 5
switch(config)# spanning-tree vlan 3 priority 4
switch(config)# spanning-tree vlan 4 priority 3
```

```

switch(config)# spanning-tree vlan 5 priority 2
switch(config)# exit
switch# show spanning-tree

Spanning tree status          : Enabled Protocol: RPVST
Extended System-id           : Enabled
Ignore PVID Inconsistency    : Disabled
Path cost method              : Long
RPVST-MSTP Interconnect VLAN : 1
RPVST-Configured VLAN        : all
RPVST-Enabled VLAN           : 1
Current Virtual Ports Count   : 28
Maximum Allowed Virtual Ports : 2048

VLAN1
  Root ID    Priority    : 32768
             MAC-Address: 38:21:c7:5c:df:c0
             Hello time(in seconds):2  Max Age(in seconds):20
             Forward Delay(in seconds):15

VLAN2
  Root ID    Priority    : 20480
             MAC-Address: 70:72:cf:38:21:e5
             This bridge is the root
             Hello time(in seconds):2  Max Age(in seconds):20
             Forward Delay(in seconds):15

  Bridge ID   Priority    : 20480
             MAC-Address: 70:72:cf:38:21:e5
             Hello time(in seconds):2  Max Age(in seconds):20
             Forward Delay(in seconds):15

Port          Role          State          Cost          Priority    Type
-----
1/1/1         Designated   Forwarding    20000         128         point_to_point

VLAN3
  Root ID    Priority    : 16384
             MAC-Address: 70:72:cf:38:21:e5
             This bridge is the root
             Hello time(in seconds):2  Max Age(in seconds):20
             Forward Delay(in seconds):15

  Bridge ID   Priority    : 16384
             MAC-Address: 70:72:cf:38:21:e5
             Hello time(in seconds):2  Max Age(in seconds):20
             Forward Delay(in seconds):15

Port          Role          State          Cost          Priority    Type
-----
1/1/1         Designated   Forwarding    20000         128         point_to_point

VLAN4
  Root ID    Priority    : 12288
             MAC-Address: 70:72:cf:38:21:e5
             This bridge is the root
             Hello time(in seconds):2  Max Age(in seconds):20
             Forward Delay(in seconds):15

  Bridge ID   Priority    : 12288
             MAC-Address: 70:72:cf:38:21:e5
             Hello time(in seconds):2  Max Age(in seconds):20

```

Forward Delay(in seconds):15					
Port	Role	State	Cost	Priority	Type
1/1/1	Designated	Forwarding	20000	128	point_to_point
VLAN5					
Root ID	Priority : 8192 MAC-Address: 70:72:cf:38:21:e5 This bridge is the root Hello time(in seconds):2 Max Age(in seconds):20 Forward Delay(in seconds):15				
Bridge ID	Priority : 8192 MAC-Address: 70:72:cf:38:21:e5 Hello time(in seconds):2 Max Age(in seconds):20 Forward Delay(in seconds):15				
Port	Role	State	Cost	Priority	Type
1/1/1	Designated	Forwarding	20000	128	point_to_point

Viewing RPVST+ information

Prerequisites

These commands are in the manager context, as indicated by the `switch#` prompt.

Procedure

To view various aspects of RPVST+ information, use the following commands.

- To view information on spanning-tree mode and the RPVST+ instances, use:
`show spanning-tree`
- To view information on spanning-tree mode and the RPVST+ instance of a specific VLAN, use:
`show spanning-tree vlan`
- To view a summary of the spanning-tree configurations related to a port, use:
`show spanning-tree summary port`
- To view a summary of the spanning-tree configurations, use:
`show spanning-tree summary root`

RPVST+ Considerations and best practices

- For the best RPVST+ experience, use at least AOS-CX 10.07.
- As the number of VLANs increase in an RPVST+ environment the consumption of switch resources increases and you should therefore consider reducing VLAN sprawl. If VLAN increases are required and can be mapped sensibly, consider using MSTP.
- Do not exceed the available number of VLANs or vPorts supported on your switch platform.
- Check the number of RPVST+ VLANs and vPorts currently in use and the maximum number available using command `show capacities-status rpvst` like this:

```
switch# show capacities-status rpvst

System Capacities Status: Filter rpvst
Capacities Status Name                                     Value Maximum
-----
Number of RPVST VLANs currently configured                3          254
Number of RPVST Vports currently configured               9        2048
```

- Only select the VLANs required on a specific link for the allowed list based on the requirement on each port. For example for a link using VLANs 10,11,12 and 15 use command `vlan trunk 10-12,15`.
- To whatever degree possible, avoid using the catch all command `vlan trunk allowed all`.
- Topology Change Notifications (TCN) are an important part of STP. However, reducing unwanted TCNs is important for things such as access ports which can go up and down with end-point attachment and detachment at the network edge. It is recommended to use command `spanning-tree port-type admin-edge` to remove unwanted TCNs from end points.
- The use of spanning tree Topology Change Notification (TCN) guard may also be used in certain circumstances using command `spanning-tree tcn-guard`.
 - If the access switch is rebooting or the link between access and core switches is flapping, then this will cause TCNs towards the network core. Any TC on any interface on the core will clear all MACs locally and propagate the TC on all other interfaces. This can cause a significant traffic disruption on the network. If the network has a loop-free topology and mac-flush is not really needed on all switches in the network, then it can be feasible to add tcn-guard on access switches facing L2 interfaces. This will avoid mac-flush and TC propagation on the core switch (STP root switch).
 - If a core or aggregation switch in the network keeps getting TC messages due to unpredictable behavior of an access switch, TCN guard can be applied (using command `spanning-tree tcn-guard`) to the core or aggregation switch on the Layer 2 link facing the access switch.
- Stability in a spanning tree environment is paramount. It is recommended that default timers be used, and any alteration of timers be carried out only under special circumstances and in consultation with experts.
- Avoid automatic placement of root bridges. To enable a deterministic, predictable, and stable network, the placement of Primary and Secondary root bridges should be considered using command `spanning-tree vlan <VALUE> priority <VALUE>`.
- To further provide stability and deterministic behavior additional security configuration should be considered, such as:
 - **root-guard:** Sets a port to ignore superior BPDUs to prevent it from becoming the root port. This is typically carried out between the core that is required to be the root and access switches to prevent ports that are not expected to originate root information such as server ports and access

switch ports.

- **rpvst-guard:** Disables the specific port if the port receives RPVST+ BPDUs. This will be on well-defined ports that are known from your network design on which you never expect RPVST+ BPDUs. For example, user access ports or ports connected to servers in the datacenter where other switches may exist, and technicians can inadvertently patch into.
- **bpdu-guard:** Disables the specific port if the port receives STP BPDUs. This is done to prevent any inadvertent spanning tree or malicious attack, or switches being connected to the network and causing STP processing. This will be on well-defined ports that are known from your network design on which you never expect BPDUs. For example, user access ports or ports connected to servers in the datacenter where other switches may exist, and technicians can inadvertently patch into.
- With VSX configuration it is advisable that either the VSX pair acts as a STP root switch or that the STP root switch is reachable only through mc-lags. An STP root switch connected to a VSX pair with standalone interfaces (non-mc-lags) is not recommended.

RPVST+ use cases

RPVST+ use case: Deterministic root bridges

As mentioned in [RPVST+ Considerations and best practices](#), the placement of root bridges is important in the Layer 2 network domain. Having deterministic Root and Secondary Root bridges is a typically-accepted design that allows you to provide predictability and protection in your network .

The Root and Secondary root are typically placed at the Core of the Layer 2 domain. [Figure 9, Deterministic root bridges \(physical\)](#) shows the physical topology and [Figure 10, Deterministic root bridges \(logical\)](#) shows the logical topology. Switch A and Switch B are the core/center of the Layer 2 domain, and they provide root redundancy for each other.

Figure 9 *Deterministic root bridges (physical)*

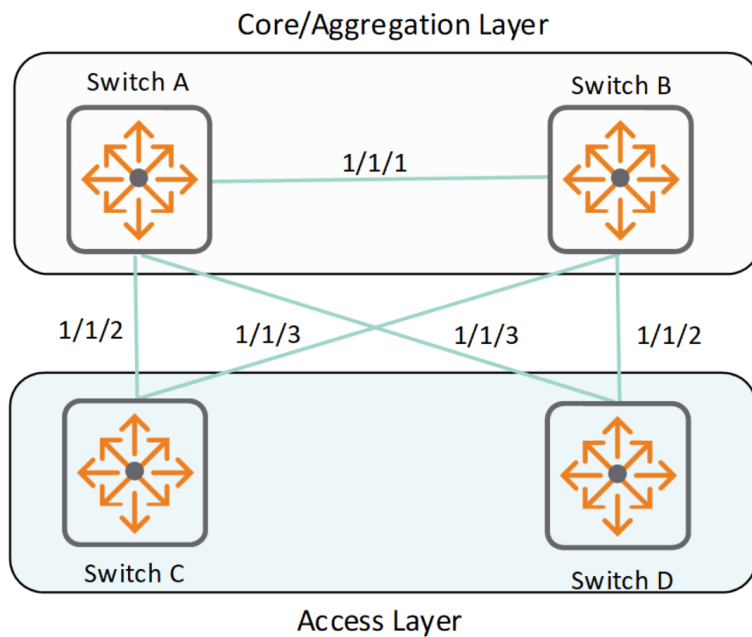
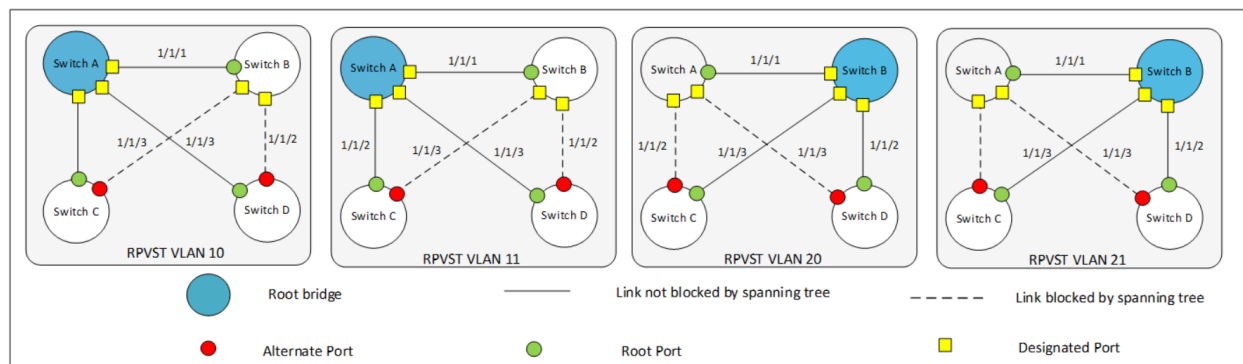


Figure 10 *Deterministic root bridges (logical)*



In this example network there are four VLANs and each VLAN has its own independent topology. The root bridges and VLANs are as follows:

- VLAN 10 Root bridge Switch A, Secondary Root bridge Switch B
- VLAN 11 Root bridge Switch A, Secondary Root bridge Switch B
- VLAN 20 Root bridge Switch B, Secondary Root bridge Switch A
- VLAN 21 Root bridge Switch B, Secondary Root bridge Switch A

Switches A through D are configured as follows:



In the following switch configuration command sequences, configuration portions (typically default) unrelated to RPVST+ are represented by an ellipsis "...". Also, descriptive comments, preceded by "<--", are included to the right of some commands.

Switch A configuration

- Add VLANs 10,11,20,21.
- Configure RPVST+ making Switch A the root for VLANs 10 and 11 and the secondary root for VLANs 20, 21.
- Allow the required VLANs for interfaces 1/1/1 to 1/1/3.

```
SwitchA#
...
vlan 10-11,20-21
spanning-tree mode rpvst          <-- Enable RPVST+
spanning-tree
spanning-tree vlan 10-11,20-21    <-- Define VLANs for RPVST+
spanning-tree vlan 10 priority 1  <-- Make Switch A Root Bridge for VLANs
spanning-tree vlan 11 priority 1
spanning-tree vlan 20 priority 2  <-- Make Switch A Secondary Root Bridge for
VLANs
spanning-tree vlan 21 priority 2
...
interface 1/1/1                   <-- Allocate required VLANs to interface
vlan trunk 10-11,20-21
vlan trunk native 1
interface 1/1/2
vlan trunk 10-11,20-21
vlan trunk native 1
interface 1/1/3
vlan trunk 10-11,20-21
vlan trunk native 1
...
```

Switch B configuration

- Add VLANs 10,11,20,21.
- Configure RPVST+ making Switch B the root for VLANs 20, 21, and the secondary root for VLANs 10, 11.
- Configure the trunk-required VLANs for interfaces 1/1/1 to 1/1/3.

```
SwitchB#
...
```

```

vlan 10-11,20-21
spanning-tree mode rpvt
spanning-tree
spanning-tree vlan 10-11,20-21
spanning-tree vlan 10 priority 2
spanning-tree vlan 11 priority 2
spanning-tree vlan 20 priority 1
spanning-tree vlan 21 priority 1
...
interface 1/1/1
vlan trunk 10-11,20-21
vlan trunk native 1
interface 1/1/2
vlan trunk 10-11,20-21
vlan trunk native 1
interface 1/1/3
vlan trunk 10-11,20-21
vlan trunk native 1
...

```

Switch C and D configuration

- Define the VLANs for RPVST+ and the trunk-required VLANs using the same configuration on both C and D except for the hostname.

```

vlan 10-11,20-21
exit
spanning-tree mode rpvt
spanning-tree
spanning-tree vlan 10-11,20-21
int 1/1/2-1/1/3
vlan trunk 10-11,20-21
vlan trunk native 1
exit
...
vlan 10-11,20-21
spanning-tree mode rpvt
spanning-tree
spanning-tree vlan 10-11,20-21
...
interface 1/1/2
vlan trunk 10-11,20-21
vlan trunk native 1
interface 1/1/3
vlan trunk 10-11,20-21
vlan trunk native 1
...

```

Checking the configuration

The applied configurations can be checked as follows:

- Checking RPVST+
- Checking that the System ID matches Root for the VLAN.

Checking Switch A

Use command `show spanning-tree summary root`. As seen here, Switch A is Root for VLAN 10 and 11, identified by the System ID, and VLAN 20 and 21 Root is another device which is expected to be Switch B based on previous configurations.

Notice the zero Root Port cost indicated in the first two rows of output.

```
SwitchA#show spanning-tree summary root
STP status      : Enabled
Protocol        : RPVST
System ID       : 08:00:09:8a:14:fa    <-- System ID
Root bridge for VLANs : 10,11        <-- Identify root bridges for VLANs
```

VLAN	Priority	Root ID	Root cost	Hello Time	Max Age	Fwd Dly	Root Port
VLAN10	4096	08:00:09:8a:14:fa	0	2	20	15	0
VLAN11	4096	08:00:09:8a:14:fa	0	2	20	15	0
VLAN20	4096	08:00:09:12:8e:9e	20000	2	20	15	1/1/1
VLAN21	4096	08:00:09:12:8e:9e	20000	2	20	15	1/1/1

Checking Switch B

As seen here, Switch B is Root for VLAN 20 and 21 identified by the System ID, and VLAN 10 and 11 Root is Switch A identified by the System ID.

```
SwitchA#show spanning-tree summary root
STP status      : Enabled
Protocol        : RPVST
System ID       : 08:00:09:12:8e:9e

Root bridge for VLANs : 20,21
```

VLAN	Priority	Root ID	Root cost	Hello Time	Max Age	Fwd Dly	Root Port
VLAN10	4096	08:00:09:8a:14:fa	20000	2	20	15	1/1/1
VLAN11	4096	08:00:09:8a:14:fa	20000	2	20	15	1/1/1
VLAN20	4096	08:00:09:12:8e:9e	0	2	20	15	0
VLAN21	4096	08:00:09:12:8e:9e	0	2	20	15	0

Checking Switches C and D

Although not illustrated, Switches C and D can be checked in a similar manner to the other switches.

Observe port behavior and state

We can observe the port behavior and state using command `show spanning-tree`. The topology in [Figure 10, Deterministic root bridges \(logical\)](#) for each switch can be observed showing a loop free Layer 2 topology. The following command sequences focus on VLAN 10.

Observing Switch A for VLAN 10

Use command `show spanning-tree`. As seen here, all ports shown on Switch A, the Root Bridge for VLAN 10, are **Designated Forwarding** as expected.

```
SwitchA# show spanning-tree vlan 10

VLAN10
Spanning tree status : Enabled Protocol: RPVST
```

```

Root ID    Priority    : 4096
MAC-Address: 08:00:09:8a:14:fa
This bridge is the root
Hello time(in seconds):2  Max Age(in seconds):20
Forward Delay(in seconds):15

Bridge ID  Priority    : 4096
MAC-Address: 08:00:09:8a:14:fa
Hello time(in seconds):2  Max Age(in seconds):20
Forward Delay(in seconds):15

Port      Role      State      Cost      Priority  Type      BPDU-Tx    BPDU-Rx    TCN-Tx     TCN-Rx
-----
1/1/1     Designated Forwarding 20000      128       P2P       2586       533        10         8
1/1/2     Designated Forwarding 20000      128       P2P       2679       434        5          7
1/1/3     Designated Forwarding 20000      128       P2P       3106       5          6          2

Number of topology changes : 6
Last topology change occurred : 4828 seconds ago

```

Observing Switch B for VLAN 10

As seen here, the Root Bridge for VLAN 10 is identified by its MAC address “08:00:09:8a:14:fa” which is Switch A. The port connecting to Switch A 1/1/1 is the Root port and **Forwarding** and the other two ports are **Designated Forwarding** leading to Switch C and D respectively. All ports follow the VLAN 10 topology (as seen in [Use case: Deterministic root bridges example network](#)) as expected.

```

SwitchB# show spanning-tree vlan 10

VLAN10
Spanning tree status : Enabled Protocol: RPVST
Root ID    Priority    : 4096
MAC-Address: 08:00:09:8a:14:fa
Hello time(in seconds):2  Max Age(in seconds):20
Forward Delay(in seconds):15

Bridge ID  Priority    : 8192
MAC-Address: 08:00:09:12:8e:9e
Hello time(in seconds):2  Max Age(in seconds):20
Forward Delay(in seconds):15

Port      Role      State      Cost      Priority  Type      BPDU-Tx    BPDU-Rx    TCN-Tx     TCN-Rx
-----
1/1/1     Root      Forwarding 20000      128       P2P       537        2770       8          9
1/1/2     Designated Forwarding 20000      128       P2P       3298       7          6          2
1/1/3     Designated Forwarding 20000      128       P2P       3298       9          9          3

Number of topology changes : 3
Last topology change occurred : 5247 seconds ago

```

Observing Switch C for VLAN 10

As seen here, the Root Bridge for VLAN 10 is identified by its MAC address "08:00:09:8a:14:fa" which is Switch A. The port connecting to Switch A 1/1/2 is the Root port and **Forwarding**, and the other port 1/1/3 towards Switch B is **Alternate Blocking** preventing a looped topology for VLAN 10. Although not illustrated, Switch D can be observed in a similar manner.

```
SwitchC# show spanning-tree vlan 10

VLAN10
Spanning tree status : Enabled Protocol: RPVST
Root ID    Priority    : 4096
           MAC-Address: 08:00:09:8a:14:fa
           Hello time(in seconds):2 Max Age(in seconds):20
           Forward Delay(in seconds):15

Bridge ID   Priority    : 32768
           MAC-Address: 08:00:09:16:7b:7e
           Hello time(in seconds):2 Max Age(in seconds):20
           Forward Delay(in seconds):15

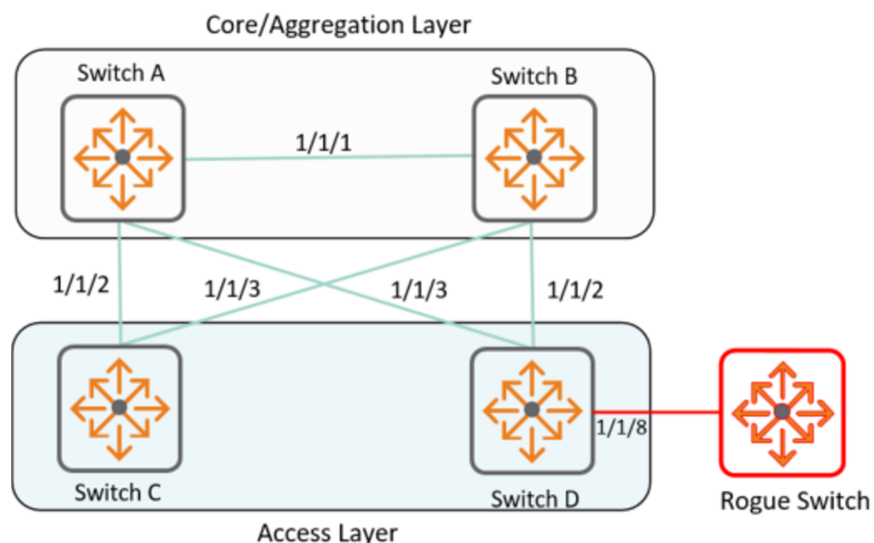
Port        Role        State        Cost        Priority    Type        BPDU-Tx    BPDU-Rx    TCN-Tx    TCN-Rx
-----
1/1/2       Root        Forwarding   20000        128        P2P         438        3553       7         4
1/1/3       Alternate   Blocking     20000        128        P2P         9          3986       3         8

Number of topology changes : 5
Last topology change occurred : 6811 seconds ago
```

RPVST+ use case: BPDU protection

Various security mechanisms are in place to protect spanning tree configurations from interference and rogue devices or unwarranted changes to the network. BPDU protection secures the active topology by preventing spoofed BPDU packets from entering the network. Typically, BPDU protection is applied on edge ports connected to end user devices that do not run STP. If STP BPDU packets are received on a protected port, BPDU guard disables the port and an alert is sent. As shown in [Figure 11, Rogue device needing BPDU guard](#) we have a rogue device attempting to connect to Switch D port 1/1/8.

Figure 11 *Rogue device needing BPDU guard*



BPDU guard is configured on switch D.

```
SwitchD#
config
interface 1/1/8
    no shutdown
    no routing
    vlan access 10
    spanning-tree bpdu-guard
exit
```

Use command `show spanning-tree summary vlan 10` to observe that port 1/1/8 is disabled because BPDU was received on it from the rogue switch.

Notice how port 1/1/8 is disabled due to "Bpdu-Error." A timeout can be configured to re-enable the port.

```
SwitchD# show spanning-tree vlan 10

VLAN10
Spanning tree status : Enabled Protocol: RPVST
  Root ID    Priority    : 4096
             MAC-Address: 08:00:09:8a:14:fa
             Hello time(in seconds):2  Max Age(in seconds):20
             Forward Delay(in seconds):15

  Bridge ID  Priority    : 32768
             MAC-Address: 08:00:09:ee:11:82
             Hello time(in seconds):2  Max Age(in seconds):20
             Forward Delay(in seconds):15

Port      Role      State      Cost      Priority  Type      BPDU-Tx  BPDU-Rx  TCN-Tx  TCN-Rx
-----
1/1/2     Root      Forwarding 20000     128      P2P      580      1237     400     395
1/1/3     Alternate Blocking 40001     128      P2P      214      1057     212     303
1/1/8     Disabled  Bpdu-Error 20000     128      P2P      81       0        0        0

Number of topology changes : 307
Last topology change occurred : 2 seconds ago
```

Use command `show int 1/1/8` to observe the interface state. Notice that port 1/1/8 is down as expected due to BPDU error.

```
SwitchD#show int 1/1/8
Interface 1/1/8 is down
Admin state is up
State information:
Link state: down
Link transitions: 0
Description:
Hardware: Ethernet, MAC Address: 08:00:09:ee:11:c4
MTU 1500
Type --
Full-duplex
qos trust none
Speed 1000 Mb/s
Auto-negotiation is off
Flow-control: off
Error-control: off
MDI mode: none
VLAN Mode: access
```

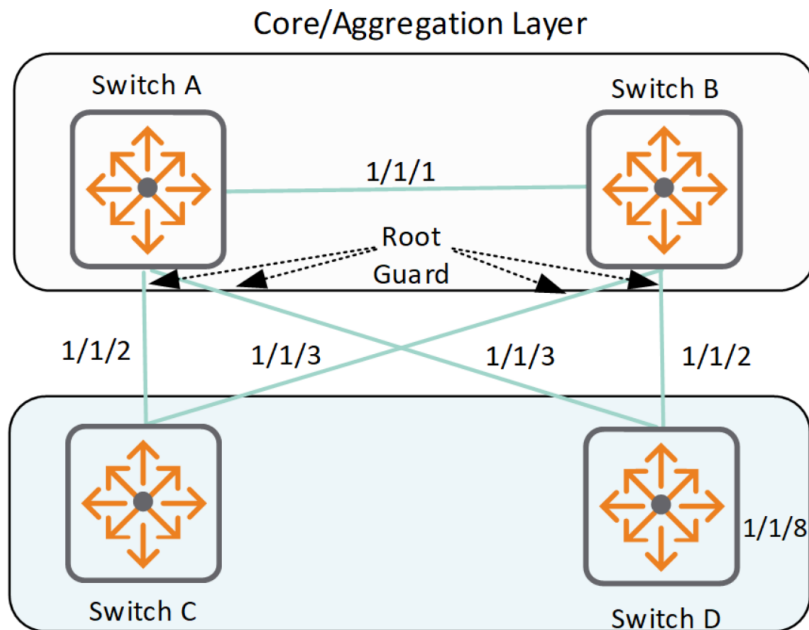
Access VLAN: 10

RPVST+ use case: Root protection

Root protection secures the active topology by preventing other switches from declaring their ability to propagate superior BPDUs, containing both better information on the root bridge and path cost to the root bridge which would normally replace the current root bridge selection.

As illustrated in [Figure 12, Root protection](#), by adding root guard on interfaces 1/1/2 and 1/1/3 of both core switches (A and B), these two switches are protected in the core and allow propagation of superior BPDUs from the access layer.

Figure 12 Root protection



Configuring Switches A and B:

```
SwitchA#  
config  
interface 1/1/2  
    spanning-tree root-guard  
exit  
interface 1/1/3  
    spanning-tree root-guard  
exit
```

```
SwitchB#  
Config  
interface 1/1/2  
    spanning-tree root-guard  
interface 1/1/3  
    spanning-tree root-guard  
exit
```

To observe the protection behavior, we can (inappropriately) make switch C the root for VLAN 10.

```
SwitchC#  
config
```

```
spanning-tree vlan 10 priority 0 <-- Make Switch C Root for VLAN 10
exit
```

Notice how as protection occurs, VLAN 10 on both Switch A and B ports show as **Alternate Root-Inc** (Alternate Root-Inconsistent). This action maintains Layer 2 stability by protecting the rest of the network from the (inaccurate) information that Switch C is sending “better” BPDUs.

```
SwitchA# show spanning-tree vlan 10
```

```
VLAN10
```

```
Spanning tree status : Enabled Protocol: RPVST
```

```
Root ID    Priority    : 4096
           MAC-Address: 08:00:09:8a:14:fa
           This bridge is the root
           Hello time(in seconds):2  Max Age(in seconds):20
           Forward Delay(in seconds):15
```

```
Bridge ID  Priority    : 4096
           MAC-Address: 08:00:09:8a:14:fa
           Hello time(in seconds):2  Max Age(in seconds):20
           Forward Delay(in seconds):15
```

Port	Role	State	Cost	Priority	Type	BPDU-Tx	BPDU-Rx	TCN-Tx	TCN-Rx
1/1/1	Designated	Forwarding	20000	128	P2P	1606	383	432	159
1/1/2	Alternate	Root-Inc	20000	128	P2P	1571	114	520	92
1/1/3	Designated	Forwarding	20000	128	P2P	1567	172	447	167

```
Number of topology changes    : 694
```

```
Last topology change occurred : 1 seconds ago
```

```
SwitchB# show spanning-tree vlan 10
```

```
VLAN10
```

```
Spanning tree status : Enabled Protocol: RPVST
```

```
Root ID    Priority    : 4096
           MAC-Address: 08:00:09:8a:14:fa
           Hello time(in seconds):2  Max Age(in seconds):20
           Forward Delay(in seconds):15
```

```
Bridge ID  Priority    : 8192
           MAC-Address: 08:00:09:12:8e:9e
           Hello time(in seconds):2  Max Age(in seconds):20
           Forward Delay(in seconds):15
```

Port	Role	State	Cost	Priority	Type	BPDU-Tx	BPDU-Rx	TCN-Tx	TCN-Rx
1/1/1	Designated	Learning	20000	128	P2P	1127	551	608	125
1/1/2	Root	Forwarding	20000	128	P2P	1865	354	569	187
1/1/3	Alternate	Root-Inc	20000	128	P2P	1717	479	627	88

```
Number of topology changes    : 763
```

```
Last topology change occurred : 2 seconds ago
```

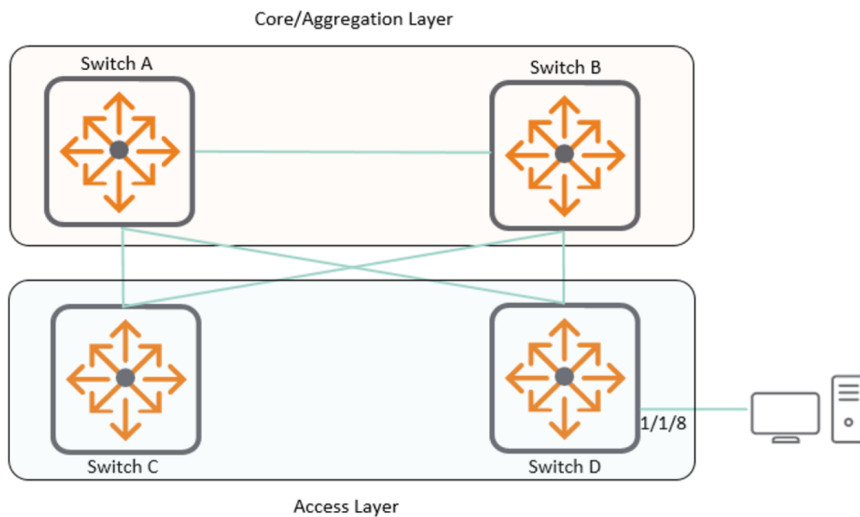


Depending on when the `show` command is executed, it may first show the protected port as **Designated Blocking** before it shows it as **Alternate Root-Inc**.

RPVST+ use case: Spanning tree on edge ports

When using spanning tree and taking into consideration the edge of the network ports that provide connectivity to end points, the network should not typically participate in spanning tree. Consider this topology that shows an endpoint connected to port 1/1/8 on Switch D:

Figure 13 *Spanning tree on edge ports*



End points that connect to ports that do participate in spanning tree (STP) may experience DHCP assignment timeouts or IP address assignment delays plus extended client onboarding time and authentication issues. These problems occur because the port participates in the full STP process. To avoid such issues consider setting the port as a spanning tree administrative edge port by using command **spanning-tree port-type admin-edge**. This command removes the port participation from STP interactions when onboarding devices, enabling quicker onboarding.



Edge ports still need to be protected from possible spanning tree attacks. For example BPDU guard can be used. See [RPVST+ use case: BPDU protection](#).

Before configuring a port as spanning tree administrative edge, the port configuration looks like this:

```
interface 1/1/8
  no shutdown
  vlan access 10
  spanning-tree bpdu-guard
```

The port State is **Forwarding** and the Type is **P2P** (Point to Point) by default.

```
switch# show spanning-tree vlan 10
```

Port	Role	State	Cost	Priority	Type	BPDU-Tx	BPDU-Rx	TCN-Tx	TCN-Rx
1/1/8	Designated	Forwarding	2000000	128	P2P	167	0	0	0

Configure the port as admin edge as follows with command **spanning-tree port-type admin-edge**:

```
interface 1/1/8
  no shutdown
  vlan access 10
  spanning-tree bpdu-guard
  spanning-tree port-type admin-edge
exit
```

Notice how that the port State is now **Forwarding** and the Type is **P2P Edge** meaning that the port will go into the forwarding state and bypass the standard STP listening and learning states.

```
switch# show spanning-tree vlan 10show spanning-tree vlan 10
```

Port	Role	State	Cost	Priority	Type	BPDU-Tx	BPDU-Rx	TCN-Tx	TCN-Rx
1/1/8	Designated	Forwarding	2000000	128	P2P Edge	347	0	0	0

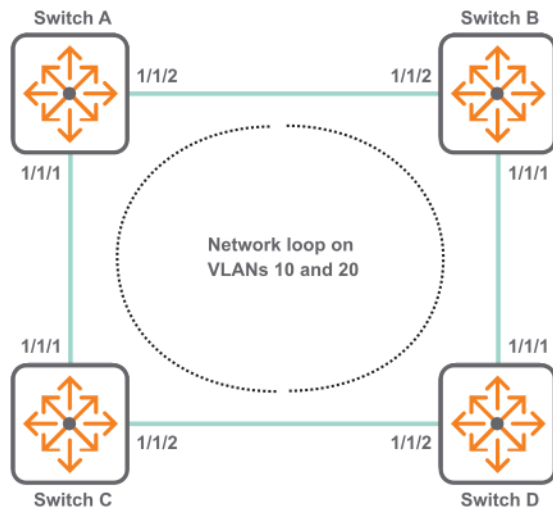
```
Number of topology changes : 0
```

```
Last topology change occurred : 0 seconds ago
```

RPVST+ use case: Preventing loops

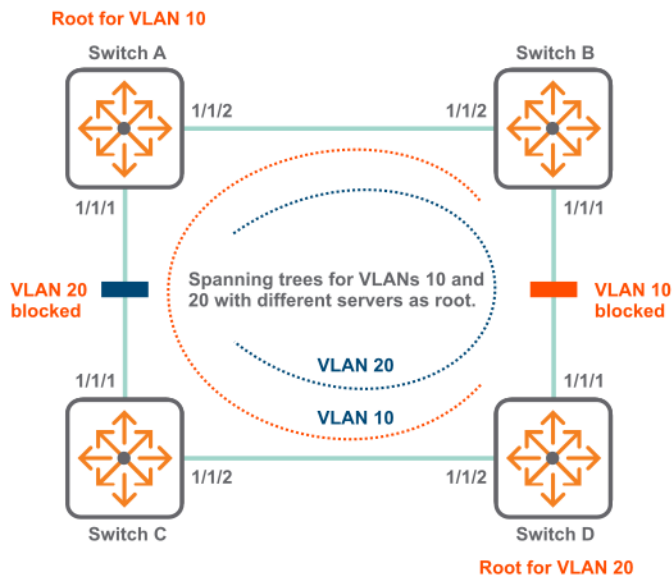
In this scenario, four switches are interconnected. VLANs 10 and 20 are defined on all switches, causing a network loop.

Figure 14 *Topology with an undesired loop*



To eliminate the loop, RPVST+ is enabled and switch A and B are defined as high-priority for VLAN 10 and 20 respectively. RPVST+ then eliminates the loop by assigning switch A as the root for VLAN 10 and switch B designated as the root for VLAN 20, and blocking access on one of the links.

Figure 15 *Topology with loop eliminated by RPVST+*



Procedure

1. Configure switch A.
 - a. Create VLANs 1, 10, and 20.

```
switch# config
switch(config)# vlan 1, 10, 20
```

- b. Enable RPVST+ and assign the VLANs 10 and 20 to it. Assign a priority of 5 to VLAN 10. This will force switch A to become the root of the spanning tree for VLAN 10.

```
switch(config)# spanning-tree mode rpvtst
switch(config)# spanning-tree
switch(config)# spanning-tree vlan 10,20
switch(config)# spanning-tree vlan 10 priority 5
```

- c. Define interfaces 1/1/1 and 1/1/2.

```
switch(config)# interface 1/1/1
switch(config-if)# no shutdown
switch(config-if)# no routing
switch(config-if)# vlan trunk native 1
switch(config-if)# vlan trunk allowed all
switch(config-if)# interface 1/1/2
switch(config-if)# no shutdown
switch(config-if)# no routing
switch(config-if)# vlan trunk native 1
switch(config-if)# vlan trunk allowed all
```

2. Configure switch B and switch C with the same settings.

- a. Create VLANs 1, 10, and 20.

```
switch# config
switch(config)# vlan 1, 10, 20
```

- b. Enable RPVST+ and assign the VLANs 10 and 20 to it.

```
switch(config)# spanning-tree mode rpvtst
switch(config)# spanning-tree
switch(config)# spanning-tree vlan 10,20
```

- c. Define interfaces 1/1/1 and 1/1/2.

```
switch(config)# interface 1/1/1
switch(config-if)# no shutdown
switch(config-if)# no routing
switch(config-if)# vlan trunk native 1
switch(config-if)# vlan trunk allowed all
switch(config-if)# interface 1/1/2
switch(config-if)# no shutdown
switch(config-if)# no routing
switch(config-if)# vlan trunk native 1
switch(config-if)# vlan trunk allowed all
```

3. Configure switch D.

- a. Create VLANs 1, 10, and 20.

```
switch# config
switch(config)# vlan 1, 10, 20
```

- b. Enable RPVST+ and assign the VLANs 10 and 20 to it. Assign a priority of 5 to VLAN 20. This will force switch D to become the root of the spanning tree for VLAN 20.

```
switch(config)# spanning-tree mode rpvtst
switch(config)# spanning-tree
switch(config)# spanning-tree vlan 10,20
switch(config)# spanning-tree vlan 20 priority 5
```

- c. Define interfaces 1/1/1 and 1/1/2.

```
switch(config)# interface 1/1/1
switch(config-if)# no shutdown
switch(config-if)# no routing
switch(config-if)# vlan trunk native 1
switch(config-if)# vlan trunk allowed all
switch(config-if)# interface 1/1/2
switch(config-if)# no shutdown
switch(config-if)# no routing
switch(config-if)# vlan trunk native 1
```

```
switch(config-if)# vlan trunk allowed all
```

RPVST+ commands

clear spanning-tree statistics

clear spanning-tree statistics [VLAN-ID]

Description

Clears the spanning tree BPDU statistics, either all statistics or those related to a specified VLAN.

Parameter	Description
VLAN-ID	Specifies the VLAN ID.

Example

Clearing all spanning tree BPDU statistics:

```
switch(config)# clear spanning-tree statistics
```

Clearing spanning tree BPDU statistics for a particular VLAN :

```
switch(config)# clear spanning-tree statistics 10
```

Command History

Release	Modification
10.07 or earlier	--

Command Information

Platforms	Command context	Authority
All platforms	config	Administrators or local user group members with execution rights for this command.

show capacities rpvst

show capacities rpvst

Description

Shows the capacities of RPVST VLANs configurable on a system and RPVST VPORTs supported in a system.

Examples

Showing capacities on aswitch:

```
switch# show capacities rpvst
System Capacities : Filter RPVST
```


Capacities Name	Value
Maximum number of RPVST VLANs configurable on the system	512
Maximum number of RPVST VPORTs supported in a system	2048

Command History

Release	Modification
10.07 or earlier	--

Command Information

Platforms	Command context	Authority
All platforms	Operator (>) or Manager (#)	Operators or Administrators or local user group members with execution rights for this command. Operators can execute this command from the operator context (>) only.

show capacities-status rpvst

show capacities-status rpvst

Description

Shows the number of RPVST VLANs and RPVST VPORTs currently configured.

Examples

Showing capacities-status on aswitch:

```
switch# show capacities-status rpvst
System Capacities Status : Filter RPVST
Capacities Status Name      Value      Maximum
-----
Number of RPVST VLANs configured    3          254
Number of RPVST VPORTs configured   9          2048
```

Command History

Release	Modification
10.07 or earlier	--

Command Information

Platforms	Command context	Authority
All platforms	Operator (>) or Manager (#)	Operators or Administrators or local user group members with execution rights for this command. Operators can execute this command from the operator context (>) only.

show spanning-tree

```
show spanning-tree [vsx-peer]
```

Description

Shows the spanning tree mode and information on the RPVST instances.

When Port security is enabled on the port and the client is not-yet authenticated, the security feature keeps the port in the `Down` state. STP also keeps the port in the `Blocking` state and the role as `Disabled` in the `show spanning-tree` command output, whereas in the hardware, the state is maintained as `Learning`. After client authentication is successful, the port state changes to `Forwarding`.

Parameter	Description
vsx-peer	Shows the output from the VSX peer switch. If the switches do not have the VSX configuration or the ISL is down, the output from the VSX peer switch is not displayed. This parameter is available on switches that support VSX.

Examples

Showing spanning tree mode and RPVST instance information:

```
switch# show spanning-tree
Spanning tree status      : Enabled Protocol: RPVST
Extended System-id       : Enabled
Ignore PVID Inconsistency : Enabled
Path cost method          : Long
RPVST-MSTP Interconnect VLAN : 1
Current Virtual Ports Count : 0
Maximum Allowed Virtual Ports : 2048

VLAN1
  Root ID    Priority    : 32768
             MAC-Address: 70:72:cf:31:c9:23
             This bridge is the root
             Hello time(in seconds):2  Max Age(in seconds):20
             Forward Delay(in seconds):15

  Bridge ID  Priority    : 32768
             MAC-Address: 70:72:cf:31:c9:23
             Hello time(in seconds):2  Max Age(in seconds):20
             Forward Delay(in seconds):15

PORT      ROLE        STATE        COST    PRIORITY  TYPE        BPDU-Tx  BPDU-Rx  TCN-Tx  TCN-Rx
-----
1/1/1     Designated Forwarding  20000    128      P2P Edge    100      60      20      10
1/1/2     Designated Forwarding  20000    128      P2P Edge    100      60      20      10
1/1/3     Designated Forwarding  20000    128      Shr         100      60      20      10
1/1/4     Designated Forwarding  20000    128      Shr Edge    100      60      20      10
1/1/5     Alternate Loop-Inc    20000    128      Shr Edge    100      60      20      10
1/1/6     Alternate Root-Inc    20000    128      Shr Edge    100      60      20      10
1/1/7     Disabled   Down         20000    128      P2P         100      60      20      10

Number of topology changes : 4
Last topology change occurred : 516 seconds ago
```

Command History

Release	Modification
10.09	A new state <code>Down</code> is added in the output.
10.07 or earlier	--

Command Information

Platforms	Command context	Authority
All platforms	Operator (>) or Manager (#)	Operators or Administrators or local user group members with execution rights for this command. Operators can execute this command from the operator context (>) only.

show spanning-tree detail

`show spanning-tree detail [vsx-peer]`

Description

Shows the detailed spanning tree mode and information on the RPVST instances.

When Port security is enabled on the port and the client is not-yet authenticated, the security feature keeps the port in the `Down` state. STP also keeps the port in the `Blocking` state and the role as `Disabled` in the `show spanning-tree` command output, whereas in the hardware, the state is maintained as `Learning`. After client authentication is successful, the port state changes to `Forwarding`.

Parameter	Description
<code>vsx-peer</code>	Shows the output from the VSX peer switch. If the switches do not have the VSX configuration or the ISL is down, the output from the VSX peer switch is not displayed. This parameter is available on switches that support VSX.

Examples

Showing spanning tree mode and detailed RPVST instance information:

```
switch# show spanning-tree detail
Spanning tree status      : Enabled Protocol: RPVST
Extended System-id       : Enabled
Ignore PVID Inconsistency : Enabled
Path cost method          : Long
RPVST-MSTP Interconnect VLAN : 1
Current Virtual Ports Count : 0
Maximum Allowed Virtual Ports : 2048

VLAN1
  Root ID    Priority    : 32768
             MAC-Address: 70:72:cf:31:c9:23
             This bridge is the root
             Hello time(in seconds):2   Max Age(in seconds):20
             Forward Delay(in seconds):15

  Bridge ID  Priority    : 32768
             MAC-Address: 70:72:cf:31:c9:23
             Hello time(in seconds):2   Max Age(in seconds):20
```

```
Forward Delay(in seconds):15

PORT      ROLE      STATE      COST      PRIORITY  TYPE      BPDU-Tx  BPDU-Rx  TCN-Tx    TCN-Rx
-----
1/1/1     Designated Forwarding 20000     128       P2P Edge  100       60       20       10
1/1/2     Designated Forwarding 20000     128       P2P      100       60       20       10
1/1/3     Designated Forwarding 20000     128       Shr      100       60       20       10
1/1/4     Designated Forwarding 20000     128       Shr Edge  100       60       20       10
1/1/5     Alternate  Loop-Inc  20000     128       Shr Edge  100       60       20       10
1/1/6     Alternate  Root-Inc  20000     128       Shr Edge  100       60       20       10
1/1/7     Disabled   Down      20000     128       P2P      100       60       20       10

Topology change flag : False
Number of topology changes : 1
Last topology change occurred : 33293 seconds ago

Port 1/1/1
Designated Root Priority      : 32768      Address: 48:0F:CF:AF:22:1D
Designated Bridge Priority    : 32768      Address: 48:0F:CF:AF:22:1D
Designated Port               : 1/1/1
Forwarding-State transitions  : 0
BPDUs sent 1582, received 1506
TCN_Tx: 10, TCN_Rx: 10

Port lag1
Designated Root Priority      : 32768      Address: 48:0F:CF:AF:22:1D
Designated Bridge Priority    : 32768      Address: 48:0F:CF:AF:22:1D
Designated Port               : lag1
Forwarding-State transitions  : 0
BPDUs sent 1402, received 1316
TCN_Tx: 10, TCN_Rx: 10
Multi-chassis role           : active
```

Command History

Release	Modification
10.09	A new state Down is added in the output.
10.07 or earlier	--

Command Information

Platforms	Command context	Authority
All platforms	Operator (>) or Manager (#)	Operators or Administrators or local user group members with execution rights for this command. Operators can execute this command from the operator context (>) only.

show spanning-tree inconsistent-ports

show spanning-tree inconsistent-ports [vlan <VLAN-ID>]

Description

Shows ports blocked by STP protection functions such as Root guard, Loop guard, BPDU guard, and RPVST guard.

Parameter	Description
<VLAN-ID>	Specifies a VLAN ID number.

Examples

Showing inconsistent port information:

```
switch# show spanning-tree inconsistent-ports
VLAN ID      Blocked Port      Reason
-----
1             1/1/1             BPDU Guard
2             1/1/1             BPDU Guard
3             1/1/1             BPDU Guard
4             1/1/1             BPDU Guard
5             1/1/1             BPDU Guard
6             1/1/1             BPDU Guard
7             1/1/1             BPDU Guard
8             1/1/1             BPDU Guard
9             1/1/1             BPDU Guard
10            1/1/1             BPDU Guard
```

Showing inconsistent port information for VLANs 1 to 4:

```
switch# show spanning-tree inconsistent-ports vlan 1-4
VLAN ID      Blocked Port      Reason
-----
1             1/1/3             Root Guard
2             1/1/7             BPDU Guard
3             1/1/9             Loop Guard
4             1/1/37            RPVST Guard
```

Command History

Release	Modification
10.07 or earlier	--

Command Information

Platforms	Command context	Authority
All platforms	Operator (>) or Manager (#)	Operators or Administrators or local user group members with execution rights for this command. Operators can execute this command from the operator context (>) only.

show spanning-tree summary port

show spanning-tree summary port

Description

Shows a summary of port-related spanning-tree configuration and status.

Example

Showing a summary of port-related spanning tree information:

```
switch# show spanning-tree summary port

STP status           : Enabled
Protocol             : RPVST
BPDU guard timeout value : None
BPDU guard enabled interfaces : 1/1/1
BPDU filter enabled interfaces : None
Root guard enabled interfaces : 1/1/3
Loop guard enabled interfaces : 1/1/2
TCN guard enabled interfaces : 1/1/1-1/1/3

Interface count by state

VLAN                Blocking Listening Learning Forwarding Down
-----
VLAN1                0         0         0         1         0
VLAN2                0         0         0         1         0
-----
Total = 2            0         0         0         2         0
```

Command History

Release	Modification
10.09	A new state Down is added in the output.
10.07 or earlier	--

Command Information

Platforms	Command context	Authority
All platforms	Operator (>) or Manager (#)	Operators or Administrators or local user group members with execution rights for this command. Operators can execute this command from the operator context (>) only.

show spanning-tree summary root

```
show spanning-tree summary root
```

Description

Shows the summary of spanning tree root and configurations for all VLANs.

Example

Showing summary of spanning tree configurations:

```
switch# show spanning-tree summary root

STP status           : Enabled
Protocol             : RPVST
System ID            : f8:60:f0:c9:70:40

Root bridge for VLANs : 1-10
```

VLAN	Priority	Root ID	Root cost	Hello Time	Max Age	Fwd Dly	Root Port
VLAN1	32768	f8:60:f0:c9:70:40	0	2	20	15	0
VLAN2	32768	f8:60:f0:c9:70:40	0	2	20	15	0
VLAN3	32768	f8:60:f0:c9:70:40	0	2	20	15	0
VLAN4	32768	f8:60:f0:c9:70:40	0	2	20	15	0
VLAN5	32768	f8:60:f0:c9:70:40	0	2	20	15	0
VLAN6	32768	f8:60:f0:c9:70:40	0	2	20	15	0
VLAN7	32768	f8:60:f0:c9:70:40	0	2	20	15	0
VLAN8	32768	f8:60:f0:c9:70:40	0	2	20	15	0
VLAN9	32768	f8:60:f0:c9:70:40	0	2	20	15	0
VLAN10	32768	f8:60:f0:c9:70:40	0	2	20	15	0

Command History

Release	Modification
10.07 or earlier	--

Command Information

Platforms	Command context	Authority
All platforms	Operator (>) or Manager (#)	Operators or Administrators or local user group members with execution rights for this command. Operators can execute this command from the operator context (>) only.

show spanning-tree vlan

show spanning-tree vlan <VLAN-ID> [vsx-peer]

Description

Displays the spanning tree mode and information on the RPVST instance of the specified VLAN.

Parameter	Description
<VLAN-ID>	Specifies the number of a VLAN.
vsx-peer	Shows the output from the VSX peer switch. If the switches do not have the VSX configuration or the ISL is down, the output from the VSX peer switch is not displayed. This parameter is available on switches that support VSX.

Examples

Showing spanning tree mode and RPVST instance information for VLAN 2:

```
switch# show spanning-tree vlan 2
VLAN2
Spanning tree status: Enabled Protocol: RPVST
  Root ID      Priority    : 32768
                MAC-Address: 70:72:cf:76:43:2a
```

```

This bridge is the root
Hello time(in seconds):2  Max Age(in seconds):20
Forward Delay(in seconds):15

Bridge ID  Priority   : 32768
MAC-Address: 70:72:cf:76:43:2a
Hello time(in seconds):2  Max Age(in seconds):20
Forward Delay(in seconds):15

PORT      ROLE      STATE      COST      PRIORITY  TYPE      BPDU-Tx  BPDU-Rx
TCN-Tx    TCN-Rx
-----
1/1/1     Designated Forwarding 20000     128       P2P Edge  100       60
20        10
1/1/2     Designated Forwarding 20000     128       P2P      100       60
20        10
1/1/3     Designated Forwarding 20000     128       Shr      100       60
20        10
1/1/4     Designated Forwarding 20000     128       Shr Edge  100       60
20        10
1/1/5     Alternate  Loop-Inc   20000     128       Shr Edge  100       60
20        10
1/1/6     Alternate  Root-Inc   20000     128       Shr Edge  100       60
20        10
1/1/7     Disabled   Down       20000     128       P2P      100       60
20        10

Number of topology changes      : 4
Last topology change occurred   : 516 seconds ago

```

Command History

Release	Modification
10.09	A new state <code>Down</code> is added in the output.
10.07 or earlier	--

Command Information

Platforms	Command context	Authority
All platforms	Operator (>) or Manager (#)	Operators or Administrators or local user group members with execution rights for this command. Operators can execute this command from the operator context (>) only.

show spanning-tree vlan detail

show spanning-tree vlan <VLAN-ID> detail [vsx-peer]

Description

Displays the spanning tree mode and information on the RPVST instance of the specified VLAN and optionally displays details on the RPVST instance for the VLAN.

Parameter	Description
<VLAN-ID>	Specifies the number of a VLAN.
vsx-peer	Shows the output from the VSX peer switch. If the switches do not have the VSX configuration or the ISL is down, the output from the VSX peer switch is not displayed. This parameter is available on switches that support VSX.

Examples

Showing spanning tree mode and detailed RPVST instance information for VLAN 2:

```
switch# show spanning-tree vlan 2 detail
VLAN2
Spanning tree status: Enabled Protocol: RPVST
  Root ID      Priority    : 32768
                MAC-Address: 70:72:cf:76:43:2a
                This bridge is the root
                Hello time(in seconds):2   Max Age(in seconds):20
                Forward Delay(in seconds):15

  Bridge ID    Priority    : 32768
                MAC-Address: 70:72:cf:76:43:2a
                Hello time(in seconds):2   Max Age(in seconds):20
                Forward Delay(in seconds):15
```

PORT	ROLE	STATE	COST	PRIORITY	TYPE	BPDU-Tx	BPDU-Rx
TCN-Tx	TCN-Rx						
-----	-----	-----	-----	-----	-----	-----	-----
-----	-----	-----	-----	-----	-----	-----	-----
1/1/1 20	Designated 10	Forwarding	20000	128	P2P Edge	100	60
1/1/2 20	Designated 10	Forwarding	20000	128	P2P	100	60
1/1/3 20	Designated 10	Forwarding	20000	128	Shr	100	60
1/1/4 20	Designated 10	Forwarding	20000	128	Shr Edge	100	60
1/1/5 20	Alternate 10	Loop-Inc	20000	128	Shr Edge	100	60
1/1/6 20	Alternate 10	Root-Inc	20000	128	Shr Edge	100	60
1/1/7 20	Disabled 10	Down	20000	128	P2P	100	60

```

Topology change flag : False
Number of topology changes : 1
Last topology change occurred : 33293 seconds ago

Port 1/1/1
Designated root has priority :32768 Address: 48:0f:cf:af:22:1d
Designated bridge has priority :32768 Address: 48:0f:cf:af:22:1d
Designated port :1
Number of transitions to forwarding state : 0
BPDUs sent 1582, received 1506
TCN_Tx: 10, TCN_Rx: 10
```

Command History

Release	Modification
10.09	A new state <code>Down</code> is added in the output.
10.07 or earlier	--

Command Information

Platforms	Command context	Authority
All platforms	Operator (>) or Manager (#)	Operators or Administrators or local user group members with execution rights for this command. Operators can execute this command from the operator context (>) only.

spanning-tree bpduguard timeout

```
spanning-tree bpduguard timeout <INTERVAL>
no spanning-tree bpduguard timeout [<INTERVAL>]
```

Description

Enables and configures the auto re-enable timeout in seconds for all interfaces with BPDU guard enabled. When an interface is disabled after receiving an unauthorized BPDU it will automatically be re-enabled after the timeout expires. The default is for the interface to stay disabled until manually re-enabled.

The `no` form of the command disables BPDU guard timeout on the interface. This is the default.

Parameter	Description
<INTERVAL>	Specifies the re-enable timeout in seconds. Range: 1 to 65535.

Example

Enabling the BPDU guard timeout on interface 1/1/1:

```
switch(config)# interface 1/1/1
switch(config-if)# spanning-tree bpduguard timeout 10
```

Disabling BPDU guard timeout on interface 1/1/1:

```
switch(config)# interface 1/1/1
switch(config-if)# no spanning-tree bpduguard
```

Command History

Release	Modification
10.07 or earlier	--

Command Information

Platforms	Command context	Authority
All platforms	config	Administrators or local user group members with execution rights for this command.

spanning-tree extend-system-id

```
spanning-tree extend-system-id {enable | disable}
no spanning-tree extend-system-id
```

Description

Configures use of extended system ID. When enabled, the VLAN ID is included in spanning tree packets. When disabled, the VLAN ID is set to NULL in the spanning tree packets.

By default, extended system ID is enabled. If you disable extended system ID, the bridge identifier field in the spanning tree packet is filled with zeros.

The `no` form of this command disables extended system ID.

Parameter	Description
enable	Specifies enabling use of extended system ID.
disable	Specifies disabling use of extended system ID.

Examples

Enabling extended system ID:

```
switch# config
switch(config)# spanning-tree extend-system-id enable
```

Disabling extended system ID:

```
switch# config
switch(config)# spanning-tree extend-system-id disable
switch(config)# no spanning-tree extend-system-id
```

Command History

Release	Modification
10.07 or earlier	--

Command Information

Platforms	Command context	Authority
All platforms	config	Administrators or local user group members with execution rights for this command.

spanning-tree ignore-pvid-inconsistency

```
spanning-tree ignore-pvid-inconsistency {enable | disable}
```

no spanning-tree ignore-pvid-inconsistency

Description

Configures port behavior when per-VLAN ID inconsistencies are present. For example, when the ports on both ends of a point-to-point link are untagged members of different VLANs, enabling this option allows RPVST+ to process untagged RPVST+ packets belonging to the peer's untagged VLAN as if they were received on the current device's untagged VLAN. When this option is disabled, RPVST+ blocks the link, causing traffic on the mismatched VLANs to be dropped.

If this option 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 port VLAN memberships is misconfigured on a switch in the network, then enabling this option prevents RPVST+ from detecting the problem, which may result in packet duplication in the network since RPVST+ would not converge correctly.

This command affects all ports on the switch belonging to VLANs on which RPVST+ is enabled.

By default ignore per-VLAN ID inconsistency is disabled.

The `no` form of this command sets the ignore per-VLAN ID inconsistencies to disabled.

Parameter	Description
enable	Specifies ignore per-VLAN ID inconsistencies and allow RPVST to run on mismatched links.
disable	Disables the ignore per-VLAN ID inconsistencies functionality.

Examples

Enabling ignore per-VLAN ID inconsistencies:

```
switch# config  
switch(config)# spanning-tree ignore-pvid-inconsistency enable
```

Disabling ignore per-VLAN ID inconsistencies:

```
switch# config  
switch(config)# spanning-tree ignore-pvid-inconsistency disable  
switch(config)# no spanning-tree ignore-pvid-inconsistency
```

Command History

Release	Modification
10.07 or earlier	--

Command Information

Platforms	Command context	Authority
All platforms	config	Administrators or local user group members with execution rights for this command.

spanning-tree link-type

```
spanning-tree link-type {point-to-point | shared}
no spanning-tree link-type
```

Description

Configures the link type of a port.

The `no` form of this command sets the spanning tree link type to the default value of `point-to-point`.

Parameter	Description
<code>point-to-point</code>	Sets the spanning tree link type as point-to-point. Use this for full-duplex ports that provide a point-to-point link to devices such as a switch, bridge, or end-node. Default.
<code>shared</code>	Sets the spanning tree link type as shared. Use this when the port is connected to a hub.

Examples

Setting spanning tree link type to shared:

```
switch(config)# interface 1/1/1
switch(config-if)# spanning-tree link-type shared
```

Setting spanning tree link type to point-to-point for a port:

```
switch(config)# interface 1/1/1
switch(config-if)# no spanning-tree link-type
```

Command History

Release	Modification
10.07 or earlier	--

Command Information

Platforms	Command context	Authority
All platforms	<code>config-if</code>	Administrators or local user group members with execution rights for this command.

spanning-tree mode

```
spanning-tree mode {mstp|rpvst}
no spanning-tree mode [mstp|rpvst]
```

Description

Sets the spanning tree mode to either MSTP mode (Multiple-instance Spanning Tree Protocol) or RPVST mode (Rapid Per VLAN Spanning Tree).

The `no` form of this command sets the spanning tree mode to the default `mstp`.

Parameter	Description
mstp	Sets the mode to MSTP (Multiple-instance Spanning Tree Protocol), which applies the STP (spanning tree protocol) separately for each set of VLANs (called an MSTI - multiple spanning tree instance).
rpvst	Sets the mode to RPVST (Rapid Per VLAN Spanning Tree).

Examples

Enabling MSTP mode:

```
switch(config)# spanning-tree mode mstp
```

Enabling RPVST mode:

```
switch(config)# spanning-tree mode rpvt
```

Command History

Release	Modification
10.07 or earlier	--

Command Information

Platforms	Command context	Authority
All platforms	config	Administrators or local user group members with execution rights for this command.

spanning-tree pathcost-type

```
spanning-tree pathcost-type {long | short}
no spanning-tree pathcost-type [long|short]
```

Description

Configures the spanning tree path cost type. The long mode provides support for the wider range of link speeds required by high-speed interfaces. All switches in the network must use the same path cost type or errors can occur in the spanning tree.

The `no` form of this command sets the spanning tree path cost type to the default `long`.

Parameter	Description
long	Specifies the spanning tree path cost type as a 32-bit value, allowing port cost values to be set in the range 1-200,000,000. Default.
short	Specifies the spanning tree path cost type as a 16-bit value, allowing port cost values to be set in the range 1-65535.

Examples

Setting spanning tree path cost type to short:

```
switch# config
switch(config)# spanning-tree pathcost-type short
```

Setting spanning tree path cost type to long:

```
switch# config
switch(config)# spanning-tree pathcost-type long
```

Setting spanning tree path cost to default of long:

```
switch# config
switch(config)# no spanning-tree pathcost-type
```

Command History

Release	Modification
10.07 or earlier	--

Command Information

Platforms	Command context	Authority
All platforms	config	Administrators or local user group members with execution rights for this command.

spanning-tree rpvst-mstp interconnect vlan

```
spanning-tree rpvst-mstp-interconnect-vlan <VLAN-ID>
no spanning-tree rpvst-mstp-interconnect-vlan [<VLAN-ID>]
```

Description

Configures the VLAN that has to be used to interconnect RPVST and MSTP domains. VLAN 1 is used by default.

The **no** form of this command sets the VLAN configuration to the default of 1.

- It is required to create the interconnect VLAN and then configure RPVST spanning tree on it.
- The same interconnect VLAN must be kept on all the switches in the network.
- Adding or deleting the interconnect VLAN triggers a re-convergence in the network.
- Deleting a VLAN that is configured as the interconnect VLAN does not reset the value to the default.

Parameter	Description
<VLAN-ID>	Specifies the number of a VLAN.

Examples

This example configures VLAN 10 to used to interconnect RPVST and MSTP domains.

```
switch#(config)# spanning-tree rpvst-mstp-interconnect-vlan 10
```

Command History

Release	Modification
10.07 or earlier	--

Command Information

Platforms	Command context	Authority
All platforms	config	Administrators or local user group members with execution rights for this command.

spanning-tree tcn-guard

```
spanning-tree tcn-guard  
no spanning-tree tcn-guard
```

Description

Disables propagation of topology change notifications (TCNs) to other STP ports. Use this when you do not want topology changes to be noticed by peer devices. By default, the propagation is enabled.

The `no` form of this command, enables propagation of topology changes which is the default.

Examples

Enabling `tcn-guard`, which disables propagation of topology changes:

```
switch(config-if)# spanning-tree tcn-guard
```

Disabling `tcn-guard`, which enables propagation of topology changes:

```
switch(config-if)# no spanning-tree tcn-guard
```

Command History

Release	Modification
10.07 or earlier	--

Command Information

Platforms	Command context	Authority
All platforms	config-if	Administrators or local user group members with execution rights for this command.

spanning-tree vlan

```
spanning-tree vlan <VLAN-LIST> [{hello-time | forward-delay | max-age | priority} <VALUE>]  
no spanning-tree vlan <VLAN-LIST> [hello-time | forward-delay | max-age | priority]
```

Description

Creates an RPVST instance for the specified VLAN. This command also allows for configuration of RPVST instance-specific time parameters.

The `no` form of this command removes the RPVST instance associated with the specified VLAN, and configures default values for RPVST instance-specific parameters.

Parameter	Description
<VLAN-LIST>	Specifies the number of a single VLAN, or a series of numbers for a range of VLANs, separated by commas (1, 2, 3, 4), dashes (1-4), or both (1-4,6).
hello-time <VALUE>	Specifies the hello-time in seconds for the RPVST instance. Range: 2-10 seconds. Default: 2 seconds.
forward-delay <VALUE>	Specifies the forward-delay time in seconds for the RPVST instance. Range: 4-30 seconds. Default: 15 seconds.
max-age <VALUE>	Specifies the maximum age time in seconds for the RPVST instance. Range: 6-40 seconds. Default: 20 seconds.
priority <VALUE>	Specifies the priority for the RPVST instance. Priority value is configured as a multiple of 4096. Range: 0-15. Default: 8 which is 32768.

Examples

Creating an RPVST instance for a list of VLANs and configuring various time parameters:

```
switch# config  
switch(config)# spanning-tree vlan 2-5  
switch(config)# spanning-tree vlan 2-5 hello-time 5  
switch(config)# spanning-tree vlan 5 max-age 10  
switch(config)# spanning-tree vlan 2-5 forward-delay 25  
switch(config)# spanning-tree vlan 2-5 priority 5
```

Removing an RPVST instance for a list of VLANs and setting various time parameters to the default:

```
switch# config  
switch(config)# no spanning-tree vlan 2-5  
switch(config)# no spanning-tree vlan 2-5 hello-time  
switch(config)# no spanning-tree vlan 2-5 forward-time  
switch(config)# no spanning-tree vlan 2-5 max-age  
switch(config)# no spanning-tree vlan 2-5 priority
```

Command History

Release	Modification
10.07 or earlier	--

Command Information

Platforms	Command context	Authority
All platforms	config	Administrators or local user group members with execution rights for this command.

spanning-tree vlan cost

```
spanning-tree vlan <VLAN-LIST> cost <PORT-COST>
no spanning-tree vlan <VLAN-LIST> cost
```

Description

Configures the spanning tree cost for the VLAN. This is the cost to reach the root port.

The `no` form of this command sets the port cost to the default value.

Parameter	Description
<VLAN-LIST>	Specifies the number of a single VLAN, or a series of numbers for a range of VLANs, separated by commas (1, 2, 3, 4), dashes (1-4), or both (1-4,6).
<PORT-COST>	Specifies the spanning tree cost for the VLAN. Range: 1-200,000,000. Default is calculated from the port link speed: <ul style="list-style-type: none">10 Mbps link speed equals a path cost of 2,000,000.100 Mbps link speed equals a path cost of 200,000.1 Gbps link speed equals a path cost of 20,000.2 Gbps link speed equals a path cost of 10,000.10 Gbps link speed equals a path cost of 2,000.100 Gbps link speed equals a path cost of 200.1 Tbps link speed equals a path cost of 20.

Examples

Setting port cost:

```
switch(config-if)# spanning-tree vlan 5 cost 100000
```

Setting port cost to the default:

```
switch(config-if)# no spanning-tree vlan 5 cost
```

Command History

Release	Modification
10.07 or earlier	--

Command Information

Platforms	Command context	Authority
All platforms	config-if	Administrators or local user group members with execution rights for this command.

spanning-tree vlan port-priority

```
spanning-tree vlan <VLAN-LIST> port-priority <PRIORITY>
no spanning-tree vlan <VLAN-LIST> port-priority
```

Description

Configures port priority. A port with the lowest priority number has the highest priority for use in forwarding traffic.

The `no` form of this command, sets the port priority to the default of 8.

Parameter	Description
<VLAN-LIST>	Specifies the number of a single VLAN, or a series of numbers for a range of VLANs, separated by commas (1, 2, 3, 4), dashes (1-4), or both (1-4,6).
<PRIORITY>	Specifies the port priority. The value, configured as a multiple of 16, helps in determining the designated port. The lower a priority value, the higher the priority. Range: 1 to 15. Default: 8.

Examples

Setting port priority:

```
switch(config-if)# spanning-tree vlan 5 port-priority 10
```

Setting port priority to the default of 8:

```
switch(config-if)# no spanning-tree vlan 5 port-priority
```

Command History

Release	Modification
10.07 or earlier	--

Command Information

Platforms	Command context	Authority
All platforms	config-if	Administrators or local user group members with execution rights for this command.

spanning-tree trap

```
spanning-tree trap {new-root | topology-change [vlan <VLAN-ID>] |
  errant-bpdu | root-guard-inconsistency | loop-guard-inconsistency}
```

```
no spanning-tree trap {new-root | topology-change [vlan <VLAN-ID>] |  
    errant-bpdu | root-guard-inconsistency | loop-guard-inconsistency}
```

Description

Enables SNMP traps for new root, topology change event, errant-bpdu received event, root-guard inconsistency, and loop-guard inconsistency notifications. It is disabled by default.

The `no` form of this command disables the notifications for SNMP traps.

Parameter	Description
<code>new-root</code>	Enables SNMP notification when a new root is elected on any PVST vlan on the switch.
<code>topology-change</code>	Enables SNMP notification when a topology change event occurred in specified PVST vlan on the switch.
<code><VLAN-ID></code>	Specifies the VLAN ID for the topology change trap. Range: 1 to 4094.
<code>errant-bpdu</code>	Enables SNMP notification when an errant bpdu is received by any PVST vlan on the switch.
<code>root-guard-inconsistency</code>	Enables SNMP notification when the root-guard finds the port inconsistent for any PVST vlan on the switch.
<code>loop-guard-inconsistency</code>	Enables SNMP notification when the loop-guard finds the port inconsistent for any PVST vlan on the switch.

Examples

Enabling the notifications for the SNMP traps:

```
switch(config)# spanning-tree trap  
    new-root          Enable notifications which are sent when a new root is  
    elected  
    topology-change   Enable notifications which are sent when a topology  
    change occurs  
    errant-bpdu       Enable notifications which are sent when an errant  
    bpdu is received  
    root-guard-inconsistency Enable notifications which are sent when root guard  
    inconsistency occurs  
    loop-guard-inconsistency Enable notifications which are sent when loop guard  
    inconsistency occurs  
switch(config)# spanning-tree trap new-root  
    <cr>  
switch(config)# spanning-tree trap topology-change  
    vlan Enable topology change notification for the specified PVST vlan id.  
switch(config)# spanning-tree trap topology-change vlan  
    <1-4094> Enable topology change information on the specified vlan id.  
switch(config)# spanning-tree trap topology-change vlan 1  
    <cr>  
switch(config)# spanning-tree trap errant-bpdu  
    <cr>  
switch(config)# spanning-tree trap root-guard-inconsistency  
    <cr>  
switch(config)# spanning-tree trap loop-guard-inconsistency  
    <cr>
```

Disabling the notifications for the SNMP traps:

```

switch(config)# no spanning-tree trap
    new-root          Disable notifications which are sent when a new root
is elected
    topology-change   Disable notifications which are sent when a topology
change occurs
    errant-bpdu       Disable notifications which are sent when an errant
bpdu is received
    root-guard-inconsistency Disable notifications which are sent when root guard
inconsistency occurs
    loop-guard-inconsistency Disable notifications which are sent when loop guard
inconsistency occurs
switch(config)# no spanning-tree trap new-root
<cr>
switch(config)# no spanning-tree trap topology-change
    instance          Disable topology change notification for the specified PVST vlan id.
switch(config)# no spanning-tree trap topology-change vlan
    <1-4094>          Disable topology change information on the specified PVST vlan id.
switch(config)# no spanning-tree trap topology-change vlan 1
<cr>
switch(config)# no spanning-tree trap errant-bpdu
<cr>
switch(config)# no spanning-tree trap root-guard-inconsistency
<cr>
switch(config)# no spanning-tree trap loop-guard-inconsistency
<cr>

```

Command History

Release	Modification
10.07 or earlier	--

Command Information

Platforms	Command context	Authority
All platforms	config	Administrators or local user group members with execution rights for this command.

MSTP protocol and feature details

Multiple-Instance spanning tree protocol (MSTP) 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.

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.

Developed based on IEEE 802.1s, MSTP overcomes the limitations of STP, RSTP, and PVST. In addition to supporting rapid network convergence, it allows data flows of different VLANs to be forwarded along separate paths. This provides a better load sharing mechanism for redundant links.

MSTP provides the following features:

- MSTP divides a switched network into multiple regions, each of which contains multiple spanning trees that are independent of one another.
- MSTP supports mapping VLANs to spanning tree instances by means of a VLAN-to-instance mapping table. MSTP can reduce communication overheads and resource usage by mapping multiple VLANs to one instance.
- MSTP prunes a loop network into a loop-free tree, which avoids proliferation and endless cycling of packets in a loop network. In addition, it supports load balancing of VLAN data by providing multiple redundant paths for data forwarding.
- MSTP is compatible with STP and RSTP, and partially compatible with PVST.
- Configuring instances is not mandatory. MSTP can work with the default instance CIST if spanning-tree is just enabled. All existing VLANs in the switch will be part of CIST.

MSTP key concepts

MSTP divides an entire Layer 2 network into multiple MST regions, which are connected by a calculated CST. Inside an MST region, multiple spanning trees, called MSTIs, are calculated. Among these MSTIs, MSTI 0 is the internal spanning tree (IST).

Figure 16 Network with four MST regions and four switches per region

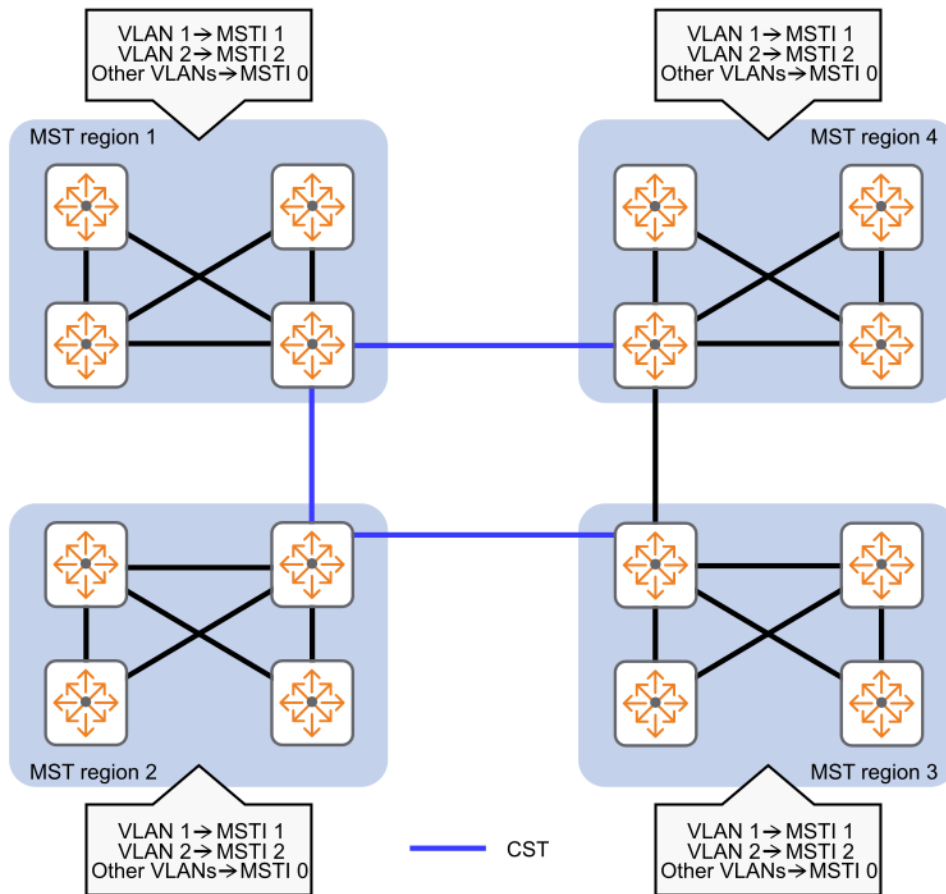
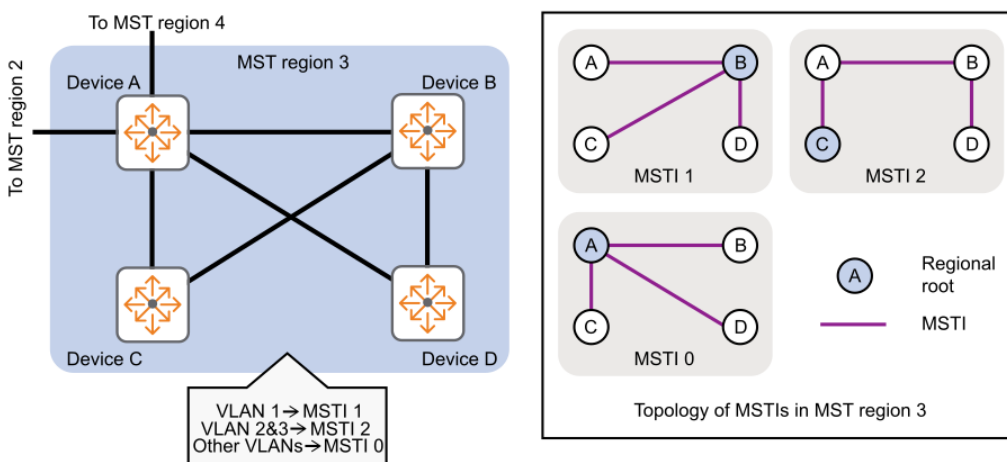


Figure 17 MST region 3



MST region

A multiple spanning tree region (MST region) consists of multiple devices in a switched network and the network segments between them. All these devices have the following characteristics:

- A spanning tree protocol enabled.
- Same region name.
- Same VLAN-to-instance mapping configuration.
- Same MSTP revision level.
- Physically linked together.

Multiple MST regions can exist in a switched network. You can assign multiple devices to the same MST region.

- The switched network comprises four MST regions, MST region 1 through MST region 4.
- All devices in each MST region have the same MST region configuration.

MSTI

MSTP can generate multiple independent spanning trees in an MST region, and each spanning tree is mapped to specific VLANs. Each spanning tree is referred to as a multiple spanning tree instance (MSTI). In the figures, MST region 3 comprises three MSTIs, MSTI 1, MSTI 2, and MSTI 0.

MSTI 0

VLAN-to-instance mapping table

As an attribute of an MST region, the VLAN-to-instance mapping table describes the mapping relationships between VLANs and MSTIs.

In the figures, the VLAN-to-instance mapping table of MST region 3 is as follows:

- VLAN 1 to MSTI 1. (Ports which are not part of any VLAN are by default part of VLAN 1.)
- VLAN 2 and VLAN 3 to MSTI 2.
- Other VLANs to MSTI 0. (VLANs that are not configured as part of any MSTI are by default part of MSTI 0.)

MSTP achieves load balancing by means of the VLAN-to-instance mapping table.

CST

The common spanning tree (CST) is a single spanning tree that connects all MST regions in a switched network. If you regard each MST region as a device, the CST is a spanning tree calculated by these devices through STP or RSTP. The blue lines in the figures represent the CST.

IST

An internal spanning tree (IST) is a spanning tree that runs in an MST region. It is also called MSTI 0, a special MSTI to which all VLANs are mapped by default. In the figures, MSTI 0 is the IST in MST region 3.

CIST

The common and internal spanning tree (CIST) is a single spanning tree that connects all devices in a switched network. It consists of the ISTs in all MST regions and the CST. In the figures, the ISTs (MSTI 0) in all MST regions plus the inter-region CST constitute the CIST of the entire network.

Regional root

The root bridge of the IST or an MSTI within an MST region is the regional root of the IST or MSTI. Based on the topology, different spanning trees in an MST region might have different regional roots, as shown in MST region 3 in the figures:

- The regional root of MSTI 1 is Device B.
- The regional root of MSTI 2 is Device C.
- The regional root of MSTI 0 (also known as the IST) is Device A.

Common root bridge

The common root bridge is the root bridge of the CIST. In the figures, the common root bridge is a device in MST region 1.

Port roles

A port can play different roles in different MSTIs. In the following figure, an MST region comprises Device A, Device B, Device C, and Device D. Port A1 and port A2 of Device A connect to the common root bridge. Port B2 and Port B3 of Device B form a loop. Port C3 and Port C4 of Device C connect to other MST regions. Port D3 of Device D directly connects to a host.

MSTP calculation involves the following port roles:

- Root port: Forwards data for a non-root bridge to the root bridge. The root bridge does not have any root port.
- Designated port: Forwards data to the downstream network segment or device.
- Alternate port: Acts as the backup port for a root port or conductor port. When the root port or conductor port is blocked, the alternate port takes over.
- Backup port: Acts as the backup port of a designated port. When the designated port is invalid, the backup port becomes the new designated port. A loop occurs when two ports of the same spanning tree device are connected, so the device blocks one of the ports. The blocked port acts as the backup.
- Edge port: Does not connect to any network device or network segment, but directly connects to a user host.
- Conductor port: Acts as a port on the shortest path from the local MST region to the common root bridge. The conductor port is not always located on the regional root. It is a root port on the IST or CIST and still a conductor port on the other MSTIs.
- Boundary port: Connects an MST region to another MST region or to an STP/RSTP-running device. In MSTP calculation, a boundary port's role on an MSTI is consistent with its role on the CIST. However, that is not true with conductor ports. A conductor port on MSTIs is a root port on the CIST.

Port states

In MSTP, a port can be in one of the following states:

- Forwarding: The port receives and sends BPDUs, learns MAC addresses, and forwards user traffic.
- Learning: The port receives and sends BPDUs, learns MAC addresses, but does not forward user traffic. Learning is an intermediate port state.
- Discarding: The port receives and sends BPDUs, but does not learn MAC addresses or forward user traffic.



When in different MSTIs, a port can be in different states.

A port state is not exclusively associated with a port role. The following table lists the port states that each port role supports. (An X indicates that the port supports this state, while a dash [—] indicates that the port does not support this state.)

Port state	Port role			
	Root port/ conductor port	Designated port	Alternate port	Backup port
Forwarding	X	X	—	—
Learning	X	X	—	—
Discarding	X	X	X	X

CIST calculation

During the CIST calculation, the following process takes place:

- The device with the highest priority is elected as the root bridge of the CIST.
- MSTP generates an IST within each MST region through calculation.
- MSTP regards each MST region as a single device and generates a CST among these MST regions through calculation.

The CST and ISTs constitute the CIST of the entire network.

MSTI calculation

Within an MST region, MSTP generates different MSTIs for different VLANs based on the VLAN-to-instance mappings. For each spanning tree, MSTP performs a separate calculation process similar to spanning tree calculation in STP.

In MSTP, a BPDU packet is forwarded along the following paths:

- Within an MST region, the packet is forwarded along the corresponding MSTI.
- Between two MST regions, the packet is forwarded along the CST.

MSTP on VSX

See the *Virtual Switching Extension (VSX) Guide* for important information when configuring MSTP with VSX.

MSTP configuration tasks

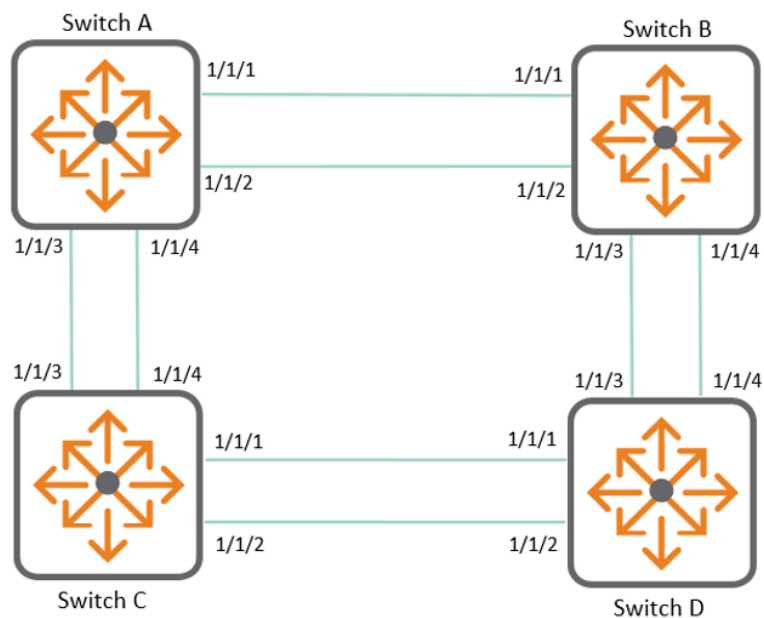
- Configuring MSTP instances is not mandatory. Instances are required only if you need to reuse the blocked links for some other VLAN path. To enable MSTP, simply configure the same 'configuration-name' across all switches and enable 'spanning-tree' and leave the configuration-revision as default. This is sufficient for MSTP.
- Ensure that the VLAN configuration in your network supports all of the forwarding paths necessary for the desired connectivity. All ports connecting one switch to another within a region and one switch to another between regions should be configured as members of all VLANs configured in the region.
- 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 MST 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 MSTI 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 MSTI.
- Determine the designated bridge and designated port for each LAN segment.
- Determine which VLANs to assign to each MST 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.
- Set the admin-edge port type to `admin-edge` for edge ports connected to end nodes.
- Set the admin-edge port type to `admin-network` for ports connected to another switch, a bridge, or a half-duplex repeater.

MSTP use cases

MSTP use case: Eliminating a loop with MSTP

In this use case, all four switches are in same region. VLANs 10, 20, 30, 40, 50, and 60 are defined on all switches, causing a network loop. The physical topology of the network looks like this:

Figure 18 *Physical topology before loop elimination*



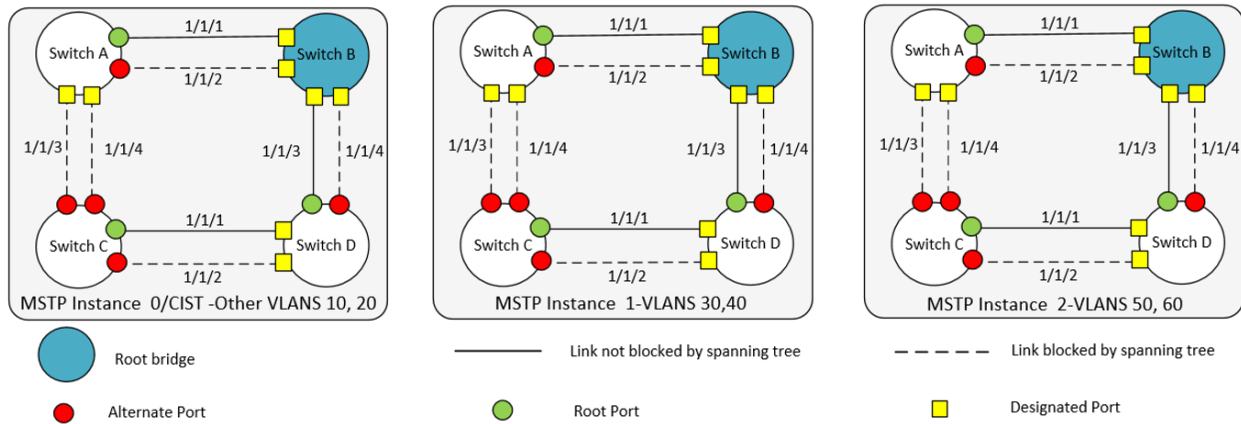
To eliminate the loop, MSTP is enabled on all the switches, with the following configuration:

- Switch B is the root for CIST, MST1, and MST2.
- CIST: VLANs 10, 20
- Instance-1: VLANs 30, 40
- Instance-2: VLANs 50, 60
- All four switches are in the same MSTP region.

To understand how MSTP works in this use case, it is useful to view each instance as a separate logical topology as illustrated in the following figures.

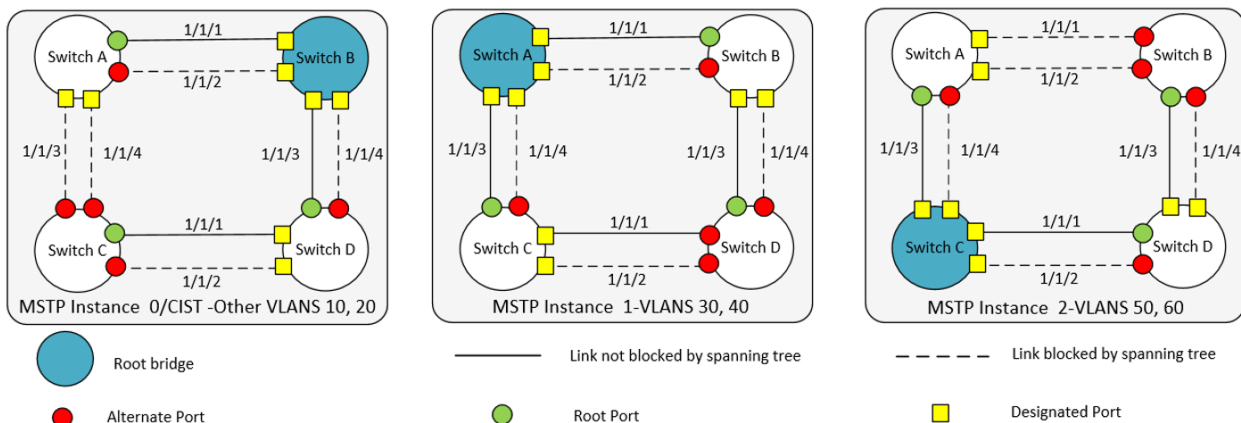
In this initial configuration, loops are avoided by blocking the alternate links for each network segment. All ports designated A (Alternate) are blocked and do not forward traffic. Although this strategy eliminates the loops, it is not the most effective way to configure the MST regions because network resources are not fully used.

Figure 19 *MSTP loop elimination, initial configuration*



By changing the root for each instance, more effective load sharing can be achieved. With this refined configuration, the links (ports) that were previously unused are now being used by different instances. Also, the network loop is eliminated and load sharing is achieved:

Figure 20 *MSTP loop elimination, refined configuration*



Procedure

Configure all switches with the same VLANs, interfaces, and spanning tree instances.

1. Create VLANs 10, 20, 30, 40, 50, and 60 and assign them to interfaces.

```
switch# config
switch(config)# vlan 10,20,30,40,50,60
switch(config)# interface 1/1/1-1/1/4
switch(config-if-<1/1/1-1/1/4>)# no shutdown
switch(config-if-<1/1/1-1/1/4>)# no routing
switch(config-if-<1/1/1-1/1/4>)# vlan trunk allowed 10,20,30,40,50,60
switch(config-if)# exit
```

2. Configure spanning tree and enable it.

```
switch(config)# spanning-tree config-name reg
switch(config)# spanning-tree config-revision 1
switch(config)# spanning-tree inst 1 vlan 30
switch(config)# spanning-tree inst 1 vlan 40
switch(config)# spanning-tree inst 2 vlan 50
switch(config)# spanning-tree inst 2 vlan 60
switch(config)# spanning-tree
```

3. On switch A, set instance 1 to priority 0.

```
switch-A(config)# spanning-tree inst 1 priority 0
```

4. On switch C, set instance 2 to priority 0.

```
switch-C(config)# spanning-tree inst 2 priority 0
```

5. On switch B set the MSTP default CIST instance priority to 0.

```
switch-B(config)# spanning-tree priority 0
```

MSTP commands

clear spanning-tree statistics

```
clear spanning-tree statistics
```

Description

Clears the spanning tree BPDU statistics.

Example

Clearing the spanning tree BPDU statistics:

```
switch(config)# clear spanning-tree statistics
```

Command History

Release	Modification
10.07 or earlier	--

Command Information

Platforms	Command context	Authority
All platforms	config	Administrators or local user group members with execution rights for this command.

show spanning-tree

```
show spanning-tree [vsx-peer]
```

Description

Shows priority, address, Hello-time, Max-age, and Forward-delay for bridge and root node.

Parameter	Description
vsx-peer	Shows the output from the VSX peer switch. If the switches do not have the VSX configuration or the ISL is down, the output from the VSX peer switch is not displayed. This parameter is available on switches that support VSX.

Example

Showing spanning tree standard information:

```
switch# show spanning-tree
Spanning tree status      : Enabled Protocol: MSTP

MST0
  Root ID
    Priority      : 32768, Root
    MAC-Address   : 48:0F:CF:AF:04:76
```

```
Hello time(in seconds):2    Max Age(in seconds):20
Forward Delay(in seconds):15
```

Bridge ID

```
Priority      : 32768
MAC-Address   : 48:0F:CF:AF:04:76
Hello time(in seconds):2    Max Age(in seconds):20
Forward Delay(in seconds):15
```

PORT	ROLE	STATE	COST	PRIORITY	TYPE	BPDU-Tx	BPDU-Rx
TCN-Tx	TCN-Rx						

1/1/1	Designated	Forwarding	20000	128	P2P Edge	100	60
20	10						
1/1/2	Designated	Forwarding	20000	128	P2P	100	60
20	10						
1/1/3	Designated	Forwarding	20000	128	Shr	100	60
20	10						
1/1/4	Designated	Forwarding	20000	128	Shr Edge	100	60
20	10						
1/1/5	Alternate	Loop-Inc	20000	128	Shr Edge	100	60
20	10						
1/1/6	Alternate	Root-Inc	20000	128	Shr Edge	100	60
20	10						
1/1/7	Root	Forwarding	2000	128	P2P	100	60
20	10						
1/1/8	Alternate	Blocking	20000	128	P2P	100	60
20	10						
1/1/9	Disabled	Down	20000	128	P2P	100	60
20	10						

```
Number of topology changes      : 4
Last topology change occurred   : 516 seconds ago
```

Command History

Release	Modification
10.09	A new state <code>Down</code> is added in the output.
10.07 or earlier	--

Command Information

Platforms	Command context	Authority
All platforms	Operator (>) or Manager (#)	Operators or Administrators or local user group members with execution rights for this command. Operators can execute this command from the operator context (>) only.

show spanning-tree detail

```
show spanning-tree detail [vsx-peer]
```

Description

Shows spanning tree detail including CIST and corresponding port information.

Parameter	Description
vsx-peer	Shows the output from the VSX peer switch. If the switches do not have the VSX configuration or the ISL is down, the output from the VSX peer switch is not displayed. This parameter is available on switches that support VSX.

Example

Showing spanning tree detailed information:

```
switch# show spanning-tree detail
Spanning tree status      : Enabled Protocol: MSTP

MST0
  Root ID
    Priority      : 32768, Root
    MAC-Address   : 48:0F:CF:AF:04:76
    Hello time(in seconds):2   Max Age(in seconds):20
    Forward Delay(in seconds):15

  Bridge ID
    Priority      : 32768
    MAC-Address   : 48:0F:CF:AF:04:76
    Hello time(in seconds):2   Max Age(in seconds):20
    Forward Delay(in seconds):15

PORT      ROLE      STATE      COST      PRIORITY  TYPE      BPDU-Tx    BPDU-Rx
TCN-Tx    TCN-Rx
-----
1/1/1     Designated Forwarding 20000     128       P2P Edge  100        60
20        10
1/1/2     Designated Forwarding 20000     128       P2P      100        60
20        10
1/1/3     Designated Forwarding 20000     128       Shr      100        60
20        10
1/1/4     Designated Forwarding 20000     128       Shr Edge  100        60
20        10
1/1/5     Alternate  Loop-Inc   20000     128       Shr Edge  100        60
20        10
1/1/6     Alternate  Root-Inc   20000     128       Shr Edge  100        60
20        10
1/1/7     Disabled   Down       20000     128       P2P      100        60
20        10

Topology change flag      : True
Number of topology changes : 4
Last topology change occurred : 516 seconds ago
Hello expiry              : 1 second
Forward delay expiry      : 18 seconds

Port 1/1/1
Designated root has priority      : 32768      Address:
48:0F:CF:AF:04:76
Designated bridge has priority    : 32768      Address:
48:0F:CF:AF:04:76
Designated port                  : 1/1/1
Number of transitions to forwarding state : 3
BPDUs sent                      : 347
BPDUs received                  : 9
```

```
TCN_Tx: 20, TCN_Rx: 10
```

```
Port 1/1/2
```

```
Designated root has priority           : 32768      Address:
```

```
48:0F:CF:AF:04:76
```

```
Designated bridge has priority         : 32768      Address:
```

```
48:0F:CF:AF:04:76
```

```
Designated port                         : 1/1/2
```

```
Number of transitions to forwarding state : 3
```

```
BPDUs sent                             : 350
```

```
BPDUs received                          : 11
```

```
TCN_Tx: 20, TCN_Rx: 10
```

```
Port lag1 ID 321
```

```
Designated root has priority           : 32768      Address:
```

```
48:0F:CF:AF:04:76
```

```
Designated bridge has priority         : 32768      Address:
```

```
48:0F:CF:AF:04:76
```

```
Designated port id                     : 321
```

```
Multi-Chassis role                     : active
```

```
Number of transitions to forwarding state : 3
```

```
BPDUs sent                             : 340
```

```
BPDUs received                          : 5
```

```
TCN_Tx: 20, TCN_Rx: 10
```

Command History

Release	Modification
10.09	A new state Down is added in the output.
10.07 or earlier	--

Command Information

Platforms	Command context	Authority
All platforms	Operator (>) or Manager (#)	Operators or Administrators or local user group members with execution rights for this command. Operators can execute this command from the operator context (>) only.

show spanning-tree inconsistent-ports

```
show spanning-tree inconsistent-ports [instance <INSTANCE-ID>]
```

Description

Shows ports blocked by STP protection functions such as Root guard, Loop guard, BPDU guard, and RPVST guard in addition to MSTI information.

Parameter	Description
<INSTANCE-ID>	Specifies the MSTP instance ID. Range: 0 to 64.

Example

Showing spanning tree inconsistent ports:

```
switch# show spanning-tree inconsistent-ports
Instance ID  Blocked Port  Reason
-----
0            1/1/13          BPDU Guard
```

Showing inconsistent port information for instances 1-4:

```
switch# show spanning-tree inconsistent-ports instance 1-4
Instance ID  Blocked Port  Reason
-----
1            1/1/3          Root Guard
2            1/1/7          BPDU Guard
3            1/1/9          Loop Guard
4            1/1/37         RPVST Guard
```

Command History

Release	Modification
10.07 or earlier	--

Command Information

Platforms	Command context	Authority
All platforms	Operator (>) or Manager (#)	Operators or Administrators or local user group members with execution rights for this command. Operators can execute this command from the operator context (>) only.

show spanning-tree mst

show spanning-tree mst [vsx-peer]

Description

Shows MSTP configuration and status information for each instance.

Parameter	Description
vsx-peer	Shows the output from the VSX peer switch. If the switches do not have the VSX configuration or the ISL is down, the output from the VSX peer switch is not displayed. This parameter is available on switches that support VSX.

Examples

Showing MSTP configuration and status information:

```
switch# show spanning-tree mst
#### MST0
Vlans mapped   : 2,4-4094
Bridge Address : 48:0F:CF:AF:04:76
Priority        : 32768
Root
```

```

Regional Root
Operational   Hello time      : 2 seconds          Forward delay: 15 seconds
               Max-age       : 20 seconds          TxHoldCount  : 6 pps
Configured    Hello time      : 2 seconds          Forward delay: 15 seconds
               Max-age       : 20 seconds          Max-Hops     : 20
Root          Address        : 48:0F:CF:AF:04:76    Priority      : 32768
               Port          : 0                   Path cost    : 0
Regional Root Address        : 48:0F:CF:AF:04:76    Priority      : 32768
               Internal cost: 0                    Rem Hops     : 20

```

PORT TCN-Tx	ROLE TCN-Rx	STATE	COST	PRIORITY	TYPE	BPDU-Tx	BPDU-Rx
1/1/1 20	Designated 10	Forwarding	20000	128	P2P Edge	100	60
1/1/2 20	Designated 10	Forwarding	20000	128	P2P	100	60
1/1/3 20	Designated 10	Forwarding	20000	128	Shr	100	60
1/1/4 20	Designated 10	Forwarding	20000	128	Shr Edge	100	60
1/1/5 20	Alternate 10	Loop-Inc	20000	128	Shr Edge	100	60
1/1/6 20	Alternate 10	Root-Inc	20000	128	Shr Edge	100	60
1/1/7 20	Disabled 10	Down	20000	128	P2P	100	60

```

Topology change flag      : True
Number of topology changes : 4
Last topology change occurred : 516 seconds ago

```

MST1

```

Vlans mapped: 1
Bridge        Address : 48:0F:CF:AF:04:76    Priority: 32768
Root          Address : 48:0F:CF:AF:04:76    Priority: 32768
               Port      : 0                   Cost      : 0
               Rem Hops: 20

```

PORT TCN-Tx	ROLE TCN-Rx	STATE	COST	PRIORITY	TYPE	BPDU-Tx	BPDU-Rx
1/1/1 20	Designated 10	Forwarding	20000	128	P2P Edge	100	60
1/1/2 20	Designated 10	Forwarding	20000	128	P2P	100	60
1/1/3 20	Designated 10	Forwarding	20000	128	Shr	100	60
1/1/4 20	Designated 10	Forwarding	20000	128	Shr Edge	100	60
1/1/5 20	Alternate 10	Loop-Inc	20000	128	Shr Edge	100	60
1/1/6 20	Alternate 10	Root-Inc	20000	128	Shr Edge	100	60
1/1/7 20	Disabled 10	Down	20000	128	P2P	100	60

```

Topology change flag      : True
Number of topology changes : 4

```

Last topology change occurred : 516 seconds ago

MST2

Vlans mapped: 3

Bridge	Address : 48:0F:CF:AF:04:76	Priority: 32768
Root	Address : 48:0F:CF:AF:04:76	Priority: 32768
	Port : 0	Cost : 0
	Rem Hops: 20	

PORT	ROLE	STATE	COST	PRIORITY	TYPE	BPDU-Tx	BPDU-Rx
TCN-Tx	TCN-Rx						
1/1/1	Designated	Forwarding	20000	128	P2P Edge	100	60
20	10						
1/1/2	Designated	Forwarding	20000	128	P2P	100	60
20	10						
1/1/3	Designated	Forwarding	20000	128	Shr	100	60
20	10						
1/1/4	Designated	Forwarding	20000	128	Shr Edge	100	60
20	10						
1/1/5	Alternate	Loop-Inc	20000	128	Shr Edge	100	60
20	10						
1/1/6	Alternate	Root-Inc	20000	128	Shr Edge	100	60
20	10						

Topology change flag : True
Number of topology changes : 4
Last topology change occurred : 516 seconds ago

Command History

Release	Modification
10.09	A new state <code>Down</code> is added in the output.
10.07 or earlier	--

Command Information

Platforms	Command context	Authority
All platforms	Operator (>) or Manager (#)	Operators or Administrators or local user group members with execution rights for this command. Operators can execute this command from the operator context (>) only.

show spanning-tree mst-config

show spanning-tree mst-config [vsx-peer]

Description

Shows MSTP instance and corresponding VLAN information.

Parameter	Description
vsx-peer	Shows the output from the VSX peer switch. If the switches do not have the VSX configuration or the ISL is down, the output from the VSX peer switch is not displayed. This parameter is available on switches that support VSX.

Examples

Showing configuration information for MST instances and corresponding VLANs:

```
switch# show spanning-tree mst-config
MST configuration information
  MST config ID       : reg
  MST config revision : 1
  MST config digest   : 2D2BC9A32097B463C48EE1817673FA2D
  Number of instances : 2

Instance ID      Member VLANs
-----
0                2,4-4094
1                1
2                3
```

Command History

Release	Modification
10.07 or earlier	--

Command Information

Platforms	Command context	Authority
All platforms	Operator (>) or Manager (#)	Operators or Administrators or local user group members with execution rights for this command. Operators can execute this command from the operator context (>) only.

show spanning-tree mst detail

show spanning-tree mst detail [vsx-peer]

Description

Shows detailed information for all MST instances.

Parameter	Description
vsx-peer	Shows the output from the VSX peer switch. If the switches do not have the VSX configuration or the ISL is down, the output from the VSX peer switch is not displayed. This parameter is available on switches that support VSX.

Example

Showing detailed information for all MST instances:

```

switch# show spanning-tree mst detail
#### MST0
Vlans mapped: 2,4-4094
Bridge Address: 48:0F:CF:AF:04:76 Priority: 32768
Root
Regional Root
Operational Hello time : 2 seconds Forward delay: 15 seconds
              Max-age : 20 seconds TxHoldCount : 6 pps
Configured Hello time : 2 seconds Forward delay: 15 seconds
              Max-age : 20 seconds Max-Hops : 20
Root Address : 48:0F:CF:AF:04:76 Priority : 32768
      Port : 0 Path cost : 0
Regional Root Address : 48:0F:CF:AF:04:76 Priority : 32768
              Internal cost: 0 Rem Hops : 20

PORT      ROLE      STATE      COST      PRIORITY  TYPE      BPDU-Tx    BPDU-Rx
  TCN-Tx    TCN-Rx
-----
1/1/1     Designated Forwarding 20000      128      P2P Edge  100        60
  20        10
1/1/2     Designated Forwarding 20000      128      P2P      100        60
  20        10
1/1/3     Designated Forwarding 20000      128      Shr      100        60
  20        10
1/1/4     Designated Forwarding 20000      128      Shr Edge  100        60
  20        10
1/1/5     Alternate Loop-Inc  20000      128      Shr Edge  100        60
  20        10
1/1/6     Alternate Root-Inc  20000      128      Shr Edge  100        60
  20        10
1/1/7     Disabled Down      20000      128      P2P      100        60
  20        10

Topology change flag : True
Number of topology changes : 4
Last topology change occurred : 516 seconds ago

Port 1/1/1
Designated root address : 48:0F:CF:AF:04:76
Designated regional root address : 48:0F:CF:AF:04:76
Designated bridge address : 48:0F:CF:AF:04:76
Priority : 32768
BPDUs sent : 638
BPDUs received : 9
Message expiry : 1 second
Forward delay expiry : 18 seconds
Forward transitions : 3
TCN_Tx: 10, TCN_Rx: 10

Port 1/1/2
Designated root address : 48:0F:CF:AF:04:76
Designated regional root address : 48:0F:CF:AF:04:76
Designated bridge address : 48:0F:CF:AF:04:76
Priority : 32768
BPDUs sent : 641
BPDUs received : 11
Message expiry : 1 second
Forward delay expiry : 18 seconds
Forward transitions : 3
TCN_Tx: 10, TCN_Rx: 10

```

```

#### MST1
Vlans mapped: 1
Bridge      Address : 48:0F:CF:AF:04:76      Priority: 32768
Root        Address : 48:0F:CF:AF:04:76      Priority: 32768
            Port      : 0                    Cost      : 0
            Rem Hops: 20

```

PORT	ROLE	STATE	COST	PRIORITY	TYPE	BPDU-Tx	BPDU-Rx
TCN-Tx	TCN-Rx						
1/1/1	Designated	Forwarding	20000	128	P2P Edge	100	60
20	10						
1/1/2	Designated	Forwarding	20000	128	P2P	100	60
20	10						
1/1/3	Designated	Forwarding	20000	128	Shr	100	60
20	10						
1/1/4	Designated	Forwarding	20000	128	Shr Edge	100	60
20	10						
1/1/5	Alternate	Loop-Inc	20000	128	Shr Edge	100	60
20	10						
1/1/6	Alternate	Root-Inc	20000	128	Shr Edge	100	60
20	10						
1/1/7	Disabled	Down	20000	128	P2P	100	60
20	10						

```

Topology change flag      : True
Number of topology changes : 4
Last topology change occurred : 516 seconds ago

```

```

Port 1/1/1
Designated root address      : 48:0F:CF:AF:04:76
Designated bridge address    : 48:0F:CF:AF:04:76
Priority                      : 32768
BPDUs sent                   : 638
BPDUs received               : 9
Message expiry                : 1 second
Forward delay expiry         : 18 seconds
Forward transitions          : 4
TCN_Tx: 10, TCN_Rx: 10

```

```

Port 1/1/2
Designated root address      : 48:0F:CF:AF:04:76
Designated bridge address    : 48:0F:CF:AF:04:76
Priority                      : 32768
BPDUs sent                   : 641
BPDUs received               : 11
Message expiry                : 1 second
Forward delay expiry         : 18 seconds
Forward transitions          : 4
TCN_Tx: 10, TCN_Rx: 10

```

```

#### MST2
Vlans mapped: 3
Bridge      Address : 48:0F:CF:AF:04:76      Priority: 32768
Root        Address : 48:0F:CF:AF:04:76      Priority: 32768
            Port      : 0                    Cost      : 0
            Rem Hops: 20

```

PORT	ROLE	STATE	COST	PRIORITY	TYPE	BPDU-Tx	BPDU-Rx
TCN-Tx	TCN-Rx						


```

-----
-----
1/1/1    Designated Forwarding 20000    128      P2P Edge 100      60
 20      10
1/1/2    Designated Forwarding 20000    128      P2P      100      60
 20      10
1/1/3    Designated Forwarding 20000    128      Shr      100      60
 20      10
1/1/4    Designated Forwarding 20000    128      Shr Edge 100      60
 20      10
1/1/5    Alternate Loop-Inc  20000    128      Shr Edge 100      60
 20      10
1/1/6    Alternate Root-Inc  20000    128      Shr Edge 100      60
 20      10
1/1/7    Disabled Down      20000    128      P2P      100      60
 20      10

Topology change flag      : True
Number of topology changes : 4
Last topology change occurred : 516 seconds ago

Port 1/1/1
Designated root address      : 48:0F:CF:AF:04:76
Designated bridge address    : 48:0F:CF:AF:04:76
Priority                      : 32768
BPDUs sent                   : 638
BPDUs received               : 9
Message expiry               : 1 second
Forward delay expiry         : 18 seconds
Forward transitions          : 3
TCN_Tx: 10, TCN_Rx: 10

Port 1/1/2
Designated root address      : 48:0F:CF:AF:04:76
Designated bridge address    : 48:0F:CF:AF:04:76
Priority                      : 32768
BPDUs sent                   : 641
BPDUs received               : 11
Message expiry               : 1 second
Forward delay expiry         : 18 seconds
Forward transitions          : 3
TCN_Tx: 10, TCN_Rx: 10

```

Command History

Release	Modification
10.09	A new state <code>Down</code> is added in the output.
10.07 or earlier	--

Command Information

Platforms	Command context	Authority
All platforms	Operator (>) or Manager (#)	Operators or Administrators or local user group members with execution rights for this command. Operators can execute this command from the operator context (>) only.

show spanning-tree mst <INSTANCE-ID>

show spanning-tree mst <INSTANCE-ID> [vsx-peer]

Description

Displays MSTP configurations for the given instance ID.

Parameter	Description
<INSTANCE-ID>	Specifies the MSTP instance number. Range: 0 to 64.
vsx-peer	Shows the output from the VSX peer switch. If the switches do not have the VSX configuration or the ISL is down, the output from the VSX peer switch is not displayed. This parameter is available on switches that support VSX.

Example

```
switch# show spanning-tree mst 1
```

```
#### MST1
```

```
Vlans mapped: 1
```

```
Bridge      Address : 48:0F:CF:AF:04:76      Priority: 32768
```

```
Root        Address : 48:0F:CF:AF:04:76      Priority: 32768
```

```
Port        : 0                      Cost    : 0
```

```
Rem Hops: 20
```

PORT	ROLE	STATE	COST	PRIORITY	TYPE	BPDU-Tx	BPDU-Rx
TCN-Tx	TCN-Rx						
1/1/1	Designated	Forwarding	20000	128	P2P Edge	100	60
20	10						
1/1/2	Designated	Forwarding	20000	128	P2P	100	60
20	10						
1/1/3	Designated	Forwarding	20000	128	Shr	100	60
20	10						
1/1/4	Designated	Forwarding	20000	128	Shr Edge	100	60
20	10						
1/1/5	Alternate	Loop-Inc	20000	128	Shr Edge	100	60
20	10						
1/1/6	Alternate	Root-Inc	20000	128	Shr Edge	100	60
20	10						
1/1/7	Disabled	Down	20000	128	P2P Bound	100	60
20	10						

```
Topology change flag      : True
```

```
Number of topology changes : 4
```

```
Last topology change occurred : 516 seconds ago
```

Command History

Release	Modification
10.09	A new state <code>Down</code> is added in the output.
10.07 or earlier	--

Command Information

Platforms	Command context	Authority
All platforms	Operator (>) or Manager (#)	Operators or Administrators or local user group members with execution rights for this command. Operators can execute this command from the operator context (>) only.

show spanning-tree mst <INSTANCE-ID> detail

show spanning-tree mst <INSTANCE-ID> detail [vsx-peer]

Description

Displays MSTP configurations for the given instance ID with corresponding port details.

Parameter	Description
<INSTANCE-ID>	Specifies the MSTP instance number. Range: 0 to 64.
vsx-peer	Shows the output from the VSX peer switch. If the switches do not have the VSX configuration or the ISL is down, the output from the VSX peer switch is not displayed. This parameter is available on switches that support VSX.

Example

```
switch# show spanning-tree mst 1 detail
```

```
#### MST1
```

```
Vlans mapped: 1
```

```
Bridge Address : 48:0F:CF:AF:04:76 Priority: 32768
```

```
Root Address : 48:0F:CF:AF:04:76 Priority: 32768
```

```
Port : 0 Cost : 0
```

```
Rem Hops: 20
```

PORT	ROLE	STATE	COST	PRIORITY	TYPE	BPDU-Tx	BPDU-Rx
TCN-Tx	TCN-Rx						
1/1/1	Designated	Forwarding	20000	128	P2P Edge	100	60
20	10						
1/1/2	Designated	Forwarding	20000	128	P2P	100	60
20	10						
1/1/3	Designated	Forwarding	20000	128	Shr	100	60
20	10						
1/1/4	Designated	Forwarding	20000	128	Shr Edge	100	60
20	10						
1/1/5	Alternate	Loop-Inc	20000	128	Shr Edge	100	60
20	10						
1/1/6	Alternate	Root-Inc	20000	128	Shr Edge	100	60
20	10						
1/1/7	Disabled	Down	20000	128	P2P Bound	100	60
20	10						

```
Topology change flag : True
```

```
Number of topology changes : 4
```

```
Last topology change occurred : 516 seconds ago
```

```
Port 1/1/1
```

```

Designated root address      : 48:0F:CF:AF:04:76
Designated bridge address   : 48:0F:CF:AF:04:76
Priority                     : 32768
BPDUs sent                  : 667
BPDUs received              : 9
Message expiry              : 0 second
Forward delay expiry        : 18 seconds
Forward transitions          : 4
TCN_Tx: 10, TCN_Rx: 10

Port 1/1/2
Designated root address      : 48:0F:CF:AF:04:76
Designated bridge address   : 48:0F:CF:AF:04:76
Priority                     : 32768
BPDUs sent                  : 670
BPDUs received              : 11
Message expiry              : 0 second
Forward delay expiry        : 18 seconds
Forward transitions          : 4
TCN_Tx: 10, TCN_Rx: 10

```

Command History

Release	Modification
10.09	A new state <code>Down</code> is added in the output.
10.07 or earlier	--

Command Information

Platforms	Command context	Authority
All platforms	Operator (>) or Manager (#)	Operators or Administrators or local user group members with execution rights for this command. Operators can execute this command from the operator context (>) only.

show spanning-tree mst interface

`show spanning-tree mst <INSTANCE-ID> interface <IFNAME> [vsx-peer]`

Description

Shows MSTP configurations for the given instance ID with corresponding port details.

Parameter	Description
<INSTANCE-ID>	Specifies the MSTP instance number. Range: 0 to 64.
<IFNAME>	Specifies an interface.
vsx-peer	Shows the output from the VSX peer switch. If the switches do not have the VSX configuration or the ISL is down, the output from the VSX peer switch is not displayed. This parameter is available on switches that support VSX.

Examples

Showing MST configuration and port details:

```
switch# show spanning-tree mst 1 interface 1/1/1
Port 1/1/1
```

Instance	Role	State	Cost	Priority	Vlans mapped
1	Designated	Forwarding	20000	128	1

Command History

Release	Modification
10.07 or earlier	--

Command Information

Platforms	Command context	Authority
All platforms	Operator (>) or Manager (#)	Operators or Administrators or local user group members with execution rights for this command. Operators can execute this command from the operator context (>) only.

show spanning-tree summary port

show spanning-tree summary port

Description

Shows spanning tree port summary information.

Example

Showing summary of spanning tree ports:

```
switch# show spanning-tree summary port
```

```
STP status           : Enabled
Protocol             : MSTP
BPDU guard timeout value : None
BPDU guard enabled interfaces : 1/1/1-1/1/9,1/1/11,1/1/13,1/1/15,1/1/17,1/1/19,
                               1/1/21,lag1,lag2
BPDU filter enabled interfaces : None
Root guard enabled interfaces : 1/1/3
Loop guard enabled interfaces : 1/1/2
TCN guard enabled interfaces : 1/1/1-1/1/3
RPVST filter enabled interfaces : 1/1/37
RPVST guard enabled interfaces : None
```

Interface count by state

Instance ID	Blocking	Listening	Learning	Forwarding	Down
0	2	0	0	15	0
1	2	0	0	15	0
2	2	0	0	15	0

Total = 3 6 0 0 45 0

Command History

Release	Modification
10.09	A new state <code>Down</code> is added in the output.
10.07 or earlier	--

Command Information

Platforms	Command context	Authority
All platforms	Operator (>) or Manager (#)	Operators or Administrators or local user group members with execution rights for this command. Operators can execute this command from the operator context (>) only.

show spanning-tree summary root

show spanning-tree summary root

Description

Shows spanning tree root summary information.

Example

Showing spanning tree root summary:

```
switch# show spanning-tree summary root

STP status           : Enabled
Protocol             : MSTP
System ID            : 70:72:cf:32:50:f5

Root bridge for STP Instance : 0,1,2

Instance ID      Priority Root ID      Root cost  Hello Time  Max Age  Fwd Dly  Root Port
-----
0                 32768 70:72:cf:32:50:f5      0          2    20    15      n/a
1                 32768 70:72:cf:32:50:f5      0          2    20    15      n/a
2                 32768 70:72:cf:32:50:f5     200         2    20    15      1/1/1
```

Command History

Release	Modification
10.07 or earlier	--

Command Information

Platforms	Command context	Authority
All platforms	Operator (>) or Manager (#)	Operators or Administrators or local user group members with execution rights for this command. Operators can execute this command from the operator context (>) only.

spanning-tree

spanning-tree
no spanning-tree

Description

Enables the spanning tree protocol on the switch.

The `no` form of this command disables the spanning tree protocol on the switch.

Examples

Enabling spanning tree:

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

Disabling spanning tree:

```
switch(config)# no spanning-tree
```

Command History

Release	Modification
10.07 or earlier	--

Command Information

Platforms	Command context	Authority
All platforms	config	Administrators or local user group members with execution rights for this command.

spanning-tree bpdud-filter

spanning-tree bpdud-filter
no spanning-tree bpdud-filter

Description

Enables the bpdud filter for the interface.

The 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, does not transmit BPDU, and stays locked in the spanning tree forwarding state. All other ports maintain their role. Typical uses for this parameter include:

- To have MSTP 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.



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, redundant links) using these ports. If you suddenly have a high load, disconnect the link and disable the BPDU filter (using the `no` command.)

The `no` form of the command sets the bpdu filter status to the default of disabled on the interface.

Examples

Enabling the bpdu filter on interface 1/1/1:

```
switch(config)# interface 1/1/1
switch(config-if)# spanning-tree bpdu-filter
```

Disabling bpdu filter on interface 1/1/1:

```
switch(config)# interface 1/1/1
switch(config-if)# no spanning-tree bpdu-filter
```

Command History

Release	Modification
10.07 or earlier	--

Command Information

Platforms	Command context	Authority
All platforms	<code>config-if</code>	Administrators or local user group members with execution rights for this command.

spanning-tree bpdu-guard

```
spanning-tree bpdu-guard
no spanning-tree bpdu-guard
```

Description

Enables the BPDU guard on the selected switch interface. When BPDU guard is enabled, interfaces receiving MSTP BPDUs become disabled.

BPDU protection is a security feature designed to protect the active MSTP topology by preventing spoofed BPDU packets from entering the MSTP domain. In a typical implementation, BPDU protection would be applied to edge ports connected to end user devices that do not run MSTP. If MSTP BPDU packets are received on a protected port, this feature disables that port and alerts the network manager using an SNMP trap.

Occasionally a hardware or software failure can cause MSTP to fail, creating 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 MSTP BPDUs.

The `no` form of the command disables BPDU guard on the selected interface.

Examples

Enabling the BPDU guard on interface 1/1/1:

```
switch(config)# interface 1/1/1
switch(config-if)# spanning-tree bpdu-guard
```

Disabling BPDU guard on interface 1/1/1:

```
switch(config)# interface 1/1/1
switch(config-if)# no spanning-tree bpdu-guard
```

Command History

Release	Modification
10.07 or earlier	--

Command Information

Platforms	Command context	Authority
All platforms	config-if	Administrators or local user group members with execution rights for this command.

spanning-tree bpdu-guard timeout

```
spanning-tree bpdu-guard timeout <INTERVAL>
no spanning-tree bpdu-guard timeout [<INTERVAL>]
```

Description

Enables and configures the auto re-enable timeout in seconds for all interfaces with BPDU guard enabled. When an interface is disabled after receiving an unauthorized BPDU it will automatically be re-enabled after the timeout expires. The default is for the interface to stay disabled until manually re-enabled.

The `no` form of the command disables BPDU guard timeout on the interface. This is the default.

Parameter	Description
<INTERVAL>	Specifies the re-enable timeout in seconds. Range: 1 to 65535.

Example

Enabling the BPDU guard timeout on interface 1/1/1:

```
switch(config)# interface 1/1/1
switch(config-if)# spanning-tree bpdu-guard timeout 10
```

Disabling BPDU guard timeout on interface 1/1/1:

```
switch(config)# interface 1/1/1
switch(config-if)# no spanning-tree bpdu-guard
```

Command History

Release	Modification
10.07 or earlier	--

Command Information

Platforms	Command context	Authority
All platforms	config	Administrators or local user group members with execution rights for this command.

spanning-tree config-name

```
spanning-tree config-name <CONFIG-NAME>
no spanning-tree config-name [<CONFIG-NAME>]
```

Description

Sets the configuration name for the MST region in which the switch resides.

All switches within an MST region must have identical configuration names. 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 `no` form of this command overwrites the currently configured name with the default name. The default name is a text string using the hexadecimal representation of the system MAC address.

Parameter	Description
<CONFIG-NAME>	Specifies the configuration name for the MST region in which the switch resides. Default: text string using the hexadecimal representation of the MAC address of the switch. Range: 1 - 32 nonblank characters (case-sensitive).

Examples

Setting the configuration name to MST0:

```
switch(config)# spanning-tree config-name MST0
```

Setting the configuration name to the default value:

```
switch(config)# no spanning-tree config-name
```

Command History

Release	Modification
10.07 or earlier	--

Command Information

Platforms	Command context	Authority
All platforms	config	Administrators or local user group members with execution rights for this command.

spanning-tree config-revision

```
spanning-tree config-revision <REVISION-NUMBER>  
no spanning-tree config-revision [<REVISION-NUMBER>]
```

Description

Configures the revision number for the MST region in which the switch resides. All switches within an MST region must have identical revision numbers. Use this setting to differentiate between region configurations. For example, when changing configuration settings within a region where you want to track the configuration versions you use, or when creating a new region from a subset of switches in a current region and you want to maintain the same region name.

The **no** form of this command overwrites the currently configured revision number of the MST region and sets it to the default value of 0.

Parameter	Description
<REVISION-NUMBER>	Specifies the revision number for the MST region in which the switch resides. Range: 0 - 65535. Default: 0.

Examples

Setting the revision to 40:

```
switch(config)# spanning-tree config-revision 40
```

Setting the revision to the default value:

```
switch(config)# no spanning-tree config-revision
```

Command History

Release	Modification
10.07 or earlier	--

Command Information

Platforms	Command context	Authority
All platforms	config	Administrators or local user group members with execution rights for this command.

spanning-tree cost

```
spanning-tree cost <PORT-COST>
no spanning-tree cost [<PORT-COST>]
```

Description

Sets individual port cost for MSTI 0.

For a given port, the path cost setting can be different for different MSTIs to which the port may belong.

The switch uses the path cost to determine which ports are the forwarding ports in the MSTI; that is, which links to use for the active topology of the MSTI and which ports to block.

Cost gets calculated based on physical interface link speed. It is not based on cumulative speed of all physical links under a lag. Therefore, the cost will be same for a 1G interface and 2x1G lag interfaces.

The `no` form of the command sets the port cost for MSTI 0 instance to the default value.

Parameter	Description
<PORT-COST>	Specifies the cost of the port for MSTI 0. Range: 1-200,000,000. Default is calculated from the port link speed: <ul style="list-style-type: none"> 10 Mbps link speed equals a path cost of 2,000,000. 100 Mbps link speed equals a path cost of 200,000. 1 Gbps link speed equals a path cost of 20,000. 10 Gbps link speed equals a path cost of 2,000. 100 Gbps link speed equals a path cost of 200. 1 Tbps link speed equals a path cost of 20.

Examples

Setting the cost to **2000** on interface **1/1/1**:

```
switch(config)# interface 1/1/1
switch(config-if)# spanning-tree cost 2000
```

Setting the cost to the default on interface **1/1/1**:

```
switch(config)# interface 1/1/1
switch(config-if)# no spanning-tree cost
```

Command History

Release	Modification
10.07 or earlier	--

Command Information

Platforms	Command context	Authority
All platforms	config-if	Administrators or local user group members with execution rights for this command.

spanning-tree forward-delay

```
spanning-tree forward-delay <DELAY-IN-SECS>
no spanning-tree forward-delay [<DELAY-IN-SECS>]
```

Description

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

The `no` form of this command sets forward delay time for the bridge to the default of 15 seconds.

Parameter	Description
<DELAY-IN-SECS>	Specifies the forward delay time in seconds. Default: 15 seconds. Range: 4-30.

Examples

Setting forward delay to 6 seconds:

```
switch(config)# spanning-tree forward-delay 6
```

Setting forward delay to the default of 15 seconds:

```
switch(config)# no spanning-tree forward-delay
```

Command History

Release	Modification
10.07 or earlier	--

Command Information

Platforms	Command context	Authority
All platforms	config	Administrators or local user group members with execution rights for this command.

spanning-tree hello-time

```
spanning-tree hello-time <HELLO-IN-SECS>
no spanning-tree hello-time [<HELLO-IN-SECS>]
```

Description

Configures the transmission interval between consecutive Bridge Protocol Data Units (BPDU) that the switch sends as a root bridge. The hello time interval is inserted in outbound BPDUs.

The `no` form of this command sets hello time to the default of 2 seconds.

Parameter	Description
<HELLO-IN-SECS>	Specifies the hello time interval in seconds. Default: 2 seconds. Range: 2-10.

Examples

Setting the hello time interval to 6 seconds:

```
switch(config)# spanning-tree hello-time 6
```

Setting the hello time interval to the default of 2 seconds:

```
switch(config)# no spanning-tree hello-time
```

Command History

Release	Modification
10.07 or earlier	--

Command Information

Platforms	Command context	Authority
All platforms	config	Administrators or local user group members with execution rights for this command.

spanning-tree instance cost

```
spanning-tree instance <INSTANCE-ID> cost <PORT-COST>
no spanning-tree instance <INSTANCE-ID> cost [<PORT-COST>]
```

Description

Sets the individual port cost for an MSTI. The switch uses the path cost to determine which links to use for the active topology of the MSTI (forwarding ports) and which ports to block. The path cost setting for a port can be different on each MSTI to which the port belongs.

The `no` form of this command sets the port cost for an MSTI to the default value.

Parameter	Description
<INSTANCE-ID>	Specifies the MSTI number. Range: 1-64.
<PORT-COST>	Specifies the cost of the port for the MSTI. Range: 1-200000000. Default value is calculated from the port link speed: <ul style="list-style-type: none"> 10 Mbps link speed equals a path cost of 2000000. 100 Mbps link speed equals a path cost of 200000. 1 Gbps link speed equals a path cost of 20000.

Examples

Setting the port **1/1/1** cost for MSTI **1** to **2000**:

```
switch(config)# interface 1/1/1
switch(config-if)# spanning-tree instance 1 cost 2000
```

Setting the port **1/1/1** cost for MSTI **1** to the default:

```
switch(config)# interface 1/1/1
switch(config-if)# no spanning-tree instance 1 cost
```

Command History

Release	Modification
10.07 or earlier	--

Command Information

Platforms	Command context	Authority
All platforms	config-if	Administrators or local user group members with execution rights for this command.

spanning-tree instance port-priority

```
spanning-tree instance <INSTANCE-ID> port-priority <PRIORITY-MULTIPLIER>
no spanning-tree instance <INSTANCE-ID> port-priority [<PRIORITY-MULTIPLIER>]
```

Description

Configures the priority as a priority multiplier 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 **no** form of this command sets the port priority to the default value of 8 for the MST instance. The default priority value is derived by multiplying 8 by 16.

Parameter	Description
<INSTANCE-ID>	Specifies the MSTP instance number. Range: 1-64.
<PRIORITY-MULTIPLIER>	Specifies the priority as a multiplier. Default: 8. Range: 0 to 15. The priority range for a port in a given MST instance is 0 to 255. However, this command specifies the priority as a multiplier (0 to 15) of 16. When you specify a priority multiplier of 0 to 15, the actual priority assigned to the switch is: (priority-multiplier) x 16.

Examples

Setting the port **1/1/1** priority for instance **1** to **8**:

```
switch(config)# interface 1/1/1
switch(config-if)# spanning-tree instance 1 port-priority 8
```

Setting the port 1/1/1 priority for instance 1 to the default:

```
switch(config)# interface 1/1/1
switch(config-if)# no spanning-tree instance 1 port-priority
```

Command History

Release	Modification
10.07 or earlier	--

Command Information

Platforms	Command context	Authority
All platforms	config-if	Administrators or local user group members with execution rights for this command.

spanning-tree instance priority

```
spanning-tree instance <INSTANCE-ID> priority <PRIORITY-MULTIPLIER>
no spanning-tree instance <INSTANCE-ID> priority [<PRIORITY-MULTIPLIER>]
```

Description

Sets the switch priority for the specified MST instance.

The **no** form of this command sets the priority for the specified instance to the default of 8.

Parameter	Description
<INSTANCE-ID>	Specifies the MSTP instance number. Range: 1 to 64.
<PRIORITY-MULTIPLIER>	Specifies the priority as a multiplier. Default: 8. Range: 0 to 15. The priority range for an MSTP switch is 0-61440. However, this command specifies the priority as a multiplier (0 - 15) of 4096.

Parameter	Description
	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.

Examples

Setting the priority multiplier for instance 1 to 5:

```
switch(config)# spanning-tree instance 1 priority 5
```

Setting the priority multiplier for instance 1 to the default of 8:

```
switch(config)# no spanning-tree instance 1 priority
```

Command History

Release	Modification
10.07 or earlier	--

Command Information

Platforms	Command context	Authority
All platforms	config	Administrators or local user group members with execution rights for this command.

spanning-tree instance vlan

```
spanning-tree instance <INSTANCE-ID> vlan <VLAN-ID>
no spanning-tree instance <INSTANCE-ID> vlan <VLAN-ID>
```

Description

Creates a new instance with VLANs mapped or maps VLANs to an existing instance.

Each instance must have at least one VLAN mapped to it. 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 on the switch.

The **no** form of this command removes the specified VLAN from the MSTP instance.

Parameter	Description
<INSTANCE-ID>	Specifies the MSTP instance number. Range: 1 to 64.
<VLAN-ID>	Specifies a VLAN ID number.

Examples

Mapping VLAN 1 to instance 1:

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

Removing VLAN 1 from instance 1:

```
switch(config)# no spanning-tree instance 1 vlan 1
```

Command History

Release	Modification
10.07 or earlier	--

Command Information

Platforms	Command context	Authority
All platforms	config	Administrators or local user group members with execution rights for this command.

spanning-tree link-type

```
spanning-tree link-type {point-to-point|shared}
```

Description

Specifies the link type of the interface, which is normally derived from the duplex setting of the port. The default setting depends on the duplex mode of the port: full-duplex ports are point-to-point, half-duplex ports are shared.

Parameter	Description
point-to-point	Specifies the link type as point-to-point.
shared	Specifies the link type as shared.

Examples

Setting the link type to point-to-point on interface **1/1/1**:

```
switch(config)# interface 1/1/1
switch(config-if)# spanning-tree link-type point-to-point
```

Setting the link type to shared on interface **1/1/1**:

```
switch(config)# interface 1/1/1
switch(config-if)# spanning-tree link-type shared
```

Command History

Release	Modification
10.07 or earlier	--

Command Information

Platforms	Command context	Authority
All platforms	config-if	Administrators or local user group members with execution rights for this command.

spanning-tree loop-guard

spanning-tree loop-guard
no spanning-tree loop-guard

Description

Enables the loop guard on the interface. STP loop guard is best applied on blocking or forwarding ports. The `no` form of the command sets the loop guard status to the default of disabled on the interface.

Usage

Occasionally a hardware or software failure can cause MSTP to fail, creating 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 MSTP BPDUs.

Loop guard causes the non-designated port to go into the MSTP 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 MSTP operation automatically.

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 to an inconsistent state, and does not forward the traffic through the link, thus avoiding loop creation.

Examples

Enabling the loop guard on interface **1/1/1**:

```
switch(config)# interface 1/1/1
switch(config-if)# spanning-tree loop-guard
```

Disabling loop guard on interface **1/1/1**:

```
switch(config)# interface 1/1/1
switch(config-if)# no spanning-tree loop-guard
```

Command History

Release	Modification
10.07 or earlier	--

Command Information

Platforms	Command context	Authority
All platforms	config-if	Administrators or local user group members with execution rights for this command.

spanning-tree max-age

```
spanning-tree max-age <AGE-IN-SECS>  
no spanning-tree max-age [<AGE-IN-SECS>]
```

Description

Sets the maximum age timer, which specifies the maximum age value that the switch inserts in outbound BPDU packets it sends as a root bridge. Max-age is the interval, specified in the BPDU, that BPDU data remains valid after its reception.

The bridge recomputes the spanning tree topology if it does not receive a new BPDU before max-age expiry.

The `no` form of this command sets the max-age value to the default of 20 seconds.

Parameter	Description
<AGE-IN-SECS>	Specifies the max-age in seconds. Range: 6 to 40. Default: 20.

Examples

Setting the max-age to 10 seconds:

```
switch(config)# spanning-tree max-age 10
```

Setting the max-age to the default of 20 seconds:

```
switch(config)# no spanning-tree max-age
```

Command History

Release	Modification
10.07 or earlier	--

Command Information

Platforms	Command context	Authority
All platforms	config	Administrators or local user group members with execution rights for this command.

spanning-tree max-hops

```
spanning-tree max-hops <HOP-COUNT>  
no spanning-tree max-hops [<HOP-COUNT>]
```

Description

Configures the max hop setting that the switch inserts into BPDUs that it sends out as the root bridge. The max hop setting determines the number of bridges in an MST region that a BPDU can traverse before it is discarded.

The `no` form of this command sets the maximum number of hops to the default of 20.

Parameter	Description
<code><HOP-COUNT></code>	Specifies the maximum number of hops. Range: 1 to 40. Default: 20.

Examples

Setting the hop count to 10:

```
switch(config)# spanning-tree max-hops 10
```

Setting the max-age to the default of 20:

```
switch(config)# no spanning-tree max-hops
```

Command History

Release	Modification
10.07 or earlier	--

Command Information

Platforms	Command context	Authority
All platforms	<code>config</code>	Administrators or local user group members with execution rights for this command.

spanning-tree mode

```
spanning-tree mode {mstp|rpvst}  
no spanning-tree mode [mstp|rpvst]
```

Description

Sets the spanning tree mode to either MSTP mode (Multiple-instance Spanning Tree Protocol) or RPVST mode (Rapid Per VLAN Spanning Tree).

The `no` form of this command sets the spanning tree mode to the default `mstp`.

Parameter	Description
<code>mstp</code>	Sets the mode to MSTP (Multiple-instance Spanning Tree Protocol), which applies the STP (spanning tree protocol) separately for each set of VLANs (called an MSTI - multiple

Parameter	Description
	spanning tree instance).
rpvst	Sets the mode to RPVST (Rapid Per VLAN Spanning Tree).

Examples

Enabling MSTP mode:

```
switch(config)# spanning-tree mode mstp
```

Enabling RPVST mode:

```
switch(config)# spanning-tree mode rpvst
```

Command History

Release	Modification
10.07 or earlier	--

Command Information

Platforms	Command context	Authority
All platforms	config	Administrators or local user group members with execution rights for this command.

spanning-tree port-priority

```
spanning-tree port-priority <PRIORITY-MULTIPLIER>
no spanning-tree port-priority [<PRIORITY-MULTIPLIER>]
```

Description

Configures the port priority. The priority of a port can be different for each MST instance to which it belongs.

The **no** form of the command sets the port priority for MST instance 0 to the default of 8. The default priority value is derived by multiplying 8 by 8. For LAG interfaces the default is 4.

Parameter	Description
<PRIORITY-MULTIPLIER>	Specifies the port priority as a multiplier. Default: 8, except for LAG interfaces where the default is 4. Range: 0 to 15. The priority range for a port in a given MSTI is 0 to 255. However, this command specifies the priority as a multiplier (0 to 15) of 16. When you specify a priority multiplier of 0 to 15, the actual priority assigned to the switch is: (priority-multiplier) x 16.

Examples

Setting the port priority to 8 on interface **1/1/1**:

```
switch(config)# interface 1/1/1
switch(config-if)# spanning-tree port-priority 8
```

Setting the port priority to the default on interface **1/1/1**:

```
switch(config)# interface 1/1/1
switch(config-if)# no spanning-tree port-priority
```

Command History

Release	Modification
10.07 or earlier	--

Command Information

Platforms	Command context	Authority
All platforms	config-if	Administrators or local user group members with execution rights for this command.

spanning-tree port-type

```
spanning-tree port-type {admin-edge|admin-network}
no spanning-tree port-type [admin-edge|admin-network]
```

Description

Sets the STP port type for the interface.

Port types include: admin-edge and admin-network.

The **no** form of the command sets the port type to the default of admin-network.

Parameter	Description
admin-edge	Specifies the port type as administrative edge. During spanning tree establishment, ports with admin-edge enabled transition immediately to the forwarding state.
admin-network	Specifies the port type as administrative network. When this option is selected, 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.

Examples

Setting the port type to admin-edge on interface **1/1/1**:

```
switch(config)# interface 1/1/1
switch(config-if)# spanning-tree port-type admin-edge
```

Setting the port type to admin-network on interface **1/1/1**:

```
switch(config)# interface 1/1/1
switch(config-if)# spanning-tree port-type admin-network
```

Setting the port type to the default of admin-network on interface 1/1/1:

```
switch(config)# interface 1/1/1
switch(config-if)# no spanning-tree port-type
```

Command History

Release	Modification
10.07 or earlier	--

Command Information

Platforms	Command context	Authority
All platforms	config-if	Administrators or local user group members with execution rights for this command.

spanning-tree priority

```
spanning-tree priority <PRIORITY-MULTIPLIER>
no spanning-tree priority [<PRIORITY-MULTIPLIER>]
```

Description

Configures 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.

The `no` form of this command sets the bridge priority to the default of 8. The default priority value is derived by multiplying 8 by 4096.

Parameter	Description
<PRIORITY-MULTIPLIER>	Specifies the priority as a multiplier. Range: 0 to 15. Default: 8. The priority range for an MSTP switch is 0-61440. However, this command specifies the priority as a multiplier (0 to 15) of 4096. That is, when you specify a priority multiplier value of 0 to 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.

Usage

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.

Examples

Setting the priority multiplier to 12:

```
switch(config)# spanning-tree priority 12
```

Setting the priority multiplier to the default of 8:

```
switch(config)# no spanning-tree priority
```

Command History

Release	Modification
10.07 or earlier	--

Command Information

Platforms	Command context	Authority
All platforms	config	Administrators or local user group members with execution rights for this command.

spanning-tree root-guard

```
spanning-tree root-guard  
no spanning-tree root-guard
```

Description

Enables the root guard on the interface.

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 MSTP 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 `no` form of the command sets the root guard status to the default of disabled on the interface.

Examples

Enabling the root guard on interface **1/1/1**:

```
switch(config)# interface 1/1/1  
switch(config-if)# spanning-tree root-guard
```

Disabling root guard on interface **1/1/1**:

```
switch(config)# interface 1/1/1  
switch(config-if)# no spanning-tree root-guard
```

Command History

Release	Modification
10.07 or earlier	--

Command Information

Platforms	Command context	Authority
All platforms	config-if	Administrators or local user group members with execution rights for this command.

spanning-tree rpvst-filter

```
spanning-tree rpvst-filter
no spanning-tree rpvst-filter
```

Description

Enables the RPVST filter for the interface. This command is only applicable to MSTP mode. It is not applicable to RPVST+ mode.

When the RPVST filter is enabled, the ingressing RPVST proprietary BPDUs are dropped after copying to CPU whereas the standard IEEE RPVST BPDUs are still allowed. This helps in preventing the flooding of RPVST proprietary BPDUs under an MSTP-RPVST interop environment.



If the neighboring switch is running RPVST then this pair of switches will not converge as RPVST BPDUs will not reach them.

If enabling RPVST filter causes a high traffic load, shutdown the port and reconfigure the BPDU filter with the CLI command: `no spanning tree rpvst-filter`.

RPVST filter is disabled by default.

Example

Enabling the RPVST filter on interface 1/1/1:

```
switch# configure terminal
switch(config)# interface 1/1/1
switch(config-if)# spanning-tree rpvst-filter
```

Disabling RPVST filter on interface 1/1/1:

```
switch# configure terminal
switch(config)# interface 1/1/1
switch(config-if)# no spanning-tree rpvst-filter
```

Command History

Release	Modification
10.07 or earlier	--

Command Information

Platforms	Command context	Authority
All platforms	config-if	Administrators or local user group members with execution rights for this command.

spanning-tree rpvst-guard

spanning-tree rpvst-guard
no spanning-tree rpvst-guard

Description

Enables RPVST guard on the switch interface. This command is only applicable to MSTP mode. It is not applicable to RPVST+ mode.

When RPVST guard is enabled on an interface, it will disable that interface if RPVST BPDUs are received on it.

The `no` form of the command sets the RPVST guard status to the default of disabled on the interface.

Example

Enabling RPVST guard on interface 1/1/1:

```
switch# configure terminal
switch(config)# interface 1/1/1
switch(config-if)# spanning-tree rpvst-guard
```

Disabling RPVST guard on interface 1/1/1:

```
switch# configure terminal
switch(config)# interface 1/1/1
switch(config-if)# no spanning-tree rpvst-guard
```

Command History

Release	Modification
10.07 or earlier	--

Command Information

Platforms	Command context	Authority
All platforms	config-if	Administrators or local user group members with execution rights for this command.

spanning-tree tcn-guard

spanning-tree tcn-guard
no spanning-tree tcn-guard

Description

Enables the TCN (Topology Change Notification) guard in the interface. When enabled for a port, the port stops propagating received topology change notifications and topology changes to other ports.

The **no** form of the command sets the TCN guard status to the default of disabled on the interface.

Examples

Enabling TCN guard on interface **1/1/1**:

```
switch(config)# interface 1/1/1
switch(config-if)# spanning-tree tcn-guard
```

Disabling TCN guard on interface **1/1/1**:

```
switch(config)# interface 1/1/1
switch(config-if)# no spanning-tree tcn-guard
```

Command History

Release	Modification
10.07 or earlier	--

Command Information

Platforms	Command context	Authority
All platforms	config-if	Administrators or local user group members with execution rights for this command.

spanning-tree transmit-hold-count

```
spanning-tree transmit-hold-count <COUNT>
no spanning-tree transmit-hold-count [<COUNT>]
```

Description

Sets the maximum number of BPDUs per second that the switch can send from an interface.

The **no** form of this command sets the transmit-hold-count to the default of 6.

Parameter	Description
<COUNT>	Specifies the number of BPDUs that can be sent per second. Range: 1 to 10. Default: 6.

Examples

Setting the transmit-hold-count to 5:

```
switch(config)# spanning-tree transmit-hold-count 5
```

Setting the transmit-hold-count to the default of 6:

```
switch(config)# no spanning-tree transmit-hold-count
```

Command History

Release	Modification
10.07 or earlier	--

Command Information

Platforms	Command context	Authority
All platforms	config	Administrators or local user group members with execution rights for this command.

spanning-tree trap

```
spanning-tree trap {new-root|topology-change [instance <INSTANCE-ID>] |  
    errant-bpdu | root-guard-inconsistency | loop-guard-inconsistency}  
no spanning-tree trap {new-root|topology-change [instance <INSTANCE-ID>] |  
    errant-bpdu | root-guard-inconsistency | loop-guard-inconsistency}
```

Description

Enables SNMP traps for new root, topology change event, errant-bpdu received event, root-guard inconsistency, and loop-guard inconsistency notifications. It is disabled by default.

The `no` form of this command disables the notifications for SNMP traps.

Parameter	Description
<code>new-root</code>	Enabling SNMP notification when a new root is elected on any MST instance on the switch.
<code>topology-change</code>	Enabling SNMP notification when a topology change event occurs in the specified MST instance on the switch.
<code><INSTANCE-ID></code>	Specifies the instance ID for the topology change trap. Range: 0 to 64.
<code>errant-bpdu</code>	Enabling SNMP notification when an errant bpdu is received by any MST instance on the switch.
<code>root-guard-inconsistency</code>	Enabling SNMP notification when the root-guard finds the port inconsistent for any MST instance on the switch.
<code>loop-guard-inconsistency</code>	Enabling SNMP notification when the loop-guard finds the port inconsistent for any MST instance on the switch.

Examples

Enabling the notifications for the SNMP traps:

```
switch(config)# spanning-tree trap  
    new-root          Enable notifications which are sent when a new root is  
    elected  
    topology-change   Enable notifications which are sent when a topology  
    change occurs  
    errant-bpdu        Enable notifications which are sent when an errant
```

```

bpdu is received
  root-guard-inconsistency  Enable notifications which are sent when root guard
inconsistency occurs
  loop-guard-inconsistency  Enable notifications which are sent when loop guard
inconsistency occurs
switch(config)# spanning-tree trap new-root
<cr>
switch(config)# spanning-tree trap topology-change
  instance  Enable topology change notification for the specified MST instance id.
switch(config)# spanning-tree trap topology-change instance
  <0-64>  Enable topology change information on the specified instance id.
switch(config)# spanning-tree trap topology-change instance 1
<cr>
switch(config)# spanning-tree trap errant-bpdu
<cr>
switch(config)# spanning-tree trap root-guard-inconsistency
<cr>
switch(config)# spanning-tree trap loop-guard-inconsistency
<cr>

```

Disabling the notifications for the SNMP traps:

```

switch(config)# no spanning-tree trap
  new-root  Disable notifications which are sent when a new root
is elected
  topology-change  Disable notifications which are sent when a topology
change occurs
  errant-bpdu  Disable notifications which are sent when an errant
bpdu is received
  root-guard-inconsistency  Disable notifications which are sent when root guard
inconsistency occurs
  loop-guard-inconsistency  Disable notifications which are sent when loop guard
inconsistency occurs
switch(config)# no spanning-tree trap new-root
<cr>
switch(config)# no spanning-tree trap topology-change
  instance  Disable topology change notification for the specified MST instance
switch(config)# no spanning-tree trap topology-change instance
  <0-64>  Disable topology change information on the specified instance id
switch(config)# no spanning-tree trap topology-change instance 1
<cr>
switch(config)# no spanning-tree trap errant-bpdu
<cr>
switch(config)# no spanning-tree trap root-guard-inconsistency
<cr>
switch(config)# no spanning-tree trap loop-guard-inconsistency
<cr>

```

Command History

Release	Modification
10.07 or earlier	--

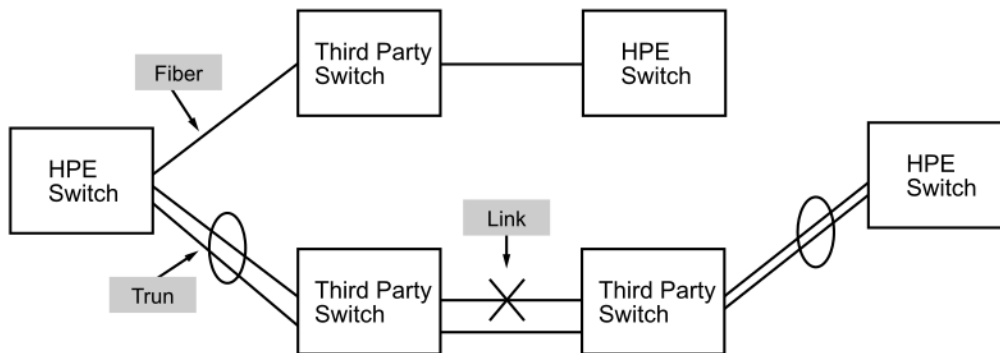
Command Information

Platforms	Command context	Authority
All platforms	<code>config</code>	Administrators or local user group members with execution rights for this command.

The Unidirectional Link Detection (UDLD) protocol enables detection of unidirectional behavior of layer 2 link. For UDLD to work, both connected devices must run the same UDLD protocol on the respective ports.

UDLD monitors the link between two network devices and blocks the ports on both ends of the link if the link fails. UDLD is particularly useful for detecting failures in fiber links and trunks.

In the following example each switch load balances traffic across two ports in a trunk group. Without the UDLD feature, a link failure on a link that is not directly attached to one of the HPE switches remains undetected. As a result, each switch continues to send traffic on the ports connected to the failed link. When UDLD is enabled on the trunk ports on each switch, the switches detect the failed link, block the ports connected to the failed link, and use the remaining ports in the trunk group to forward the traffic.



Similarly, UDLD is effective for monitoring fiber optic links that use two uni-direction fibers to transmit and receive packets. Without UDLD, if a fiber breaks in one direction, a fiber port may assume the link is still good (because the other direction is operating normally) and continue to send traffic on the connected ports. UDLD-enabled ports; however, will prevent traffic from being sent across a bad link by blocking the ports in the event that either the individual transmitter or receiver for that connection fails. Ports enabled for UDLD exchange health-check packets once every seven seconds (the link-keepalive interval). If a port does not receive a health-check packet from the port at the other end of the link within the keepalive interval, the port waits for four more intervals. If the port still does not receive a health-check packet after waiting for five intervals, the port concludes that the link has failed and blocks the UDLD-enabled port.

When a port is blocked by UDLD, the event is recorded in the switch log and other port blocking protocols, like spanning tree or meshing, will not use the bad link to load balance packets. The port will remain blocked until the link is unplugged, disabled, or fixed. The port can also be unblocked by disabling UDLD on the port.

Port blocking behavior is dependant on the UDLD mode in use. The previous paragraphs describe RFC5171 Aggressive mode. Other modes behave as follows:

- RFC 5171 normal: The port is not blocked but a notification is triggered.
- Aruba OS verify-then-forward: The links are considered blocked until bi-directionality is confirmed. After a link is considered bidirectional, if the retries are met and no packets are received, the link is marked as blocked.
- Aruba OS forward-then-verify: The links start up as unblocked. After a link is considered bidirectional, if the retries are met and no packets are received, the link is marked as blocked.

Configuring UDLD

Procedure

1. Enable UDLD on an interface with the command `udld`.
2. For most deployments, the default values for the following settings do not need to be changed. If your deployment requires different settings, change the default values with the indicated command:

UDLD setting	Default value	Command to change it
Packet transmission delay interval.	7000 ms	<code>udld interval</code>
Operating mode.	Interconnect with HPE PVOS/Brocade/Foundry switches in forward-then-verify mode.	<code>udld mode</code>
Retry count.	4	<code>udld retries</code>

3. Review UDLD configuration settings with the command `show udld`.

Example

This example creates the following configuration:

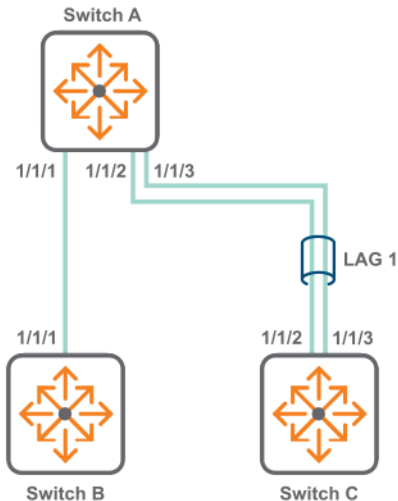
- Enables UDLD on interface **1/1/1**.
- Sets the UDLD mode to **rfc5171 aggressive**.
- Sets the UDLD interval to **1000**.
- Sets the UDLD retries to **3**.

```
switch(config)# interface 1/1/1
switch(config-if)# mode rfc5171 aggressive
switch(config-if)# interval 10000
switch(config-if)# retries 3
switch(config-if)# udld
switch(config-if)# quit
switch(config)# show udld interface 1/1/1
Interface 1/1/1
Config: enabled
State: active
Substate: bidirectional
Link: unblock
Version: rfc5171
Mode: aggressive
```

```
Interval: 10000 milliseconds
Retries: 3
Tx: 0 packets
Rx: 0 packets, 0 discarded packets, 0 dropped packets
Port transitions: 0
```

UDLD scenario

This scenario describes how to use UDLD on a single physical interface as well as a LAG.



Procedure

1. On switch A:
 - a. Configure the UDLD session between switch A and B.

```
switch# config
switch(config-if)# interface 1/1/1
switch(config-if)# udld
switch(config-if)# exit
switch(config)#
```

- b. Configure the UDLD session between switch A and C.

```
switch(config)# interface 1/1/2
switch(config-if)# no shutdown
switch(config-if)# no routing
switch(config-if)# lag 1
switch(config-if)# udld interval 400
switch(config-if)# udld mode aruba-os verify-then-forward
switch(config-if)# udld retries 5
switch(config)# exit
switch(config)# interface 1/1/3
switch(config-if)# no shutdown
switch(config-if)# no routing
switch(config-if)# lag 1
switch(config-if)# udld interval 400
switch(config-if)# udld mode aruba-os verify-then-forward
```

```
switch(config-if) # udld retries 5
```

2. On switch B, configure the UDLD session between switch B and A.

```
switch# config
switch(config-if) # interface 1/1/1
switch(config-if) # udld
switch(config-if) # exit
```

3. On switch C, configure the UDLD session between switch C and A.

```
switch# config
switch(config) # interface 1/1/2
switch(config-if) # no shutdown
switch(config-if) # no routing
switch(config-if) # lag 1
switch(config-if) # udld interval 400
switch(config-if) # udld mode aruba-os verify-then-forward
switch(config-if) # udld retries 5
switch(config) # exit
switch(config) # interface 1/1/3
switch(config-if) # no shutdown
switch(config-if) # no routing
switch(config-if) # lag 1
switch(config-if) # udld interval 400
switch(config-if) # udld mode aruba-os verify-then-forward
switch(config-if) # udld retries 5
```

4. On switch A, verify UDLD configuration by running the command `show udld`. (A packet must arrive on each switch for it to unblock the interface.)

```
switch# show udld
Abbreviations:
  VTF - Verify then forward      FTV - Forward then verify
  NOR - RFC 5171 normal          AGG - RFC 5171 aggressive

-----
Interface  UDLD      UDLD      UDLD      UDLD      Mode Interval
Retries Tx    Rx    Rx      Rx      Transitions
          Config  State  Substate Link
          Pkts  Pkts  Pkts disc. Pkts drop.
-----
1/1/1      enabled  active  Bidirectional  unblock  FTV  7000      4
0          0      0      0          0
1/1/2      enabled  active  Bidirectional  unblock  VTF  400       5
2          2      0      0          1
1/1/3      enabled  active  Bidirectional  unblock  VTF  400       5
2          2      0      0          1
```

UDLD commands

clear udld statistics

```
clear udld statistics [interface <INTERFACE-NAME>]
```

Description

Clears UDLD statistics for all interfaces or a specific interface.

Examples

Clearing all UDLD statistics on all interfaces:

```
switch# clear udld statistics
```

Clearing all UDLD statistics on interface 1/1/1:

```
switch# clear udld statistics interface 1/1/1
```

Command History

Release	Modification
10.07 or earlier	--

Command Information

Platforms	Command context	Authority
All platforms	Manager (#)	Administrators or local user group members with execution rights for this command.

show udld

```
show udld [interface <INTERFACE-NAME>] [vsx-peer]
```

Description

Displays UDLD information for all interfaces or for a specific interface.

Parameter	Description
interface <INTERFACE-NAME>	<p>Specifies the name of a logical interface on the switch, which can be:</p> <ul style="list-style-type: none">■ An Ethernet interface associated with a physical port. Use the format <code>member/slot/port</code> (for example, <code>1/3/1</code>).■ UDLD runs only on physical interfaces. LAGs, tunnels, and the like are not supported. However, UDLD can be configured individually on each port of a LAG or trunk group. Configuring UDLD on a trunk group primary port enables UDLD on that port only.
vsx-peer	<p>Shows the output from the VSX peer switch. If the switches do not have the VSX configuration or the ISL is down, the output from the VSX peer switch is not displayed. This parameter is available on switches that support VSX.</p>

Examples

Displaying all UDLD information:

```
switch# show udld
```

Abbreviations:

VTF - Verify-then-forward FTV - Forward-then-verify
NOR - RFC 5171 normal AGG - RFC 5171 aggressive

Interface	UDLD Config	UDLD State	UDLD Substate	UDLD Link	Mode	Interval
1/1/1	Disabled	Inactive	Undetermined	Unblock	FTV	8000
1/1/2	Enabled	Active	Bidirectional	Unblock	FTV	7000
1/1/3	Enabled	Active	Blocked	Block	FTV	7000
1/1/4	Enabled	Inactive	Uninitialized	Unblock	NOR	7000
1/1/5	Enabled	Active	ErrDisabled	Block	AGG	7000
1/1/6	Disabled	Active	Detection	Unblock	NOR	7000

Retries	Tx Pkts	Rx Pkts	Rx Pkts disc.	Rx Pkts drop.	Transitions
4	4	54	123	123	1
7	1234567	1548421	23214	1878981	3
4	3	77871	2157	81878	1
5	50	0	0	0	0
3	150	25	0	2	1
3	6	54	123	23	1

Displaying information for interface **1/1/1**:

```
switch# show udld interface 1/1/1
```

Interface 1/1/1
Config: Enabled
State: Active
Substate: Bidirectional
Link: Unblock
Version: Aruba OS
Mode: Forward then verify
Interval: 7000 milliseconds
Retries: 7
Tx: 1234567 packets
Rx: 1548421 packets, 23214 discarded packets, 1878981 dropped packets
Port transitions: 3

Displaying the UDLD enable interfaces information:

```
switch# show udld enabled
```

Abbreviations:

VTF - Verify-then-forward FTV - Forward-then-verify
NOR - RFC 5171 normal AGG - RFC 5171 aggressive

Interface	UDLD	UDLD	UDLD	UDLD	Mode	Interval	Retries	Tx
Rx	Rx	Rx	Transitions					
Pkts	Config	State	Substate	Link				
Pkts	Pkts	Pkts disc.	Pkts drop.					
2	Enabled	Active	Bidirectional	Unblock	FTV	7000	7	
1234567	1548421	23214	1878981	3				
3	Enabled	Active	Blocked	Block	FTV	7000	4	3
	77871	2157	81878	1				
4	Enabled	Inactive	Uninitialized	Unblock	NOR	7000	5	50
	0	0	0	0				
5	Enabled	Active	ErrDisabled	Block	AGG	7000	3	
150	25	0	2	1				

Command History

Release	Modification
10.07 or earlier	--

Command Information

Platforms	Command context	Authority
All platforms	Operator (>) or Manager (#)	Operators or Administrators or local user group members with execution rights for this command. Operators can execute this command from the operator context (>) only.

udld

```
udld [disable]
no udld [disable]
```

Description

Enables UDLD support on a physical interface. UDLD is disabled by default. UDLD is configured on a per-port basis and must be enabled at both ends of the link.

UDLD runs only on physical interfaces. LAGs, tunnels, and the like are not supported. However, UDLD can be configured individually on each port of a LAG or trunk group. Configuring UDLD on a trunk group's primary port enables UDLD on that port only.

The `no` form of this command disables UDLD support and resets all configuration values to their default settings.

Parameter	Description
<i>disable</i>	Disables UDLD on the interface but retains all UDLD configuration settings.

Examples

Enabling UDLD on interface **1/1/1**:

```
switch(config)# interface 1/1/1
switch(config-if)# udld
```

Disabling UDLD on interface **1/1/1**:

```
switch(config)# interface 1/1/1
switch(config-if)# no udld
```

Command History

Release	Modification
10.07 or earlier	--

Command Information

Platforms	Command context	Authority
All platforms	config-if	Administrators or local user group members with execution rights for this command.

udld interval

```
udld interval <TIME>
no udld interval [<TIME>]
```

Description

Sets the packet transmission interval.

The **no** form of this command sets the packet transmission interval to the default value of 7000 ms.

The allowed values vary depending on the operation mode.

The default interval is 7000 ms (7 seconds) for both ArubaOS-Switch and RFC5171 operation modes.

Values must be specified as multiples of 10 ms (7000 ms is allowed but 7005 ms is not a valid setting).



Sessions under 100ms total detection time are susceptible to increasing processing load on the system. It is advisable to experiment with values that provide adequate detection times and system/protocol stability. Aruba recommends additional testing prior to configuring these sessions on a production environment.

However, these settings are recommended for specific deployments only, such as using UDLD for Ethernet Ring Protection Switching (ERPS) link-failure detection. The minimum detection time appropriate for your environment depends on the specific device family and configuration on which the protocol and system load is running. Aruba recommends additional testing for these configurations. During testing, monitor for unexpected false positive detections (i.e., UDLD records a failure when there was not any) on the interfaces running UDLD. Such false positive failures are an indication that the interval configuration requires tuning and that the system load might not allow such configuration.



When configuring detection times under 100ms for LAG interfaces, consider adding the interface first to the LAG and then enabling UDLD in the interface, to avoid false positive link failure detections. Adding an interface to a LAG causes momentary control plane traffic interruption for up to 100ms, which UDLD detects as a link failure if the detection time is following the control traffic interruption interval.

Parameter	Description
<TIME>	Specifies the packet transmission interval. Range: 200 ms to 90000 ms (in increments of 10).

Examples

Setting the packet transmission interval to **1000** ms on interface **1/1/1**.

```
switch(config)# interface 1/1/1
switch(config-if)# udld interval 1000
```

Setting the packet transmission interval on interface **1/1/1** to the default value.

```
switch(config)# interface 1/1/1
switch(config-if)# no udld interval
```

Trying to set the packet interval to 1055 ms on interface 1 is rejected because the interval must be specified as a multiple of 10:

```
switch(config)# interface 1
switch(config-if)# udld interval 1055
Invalid interval. The interval value must be between 20ms and 90000ms and should
be
specified as a multiple of 10, for example: 20, 100, 3000 or 90000.
```

Trying to set the packet interval to less than 7000 ms on interface 1 is rejected if using the RFC5171 mode.

```
switch(config)# interface 1
switch(config-if)# udld mode rfc5171 normal
switch(config-if)# udld interval 1000
Invalid interval. The interval must be equal or greater than 7000ms.
```

Command History

Release	Modification
10.07 or earlier	--

Command Information

Platforms	Command context	Authority
All platforms	config-if	Administrators or local user group members with execution rights for this command.

udld mode

```
udld mode aruba-os {verify-then-forward | forward-then-verify}
udld mode rfc5171 <RFC5171-MODE>
no udld mode [[aruba-os [verify-then-forward | forward-then-verify]] | [rfc5171
[<RFC5171-MODE>]]]
```

Description

Sets the operating mode.

The `no` form of this command sets the operating mode to the default value of `aruba-os` and `forward-then-verify`.

Parameter	Description
<code>aruba-os {verify-then-forward forward-then-verify}</code>	Selects the ArubaOS mode to use. Use this mode when interconnecting with HPE PVOS/Brocade/Foundry switches.
<code>verify-then-forward</code>	In this mode: <ul style="list-style-type: none"> ■ Interfaces start as unblocked. ■ Once an interface is determined to be bidirectional, it is blocked if the retry limit is reached without receiving any UDLD packets. ■ Interfaces automatically unblock if a UDLD packet is received. ■ On failover, the UDLD state does not change if the (interval * retries) time is around 6 seconds.
<code>forward-then-verify</code>	In this mode: <ul style="list-style-type: none"> ■ Interfaces start as unblocked. ■ Interfaces transition to the unblocked state when receiving UDLD packets. ■ Once an interface is determined to be bidirectional, it is blocked if the retry limit is reached without receiving any UDLD packets. ■ Interfaces automatically unblock if a UDLD packet is received.
<code>rfc5171 <RFC5171-MODE></code>	Selects the RFC5171 mode to use. Use this mode when interconnecting with third-party switches.
<code>normal</code>	In this mode: <ul style="list-style-type: none"> ■ Interfaces start as unblocked. ■ Interfaces do not block when the retry limit is reached without receiving any UDLD packets (plus 8 extra packets sent to the peer). Instead, an event is generated. ■ Interfaces automatically unblock if a UDLD packet is received.
<code>aggressive</code>	In this mode: <ul style="list-style-type: none"> ■ Interfaces start as unblocked.

Parameter	Description
	<ul style="list-style-type: none"> Once an interface is determined to be bidirectional, an interface will block when the retry limit is reached without receiving any UDLD packets (plus 8 extra packets sent to the peer). Interfaces implement a limited/reduced errDisabled recovery mechanism. When the interface's state goes to errDisabled, a maximum of 3 attempts (5 minutes apart) are triggered to try and bring up the interface in case the remote endpoint is still sending UDLD packets. After these 3 retries, the interface will remain blocked even if UDLD packets are received. The only way to unblock the interface when this occurs is to disable (and optionally re-enable) UDLD on the interface. The retry limit is reset once the interface becomes unblocked.

Examples

Setting the operating mode to **aruba-os** and **forward-then-verify** on interface **1/1/1**:

```
switch(config)# interface 1/1/1
switch(config-if)# udld mode aruba-os forward-then-verify
```

Setting the operating mode to **rfc5171** and **aggressive** on interface **1/1/1**:

```
switch(config)# interface 1/1/1
switch(config-if)# udld mode rfc5171 aggressive
```

Setting the operating mode on interface **1/1/1** to the default value:

```
switch(config)# interface 1/1/1
switch(config-if)# no udld mode
```

Command History

Release	Modification
10.07 or earlier	--

Command Information

Platforms	Command context	Authority
All platforms	config-if	Administrators or local user group members with execution rights for this command.

udld retries

```
udld retries <COUNT>
no udld retries [<COUNT>]
```

Description

Sets the UDLD retry count.

The `no` form of this command sets the retry count to the default of 4.

Parameter	Description
<code><COUNT></code>	Specifies the UDLD retry count. Range: 3 to 10. Default: 4.

Examples

Setting the UDLD retry count to **5** on interface **1/1/1**:

```
switch(config)# interface 1/1/1
switch(config-if)# udld retries 5
```

Setting the UDLD retry count on interface **1/1/1** to the default value:

```
switch(config)# interface 1/1/1
switch(config-if)# no udld retries
```

Command History

Release	Modification
10.07 or earlier	--

Command Information

Platforms	Command context	Authority
All platforms	<code>config-if</code>	Administrators or local user group members with execution rights for this command.

Accessing Aruba Support

Aruba Support Services	https://www.arubanetworks.com/support-services/
AOS-CX Switch Software Documentation Portal	https://www.arubanetworks.com/techdocs/AOS-CX/help_portal/Content/home.htm
Aruba Support Portal	https://asp.arubanetworks.com/
North America telephone	1-800-943-4526 (US & Canada Toll-Free Number) +1-408-754-1200 (Primary - Toll Number) +1-650-385-6582 (Backup - Toll Number - Use only when all other numbers are not working)
International telephone	https://www.arubanetworks.com/support-services/contact-support/

Be sure to collect the following information before contacting Support:

- 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

Other useful sites

Other websites that can be used to find information:

Airheads social forums and Knowledge Base	https://community.arubanetworks.com/
AOS-CX Switch Software Documentation Portal	https://www.arubanetworks.com/techdocs/AOS-CX/help_portal/Content/home.htm
Aruba Hardware Documentation and Translations	https://www.arubanetworks.com/techdocs/hardware/DocumentationPortal/Content/home.htm

Portal	
Aruba software	https://asp.arubanetworks.com/downloads
Software licensing	https://lms.arubanetworks.com/
End-of-Life information	https://www.arubanetworks.com/support-services/end-of-life/
Aruba Developer Hub	https://developer.arubanetworks.com/

Accessing Updates

You can access updates from the Aruba Support Portal or the HPE My Networking Website.

Aruba Support Portal

<https://asp.arubanetworks.com/downloads>

If you are unable to find your product in the Aruba Support Portal, you may need to search My Networking, where older networking products can be found:

My Networking

<https://www.hpe.com/networking/support>

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:

<https://support.hpe.com/portal/site/hpsc/aae/home/>

Access to some updates might require product entitlement when accessed through the Hewlett Packard Enterprise Support Center. You must have an HP Passport set up with relevant entitlements.

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 subscribe to eNewsletters and alerts:

<https://asp.arubanetworks.com/notifications/subscriptions> (requires an active Aruba Support Portal (ASP) account to manage subscriptions). Security notices are viewable without an ASP account.

Warranty Information

To view warranty information for your product, go to <https://www.arubanetworks.com/support-services/product-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 <https://www.hpe.com/support/Safety-Compliance-EnterpriseProducts>

Additional regulatory information

Aruba is committed to providing our customers with information about the chemical substances in our products as needed to comply with legal requirements, environmental data (company programs, product recycling, energy efficiency), and safety information and compliance data, (RoHS and WEEE). For more information, see <https://www.arubanetworks.com/company/about-us/environmental-citizenship/>.

Documentation Feedback

Aruba 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-switching@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.

spanning-tree rpvst-mstp interconnect-vlan

```
spanning-tree rpvst-mstp-interconnect-vlan <VLAN-ID>
no spanning-tree rpvst-mstp-interconnect-vlan [<VLAN-ID>]
```

Description

Configures the VLAN that has to be used to interconnect RPVST and MSTP domains. VLAN 1 is used by default.

The `no` form of this command sets the VLAN configuration to the default of 1.

- It is required to create the interconnect VLAN and then configure RPVST spanning tree on it.
- The same interconnect VLAN must be kept on all the switches in the network.
- Adding or deleting the interconnect VLAN triggers a re-convergence in the network.
- Deleting a VLAN that is configured as the interconnect VLAN does not reset the value to the default.

Parameter	Description
<VLAN-ID>	Specifies the number of a VLAN.

Examples

This example configures VLAN 10 to used to interconnect RPVST and MSTP domains.

```
switch#(config)# spanning-tree rpvst-mstp-interconnect-vlan 10
```

Command History

Release	Modification
10.07 or earlier	--

Command Information

Platforms	Command context	Authority
All platforms	<code>config</code>	Administrators or local user group members with execution rights for this command.