CONTENTS

CHAPTER 1
New and Changed Information 1

CHAPTER 2
Overview 3
Layer 2 Ethernet Switching Overview 3
VLANs 3
Private VLANs 4
Spanning Tree 4
STP Overview 4
Rapid PVST+ 5
MST 5
STP Extensions 5

CHAPTER 3
Configuring VLANs 7
Information About VLANs 7
Understanding VLANs 7
VLAN Ranges 8
Creating, Deleting, and Modifying VLANs 9
About the VLAN Trunking Protocol 10
Guidelines and Limitations for VTP 10
Configuring a VLAN 11
Creating and Deleting a VLAN 11
Configuring a VLAN 12
Adding Ports to a VLAN 13
Triggering the VLAN Membership Consistency Checker 13
Configuring a VLAN as a Routed SVI 14
Configuring a VLAN as a Management SVI 15
Configuring VTP 15
Contents

CHAPTER 4

Configuring Private VLANs 19

Information About Private VLANs 19

Primary and Secondary VLANs in Private VLANs 20

Private VLAN Ports 20

Primary, Isolated, and Community Private VLANs 21

Associating Primary and Secondary VLANs 22

Broadcast Traffic in Private VLANs 23

Private VLAN Port Isolation 23

Guidelines and Limitations for Private VLANs 23

Configuring a Private VLAN 24

Enabling Private VLANs 24

Configuring a VLAN as a Private VLAN 24

Associating Secondary VLANs with a Primary Private VLAN 25

Configuring an Interface as a Private VLAN Host Port 26

Configuring an Interface as a Private VLAN Promiscuous Port 27

Configuring a Layer 2 Interface as a Private VLAN Isolated Trunk Port 28

Configuring a Layer 2 Interface as a Private VLAN Promiscuous Trunk Port 33

Configuring VLAN Mapping for Selective Q-in-Q on a Private VLAN Secondary Trunk Port 35

Verifying the Private VLAN Configuration 36

CHAPTER 5

Configuring Access and Trunk Interfaces 39

Information About Access and Trunk Interfaces 39

Understanding Access and Trunk Interfaces 39

Understanding IEEE 802.1Q Encapsulation 40

Understanding Access VLANs 41

Understanding the Native VLAN ID for Trunk Ports 42

Understanding Allowed VLANs 42

Understanding Native 802.1Q VLANs 42

Configuring Access and Trunk Interfaces 43

Configuring a LAN Interface as an Ethernet Access Port 43

Configuring Access Host Ports 44

Configuring Trunk Ports 44

Verifying the VLAN Configuration 17
Configuring the Native VLAN for 802.1Q Trunking Ports 45
Configuring the Allowed VLANs for Trunking Ports 45
Configuring Native 802.1Q VLANs 46
Verifying the Interface Configuration 47

CHAPTER 6

Configuring Switching Modes 49
Information About Switching Modes 49
Guidelines and Limitations for Switching Modes 50
Licensing Requirements for Switching Modes 51
Default Settings for Switching Modes 51
Configuring Switching Modes 51
  Enabling Store-and-Forward Switching 51
  Reenabling Cut-Through Switching 52

CHAPTER 7

Configuring Rapid PVST+ 53
Information About Rapid PVST+ 53
Understanding STP 53
  STP Overview 53
  Understanding How a Topology is Created 54
Understanding the Bridge ID 54
  Bridge Priority Value 54
  Extended System ID 55
  STP MAC Address Allocation 55
Understanding BPDUs 56
Election of the Root Bridge 57
Creating the Spanning Tree Topology 57
Understanding Rapid PVST+ 58
  Rapid PVST+ Overview 58
  Rapid PVST+ BPDUs 59
Proposal and Agreement Handshake 60
Protocol Timers 61
Port Roles 61
Port States 62
  Rapid PVST+ Port State Overview 62
  Blocking State 63
Learning State 63
Forwarding State 63
Disabled State 64
Summary of Port States 64
Synchronization of Port Roles 64
Processing Superior BPDU Information 65
Processing Inferior BPDU Information 66
Spanning-Tree Dispute Mechanism 66
Port Cost 66
Port Priority 67
Rapid PVST+ and IEEE 802.1Q Trunks 67
Rapid PVST+ Interoperation with Legacy 802.1D STP 67
Rapid PVST+ Interoperation with 802.1s MST 68
Configuring Rapid PVST+ 68
Enabling Rapid PVST+ 68
Enabling Rapid PVST+ per VLAN 69
Configuring the Root Bridge ID 70
Configuring a Secondary Root Bridge 71
Configuring the Rapid PVST+ Port Priority 72
Configuring the Rapid PVST+ Path-Cost Method and Port Cost 73
Configuring the Rapid PVST+ Bridge Priority of a VLAN 73
Configuring the Rapid PVST+ Hello Time for a VLAN 74
Configuring the Rapid PVST+ Forward Delay Time for a VLAN 75
Configuring the Rapid PVST+ Maximum Age Time for a VLAN 75
Specifying the Link Type 75
Restarting the Protocol 76
Verifying the Rapid PVST+ Configuration 76
Triggering the VLAN STP State Consistency Checker 77

CHAPTER 8

Configuring Multiple Spanning Tree 79
Information About MST 79
MST Overview 79
MST Regions 80
MST BPDUs 80
MST Configuration Information 81
IST, CIST, and CST  81
  IST, CIST, and CST Overview  81
  Spanning Tree Operation Within an MST Region  82
  Spanning Tree Operations Between MST Regions  82
  MST Terminology  83
Hop Count  84
Boundary Ports  84
Spanning-Tree Dispute Mechanism  85
Port Cost and Port Priority  86
Interoperability with IEEE 802.1D  86
Interoperability with Rapid PVST+: Understanding PVST Simulation  87
Configuring MST  87
  MST Configuration Guidelines  87
  Enabling MST  87
  Entering MST Configuration Mode  88
  Specifying the MST Name  89
  Specifying the MST Configuration Revision Number  90
  Specifying the Configuration on an MST Region  90
  Mapping and Unmapping VLANs to MST Instances  92
  Mapping Secondary VLANs to Same MSTI as Primary VLANs for Private VLANs  92
  Configuring the Root Bridge  93
  Configuring a Secondary Root Bridge  94
  Configuring the Port Priority  95
  Configuring the Port Cost  96
  Configuring the Switch Priority  96
  Configuring the Hello Time  97
  Configuring the Forwarding-Delay Time  98
  Configuring the Maximum-Aging Time  98
  Configuring the Maximum-Hop Count  99
  Configuring PVST Simulation Globally  99
  Configuring PVST Simulation Per Port  100
  Specifying the Link Type  101
  Restarting the Protocol  101
  Verifying the MST Configuration  102
CHAPTER 9 Configuring STP Extensions 103

Information About STP Extensions 103

Overview 103

Understanding STP Port Types 103

Spanning Tree Edge Ports 104

Spanning Tree Network Ports 104

Spanning Tree Normal Ports 104

Understanding Bridge Assurance 104

Understanding BPDU Guard 105

Understanding BPDU Filtering 105

Understanding Loop Guard 106

Understanding Root Guard 107

Configuring STP Extensions 107

STP Extensions Configuration Guidelines 107

Configuring Spanning Tree Port Types Globally 108

Configuring Spanning Tree Edge Ports on Specified Interfaces 109

Configuring Spanning Tree Network Ports on Specified Interfaces 110

Enabling BPDU Guard Globally 111

Enabling BPDU Guard on Specified Interfaces 111

Enabling BPDU Filtering Globally 112

Enabling BPDU Filtering on Specified Interfaces 113

Enabling Loop Guard Globally 114

Enabling Loop Guard or Root Guard on Specified Interfaces 115

Verifying the STP Extension Configuration 116

Troubleshooting Loop Detection Error Message 116

Generating Syslog Error Messages 117

CHAPTER 10 Configuring LLDP 119

Information About Global LLDP Commands 119

Configuring LLDP 120

Information About LLDP Management TLV IP Addresses 121

Configuring LLDP Management TLV IP Addresses on an Interface 123

Configuring Interface LLDP 124

MIBs for LLDP 125
New and Changed Information

This chapter contains the following sections:

- New and Changed Information, page 1

### Table 1: New and Changed Features

<table>
<thead>
<tr>
<th>Feature</th>
<th>Description</th>
<th>Added or Changed in Release</th>
<th>Where Documented</th>
</tr>
</thead>
<tbody>
<tr>
<td>LLDP TLV Management</td>
<td>You can now send two TLVs. One for IPv4 and the other for IPv6.</td>
<td>6.0(2)U4(1)</td>
<td>Information About LLDP Management TLV IP Addresses, on page 121</td>
</tr>
</tbody>
</table>
| LLDP TLV Management Assignment   | You can now explicitly specify the management IP address to be sent in the LLDP management TLV.  
<pre><code>                                |                             | 6.0(2)U3(1)                 | Configuring LLDP Management TLV IP Addresses on an Interface, on page 123       |
</code></pre>
<p>|                                  | This address can be one of the following:                                   |                             |                                                                                  |
|                                  |     • IP address of a port                                                 |                             |                                                                                  |
|                                  |     • IP address of a VLAN (SVI)                                            |                             |                                                                                  |
| MAC Move Threshold Action        | You can now configure the action of bringing down the port with the lower interface index when a MAC address move loop is detected between two ports. | 6.0(2)U3(1)                 | Configuring MAC Move Loop Detection, on page 130                                  |</p>
<table>
<thead>
<tr>
<th>Feature</th>
<th>Description</th>
<th>Added or Changed in Release</th>
<th>Where Documented</th>
</tr>
</thead>
</table>
| Storm Control Enhancements | You can configure traffic storm control to perform the following optional corrective actions when traffic exceeds the configured level:  
  • Shut down  
  • Trap | 6.0(2)U3(1)                  | Configuring Traffic Storm Control, on page 146 |
| Consistency Checkers | The following consistency checkers were introduced to check for consistency and to display the results:  
  • Layer 2 Consistency Checker  
  • VLAN Membership Consistency Checker  
  • VLAN STP State Consistency Checker | 6.0(2)U2(1)                  | • Triggering the Layer 2 Consistency Checker, on page 132  
 • Triggering the VLAN Membership Consistency Checker, on page 13  
 • Triggering the VLAN STP State Consistency Checker, on page 77 |
| vPC                     | Added for vPC                                                               | 6.0(2)U1(1)                  | Configuring IGMP Snooping Parameters, on page 138                |
Overview

This chapter contains the following sections:

- Layer 2 Ethernet Switching Overview, page 3
- VLANs, page 3
- Private VLANs, page 4
- Spanning Tree, page 4

Layer 2 Ethernet Switching Overview

The device supports simultaneous, parallel connections between Layer 2 Ethernet segments. Switched connections between Ethernet segments last only for the duration of the packet. New connections can be made between different segments for the next packet.

The device assigns a domain (for example, a server) to each device to solve traffic congestion caused by high-bandwidth devices and large number of users.

Because collisions cause significant congestion in Ethernet networks, an effective solution is full-duplex communication. Typically, 10/100-Mbps Ethernet operates in half-duplex mode, which means that stations can either receive or transmit. In full-duplex mode, which is configurable on these interfaces, two stations can transmit and receive at the same time. When packets can flow in both directions simultaneously, the effective Ethernet bandwidth doubles. 1/10-Gigabit Ethernet operates in full-duplex only.

VLANs

A VLAN is a switched network that is logically segmented by function, project team, or application, without regard to the physical locations of the users. VLANs have the same attributes as physical LANs, but you can group end stations even if they are not physically located on the same LAN segment.

Any switch port can belong to a VLAN, and unicast, broadcast, and multicast packets are forwarded and flooded only to end stations in that VLAN. Each VLAN is considered as a logical network, and packets destined for stations that do not belong to the VLAN must be forwarded through a bridge or a router.

All ports are assigned to the default VLAN (VLAN1) when the device comes up.
The devices support 4094 VLANs in accordance with the IEEE 802.1Q standard. These VLANs are organized into several ranges, and you use each range slightly differently. Some of these VLANs are reserved for internal use by the device and are not available for configuration.

---

**Note**

Inter-Switch Link (ISL) trunking is not supported.

---

### Private VLANs

Private VLANs provide traffic separation and security at the Layer 2 level.

A private VLAN is one or more pairs of a primary VLAN and a secondary VLAN, all with the same primary VLAN. The two types of secondary VLANs are isolated and community VLANs. Hosts on isolated VLANs communicate only with hosts in the primary VLAN. Hosts in a community VLAN can communicate only among themselves and with hosts in the primary VLAN but not with hosts in isolated VLANs or in other community VLANs.

Regardless of the combination of isolated and community secondary VLANs, all interfaces within the primary VLAN comprise one Layer 2 domain, and therefore, require only one IP subnet.

### Spanning Tree

This section discusses the implementation of the Spanning Tree Protocol (STP).

#### STP Overview

STP provides a loop-free network at the Layer 2 level. Layer 2 LAN ports send and receive STP frames, which are called Bridge Protocol Data Units (BPDUs), at regular intervals. Network devices do not forward these frames but use the frames to construct a loop-free path.

802.1D is the original standard for STP, and many improvements have enhanced the basic loop-free STP. You can create a separate loop-free path for each VLAN, which is named Per VLAN Spanning Tree (PVST+). Additionally, the entire standard was reworked to make the loop-free convergence process faster to keep up with the faster equipment. This STP standard with faster convergence is the 802.1w standard, which is known as Rapid Spanning Tree (RSTP).

Finally, the 802.1s standard, Multiple Spanning Trees (MST), allows you to map multiple VLANs into a single spanning tree instance. Each instance runs an independent spanning tree topology.

Although the software can interoperate with legacy 802.1D systems, the device runs Rapid PVST+ and MST. You can use either Rapid PVST+ or MST in a given VDC; you cannot mix both in one VDC. Rapid PVST+ is the default STP protocol.

---

**Note**

Cisco NX-OS uses the extended system ID and MAC address reduction; you cannot disable these features.

In addition, Cisco has created some proprietary features to enhance the spanning tree activities.
Rapid PVST+

Rapid PVST+ is the default spanning tree mode for the software and is enabled by default on the default VLAN and all newly created VLANs.

A single instance, or topology, of RSTP runs on each configured VLAN, and each Rapid PVST+ instance on a VLAN has a single root device. You can enable and disable STP on a per-VLAN basis when you are running Rapid PVST+.

MST

The software also supports MST. The multiple independent spanning tree topologies enabled by MST provide multiple forwarding paths for data traffic, enable load balancing, and reduce the number of STP instances required to support a large number of VLANs.

MST incorporates RSTP, so it also allows rapid convergence. MST improves the fault tolerance of the network because a failure in one instance (forwarding path) does not affect other instances (forwarding paths).

Note

Changing the spanning tree mode disrupts the traffic because all spanning tree instances are stopped for the previous mode and started for the new mode.

STP Extensions

The software supports the following Cisco proprietary features:

- Spanning tree port types—The default spanning tree port type is normal. You can configure interfaces connected to Layer 2 hosts as edge ports and interfaces connected to Layer 2 switches or bridges as network ports.

- Bridge Assurance—Once you configure a port as a network port, Bridge Assurance sends BPDUs on all ports and moves a port into the blocking state if it no longer receives BPDUs. This enhancement is available only when you are running Rapid PVST+ or MST.

- BPDU Guard—BPDU Guard shuts down the port if that port receives a BPDU.

- BPDU Filter—BPDU Filter suppresses sending and receiving BPDUs on the port.

- Loop Guard—Loop guard prevents the occurrence of loop bridging because of unidirectional link failure in a point-to-point link.

- Root Guard—Root guard prevents a port from becoming a root port or a blocked port. If you configure a port with root guard then the port receives a superior BPDU and it immediately goes to root-inconsistent (blocked) state.
CHAPTER 3

Configuring VLANs

This chapter contains the following sections:

- Information About VLANs, page 7
- Configuring a VLAN, page 11

Information About VLANs

Understanding VLANs

A VLAN is a group of end stations in a switched network that is logically segmented by function or application, without regard to the physical locations of the users. VLANs have the same attributes as physical LANs, but you can group end stations even if they are not physically located on the same LAN segment.

Any switch port can belong to a VLAN, and unicast, broadcast, and multicast packets are forwarded and flooded only to end stations in that VLAN. Each VLAN is considered as a logical network, and packets destined for stations that do not belong to the VLAN must be forwarded through a router. The following figure shows VLANs as logical networks. The stations in the engineering department are assigned to one VLAN,
the stations in the marketing department are assigned to another VLAN, and the stations in the accounting department are assigned to another VLAN.

**Figure 1: VLANs as Logically Defined Networks**

VLANs are usually associated with IP subnetworks. For example, all the end stations in a particular IP subnet belong to the same VLAN. To communicate between VLANs, you must route the traffic.

By default, a newly created VLAN is operational; that is, the newly created VLAN is in the no shutdown condition. Additionally, you can configure VLANs to be in the active state, which is passing traffic, or the suspended state, in which the VLANs are not passing packets. By default, the VLANs are in the active state and pass traffic.

**VLAN Ranges**

**Note**

The extended system ID is always automatically enabled in Cisco NX-OS devices.

The device supports up to 4094 VLANs in accordance with the IEEE 802.1Q standard. The software organizes these VLANs into ranges, and you use each range slightly differently.

For information about configuration limits, see the configuration limits documentation for your switch.

This table describes the VLAN ranges.
<table>
<thead>
<tr>
<th>VLANs Numbers</th>
<th>Range</th>
<th>Usage</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Normal</td>
<td>Cisco default. You can use this VLAN, but you cannot modify or delete it.</td>
</tr>
<tr>
<td>2 to 1005</td>
<td>Normal</td>
<td>You can create, use, modify, and delete these VLANs.</td>
</tr>
</tbody>
</table>
| 1006 to 3967 and 4048 to 4093 | Extended | You can create, name, and use these VLANs. You cannot change the following parameters:  
  • The state is always active.  
  • The VLAN is always enabled. You cannot shut down these VLANs. |
| 3968 to 4047 and 4094 | Internally allocated | These 80 VLANs and VLAN 4094 are allocated for internal device use. You cannot create, delete, or modify any VLANs within the block reserved for internal use. |

The software allocates a group of VLAN numbers for features such as multicast and diagnostics that need to use internal VLANs for their operation. You cannot use, modify, or delete any of the VLANs in the reserved group. You can display the VLANs that are allocated internally and their associated use.

**Creating, Deleting, and Modifying VLANs**

VLANs are numbered from 1 to 4094. All configured ports belong to the default VLAN when you first bring up the switch. The default VLAN (VLAN1) uses only default values. You cannot create, delete, or suspend activity in the default VLAN.

You create a VLAN by assigning a number to it. You can delete VLANs as well as move them from the active operational state to the suspended operational state. If you attempt to create a VLAN with an existing VLAN ID, the switch goes into the VLAN submode but does not create the same VLAN again.

Newly created VLANs remain unused until ports are assigned to the specific VLAN. All the ports are assigned to VLAN1 by default.

Depending on the range of the VLAN, you can configure the following parameters for VLANs (except the default VLAN):

- VLAN name
- Shutdown or not shutdown
When you delete a specified VLAN, the ports associated to that VLAN are shut down and no traffic flows. However, the system retains all the VLAN-to-port mapping for that VLAN, and when you reenable, or recreate, the specified VLAN, the system automatically reinstates all the original ports to that VLAN.

---

**Note**

Commands entered in the VLAN configuration submode are immediately executed.

VLANs 3968 to 4047 and 4094 are reserved for internal use; these VLANs cannot be changed or used.

---

## About the VLAN Trunking Protocol

VTP is a distributed VLAN database management protocol that synchronizes the VTP VLAN database across domains. A VTP domain includes one or more network switches that share the same VTP domain name and that are connected with trunk interfaces. Each device can be in one VTP domain, Layer 2 trunk interfaces, and Layer 2 port channels.

### Guidelines and Limitations for VTP

VTP has the following configuration guidelines and limitations:

- VLAN 1 is required on all trunk ports used for switch interconnects if VTP is supported in the network. Disabling VLAN 1 from any of these ports prevents VTP from functioning properly.
- If you enable VTP, you must configure either version 1 or version 2.
- The `show running-configuration` command does not show VLAN or VTP configuration information for VLANs 1 to 1000.
- VTP pruning is not supported.
- You must enter the `copy running-config startup-config` command followed by a reload after changing a reserved VLAN range. For example:

  ```
  switch(config)# system vlan 2000 reserve
  This will delete all configs on vlans 2000-2127. Continue anyway? (y/n) [no] y
  ```

  After the switch reload, VLANs 2000 to 2127 are reserved for internal use, which requires that you enter the `copy running-config startup-config` command before the switch reload. Creating VLANs within this range is not allowed.
- SNMP can perform GET and SET operations on the CISCO-VTP-MIB objects.
- VTP server mode and VTP client mode are not supported. The only supported mode is transparent mode, which is the default mode.
Configuring a VLAN

Creating and Deleting a VLAN

You can create or delete all VLANs except the default VLAN and those VLANs that are internally allocated for use by the switch. Once a VLAN is created, it is automatically in the active state.

When you delete a VLAN, ports associated to that VLAN shut down. The traffic does not flow and the packets are dropped.

Procedure

<table>
<thead>
<tr>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Step 1</strong> switch# configure terminal</td>
<td>Enters global configuration mode.</td>
</tr>
<tr>
<td><strong>Step 2</strong> switch(config)# vlan {vlan-id</td>
<td>vlan-range}</td>
</tr>
<tr>
<td><strong>Step 3</strong> switch(config-vlan)# no vlan {vlan-id</td>
<td>vlan-range}</td>
</tr>
</tbody>
</table>

This example shows how to create a range of VLANs from 15 to 20:

```
switch# configure terminal
switch(config)# vlan 15-20
```

You can also create and delete VLANs in the VLAN configuration submode.
Configuring a VLAN

To configure or modify the VLAN for the following parameters, you must be in the VLAN configuration submode:

- Name
- Shut down

Note

You cannot create, delete, or modify the default VLAN or the internally allocated VLANs. Additionally, some of these parameters cannot be modified on some VLANs.

Procedure

<table>
<thead>
<tr>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Step 1</strong></td>
<td>switch# configure terminal</td>
</tr>
<tr>
<td><strong>Step 2</strong></td>
<td>switch(config)# vlan {vlan-id</td>
</tr>
<tr>
<td><strong>Step 3</strong></td>
<td>switch(config-vlan)# name vlan-name</td>
</tr>
<tr>
<td><strong>Step 4</strong></td>
<td>switch(config-vlan)# state {active</td>
</tr>
<tr>
<td><strong>Step 5</strong></td>
<td>switch(config-vlan)# no shutdown</td>
</tr>
</tbody>
</table>

This example shows how to configure optional parameters for VLAN 5:

```
switch# configure terminal
switch(config)# vlan 5
switch(config-vlan)# name accounting
switch(config-vlan)# state active
switch(config-vlan)# no shutdown
```
Adding Ports to a VLAN

After you have completed the configuration of a VLAN, assign ports to it.

**Procedure**

<table>
<thead>
<tr>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Step 1</strong></td>
<td>switch# configure terminal</td>
</tr>
<tr>
<td></td>
<td>Enters global configuration mode.</td>
</tr>
<tr>
<td><strong>Step 2</strong></td>
<td>switch(config)# interface {ethernet slot/port</td>
</tr>
<tr>
<td></td>
<td>Specifies the interface to configure, and enters the interface configuration mode. The interface can be a physical Ethernet port or an EtherChannel.</td>
</tr>
<tr>
<td><strong>Step 3</strong></td>
<td>switch(config-if)# switchport access vlan vlan-id</td>
</tr>
<tr>
<td></td>
<td>Sets the access mode of the interface to the specified VLAN.</td>
</tr>
</tbody>
</table>

This example shows how to configure an Ethernet interface to join VLAN 5:

```plaintext
switch# configure terminal
switch(config)# interface ethernet 1/13
switch(config-if)# switchport access vlan 5
```

**Triggering the VLAN Membership Consistency Checker**

You can manually trigger the VLAN Membership consistency checker to compare the hardware and software configuration of all ports in a VLAN and display the results. To manually trigger the VLAN Membership consistency checker and display the results, use the following command in any mode:

**Procedure**

<table>
<thead>
<tr>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Step 1</strong></td>
<td>switch# show consistency-checker membership vlan vlan-id</td>
</tr>
<tr>
<td></td>
<td>Starts a VLAN Membership consistency check on the member ports of vlan-id and displays the results.</td>
</tr>
</tbody>
</table>

This example shows how to trigger a VLAN Membership consistency check and display the results:

```plaintext
switch# show consistency-checker membership vlan 2
Checks: Port membership of Vlan Vlan 2 : Consistency Check: PASSED Vlan:2, Hardware state consistent for:
  Ethernet1/18
  Ethernet1/20
  Ethernet1/29
  Ethernet1/30
  Ethernet1/31
  Ethernet1/32
  Ethernet1/33
  Ethernet1/34
```
Configuring a VLAN as a Routed SVI

You can configure a VLAN to be a routed switch virtual interface (SVI).

**Before You Begin**

- Install the Layer 3 license. For more information, see *License and Copyright Information for Cisco NX-OS Software* available at the following URL: http://www.cisco.com/en/US/docs/switches/datacenter/sw/4_0/nx-os/license_agreement/nx-ossw_lisns.html.
- Make sure you understand the guidelines and limitations of this feature.

**Procedure**

<table>
<thead>
<tr>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Step 1</strong></td>
<td>switch# configure terminal</td>
</tr>
<tr>
<td><strong>Step 2</strong></td>
<td>switch(config)# feature interface-vlan</td>
</tr>
<tr>
<td><strong>Step 3</strong></td>
<td>switch(config)# interface-vlan vlan-id</td>
</tr>
<tr>
<td><strong>Step 4</strong></td>
<td>switch(config-if)# copy running-config startup-config</td>
</tr>
</tbody>
</table>

This example shows how to configure a VLAN as a routed SVI:

```
switch# configure terminal
switch(config)# feature interface-vlan
switch(config)# interface vlan 5
switch(config-if)# copy running-config startup-config
switch(config-if)#
```

This example shows how to remove the routed SVI function from a VLAN:

```
switch# configure terminal
switch(config)# no interface vlan 5
switch(config-if)# copy running-config startup-config
switch(config-if)#
```
What to Do Next
You can configure routing protocols on this interface.

Configuring a VLAN as a Management SVI
You can configure a VLAN to be a management switch virtual interface (SVI).

Procedure

<table>
<thead>
<tr>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td>Step 1</td>
<td>Enters global configuration mode.</td>
</tr>
<tr>
<td>switch# configure terminal</td>
<td></td>
</tr>
<tr>
<td>Step 2</td>
<td>Enables the creation of SVIs.</td>
</tr>
<tr>
<td>switch(config)# feature interface-vlan</td>
<td></td>
</tr>
<tr>
<td>Step 3</td>
<td>Creates a VLAN interface (SVI) and configures the SVI to be used for in-band management.</td>
</tr>
<tr>
<td>switch(config)# interface-vlan vlan-id management</td>
<td></td>
</tr>
<tr>
<td>Step 4</td>
<td>Copies the running configuration to the startup configuration.</td>
</tr>
<tr>
<td>switch(config-if)# copy running-config startup-config</td>
<td></td>
</tr>
</tbody>
</table>

This example shows how to configure a VLAN as a management SVI:
```
switch# configure terminal
switch(config)# feature interface-vlan
switch(config)# interface vlan 5
switch(config-if)# management
switch(config-if)# copy running-config startup-config
```

This example shows how to remove the management function from an SVI:
```
switch# configure terminal
switch(config)# interface vlan 5
switch(config-if)# no management
switch(config-if)# copy running-config startup-config
```

Configuring VTP
You can enable and configure VTP. If you enable VTP, you must configure either version 1 or version 2. If you are using VTP in a Token Ring environment, you must use version 2.

Procedure

<table>
<thead>
<tr>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td>Step 1</td>
<td>Enters global configuration mode.</td>
</tr>
<tr>
<td>switch# configure terminal</td>
<td></td>
</tr>
<tr>
<td>Step 2</td>
<td>Enables VTP on the device. The default is disabled.</td>
</tr>
<tr>
<td>switch(config)# feature vtp</td>
<td></td>
</tr>
<tr>
<td>Step 3</td>
<td>switch(config)# vtp domain domain-name</td>
</tr>
<tr>
<td>-----------</td>
<td>----------------------------------------</td>
</tr>
<tr>
<td>Step 4</td>
<td>switch(config)# vtp version {1</td>
</tr>
<tr>
<td>Step 5</td>
<td>switch(config)# vtp file file-name</td>
</tr>
<tr>
<td>Step 6</td>
<td>switch(config)# vtp password password-value</td>
</tr>
<tr>
<td>Step 7</td>
<td>switch(config)# exit</td>
</tr>
<tr>
<td>Step 8</td>
<td>switch# show vtp status</td>
</tr>
<tr>
<td>Step 9</td>
<td>switch# show vtp counters</td>
</tr>
<tr>
<td>Step 10</td>
<td>switch# show vtp interface</td>
</tr>
<tr>
<td>Step 11</td>
<td>switch# show vtp password</td>
</tr>
<tr>
<td>Step 12</td>
<td>switch# copy running-config startup-config</td>
</tr>
</tbody>
</table>

This example shows how to configure VTP for the device:

```
switch# configure terminal
switch(config)# feature vtp
switch(config)# vtp domain accounting
switch(config)# vtp version 2
switch(config)# exit
switch#
```

This example shows the VTP status and that the switch is capable of supporting Version 2 and that the switch is running Version 1:

```
switch(config)# show vtp status
VTP Status Information
-----------------------
VTP Version : 2 (capable)
Configuration Revision : 0
Maximum VLANs supported locally : 1005
Number of existing VLANs : 502
VTP Operating Mode : Transparent
VTP Domain Name : 
VTP Pruning Mode : Disabled (Operationally Disabled)
VTP V2 Mode : Disabled
```
Verifying the VLAN Configuration

Use one of the following commands to verify the configuration:

<table>
<thead>
<tr>
<th>Command</th>
<th>Purpose</th>
</tr>
</thead>
</table>
| switch# show running-config vlan [vlan_id | vlan_range] | Displays VLAN information.  
**Note** Starting with Release 7.0(3)I2(1), the output of show running-config vlan displays the VLAN output multiple times. There is no functional impact of this behavior change. |

<table>
<thead>
<tr>
<th>Command</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td>switch# show vlan [brief</td>
<td>id [vlan_id</td>
</tr>
</tbody>
</table>
Verifying the VLAN Configuration
Chapter 4
Configuring Private VLANs

This chapter contains the following sections:

- Information About Private VLANs, page 19
- Guidelines and Limitations for Private VLANs, page 23
- Configuring a Private VLAN, page 24
- Configuring VLAN Mapping for Selective Q-in-Q on a Private VLAN Secondary Trunk Port, page 35
- Verifying the Private VLAN Configuration, page 36

Information About Private VLANs

A private VLAN (PVLAN) partitions the Ethernet broadcast domain of a VLAN into subdomains, allowing you to isolate the ports on the switch from each other. A subdomain consists of a primary VLAN and one or more secondary VLANs (see the following figure). All VLANs in a PVLAN domain share the same primary VLAN. The secondary VLAN ID differentiates one subdomain from another. The secondary VLANs can either be isolated VLANs or community VLANs. A host on an isolated VLAN can communicate only with
the associated promiscuous port in its primary VLAN. Hosts on community VLANs can communicate among themselves and with their associated promiscuous port but not with ports in other community VLANs.

Figure 2: Private VLAN Domain

Note
You must first create the VLAN before you can convert it to a PVLAN, either primary or secondary.

Primary and Secondary VLANs in Private VLANs

A private VLAN domain has only one primary VLAN. Each port in a private VLAN domain is a member of the primary VLAN; the primary VLAN is the entire private VLAN domain.

Secondary VLANs provide isolation between ports within the same private VLAN domain. The following two types are secondary VLANs within a primary VLAN:

- Isolated VLANs—Ports within an isolated VLAN cannot communicate directly with each other at the Layer 2 level.
- Community VLANs—Ports within a community VLAN can communicate with each other but cannot communicate with ports in other community VLANs or in any isolated VLANs at the Layer 2 level.

Private VLAN Ports

The three types of PVLAN ports are as follows:
• Promiscuous port—A promiscuous port belongs to the primary VLAN. The promiscuous port can communicate with all interfaces, including the community and isolated host ports, that belong to those secondary VLANs associated to the promiscuous port and associated with the primary VLAN. You can have several promiscuous ports in a primary VLAN. Each promiscuous port can have several secondary VLANs or no secondary VLANs that are associated to that port. You can associate a secondary VLAN to more than one promiscuous port, as long as the promiscuous port and secondary VLANs are within the same primary VLAN. You can also have secondary VLANs that are not associated to any promiscuous port.

A promiscuous port can be configured as an access port.

• Isolated port—An isolated port is a host port that belongs to an isolated secondary VLAN. This port has complete isolation from other ports within the same PVLAN domain, except that it can communicate with associated promiscuous ports. PVLANs block all traffic to isolated ports except traffic from promiscuous ports. Traffic received from an isolated port is forwarded only to promiscuous ports. You can have more than one isolated port in a specified isolated VLAN. Each port is completely isolated from all other ports in the isolated VLAN.

An isolated port can be configured an access port.

• Community port—A community port is a host port that belongs to a community secondary VLAN. Community ports communicate with other ports in the same community VLAN and with associated promiscuous ports. These interfaces are isolated from all other interfaces in other communities and from all isolated ports within the PVLAN domain.

A community port must be configured as an access port.

Primary, Isolated, and Community Private VLANs

Primary VLANs and the two types of secondary VLANs (isolated and community) have these characteristics:

• Primary VLAN—The primary VLAN carries traffic from the promiscuous ports to the host ports, both isolated and community, and to other promiscuous ports.

• Isolated VLAN—An isolated VLAN is a secondary VLAN that carries unidirectional traffic upstream from the hosts toward the promiscuous ports. You can configure only one isolated VLAN in a PVLAN domain. An isolated VLAN can have several isolated ports. The traffic from each isolated port also remains completely separate.

• Community VLAN—A community VLAN is a secondary VLAN that carries upstream traffic from the community ports to the promiscuous port and to other host ports in the same community. You can configure multiple community VLANs in a PVLAN domain. The ports within one community can communicate, but these ports cannot communicate with ports in any other community or isolated VLAN in the private VLAN.
The following figure shows the traffic flows within a PVLAN, along with the types of VLANs and types of ports.

**Figure 3: Private VLAN Traffic Flows**

The PVLAN traffic flows are unidirectional from the host ports to the promiscuous ports. Traffic received on primary VLAN enforces no separation and forwarding is done as in a normal VLAN.

**Note**
A promiscuous access port can serve only one primary VLAN and multiple secondary VLANs (community and isolated VLANs). With a promiscuous port, you can connect a wide range of devices as access points to a PVLAN. For example, you can use a promiscuous port to monitor or back up all the PVLAN servers from an administration workstation.

In a switched environment, you can assign an individual PVLAN and associated IP subnet to each individual or common group of end stations.

### Associating Primary and Secondary VLANs

To allow host ports in secondary VLANs to communicate outside the PVLAN, you associate secondary VLANs to the primary VLAN. If the association is not operational, the host ports (community and isolated ports) in the secondary VLAN are brought down.

**Note**
You can associate a secondary VLAN with only one primary VLAN.

For an association to be operational, the following conditions must be met:
• The primary VLAN must exist and be configured as a primary VLAN.

• The secondary VLAN must exist and be configured as either an isolated or community VLAN.

Note
Use the `show vlan private-vlan` command to verify that the association is operational. The switch does not display an error message when the association is nonoperational.

If you delete either the primary or secondary VLAN, the ports that are associated with the VLAN become inactive. Use the `no private-vlan` command to return the VLAN to the normal mode. All primary and secondary associations on that VLAN are suspended, but the interfaces remain in PVLAN mode. When you convert the VLAN back to PVLAN mode, the original associations are reinstated.

If you enter the `no vlan` command for the primary VLAN, all PVLAN associations with that VLAN are deleted. However, if you enter the `no vlan` command for a secondary VLAN, the PVLAN associations with that VLAN are suspended and are restored when you recreate the specified VLAN and configure it as the previous secondary VLAN.

In order to change the association between a secondary and primary VLAN, you must first remove the current association and then add the desired association.

Broadcast Traffic in Private VLANs

Broadcast traffic from ports in a private VLAN flows in the following ways:

• The broadcast traffic flows from a promiscuous port to all ports in the primary VLAN (which includes all the ports in the community and isolated VLANs). This broadcast traffic is distributed to all ports within the primary VLAN, including those ports that are not configured with private VLAN parameters.

• The broadcast traffic from an isolated port is distributed only to those promiscuous ports in the primary VLAN that are associated to that isolated port.

• The broadcast traffic from community ports is distributed to all ports within the port’s community and to all promiscuous ports that are associated to the community port. The broadcast packets are not distributed to any other communities within the primary VLAN or to any isolated ports.

Private VLAN Port Isolation

You can use PVLANs to control access to end stations as follows:

• Configure selected interfaces connected to end stations as isolated ports to prevent any communication. For example, if the end stations are servers, this configuration prevents communication between the servers.

• Configure interfaces connected to default gateways and selected end stations (for example, backup servers) as promiscuous ports to allow all end stations access to a default gateway.

Guidelines and Limitations for Private VLANs

When configuring PVLANs, follow these guidelines:
• You must have already created the VLAN before you can assign the specified VLAN as a private VLAN.

• You must enable PVLANs before the switch can apply the PVLAN functionality.

• You cannot disable PVLANs if the switch has any operational ports in a PVLAN mode.

• Enter the `private-vlan synchronize` command from within the Multiple Spanning Tree (MST) region definition to map the secondary VLANs to the same MST instance as the primary VLAN.

• You cannot connect a second switch to a promiscuous or isolated PVLAN trunk. The promiscuous or isolated PVLAN trunk is supported only on host-switch.

• Private VLANs do not provide port mode support for port channels.

• Private VLANs do not provide port mode support for virtual port channels (vPCs) interfaces.

## Configuring a Private VLAN

### Enabling Private VLANs

You must enable PVLANs on the switch to use the PVLAN functionality.

| Note | The PVLAN commands do not appear until you enable the PVLAN feature. |

| Procedure |

<table>
<thead>
<tr>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Step 1</strong></td>
<td>switch# configure terminal</td>
</tr>
<tr>
<td></td>
<td>Enters global configuration mode.</td>
</tr>
<tr>
<td><strong>Step 2</strong></td>
<td>switch(config)# feature private-vlan</td>
</tr>
<tr>
<td></td>
<td>Enables the PVLAN feature on the switch.</td>
</tr>
<tr>
<td><strong>Step 3</strong></td>
<td>switch(config)# no feature private-vlan</td>
</tr>
<tr>
<td></td>
<td>(Optional) Disables the PVLAN feature on the switch.</td>
</tr>
<tr>
<td></td>
<td><strong>Note</strong> You cannot disable PVLANs if there are operational ports on the switch that are in PVLAN mode.</td>
</tr>
</tbody>
</table>

This example shows how to enable the PVLAN feature on the switch:

```
switch# configure terminal
switch(config)# feature private-vlan
```

### Configuring a VLAN as a Private VLAN

To create a PVLAN, you first create a VLAN, and then configure that VLAN to be a PVLAN.
Before You Begin

Ensure that the PVLAN feature is enabled.

Procedure

<table>
<thead>
<tr>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Step 1</strong></td>
<td>switch# configure terminal</td>
</tr>
<tr>
<td><strong>Step 2</strong></td>
<td>switch(config)# vlan {vlan-id</td>
</tr>
<tr>
<td><strong>Step 3</strong></td>
<td>switch(config-vlan)# private-vlan {community</td>
</tr>
<tr>
<td><strong>Step 4</strong></td>
<td>switch(config-vlan)# no private-vlan {community</td>
</tr>
</tbody>
</table>

This example shows how to assign VLAN 5 to a PVLAN as the primary VLAN:

`switch# configure terminal
switch(config)# vlan 5
switch(config-vlan)# private-vlan primary`

This example shows how to assign VLAN 100 to a PVLAN as a community VLAN:

`switch# configure terminal
switch(config)# vlan 100
switch(config-vlan)# private-vlan community`

This example shows how to assign VLAN 200 to a PVLAN as an isolated VLAN:

`switch# configure terminal
switch(config)# vlan 200
switch(config-vlan)# private-vlan isolated`

**Associating Secondary VLANs with a Primary Private VLAN**

When you associate secondary VLANs with a primary VLAN, follow these guidelines:

- The `secondary-vlan-list` parameter cannot contain spaces. It can contain multiple comma-separated items. Each item can be a single secondary VLAN ID or a hyphenated range of secondary VLAN IDs.

- The `secondary-vlan-list` parameter can contain multiple community VLAN IDs and one isolated VLAN ID.

- Enter a `secondary-vlan-list` or use the `add` keyword with a `secondary-vlan-list` to associate secondary VLANs with a primary VLAN.
Use the `remove` keyword with a `secondary-vlan-list` to clear the association between secondary VLANs and a primary VLAN.

You change the association between a secondary and primary VLAN by removing the existing association and then adding the desired association.

If you delete either the primary or secondary VLAN, the VLAN becomes inactive on the port where the association is configured. When you enter the `no private-vlan` command, the VLAN returns to the normal VLAN mode. All primary and secondary associations on that VLAN are suspended, but the interfaces remain in PVLAN mode. If you again convert the specified VLAN to PVLAN mode, the original associations are reinstated.

If you enter the `no vlan` command for the primary VLAN, all PVLAN associations with that VLAN are lost. However, if you enter the `no vlan` command for a secondary VLAN, the PVLAN associations with that VLAN are suspended and are reinstated when you recreate the specified VLAN and configure it as the previous secondary VLAN.

**Before You Begin**

Ensure that the PVLAN feature is enabled.

**Procedure**

<table>
<thead>
<tr>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Step 1</strong></td>
<td>switch# configure terminal</td>
</tr>
<tr>
<td><strong>Step 2</strong></td>
<td>switch(config)# vlan primary-vlan-id</td>
</tr>
<tr>
<td><strong>Step 3</strong></td>
<td>switch(config-vlan)# private-vlan association {[add] secondary-vlan-list</td>
</tr>
<tr>
<td><strong>Step 4</strong></td>
<td>switch(config-vlan)# no private-vlan association</td>
</tr>
</tbody>
</table>

This example shows how to associate community VLANs 100 through 110 and isolated VLAN 200 with primary VLAN 5:

```
switch# configure terminal
switch(config)# vlan 5
switch(config-vlan)# private-vlan association 100-110, 200
```

**Configuring an Interface as a Private VLAN Host Port**

In PVLANs, host ports are part of the secondary VLANs, which are either community VLANs or isolated VLANs. Configuring a PVLAN host port involves two steps. First, you define the port as a PVLAN host port and then you configure a host association between the primary and secondary VLANs.
We recommend that you enable BPDU Guard on all interfaces configured as a host ports.

Before You Begin

Ensure that the PVLAN feature is enabled.

Procedure

<table>
<thead>
<tr>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td>Step 1    switch# configure terminal</td>
<td>Enters global configuration mode.</td>
</tr>
<tr>
<td>Step 2    switch(config)# interface type [chassis/slot/port]</td>
<td>Selects the port to configure as a PVLAN host port. This port can be on a FEX (identified by the chassis option).</td>
</tr>
<tr>
<td>Step 3    switch(config-if)# switchport mode private-vlan host</td>
<td>Configures the port as a host port for a PVLAN.</td>
</tr>
<tr>
<td>Step 4    switch(config-if)# switchport private-vlan host-association {primary-vlan-id} {secondary-vlan-id}</td>
<td>Associates the port with the primary and secondary VLANs of a PVLAN. The secondary VLAN can be either an isolated or community VLAN.</td>
</tr>
<tr>
<td>Step 5    switch(config-if)# no switchport private-vlan host-association</td>
<td>(Optional) Removes the PVLAN association from the port.</td>
</tr>
</tbody>
</table>

This example shows how to configure Ethernet port 1/12 as a host port for a PVLAN and associate it to primary VLAN 5 and secondary VLAN 101:

```
switch# configure terminal
switch(config)# interface ethernet 1/12
switch(config-if)# switchport mode private-vlan host
switch(config-if)# switchport private-vlan host-association 5 101
```

Configuring an Interface as a Private VLAN Promiscuous Port

In a PVLAN domain, promiscuous ports are part of the primary VLAN. Configuring a promiscuous port involves two steps. First, you define the port as a promiscuous port and then you configure the mapping between a secondary VLAN and the primary VLAN.

Before You Begin

Ensure that the PVLAN feature is enabled.
Procedure

<table>
<thead>
<tr>
<th>Step</th>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>switch# configure terminal</td>
<td>Enters global configuration mode.</td>
</tr>
<tr>
<td>2</td>
<td>switch(config)# interface type slot/port</td>
<td>Selects the port to configure as a PVLAN promiscuous port. A physical interface is required. This port cannot be on a FEX.</td>
</tr>
<tr>
<td>3</td>
<td>switch(config-if)# switchport mode private-vlan promiscuous</td>
<td>Configures the port as a promiscuous port for a PVLAN. You can only enable a physical Ethernet port as the promiscuous port.</td>
</tr>
<tr>
<td>4</td>
<td>switch(config-if)# switchport private-vlan mapping {primary-vlan-id} [secondary-vlan-list]</td>
<td>Configures the port as a promiscuous port and associates the specified port with a primary VLAN and a selected list of secondary VLANs. The secondary VLAN can be either an isolated or community VLAN.</td>
</tr>
<tr>
<td>5</td>
<td>switch(config-if)# no switchport private-vlan mapping</td>
<td>(Optional) Clears the mapping from the PVLAN.</td>
</tr>
</tbody>
</table>

This example shows how to configure Ethernet interface 1/4 as a promiscuous port associated with primary VLAN 5 and secondary isolated VLAN 200:

```
switch# configure terminal
switch(config)# interface ethernet 1/4
switch(config-if)# switchport mode private-vlan promiscuous
switch(config-if)# switchport private-vlan mapping 5 200
```

**Configuring a Layer 2 Interface as a Private VLAN Isolated Trunk Port**

You can configure a Layer 2 interface as a private VLAN isolated trunk port. These isolated trunk ports carry traffic for multiple secondary VLANs as well as normal VLANs.

**Note**
You must associate the primary and secondary VLANs before they become operational on the private VLAN isolated trunk port.

**Before You Begin**
Ensure that the private VLAN feature is enabled.
### Configuring a Layer 2 Interface as a Private VLAN Isolated Trunk Port

#### Procedure

<table>
<thead>
<tr>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Step 1</strong></td>
<td><strong>config t</strong> Enters configuration mode.</td>
</tr>
</tbody>
</table>
| **Example:** | switch(config)#
| **Step 2** | **interface type slot/port** Selects the Layer 2 port to configure as a private VLAN isolated trunk port. |
| **Example:** | switch(config)# interface ethernet 2/11 switch(config-if)#
| **Step 3** | **switchport** Configures the Layer 2 port as a switch port. |
| **Example:** | switch(config-if)# switchport switch(config-if)#
| **Step 4** | **switchport mode private-vlan trunk secondary** Configures the Layer 2 port as an isolated trunk port to carry traffic for multiple isolated VLANs. **Note** You cannot put community VLANs into the isolated trunk port. |
| **Example:** | switch(config-if)# switchport mode private-vlan trunk secondary switch(config-if)#
| **Step 5** | **switchport private-vlan trunk native vlan vlan-id** (Optional) Sets the native VLAN for the 802.1Q trunk. Valid values are from 1 to 3968 and 4048 to 4093. The default value is 1. **Note** If you are using a private VLAN as the native VLAN for the isolated trunk port, you must enter a value for a secondary VLAN or a normal VLAN; you cannot configure a primary VLAN as the native VLAN. |
| **Example:** | switch(config-if)# switchport private-vlan trunk native vlan 5
| **Step 6** | **switchport private-vlan trunk allowed vlan add vlan-list | all | except vlan-list | none | remove vlan-list** Sets the allowed VLANs for the private VLAN isolated trunk interface. Valid values are from 1 to 3968 and 4048 to 4093. When you map the private primary and secondary VLANs to the isolated trunk port, the system automatically puts all the primary VLANs into the allowed VLAN list for this port. |
| **Example:** | switch(config-if)# switchport private-vlan trunk allowed vlan add 1 switch(config-if)#

---

**Cisco Nexus 3000 Series NX-OS Layer 2 Switching Configuration Guide, Release 6.x**  
OL-29545-03
<table>
<thead>
<tr>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td><strong>Note</strong> Ensure that the native VLAN is part of the allowed VLAN list. The default for this command is to allow no VLANs on this interface, so you must configure the native VLAN as an allowed VLAN, unless it is already added as an associated VLAN, to pass native VLAN traffic. <strong>Note</strong> Do not configure primary VLAN as part of allowed VLAN list.</td>
</tr>
</tbody>
</table>

**Step 7** Enter one of the following commands
### Command or Action

<table>
<thead>
<tr>
<th>Option</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>switchport private-vlan association trunk {primary-vlan-id} {secondary-vlan-id}</code></td>
<td>Associates the Layer 2 isolated trunk port with the primary and secondary VLANs of private VLANs. The secondary VLAN must be an isolated VLAN. You can associate a maximum of 16 private VLAN primary and secondary pairs on each isolated trunk port. You must reenter the command for each pair of primary and secondary VLANs that you are working with. <strong>Note</strong> Each secondary VLAN on an isolated trunk port must be associated with a different primary VLAN. You cannot put two isolated VLANs that are associated with the same primary VLAN into a private VLAN isolated trunk port. If you do, the last entry overwrites the previous entry.</td>
</tr>
<tr>
<td><code>no switchport private-vlan association trunk [primary-vlan-id [secondary-vlan-id]]</code></td>
<td>Removes the private VLAN association from the private VLAN isolated trunk port.</td>
</tr>
</tbody>
</table>
### Configuring a Layer 2 Interface as a Private VLAN Isolated Trunk Port

<table>
<thead>
<tr>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Example:</strong></td>
<td></td>
</tr>
<tr>
<td>switch(config-if)# switchport private-vlan association trunk 10 101</td>
<td></td>
</tr>
<tr>
<td>switch(config-if)#</td>
<td></td>
</tr>
</tbody>
</table>

**Step 8**

```plaintext
[no] switchport private-vlan association trunk
{primary-vlan-id [secondary-vlan-id]}
```

**Example:**

`switch(config-if)# switchport private-vlan association trunk 10 101
switch(config-if)#`

- **Purpose:** Associate the Layer 2 isolated trunk port with the primary and secondary VLANs of private VLANs. The secondary VLAN must be an isolated VLAN. You can associate a maximum of 16 private VLAN primary and secondary pairs on each isolated trunk port. You must reenter the command for each pair of primary and secondary VLANs that you are working with.

**Note:** Each secondary VLAN on an isolated trunk port must be associated with a different primary VLAN. You cannot put two isolated VLANs that are associated with the same primary VLAN into a private VLAN isolated trunk port. If you do, the last entry overwrites the previous entry.

- **Purpose:** Remove the private VLAN association from the private VLAN isolated trunk port.

**Step 9**

```plaintext
exit
```

**Example:**

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

- **Purpose:** Exits the interface configuration mode.

**Step 10**

```plaintext
show interface switchport
```

**Example:**

`switch# show interface switchport`

- **Purpose:** Displays information on all interfaces configured as switch ports.

**Step 11**

```plaintext
copy running-config startup-config
```

**Example:**

`switch(config)# copy running-config startup-config`

- **Purpose:** Copies the running configuration to the startup configuration.

This example shows how to configure the Layer 2 port 2/1 as a private VLAN isolated trunk port associated with three different primary VLANs and an associated secondary VLAN:

```plaintext
switch# config t
switch(config)# interface ethernet 2/1
switch(config-if)# switchport mode private-vlan trunk
switch(config-if)# switchport private-vlan trunk allowed vlan add 1
switch(config-if)# switchport private-vlan association trunk 10 101
switch(config-if)# switchport private-vlan association trunk 20 201
```
### Configuring a Layer 2 Interface as a Private VLAN Promiscuous Trunk Port

You can configure a Layer 2 interface as a private VLAN promiscuous trunk port and then associate that promiscuous trunk port with multiple primary VLANs. These promiscuous trunk ports carry traffic for multiple primary VLANs as well as normal VLANs.

**Note**

You must associate the primary and secondary VLANs before they become operational on the private VLAN promiscuous trunk port.

**Before You Begin**

Ensure that the private VLAN feature is enabled.

**Procedure**

<table>
<thead>
<tr>
<th>Step</th>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td>Step 1</td>
<td><code>config t</code></td>
<td>Enters configuration mode.</td>
</tr>
<tr>
<td></td>
<td><strong>Example:</strong></td>
<td></td>
</tr>
<tr>
<td></td>
<td>switch# config t</td>
<td></td>
</tr>
<tr>
<td></td>
<td>switch(config)#</td>
<td></td>
</tr>
<tr>
<td>Step 2</td>
<td><code>interface {type slot/port}</code></td>
<td>Selects the Layer 2 port to configure as a private VLAN promiscuous trunk port.</td>
</tr>
<tr>
<td></td>
<td><strong>Example:</strong></td>
<td></td>
</tr>
<tr>
<td></td>
<td>switch(config)# interface ethernet 2/1</td>
<td></td>
</tr>
<tr>
<td></td>
<td>switch(config-if)#</td>
<td></td>
</tr>
<tr>
<td>Step 3</td>
<td><code>switchport</code></td>
<td>Configures the Layer 2 port as a switch port.</td>
</tr>
<tr>
<td></td>
<td><strong>Example:</strong></td>
<td></td>
</tr>
<tr>
<td></td>
<td>switch(config-if)# switchport</td>
<td></td>
</tr>
<tr>
<td></td>
<td>switch(config-if)#</td>
<td></td>
</tr>
<tr>
<td>Step 4</td>
<td><code>switchport mode private-vlan trunk promiscuous</code></td>
<td>Configures the Layer 2 port as a promiscuous trunk port to carry traffic for multiple private VLANs as well as normal VLANs.</td>
</tr>
<tr>
<td></td>
<td><strong>Example:</strong></td>
<td></td>
</tr>
<tr>
<td></td>
<td>switch(config-if)# switchport mode</td>
<td></td>
</tr>
<tr>
<td></td>
<td>private-vlan trunk promiscuous</td>
<td></td>
</tr>
<tr>
<td></td>
<td>switch(config-if)#</td>
<td></td>
</tr>
<tr>
<td>Step 5</td>
<td><code>switchport private-vlan trunk native vlan vlan-id</code></td>
<td>(Optional) Sets the native VLAN for the 802.1Q trunk. Valid values are from 1 to 3968 and 4048 to 4093. The default value is 1.</td>
</tr>
<tr>
<td></td>
<td><strong>Example:</strong></td>
<td></td>
</tr>
<tr>
<td></td>
<td>switch(config-if)# switchport</td>
<td></td>
</tr>
<tr>
<td></td>
<td>private-vlan trunk native vlan 5</td>
<td></td>
</tr>
</tbody>
</table>
### Configuring a Layer 2 Interface as a Private VLAN Promiscuous Trunk Port

<table>
<thead>
<tr>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Note</strong></td>
<td>If you are using a private VLAN as the native VLAN for the promiscuous trunk port, you must enter a value for a primary VLAN or a normal VLAN; you cannot configure a secondary VLAN as the native VLAN.</td>
</tr>
<tr>
<td><strong>Step 6</strong></td>
<td>switchport mode private-vlan trunk allowed vlan {add vlan-list</td>
</tr>
<tr>
<td>Example:</td>
<td>switch(config-if)# switchport private-vlan trunk allowed vlan add 1 switch(config-if)#</td>
</tr>
<tr>
<td></td>
<td>Sets the allowed VLANs for the private VLAN promiscuous trunk interface. Valid values are from 1 to 3968 and 4048 to 4093. When you map the private primary and secondary VLANs to the promiscuous trunk port, the system automatically puts all the primary VLANs into the allowed VLAN list for this port.</td>
</tr>
<tr>
<td><strong>Note</strong></td>
<td>Ensure that the native VLAN is part of the allowed VLAN list. The default for this command is to allow no VLANs on this interface, so you must configure the native VLAN as an allowed VLAN, unless it is already added as an associated VLAN, to pass native VLAN traffic.</td>
</tr>
<tr>
<td><strong>Step 7</strong></td>
<td>[no]switchport private-vlan mapping trunk primary-vlan-id {add secondary-vlan-id }</td>
</tr>
<tr>
<td>Example:</td>
<td>switch(config-if)# switchport private-vlan mapping trunk 4 add 5 switch(config-if)#</td>
</tr>
<tr>
<td></td>
<td>Map or remove the mapping for the promiscuous trunk port with the primary VLAN and a selected list of associated secondary VLANs. The secondary VLAN can be either an isolated or community VLAN. The private VLAN association between primary and secondary VLANs must be operational to pass traffic. You can map a maximum of 16 private VLAN primary and secondary pairs on each promiscuous trunk port. You must reenter the command for each primary VLAN that you are working with.</td>
</tr>
<tr>
<td>or</td>
<td>Remove the private VLAN promiscuous trunk mappings from the interface.</td>
</tr>
<tr>
<td><strong>Step 8</strong></td>
<td>exit</td>
</tr>
<tr>
<td>Example:</td>
<td>switch(config-if)# exit switch(config)#</td>
</tr>
<tr>
<td></td>
<td>Exits the interface configuration mode.</td>
</tr>
<tr>
<td><strong>Step 9</strong></td>
<td>show interface switchport</td>
</tr>
<tr>
<td>Example:</td>
<td>switch# show interface switchport</td>
</tr>
<tr>
<td></td>
<td>(Optional) Displays information on all interfaces configured as switch ports.</td>
</tr>
</tbody>
</table>
Purpose
Command or Action | Purpose
--- | ---
**Step 10** | copy running-config startup-config
**Example:** | switch(config)# copy running-config startup-config
(Optional) Copies the running configuration to the startup configuration.

This example shows how to configure the Layer 2 port 2/1 as a promiscuous trunk port associated with two primary VLANs and their associated secondary VLANs:

```
switch# configure terminal
switch(config)# interface ethernet 2/1
switch(config-if)# switchport
switch(config-if)# switchport mode private-vlan trunk promiscuous
switch(config-if)# switchport private-vlan trunk allowed vlan add 1
switch(config-if)# switchport private-vlan trunk mapping trunk 2 add 3
switch(config-if)# switchport private-vlan trunk mapping trunk 4 add 5
switch(config-if)# switchport private-vlan mapping trunk 1 add 20
switch(config-if)# exit
```

### Configuring VLAN Mapping for Selective Q-in-Q on a Private VLAN Secondary Trunk Port

To configure VLAN mapping for selective Q-in-Q on a Private VLAN secondary trunk port, complete the following steps.

**Procedure**

<table>
<thead>
<tr>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Step 1</strong></td>
<td>switch# configure terminal</td>
</tr>
<tr>
<td></td>
<td>Enters global configuration mode.</td>
</tr>
<tr>
<td><strong>Step 2</strong></td>
<td>switch(config)# interface interface-id</td>
</tr>
<tr>
<td></td>
<td>Enters interface configuration mode for the interface connected to the service provider network. You can enter a physical interface or an EtherChannel port channel.</td>
</tr>
<tr>
<td><strong>Step 3</strong></td>
<td>switch(config-if)# switchport mode private-vlan trunk secondary</td>
</tr>
<tr>
<td></td>
<td>Configure the interface as a Private VLAN secondary trunk port.</td>
</tr>
<tr>
<td><strong>Step 4</strong></td>
<td>switch(config-if)# switchport vlan mapping vlan-id-range all dot1q-tunnel outer vlan-id</td>
</tr>
</tbody>
</table>
| | Enters the VLAN IDs to be mapped:
| | • vlan-id-range—The customer VLAN ID range (C-VLAN) entering the switch from the customer network. The range is from 1 to 4094. You can enter a string of VLAN-IDs.
| | • all—Accepts all the VLAN IDs (C-VLAN) entering the switch from the customer network. |
Verifying the Private VLAN Configuration

Use the `no switchport vlan mapping vlan-id-range dot1q-tunnel outer vlan-id` command to remove the VLAN mapping configuration.

The following example shows how to drop all VLANs other than the configured mapping and allowed VLANs.

```
switch(config)# interface port-channel201
switch(config-if)# switchport mode private-vlan trunk secondary
switch(config-if)# switchport private-vlan trunk native vlan 2
switch(config-if)# switchport private-vlan trunk allowed vlan 2,500
switch(config-if)# switchport private-vlan association trunk 101 201
switch(config-if)# switchport private-vlan association trunk 102 202
switch(config-if)# switchport private-vlan association trunk 103 203
switch(config-if)# switchport private-vlan association trunk 104 204
switch(config-if)# switchport vlan mapping dot1q-tunnel allowed-vlan 2,201-204
switch(config-if)# switchport vlan mapping 300-400 dot1q-tunnel 500
switch(config-if)# spanning-tree port type edge trunk
switch(config-if)# spanning-tree bpdufilter enable
switch(config-if)# vpc 201
```

Verifying the Private VLAN Configuration

Use the following commands to display PVLAN configuration information.

<table>
<thead>
<tr>
<th>Command</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>switch# show feature</code></td>
<td>Displays the features enabled on the switch.</td>
</tr>
<tr>
<td><code>switch# show interface switchport</code></td>
<td>Displays information on all interfaces configured as switch ports.</td>
</tr>
<tr>
<td><code>switch# show vlan private-vlan [type]</code></td>
<td>Displays the status of the PVLAN.</td>
</tr>
</tbody>
</table>
This example shows how to display the PVLAN configuration:

```
switch# show vlan private-vlan
Primary Secondary Type Ports
------- --------- --------------- -------------------------------------------
5 100 community
5 101 community Eth1/12, Eth100/1/1
5 102 community
5 110 community
5 200 isolated Eth1/2
switch# show vlan private-vlan type
Vlan Type
---- -----------------
5 primary
100 community
101 community
102 community
110 community
200 isolated
```

This example shows how to display enabled features (some of the output has been removed for brevity):

```
switch# show feature
Feature Name Instance State
-------------------- -------- --------
fcsp 1 enabled
...
interface-vlan 1 enabled
private-vlan 1 enabled
udld 1 disabled
...
Verifying the Private VLAN Configuration
Configuring Access and Trunk Interfaces

This chapter contains the following sections:

- Information About Access and Trunk Interfaces, page 39
- Configuring Access and Trunk Interfaces, page 43
- Verifying the Interface Configuration, page 47

Information About Access and Trunk Interfaces

Understanding Access and Trunk Interfaces

Ethernet interfaces can be configured either as access ports or a trunk ports, as follows:

- An access port can have only one VLAN configured on the interface; it can carry traffic for only one VLAN.
- A trunk port can have two or more VLANs configured on the interface; it can carry traffic for several VLANs simultaneously.

Note

Cisco NX-OS supports only IEEE 802.1Q-type VLAN trunk encapsulation.
The following figure shows how you can use trunk ports in the network. The trunk port carries traffic for two or more VLANs.

*Figure 4: Devices in a Trunking Environment*

In order to correctly deliver the traffic on a trunk port with several VLANs, the device uses the IEEE 802.1Q encapsulation or tagging method.

To optimize the performance on access ports, you can configure the port as a host port. Once the port is configured as a host port, it is automatically set as an access port, and channel grouping is disabled. Use the host designation to decrease the time it takes the designated port to begin to forward packets.

---

**Note**

Only an end station can be set as a host port; you will receive an error message if you attempt to configure other ports as hosts.

If an access port receives a packet with an 802.1Q tag in the header other than the access VLAN value, that port drops the packet without learning its MAC source address.

---

**Note**

An Ethernet interface can function as either an access port or a trunk port; it cannot function as both port types simultaneously.

---

**Understanding IEEE 802.1Q Encapsulation**

A trunk is a point-to-point link between the device and another networking device. Trunks carry the traffic of multiple VLANs over a single link and allow you to extend VLANs across an entire network.

To correctly deliver the traffic on a trunk port with several VLANs, the device uses the IEEE 802.1Q encapsulation (tagging) method. This tag carries information about the specific VLAN to which the frame
and packet belong. This method allows packets that are encapsulated for several different VLANs to traverse the same port and maintain traffic separation between the VLANs.

Figure 5: Header Without and With 802.1Q Tag Included

<table>
<thead>
<tr>
<th>Preamble (7 - bytes)</th>
<th>Start Frame Deimiter (1-byte)</th>
<th>Dest. MAC Address (6-bytes)</th>
<th>Source MAC Address (6-bytes)</th>
<th>Length / Type (2-bytes)</th>
<th>MAC Client Data (0-n bytes)</th>
<th>Pad (0-p bytes)</th>
<th>Frame Check Sequence (4-bytes)</th>
</tr>
</thead>
</table>

3 bits = User Priority field
1 bit = Canonical Format Identifier (CFI)
12 bits = VLAN Identifier (VLAN ID)

Understanding Access VLANs

When you configure a port in access mode, you can specify which VLAN will carry the traffic for that interface. If you do not configure the VLAN for a port in access mode, or an access port, the interface carries traffic for the default VLAN (VLAN1).

You can change the access port membership in a VLAN by specifying the new VLAN. You must create the VLAN before you can assign it as an access VLAN for an access port. If you change the access VLAN on an access port to a VLAN that is not yet created, the system will shut that access port down.

Note
If you change the VLAN on an access port or a trunk port it will flap the interface. However, if the port is part of a vPC, then first change the native VLAN on the secondary vPC, and then to primary vPC.

If an access port receives a packet with an 802.1Q tag in the header other than the access VLAN value, that port drops the packet without learning its MAC source address.

Note
If you assign an access VLAN that is also a primary VLAN for a private VLAN, all access ports with that access VLAN will also receive all the broadcast traffic for the primary VLAN in the private VLAN mode.
Understanding the Native VLAN ID for Trunk Ports

A trunk port can carry untagged packets simultaneously with the 802.1Q tagged packets. When you assign a
default port VLAN ID to the trunk port, all untagged traffic travels on the default port VLAN ID for the trunk
port, and all untagged traffic is assumed to belong to this VLAN. This VLAN is referred to as the native
VLAN ID for a trunk port. The native VLAN ID is the VLAN that carries untagged traffic on trunk ports.

The trunk port sends an egressing packet with a VLAN that is equal to the default port VLAN ID as untagged;
all the other egressing packets are tagged by the trunk port. If you do not configure a native VLAN ID, the
trunk port uses the default VLAN.

Note
Native VLAN ID numbers must match on both ends of the trunk.

Understanding Allowed VLANs

By default, a trunk port sends traffic to and receives traffic from all VLANs. All VLAN IDs are allowed on
each trunk. However, you can remove VLANs from this inclusive list to prevent traffic from the specified
VLANs from passing over the trunk. You can add any specific VLANs later that you may want the trunk to
carry traffic for back to the list.

To partition the Spanning Tree Protocol (STP) topology for the default VLAN, you can remove VLAN1 from
the list of allowed VLANs. Otherwise, VLAN1, which is enabled on all ports by default, will have a very big
STP topology, which can result in problems during STP convergence. When you remove VLAN1, all data
traffic for VLAN1 on this port is blocked, but the control traffic continues to move on the port.

Understanding Native 802.1Q VLANs

To provide additional security for traffic passing through an 802.1Q trunk port, the `vlan dot1q tag native`
command was introduced. This feature provides a means to ensure that all packets going out of a 802.1Q
trunk port are tagged and to prevent reception of untagged packets on the 802.1Q trunk port.

Without this feature, all tagged ingress frames received on a 802.1Q trunk port are accepted as long as they
fall inside the allowed VLAN list and their tags are preserved. Untagged frames are tagged with the native
VLAN ID of the trunk port before further processing. Only those egress frames whose VLAN tags are inside
the allowed range for that 802.1Q trunk port are received. If the VLAN tag on a frame happens to match that
of the native VLAN on the trunk port, the tag is stripped off and the frame is sent untagged.

This behavior could potentially be exploited to introduce "VLAN hopping" in which a hacker could try and
have a frame jump to a different VLAN. It is also possible for traffic to become part of the native VLAN by
sending untagged packets into an 802.1Q trunk port.

To address the above issues, the `vlan dot1q tag native` command performs the following functions:

- On the ingress side, all untagged data traffic is dropped.
- On the egress side, all traffic is tagged. If traffic belongs to native VLAN it is tagged with the native
  VLAN ID.

This feature is supported on all the directly connected Ethernet and Port Channel interfaces.
Cisco NX-OS Release 6.0(2)U2(1) introduces the tx-only option, which allows both tagged and untagged packets at ingress. You can use the `vlan dot1q tag native tx-only` command to perform the following functions:

- On the ingress side, allow both tagged and untagged traffic.
- On the egress side, tag all traffic with the native vlan dot1q tag. For previously tagged frames, the existing dot1q tag is preserved. Untagged frames are tagged with the native vlan dot1q tag.

**Note**

You can enable the `vlan dot1q tag native` command by entering the command in the global configuration mode.

---

# Configuring Access and Trunk Interfaces

## Configuring a LAN Interface as an Ethernet Access Port

You can configure an Ethernet interface as an access port. An access port transmits packets on only one, untagged VLAN. You specify which VLAN traffic that the interface carries. If you do not specify a VLAN for an access port, the interface carries traffic only on the default VLAN. The default VLAN is VLAN1.

The VLAN must exist before you can specify that VLAN as an access VLAN. The system shuts down an access port that is assigned to an access VLAN that does not exist.

**Procedure**

<table>
<thead>
<tr>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Step 1</strong></td>
<td>switch# configure terminal</td>
</tr>
<tr>
<td></td>
<td>Enters global configuration mode.</td>
</tr>
<tr>
<td><strong>Step 2</strong></td>
<td>switch(config)# interface {{type slot/port}</td>
</tr>
<tr>
<td></td>
<td>Specifies an interface to configure, and enters interface configuration mode.</td>
</tr>
<tr>
<td><strong>Step 3</strong></td>
<td>switch(config-if)# switchport mode {access</td>
</tr>
<tr>
<td></td>
<td>Sets the interface as a nontrunking nontagged single-VLAN Ethernet interface. An access port can carry traffic in one VLAN only. By default, an access port carries traffic for VLAN1; to set the access port to carry traffic for a different VLAN, use the switchport access vlan command.</td>
</tr>
<tr>
<td><strong>Step 4</strong></td>
<td>switch(config-if)# switchport access vlan vlan-id</td>
</tr>
<tr>
<td></td>
<td>Specifies the VLAN for which this access port will carry traffic. If you do not enter this command, the access port carries traffic on VLAN1 only; use this command to change the VLAN for which the access port carries traffic.</td>
</tr>
</tbody>
</table>

This example shows how to set an interface as an Ethernet access port that carries traffic for a specific VLAN only:

```
switch# configure terminal
switch(config)# interface ethernet 1/10
```
Configuring Access Host Ports

By using a switchport host, you can make an access port a spanning-tree edge port, and enable BPDU Filtering and BPDU Guard at the same time.

Before You Begin
Ensure that you are configuring the correct interface; it must be an interface that is connected to an end station.

Procedure

<table>
<thead>
<tr>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Step 1</strong></td>
<td>switch# configure terminal</td>
</tr>
<tr>
<td><strong>Step 2</strong></td>
<td>switch(config)# interface type slot/port</td>
</tr>
<tr>
<td><strong>Step 3</strong></td>
<td>switch(config-if)# switchport host</td>
</tr>
</tbody>
</table>

Note: Apply this command only to switchports that connect to hosts.

This example shows how to set an interface as an Ethernet access host port with EtherChannel disabled:

```
switch# configure terminal
switch(config)# interface ethernet 1/10
switch(config-if)# switchport host
```

Configuring Trunk Ports

You can configure an Ethernet port as a trunk port; a trunk port transmits untagged packets for the native VLAN plus encapsulated, tagged, packets for multiple VLANs.

Note: Cisco NX-OS supports only 802.1Q encapsulation.

Procedure

<table>
<thead>
<tr>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Step 1</strong></td>
<td>switch# configure terminal</td>
</tr>
</tbody>
</table>
### Configuring Access and Trunk Interfaces

<table>
<thead>
<tr>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Step 2</strong></td>
<td>Specifies an interface to configure, and enters interface configuration mode.</td>
</tr>
<tr>
<td>switch(config)# interface {type slot/port</td>
<td>port-channel number}</td>
</tr>
</tbody>
</table>

**Step 3**

Sets the interface as an Ethernet trunk port. A trunk port can carry traffic in one or more VLANs on the same physical link (VLANs are based on the trunk-allowed VLANs list). By default, a trunk interface can carry traffic for all VLANs. To specify that only certain VLANs are allowed on the specified trunk, use the `switchport trunk allowed vlan` command.

**Step 2**

Sets the interface as an Ethernet trunk port. A trunk port can carry traffic in one or more VLANs on the same physical link (VLANs are based on the trunk-allowed VLANs list). By default, a trunk interface can carry traffic for all VLANs. To specify that only certain VLANs are allowed on the specified trunk, use the `switchport trunk allowed vlan` command.

<table>
<thead>
<tr>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td>switch(config-if)# switchport mode {access</td>
<td>trunk}</td>
</tr>
</tbody>
</table>

This example shows how to set an interface as an Ethernet trunk port:

```
switch# configure terminal
switch(config)# interface ethernet 1/3
switch(config-if)# switchport mode trunk
```

### Configuring the Native VLAN for 802.1Q Trunking Ports

If you do not configure this parameter, the trunk port uses the default VLAN as the native VLAN ID.

**Procedure**

<table>
<thead>
<tr>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Step 1</strong></td>
<td>Enters global configuration mode.</td>
</tr>
<tr>
<td>switch# configure terminal</td>
<td></td>
</tr>
<tr>
<td><strong>Step 2</strong></td>
<td>Specifies an interface to configure, and enters interface configuration mode.</td>
</tr>
<tr>
<td>switch(config)# interface {type slot/port</td>
<td>port-channel number}</td>
</tr>
<tr>
<td><strong>Step 3</strong></td>
<td>Sets the native VLAN for the 802.1Q trunk. Valid values are from 1 to 4094, except those VLANs reserved for internal use. The default value is VLAN1.</td>
</tr>
<tr>
<td>switch(config-if)# switchport trunk native vlan vlan-id</td>
<td></td>
</tr>
</tbody>
</table>

This example shows how to set the native VLAN for an Ethernet trunk port:

```
switch# configure terminal
switch(config)# interface ethernet 1/3
switch(config-if)# switchport trunk native vlan 5
```

### Configuring the Allowed VLANs for Trunking Ports

You can specify the IDs for the VLANs that are allowed on the specific trunk port.

Before you configure the allowed VLANs for the specified trunk ports, ensure that you are configuring the correct interfaces and that the interfaces are trunks.
### Procedure

<table>
<thead>
<tr>
<th>Step</th>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td>Step 1</td>
<td>switch# configure terminal</td>
<td>Enters global configuration mode.</td>
</tr>
<tr>
<td>Step 2</td>
<td>switch(config)# interface {type slot/port</td>
<td>port-channel number}</td>
</tr>
<tr>
<td>Step 3</td>
<td>switch(config-if)# switchport trunk allowed vlan {vlan-list all</td>
<td>none [add</td>
</tr>
</tbody>
</table>

**Note** You cannot add internally allocated VLANs as allowed VLANs on trunk ports. The system returns a message if you attempt to list an internally allocated VLAN as an allowed VLAN.

This example shows how to add VLANs to the list of allowed VLANs on an Ethernet trunk port:

```
switch# configure terminal
switch(config)# interface ethernet 1/3
switch(config-if)# switchport trunk allow vlan 15-20
```

### Configuring Native 802.1Q VLANs

Typically, you configure 802.1Q trunks with a native VLAN ID, which strips tagging from all packets on that VLAN. This configuration allows all untagged traffic and control traffic to transit the Cisco Nexus device. Packets that enter the switch with 802.1Q tags that match the native VLAN ID value are similarly stripped of tagging.

To maintain the tagging on the native VLAN and drop untagged traffic, enter the `vlan dot1q tag native` command. The switch will tag the traffic received on the native VLAN and admit only 802.1Q-tagged frames, dropping any untagged traffic in the native VLAN.

To maintain the tagging on the native VLAN and allow both tagged and untagged traffic, use the `vlan dot1q tag native tx-only` command.

Control traffic continues to be accepted untagged on the native VLAN on a trunked port, even when the `vlan dot1q tag native` command is enabled.

**Note** The `vlan dot1q tag native` command is enabled on global basis.
Procedure

<table>
<thead>
<tr>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Step 1</strong> switch# configure terminal</td>
<td>Enters global configuration mode.</td>
</tr>
<tr>
<td><strong>Step 2</strong> switch(config)# vlan dot1q tag native [tx-only]</td>
<td>Enables dot1q (IEEE 802.1Q) tagging for all native VLANs on all trunked ports on the Cisco Nexus device. By default, this feature is disabled.</td>
</tr>
<tr>
<td><strong>Step 3</strong> switch(config)# no vlan dot1q tag native [tx-only]</td>
<td>(Optional) Enables dot1q (IEEE 802.1Q) tagging for all native VLANs on all trunked ports on the switch.</td>
</tr>
<tr>
<td><strong>Step 4</strong> switch# show vlan dot1q tag native</td>
<td>(Optional) Displays the status of tagging on the native VLANs.</td>
</tr>
</tbody>
</table>

This example shows how to enable 802.1Q tagging on the switch:

```
switch# configure terminal
switch(config)# vlan dot1q tag native
switch(config)# exit
switch# show vlan dot1q tag native
vlan dot1q native tag is enabled
```

Verifying the Interface Configuration

Use the following commands to display access and trunk interface configuration information.

<table>
<thead>
<tr>
<th>Command</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td>switch# show interface</td>
<td>Displays the interface configuration</td>
</tr>
<tr>
<td>switch# show interface switchport</td>
<td>Displays information for all Ethernet interfaces, including access and trunk interfaces.</td>
</tr>
<tr>
<td>switch# show interface brief</td>
<td>Displays interface configuration information.</td>
</tr>
</tbody>
</table>
Configuring Switching Modes

This chapter contains the following sections:

- Information About Switching Modes, page 49
- Guidelines and Limitations for Switching Modes, page 50
- Licensing Requirements for Switching Modes, page 51
- Default Settings for Switching Modes, page 51
- Configuring Switching Modes, page 51

Information About Switching Modes

The switching mode determines whether the switch begins forwarding the frame as soon as the switch has read the destination details in the packet header or waits until the entire frame has been received and checked for cyclic redundancy check (CRC) errors before forwarding them to the network.

The switching mode is applicable to all packets being switched or routed through the hardware and can be saved persistently through reboots and restarts.

The switch operates in either of the following switching modes:

**Cut-Through Switching Mode**

Cut-through switching mode is enabled by default. Switches operating in cut-through switching mode start forwarding the frame as soon as the switch has read the destination details in the packet header. A switch in cut-through mode forwards the data before it has completed receiving the entire frame.

The switching speed in cut-through mode is faster than the switching speed in store-and-forward switching mode.

**Store-and-Forward Switching Mode**

When store-and-forward switching is enabled, the switch checks each frame for cyclic redundancy check (CRC) errors before forwarding them to the network. Each frame is stored until the entire frame has been received and checked.
Because it waits to forward the frame until the entire frame has been received and checked, the switching speed in store-and-forward switching mode is slower than the switching speed in cut-through switching mode.

**Guidelines and Limitations for Switching Modes**

Consider the following guidelines and limitations for each of the switching modes:

**Cut-Through Switching Mode Guidelines and Limitations**

- Cisco Nexus 3500 Series switches do not support switching modes in Cisco NX-OS Release 7.0(3)I7(2) and later releases.
- Packets with frame check sequence (FCS) errors are dropped.
  - For the Cisco Nexus 3064PQ platform, packets smaller than or equal to 768 bytes are dropped.
  - For Cisco Nexus 3016, 3064E, 3064X, and 3048 platforms, packets smaller than or equal to 560 bytes are dropped.
  - For the Cisco Nexus 3064PQ platform, packets larger than 769 or equal to bytes are forwarded.
  - For 3016, 3064E, 3064X, and 3048 platforms, packets larger than or equal to 561 bytes are forwarded.
- Packets with FCS errors are not mirrored if SPAN is configured.

**Store-and-Forward Switching Mode Guidelines and Limitations**

- Packets with FCS errors are dropped.
- Packets with FCS errors are not mirrored if SPAN is configured.
- The CPU port always operates in store-and-forward mode. Any packets forwarded to the CPU with FCS errors are dropped.
- The store-and-forward mode activates automatically for a port when the switch identifies that the ingress rate is less than the switching capacity of the egress port. For example, when the port ingress rate is 1 GB, the switching capacity of the egress port is 10 GB.

Cisco Nexus 3000 Series switches perform cut-through switching if the bits are serialized-in at the same or greater speed than they are serialized-out.

<table>
<thead>
<tr>
<th>In-Out</th>
<th>Cisco Nexus 3000 Switching Mode</th>
</tr>
</thead>
<tbody>
<tr>
<td>1 GB to 1 GB</td>
<td>cut-through</td>
</tr>
<tr>
<td>1 GB to 10 GB</td>
<td>store-and-forward</td>
</tr>
<tr>
<td>1 GB to 40 GB</td>
<td>store-and-forward</td>
</tr>
<tr>
<td>10 GB to 1 GB</td>
<td>cut-through</td>
</tr>
<tr>
<td>10 GB to 10 GB</td>
<td>cut-through</td>
</tr>
</tbody>
</table>
### In-Out | Cisco Nexus 3000 Switching Mode
--- | ---
10 GB to 40 GB | store-and-forward
40 GB to 1 GB | cut-through
40 GB to 10 GB | cut-through
40 GB to 40 GB | cut-through

**Note**
The global configuration does not change, even if store-and-forward mode is activated for an oversubscribed port.

---

**Licensing Requirements for Switching Modes**

Cut-through switching mode and store-and-forward switching modes do not require licenses. Any feature not included in a license package is bundled with the Cisco NX-OS system images and is provided at no extra charge to you. For a complete explanation of the Cisco NX-OS licensing scheme, see the *Cisco NX-OS Licensing Guide*.

---

**Default Settings for Switching Modes**

Cut-through switching is enabled by default.

---

**Configuring Switching Modes**

**Enabling Store-and-Forward Switching**

**Procedure**

<table>
<thead>
<tr>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td>switch# configure terminal</td>
<td>Enters global configuration mode.</td>
</tr>
<tr>
<td>switch(config) # switching-mode store-forward</td>
<td>Enables store-and-forward switching mode.</td>
</tr>
</tbody>
</table>
### Reenabling Cut-Through Switching

Cut-through switching is enabled by default. To reenable cut-through switching, use the `no` form of the `switching-mode store-forward` command.

#### Procedure

<table>
<thead>
<tr>
<th>Step</th>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td>Step 1</td>
<td><code>switch# configure terminal</code></td>
<td>Enters global configuration mode.</td>
</tr>
<tr>
<td>Step 2</td>
<td><code>switch(config)# no switching-mode store-forward</code></td>
<td>Disables store-and-forward switching mode. Enables cut-through switching mode.</td>
</tr>
<tr>
<td>Step 3</td>
<td><code>switch(config)# copy running-config startup-config</code></td>
<td>(Optional) Saves the change persistently through reboots and restarts by copying the running configuration to the startup configuration.</td>
</tr>
</tbody>
</table>

This example shows how to reenable cut-through switching:

```
switch# configure terminal
switch(config) # no switching-mode store-forward
switch(config) #
```
Configuring Rapid PVST+

This chapter contains the following sections:

- Information About Rapid PVST+, page 53
- Configuring Rapid PVST+, page 68
- Verifying the Rapid PVST+ Configuration, page 76
- Triggering the VLAN STP State Consistency Checker, page 77

Information About Rapid PVST+

The Rapid PVST+ protocol is the IEEE 802.1w standard, Rapid Spanning Tree Protocol (RSTP), implemented on a per VLAN basis. Rapid PVST+ interoperates with the IEEE 802.1D standard, which mandates a single STP instance for all VLANs, rather than per VLAN.

Rapid PVST+ is enabled by default on the default VLAN (VLAN1) and on all newly created VLANs in the software. Rapid PVST+ interoperates with switches that run legacy IEEE 802.1D STP.

RSTP is an improvement on the original STP standard, 802.1D, which allows faster convergence.

Note

Spanning tree is used to refer to IEEE 802.1w and IEEE 802.1s. If the text is discussing the IEEE 802.1D Spanning Tree Protocol, 802.1D is stated specifically.

Understanding STP

STP Overview

For an Ethernet network to function properly, only one active path can exist between any two stations.

When you create fault-tolerant internetworks, you must have a loop-free path between all nodes in a network. The STP algorithm calculates the best loop-free path throughout a switched network. LAN ports send and receive STP frames, which are called Bridge Protocol Data Units (BPDUs), at regular intervals. Switches do not forward these frames but use the frames to construct a loop-free path.
Multiple active paths between end stations cause loops in the network. If a loop exists in the network, end stations might receive duplicate messages and switches might learn end station MAC addresses on multiple LAN ports. These conditions result in a broadcast storm, which creates an unstable network.

STP defines a tree with a root bridge and a loop-free path from the root to all switches in the network. STP forces redundant data paths into a blocked state. If a network segment in the spanning tree fails and a redundant path exists, the STP algorithm recalculates the spanning tree topology and activates the blocked path.

When two LAN ports on a switch are part of a loop, the STP port priority and port path cost setting determine which port on the switch is put in the forwarding state and which port is put in the blocking state.

**Understanding How a Topology is Created**

All switches in an extended LAN that participate in a spanning tree gather information about other switches in the network by exchanging of BPDUs. This exchange of BPDUs results in the following actions:

- The system elects a unique root switch for the spanning tree network topology.
- The system elects a designated switch for each LAN segment.
- The system eliminates any loops in the switched network by placing redundant interfaces in a backup state; all paths that are not needed to reach the root switch from anywhere in the switched network are placed in an STP-blocked state.

The topology on an active switched network is determined by the following:

- The unique switch identifier Media Access Control (MAC) address of the switch that is associated with each switch
- The path cost to the root that is associated with each interface
- The port identifier that is associated with each interface

In a switched network, the root switch is the logical center of the spanning tree topology. STP uses BPDUs to elect the root switch and root port for the switched network, as well as the root port and designated port for each switched segment.

**Understanding the Bridge ID**

Each VLAN on each switch has a unique 64-bit bridge ID that consists of a bridge priority value, an extended system ID (IEEE 802.1t), and an STP MAC address allocation.

**Bridge Priority Value**

The bridge priority is a 4-bit value when the extended system ID is enabled.

**Note**

In Cisco NX-OS, the extended system ID is always enabled; you cannot disable the extended system ID.
Extended System ID

A 12-bit extended system ID field is part of the bridge ID.

Figure 6: Bridge ID with Extended System ID

The switches always use the 12-bit extended system ID.

Combined with the bridge ID, the system ID extension functions as the unique identifier for a VLAN.

Table 3: Bridge Priority Value and Extended System ID with the Extended System ID Enabled

<table>
<thead>
<tr>
<th>Bridge Priority Value</th>
<th>Extended System ID (Set Equal to the VLAN ID)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Bit 16</td>
<td>Bit 15 Bit 14 Bit 13 Bit 12 Bit 11 Bit 10 Bit 9 Bit 8 Bit 7 Bit 6 Bit 5 Bit 4 Bit 3 Bit 2 Bit 1</td>
</tr>
<tr>
<td>32768</td>
<td>16384 8192 4096 2048 1024 512 256 128 64 32 16 8 4 2 1</td>
</tr>
</tbody>
</table>

STP MAC Address Allocation

Extended system ID and MAC address reduction is always enabled on the software.

With MAC address reduction enabled on any switch, you should also enable MAC address reduction on all other connected switches to avoid undesirable root bridge election and spanning tree topology issues.

When MAC address reduction is enabled, the root bridge priority becomes a multiple of 4096 plus the VLAN ID. You can only specify a switch bridge ID (used by the spanning tree algorithm to determine the identity of the root bridge, the lowest being preferred) as a multiple of 4096. Only the following values are possible:

- 0
- 4096
- 8192
- 12288
- 16384
- 20480
- 24576
- 28672
- 32768
STP uses the extended system ID plus a MAC address to make the bridge ID unique for each VLAN.

**Note** If another bridge in the same spanning tree domain does not run the MAC address reduction feature, it could achieve root bridge ownership because its bridge ID may fall between the values specified by the MAC address reduction feature.

**Understanding BPDUs**

Switches transmit bridge protocol data units (BPDUs) throughout the STP instance. Each switch sends configuration BPDUs to communicate and compute the spanning tree topology. Each configuration BPDU contains the following minimal information:

- The unique bridge ID of the switch that the transmitting switch determines is the root bridge
- The STP path cost to the root
- The bridge ID of the transmitting bridge
- Message age
- The identifier of the transmitting port
- Values for the hello, forward delay, and max-age protocol timer
- Additional information for STP extension protocols

When a switch transmits a Rapid PVST+ BPDU frame, all switches connected to the VLAN on which the frame is transmitted receive the BPDU. When a switch receives a BPDU, it does not forward the frame but instead uses the information in the frame to calculate a BPDU, and, if the topology changes, initiate a BPDU transmission.

A BPDU exchange results in the following:

- One switch is elected as the root bridge.
- The shortest distance to the root bridge is calculated for each switch based on the path cost.
- A designated bridge for each LAN segment is selected. This is the switch closest to the root bridge through which frames are forwarded to the root.
- A root port is selected. This is the port providing the best path from the bridge to the root bridge.
- Ports included in the spanning tree are selected.
Election of the Root Bridge

For each VLAN, the switch with the lowest numerical value of the bridge ID is elected as the root bridge. If all switches are configured with the default priority (32768), the switch with the lowest MAC address in the VLAN becomes the root bridge. The bridge priority value occupies the most significant bits of the bridge ID.

When you change the bridge priority value, you change the probability that the switch will be elected as the root bridge. Configuring a lower value increases the probability; a higher value decreases the probability.

The STP root bridge is the logical center of each spanning tree topology in a network. All paths that are not needed to reach the root bridge from anywhere in the network are placed in STP blocking mode.

BPDUs contain information about the transmitting bridge and its ports, including bridge and MAC addresses, bridge priority, port priority, and path cost. STP uses this information to elect the root bridge for the STP instance, to elect the root port leading to the root bridge, and to determine the designated port for each segment.

Creating the Spanning Tree Topology

In the following figure, Switch A is elected as the root bridge because the bridge priority of all the switches is set to the default (32768) and Switch A has the lowest MAC address. However, due to traffic patterns, the number of forwarding ports, or link types, Switch A might not be the ideal root bridge. By increasing the priority (lowering the numerical value) of the ideal switch so that it becomes the root bridge, you force an STP recalculation to form a new spanning tree topology with the ideal switch as the root.

![Figure 7: Spanning Tree Topology](image)

When the spanning tree topology is calculated based on default parameters, the path between source and destination end stations in a switched network might not be ideal. For instance, connecting higher-speed links to a port that has a higher number than the current root port can cause a root-port change. The goal is to make the fastest link the root port.

For example, assume that one port on Switch B is a fiber-optic link, and another port on Switch B (an unshielded twisted-pair [UTP] link) is the root port. Network traffic might be more efficient over the high-speed fiber-optic link. By changing the STP port priority on the fiber-optic port to a higher priority (lower numerical value) than the root port, the fiber-optic port becomes the new root port.
Understanding Rapid PVST+

Rapid PVST+ Overview

Rapid PVST+ is the IEEE 802.1w (RSTP) standard implemented per VLAN. A single instance of STP runs on each configured VLAN (if you do not manually disable STP). Each Rapid PVST+ instance on a VLAN has a single root switch. You can enable and disable STP on a per-VLAN basis when you are running Rapid PVST+.

Note

Rapid PVST+ is the default STP mode for the switch.

Rapid PVST+ uses point-to-point wiring to provide rapid convergence of the spanning tree. The spanning tree reconfiguration can occur in less than 1 second with Rapid PVST+ (in contrast to 50 seconds with the default settings in the 802.1D STP).

Note

Rapid PVST+ supports one STP instance for each VLAN.

Using Rapid PVST+, STP convergence occurs rapidly. Each designated or root port in the STP sends out a BPDU every 2 seconds by default. On a designated or root port in the topology, if hello messages are missed three consecutive times, or if the maximum age expires, the port immediately flushes all protocol information in the table. A port considers that it loses connectivity to its direct neighbor root or designated port if it misses three BPDUs or if the maximum age expires. This rapid aging of the protocol information allows quick failure detection. The switch automatically checks the PVID.

Rapid PVST+ provides for rapid recovery of connectivity following the failure of a network device, a switch port, or a LAN. It provides rapid convergence for edge ports, new root ports, and ports connected through point-to-point links as follows:

• Edge ports—When you configure a port as an edge port on an RSTP switch, the edge port immediately transitions to the forwarding state. (This immediate transition was previously a Cisco-proprietary feature named PortFast.) You should only configure on ports that connect to a single end station as edge ports. Edge ports do not generate topology changes when the link changes.

Enter the `spanning-tree port type` interface configuration command to configure a port as an STP edge port.

Note

We recommend that you configure all ports connected to a host as edge ports.

• Root ports—If Rapid PVST+ selects a new root port, it blocks the old root port and immediately transitions the new root port to the forwarding state.

• Point-to-point links—If you connect a port to another port through a point-to-point link and the local port becomes a designated port, it negotiates a rapid transition with the other port by using the proposal-agreement handshake to ensure a loop-free topology.

Rapid PVST+ achieves rapid transition to the forwarding state only on edge ports and point-to-point links. Although the link type is configurable, the system automatically derives the link type information from the
duplex setting of the port. Full-duplex ports are assumed to be point-to-point ports, while half-duplex ports are assumed to be shared ports.

Edge ports do not generate topology changes, but all other designated and root ports generate a topology change (TC) BPDU when they either fail to receive three consecutive BPDUs from the directly connected neighbor or the maximum age times out. At this point, the designated or root port sends out a BPDU with the TC flag set. The BPDUs continue to set the TC flag as long as the TC While timer runs on that port. The value of the TC While timer is the value set for the hello time plus 1 second. The initial detector of the topology change immediately floods this information throughout the entire topology.

When Rapid PVST+ detects a topology change, the protocol does the following:

- Starts the TC While timer with a value equal to twice the hello time for all the non-edge root and designated ports, if necessary.
- Flushes the MAC addresses associated with all these ports.

The topology change notification floods quickly across the entire topology. The system flushes dynamic entries immediately on a per-port basis when it receives a topology change.

The TCA flag is used only when the switch is interacting with switches that are running legacy 802.1D STP.

The proposal and agreement sequence then quickly propagates toward the edge of the network and quickly restores connectivity after a topology change.

### Rapid PVST+ BPDUs

Rapid PVST+ and 802.1w use all six bits of the flag byte to add the role and state of the port that originates the BPDU and the proposal and agreement handshake. The following figure shows the use of the BPDU flags in Rapid PVST+.

![Figure 8: Rapid PVST+ Flag Byte in BPDU](image)

Another important change is that the Rapid PVST+ BPDU is type 2, version 2, which makes it possible for the switch to detect connected legacy (802.1D) bridges. The BPDU for 802.1D is version 0.
Proposal and Agreement Handshake

As shown in the following figure, Switch A is connected to Switch B through a point-to-point link, and all of the ports are in the blocking state. Assume that the priority of Switch A is a smaller numerical value than the priority of Switch B.

Figure 9: Proposal and Agreement Handshaking for Rapid Convergence

Switch A sends a proposal message (a configuration BPDU with the proposal flag set) to Switch B, proposing itself as the designated switch.

After receiving the proposal message, Switch B selects as its new root port the port from which the proposal message was received, forces all non-edge ports to the blocking state, and sends an agreement message (a BPDU with the agreement flag set) through its new root port.

After receiving the agreement message from Switch B, Switch A also immediately transitions its designated port to the forwarding state. No loops in the network can form because Switch B blocked all of its non-edge ports and because there is a point-to-point link between Switches A and B.

When Switch C connects to Switch B, a similar set of handshaking messages are exchanged. Switch C selects the port connected to Switch B as its root port, and both ends of the link immediately transition to the forwarding state. With each iteration of this handshaking process, one more network device joins the active topology. As the network converges, this proposal-agreement handshaking progresses from the root toward the leaves of the spanning tree.

The switch learns the link type from the port duplex mode: a full-duplex port is considered to have a point-to-point connection and a half-duplex port is considered to have a shared connection. You can override the default setting that is controlled by the duplex setting by entering the `spanning-tree link-type` interface configuration command.

This proposal/agreement handshake is initiated only when a non-edge port moves from the blocking to the forwarding state. The handshaking process then proliferates step-by-step throughout the topology.
Protocol Timers

The following table describes the protocol timers that affect the Rapid PVST+ performance.

**Table 4: Rapid PVST+ Protocol Timers**

<table>
<thead>
<tr>
<th>Variable</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Hello timer</td>
<td>Determines how often each switch broadcasts BPDUs to other switches. The default is 2 seconds, and the range is from 1 to 10.</td>
</tr>
<tr>
<td>Forward delay timer</td>
<td>Determines how long each of the listening and learning states last before the port begins forwarding. This timer is generally not used by the protocol but is used as a backup. The default is 15 seconds, and the range is from 4 to 30 seconds.</td>
</tr>
<tr>
<td>Maximum age timer</td>
<td>Determines the amount of time protocol information received on an port is stored by the switch. This timer is generally not used by the protocol, but it is used when interoperating with 802.1D spanning tree. The default is 20 seconds; the range is from 6 to 40 seconds.</td>
</tr>
</tbody>
</table>

Port Roles

Rapid PVST+ provides rapid convergence of the spanning tree by assigning port roles and learning the active topology. Rapid PVST+ builds upon the 802.1D STP to select the switch with the highest priority (lowest numerical priority value) as the root bridge. Rapid PVST+ then assigns one of these port roles to individual ports:

- **Root port**—Provides the best path (lowest cost) when the switch forwards packets to the root bridge.
- **Designated port**—Connects to the designated switch, which incurs the lowest path cost when forwarding packets from that LAN to the root bridge. The port through which the designated switch is attached to the LAN is called the designated port.
- **Alternate port**—Offers an alternate path toward the root bridge to the path provided by the current root port. An alternate port provides a path to another switch in the topology.
- **Backup port**—Acts as a backup for the path provided by a designated port toward the leaves of the spanning tree. A backup port can exist only when two ports are connected in a loopback by a point-to-point link or when a switch has two or more connections to a shared LAN segment. A backup port provides another path in the topology to the switch.
- **Disabled port**—Has no role within the operation of the spanning tree.

In a stable topology with consistent port roles throughout the network, Rapid PVST+ ensures that every root port and designated port immediately transition to the forwarding state while all alternate and backup ports
are always in the blocking state. Designated ports start in the blocking state. The port state controls the operation of the forwarding and learning processes.

A port with the root or a designated port role is included in the active topology. A port with the alternate or backup port role is excluded from the active topology (see the following figure).

*Figure 10: Sample Topology Demonstrating Port Roles*

---

**Port States**

**Rapid PVST+ Port State Overview**

Propagation delays can occur when protocol information passes through a switched LAN. As a result, topology changes can take place at different times and at different places in a switched network. When a LAN port transitions directly from nonparticipation in the spanning tree topology to the forwarding state, it can create temporary data loops. Ports must wait for new topology information to propagate through the switched LAN before starting to forward frames.

Each LAN port on a software using Rapid PVST+ or MST exists in one of the following four states:

- **Blocking**—The LAN port does not participate in frame forwarding.
- **Learning**—The LAN port prepares to participate in frame forwarding.
- **Forwarding**—The LAN port forwards frames.
- **Disabled**—The LAN port does not participate in STP and is not forwarding frames.

When you enable Rapid PVST+, every port in the software, VLAN, and network goes through the blocking state and the transitory states of learning at power up. If properly configured, each LAN port stabilizes to the forwarding or blocking state.
When the STP algorithm places a LAN port in the forwarding state, the following process occurs:

- The LAN port is put into the blocking state while it waits for protocol information that suggests it should go to the learning state.
- The LAN port waits for the forward delay timer to expire, moves the LAN port to the learning state, and restarts the forward delay timer.
- In the learning state, the LAN port continues to block frame forwarding as it learns the end station location information for the forwarding database.
- The LAN port waits for the forward delay timer to expire and then moves the LAN port to the forwarding state, where both learning and frame forwarding are enabled.

**Blocking State**

A LAN port in the blocking state does not participate in frame forwarding.

A LAN port in the blocking state performs as follows:

- Discards frames received from the attached segment.
- Discards frames switched from another port for forwarding.
- Does not incorporate the end station location into its address database. (There is no learning on a blocking LAN port, so there is no address database update.)
- Receives BPDUs and directs them to the system module.
- Receives, processes, and transmits BPDUs received from the system module.
- Receives and responds to network management messages.

**Learning State**

A LAN port in the learning state prepares to participate in frame forwarding by learning the MAC addresses for the frames. The LAN port enters the learning state from the blocking state.

A LAN port in the learning state performs as follows:

- Discards frames received from the attached segment.
- Discards frames switched from another port for forwarding.
- Incorporates the end station location into its address database.
- Receives BPDUs and directs them to the system module.
- Receives, processes, and transmits BPDUs received from the system module.
- Receives and responds to network management messages.

**Forwarding State**

A LAN port in the forwarding state forwards frames. The LAN port enters the forwarding state from the learning state.

A LAN port in the forwarding state performs as follows:
• Forwards frames received from the attached segment.
• Forwards frames switched from another port for forwarding.
• Incorporates the end station location information into its address database.
• Receives BPDUs and directs them to the system module.
• Processes BPDUs received from the system module.
• Receives and responds to network management messages.

Disabled State
A LAN port in the disabled state does not participate in frame forwarding or STP. A LAN port in the disabled state is virtually nonoperational.

A disabled LAN port performs as follows:
• Discards frames received from the attached segment.
• Discards frames switched from another port for forwarding.
• Does not incorporate the end station location into its address database. (There is no learning, so there is no address database update.)
• Does not receive BPDUs from neighbors.
• Does not receive BPDUs for transmission from the system module.

Summary of Port States
The following table lists the possible operational and Rapid PVST+ states for ports and the corresponding inclusion in the active topology.

Table 5: Port State Active Topology

<table>
<thead>
<tr>
<th>Operational Status</th>
<th>Port State</th>
<th>Is Port Included in the Active Topology?</th>
</tr>
</thead>
<tbody>
<tr>
<td>Enabled</td>
<td>Blocking</td>
<td>No</td>
</tr>
<tr>
<td>Enabled</td>
<td>Learning</td>
<td>Yes</td>
</tr>
<tr>
<td>Enabled</td>
<td>Forwarding</td>
<td>Yes</td>
</tr>
<tr>
<td>Disabled</td>
<td>Disabled</td>
<td>No</td>
</tr>
</tbody>
</table>

Synchronization of Port Roles
When the switch receives a proposal message on one of its ports and that port is selected as the new root port, Rapid PVST+ forces all other ports to synchronize with the new root information.
The switch is synchronized with superior root information received on the root port if all other ports are synchronized. An individual port on the switch is synchronized if either of the following applies:

- That port is in the blocking state.
- It is an edge port (a port configured to be at the edge of the network).

If a designated port is in the forwarding state and is not configured as an edge port, it transitions to the blocking state when the Rapid PVST+ forces it to synchronize with new root information. In general, when the Rapid PVST+ forces a port to synchronize with root information and the port does not satisfy any of the above conditions, its port state is set to blocking.

After ensuring that all of the ports are synchronized, the switch sends an agreement message to the designated switch that corresponds to its root port. When the switches connected by a point-to-point link are in agreement about their port roles, Rapid PVST+ immediately transitions the port states to the forwarding state. The sequence of events is shown in the following figure.

**Figure 11: Sequence of Events During Rapid Convergence**

![Sequence of Events During Rapid Convergence Diagram]

**Processing Superior BPDU Information**

A superior BPDU is a BPDU with root information (such as a lower switch ID or lower path cost) that is superior to what is currently stored for the port.

If a port receives a superior BPDU, Rapid PVST+ triggers a reconfiguration. If the port is proposed and is selected as the new root port, Rapid PVST+ forces all the other ports to synchronize.

If the received BPDU is a Rapid PVST+ BPDU with the proposal flag set, the switch sends an agreement message after all of the other ports are synchronized. The new root port transitions to the forwarding state as soon as the previous port reaches the blocking state.

If the superior information received on the port causes the port to become a backup port or an alternate port, Rapid PVST+ sets the port to the blocking state and sends an agreement message. The designated port continues sending BPDUs with the proposal flag set until the forward-delay timer expires. At that time, the port transitions to the forwarding state.
Processing Inferior BPDU Information

An inferior BPDU is a BPDU with root information (such as a higher switch ID or higher path cost) that is inferior to what is currently stored for the port.

If a designated port receives an inferior BPDU, it immediately replies with its own information.

Spanning-Tree Dispute Mechanism

The software checks the consistency of the port role and state in the received BPDUs to detect unidirectional link failures that could cause bridging loops.

When a designated port detects a conflict, it keeps its role, but reverts to a discarding state because disrupting connectivity in case of inconsistency is preferable to opening a bridging loop.

The following figure illustrates a unidirectional link failure that typically creates a bridging loop. Switch A is the root bridge, and its BPDUs are lost on the link leading to Switch B. The 802.1w-standard BPDUs include the role and state of the sending port. With this information, Switch A can detect that Switch B does not react to the superior BPDUs it sends and that Switch B is the designated, not root port. As a result, Switch A blocks (or keeps blocking) its port, which prevents the bridging loop. The block is shown as an STP dispute.

Figure 12: Detecting Unidirectional Link Failure

Port Cost

Rapid PVST+ uses the short (16-bit) path-cost method to calculate the cost by default. With the short path-cost method, you can assign any value in the range of 1 to 65535. However, you can configure the switch to use the long (32-bit) path-cost method, which allows you to assign any value in the range of 1 to 200,000,000. You configure the path-cost calculation method globally.

The STP port path-cost default value is determined from the media speed and path-cost calculation method of a LAN interface. If a loop occurs, STP considers the port cost when selecting a LAN interface to put into the forwarding state.

Table 6: Default Port Cost

<table>
<thead>
<tr>
<th>Bandwidth</th>
<th>Short Path-Cost Method of Port Cost</th>
<th>Long Path-Cost Method of Port Cost</th>
</tr>
</thead>
<tbody>
<tr>
<td>10 Mbps</td>
<td>100</td>
<td>2,000,000</td>
</tr>
</tbody>
</table>
### Bandwidth Cost Comparison

<table>
<thead>
<tr>
<th>Bandwidth</th>
<th>Short Path-Cost Method of Port Cost</th>
<th>Long Path-Cost Method of Port Cost</th>
</tr>
</thead>
<tbody>
<tr>
<td>100 Mbps</td>
<td>19</td>
<td>200,000</td>
</tr>
<tr>
<td>1 Gigabit Ethernet</td>
<td>4</td>
<td>20,000</td>
</tr>
<tr>
<td>10 Gigabit Ethernet</td>
<td>2</td>
<td>2,000</td>
</tr>
</tbody>
</table>

You can assign lower cost values to LAN interfaces that you want STP to select first and higher cost values to LAN interfaces that you want STP to select last. If all LAN interfaces have the same cost value, STP puts the LAN interface with the lowest LAN interface number in the forwarding state and blocks other LAN interfaces.

On access ports, you assign the port cost by the port. On trunk ports, you assign the port cost by the VLAN; you can configure the same port cost to all the VLANs on a trunk port.

### Port Priority

If a loop occurs and multiple ports have the same path cost, Rapid PVST+ considers the port priority when selecting which LAN port to put into the forwarding state. You can assign lower priority values to LAN ports that you want Rapid PVST+ to select first and higher priority values to LAN ports that you want Rapid PVST+ to select last.

If all LAN ports have the same priority value, Rapid PVST+ puts the LAN port with the lowest LAN port number in the forwarding state and blocks other LAN ports. The possible priority range is from 0 through 224 (the default is 128), configurable in increments of 32. The software uses the port priority value when the LAN port is configured as an access port and uses the VLAN port priority values when the LAN port is configured as a trunk port.

### Rapid PVST+ and IEEE 802.1Q Trunks

In a network of Cisco switches connected through 802.1Q trunks, the switches maintain one instance of STP for each VLAN allowed on the trunks. However, non-Cisco 802.1Q switches maintain only one instance of STP for all VLANs allowed on the trunks.

When you connect a Cisco switch to a non-Cisco switch through an 802.1Q trunk, the Cisco switch combines the STP instance of the 802.1Q VLAN of the trunk with the STP instance of the non-Cisco 802.1Q switch. However, all per-VLAN STP information that is maintained by Cisco switches is separated by a cloud of non-Cisco 802.1Q switches. The non-Cisco 802.1Q cloud that separates the Cisco switches is treated as a single trunk link between the switches.

### Rapid PVST+ Interoperation with Legacy 802.1D STP

Rapid PVST+ can interoperate with switches that are running the legacy 802.1D protocol. The switch knows that it is interoperating with equipment running 802.1D when it receives a BPDU version 0. The BPDUs for Rapid PVST+ are version 2. If the BPDU received is an 802.1w BPDU version 2 with the proposal flag set, the switch sends an agreement message after all of the other ports are synchronized. If the BPDU is an 802.1D
BPDU version 0, the switch does not set the proposal flag and starts the forward-delay timer for the port. The new root port requires twice the forward-delay time to transition to the forwarding state.

The switch interoperates with legacy 802.1D switches as follows:

- **Notification**—Unlike 802.1D BPDUs, 802.1w does not use TCN BPDUs. However, for interoperability with 802.1D switches, Cisco NX-OS processes and generates TCN BPDUs.

- **Acknowledgement**—When an 802.1w switch receives a TCN message on a designated port from an 802.1D switch, it replies with an 802.1D configuration BPDU with the TCA bit set. However, if the TC-while timer (the same as the TC timer in 802.1D) is active on a root port connected to an 802.1D switch and a configuration BPDU with the TCA set is received, the TC-while timer is reset.

This method of operation is required only for 802.1D switches. The 802.1w BPDUs do not have the TCA bit set.

- **Protocol migration**—For backward compatibility with 802.1D switches, 802.1w selectively sends 802.1D configuration BPDUs and TCN BPDUs on a per-port basis.

When a port is initialized, the migrate-delay timer is started (specifies the minimum time during which 802.1w BPDUs are sent), and 802.1w BPDUs are sent. While this timer is active, the switch processes all BPDUs received on that port and ignores the protocol type.

If the switch receives an 802.1D BPDU after the port migration-delay timer has expired, it assumes that it is connected to an 802.1D switch and starts using only 802.1D BPDUs. However, if the 802.1w switch is using 802.1D BPDUs on a port and receives an 802.1w BPDU after the timer has expired, it restarts the timer and starts using 802.1w BPDUs on that port.

---

**Note**

If you want all switches to renegotiate the protocol, you must restart Rapid PVST+.

---

### Rapid PVST+ Interoperation with 802.1s MST

Rapid PVST+ interoperates seamlessly with the IEEE 802.1s Multiple Spanning Tree (MST) standard. No user configuration is needed.

### Configuring Rapid PVST+

Rapid PVST+, which has the 802.1w standard applied to the Rapid PVST+ protocol, is the default STP setting in the software.

You enable Rapid PVST+ on a per-VLAN basis. The software maintains a separate instance of STP for each VLAN (except on those VLANs on which you disable STP). By default, Rapid PVST+ is enabled on the default VLAN and on each VLAN that you create.

### Enabling Rapid PVST+

Once you enable Rapid PVST+ on the switch, you must enable Rapid PVST+ on the specified VLANs. Rapid PVST+ is the default STP mode. You cannot simultaneously run MST and Rapid PVST+.
Changing the spanning tree mode disrupts traffic because all spanning tree instances are stopped for the previous mode and started for the new mode.

Procedure

<table>
<thead>
<tr>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Step 1</strong> switch# configure terminal</td>
<td>Enters global configuration mode.</td>
</tr>
<tr>
<td><strong>Step 2</strong> switch(config)# spanning-tree mode rapid-pvst</td>
<td>Enables Rapid PVST+ on the switch. Rapid PVST+ is the default spanning tree mode.</td>
</tr>
</tbody>
</table>

Note: Changing the spanning tree mode disrupts traffic because all spanning tree instances are stopped for the previous mode and started for the new mode.

This example shows how to enable Rapid PVST+ on the switch:

```
switch# configure terminal
switch(config)# spanning-tree mode rapid-pvst
```

Note: Because STP is enabled by default, entering the `show running-config` command to view the resulting configuration does not display the command that you entered to enable Rapid PVST+.

Enabling Rapid PVST+ per VLAN

You can enable or disable Rapid PVST+ on each VLAN.

Note: Rapid PVST+ is enabled by default on the default VLAN and on all VLANs that you create.

Procedure

<table>
<thead>
<tr>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Step 1</strong> switch# configure terminal</td>
<td>Enters global configuration mode.</td>
</tr>
<tr>
<td><strong>Step 2</strong> switch(config)# spanning-tree vlan-range</td>
<td>Enables Rapid PVST+ (default STP) on a per VLAN basis. The <code>vlan-range</code> value can be 2 through 4094 (except reserved VLAN values).</td>
</tr>
<tr>
<td><strong>Step 3</strong> switch(config)# no spanning-tree vlan-range</td>
<td>(Optional) Disables Rapid PVST+ on the specified VLAN.</td>
</tr>
</tbody>
</table>

Cisco Nexus 3000 Series NX-OS Layer 2 Switching Configuration Guide, Release 6.x
### Configuring the Root Bridge ID

The software maintains a separate instance of STP for each active VLAN in Rapid PVST+. For each VLAN, the switch with the lowest bridge ID becomes the root bridge for that VLAN.

To configure a VLAN instance to become the root bridge, modify the bridge priority from the default value (32768) to a significantly lower value.

When you enter the `spanning-tree vlan vlan_ID root` command, the switch checks the bridge priority of the current root bridges for each VLAN. The switch sets the bridge priority for the specified VLANs to 24576 if this value will cause the switch to become the root for the specified VLANs. If any root bridge for the specified VLANs has a bridge priority lower than 24576, the switch sets the bridge priority for the specified VLANs to 4096 less than the lowest bridge priority.

**Note**

The `spanning-tree vlan vlan_ID root` command fails if the value required to be the root bridge is less than 1.

**Caution**

The root bridge for each instance of STP should be a backbone or distribution switch. Do not configure an access switch as the STP primary root.

Enter the `diameter` keyword to specify the network diameter (that is, the maximum number of bridge hops between any two end stations in the network). When you specify the network diameter, the software automatically selects an optimal hello time, forward delay time, and maximum age time for a network of that diameter, which can significantly reduce the STP convergence time. You can enter the `hello-time` keyword to override the automatically calculated hello time.
With the switch configured as the root bridge, do not manually configure the hello time, forward-delay time, and maximum-age time using the `spanning-tree mst hello-time`, `spanning-tree mst forward-time`, and `spanning-tree mst max-age` configuration commands.

**Note**

With the switch configured as the root bridge, do not manually configure the hello time, forward-delay time, and maximum-age time using the `spanning-tree mst hello-time`, `spanning-tree mst forward-time`, and `spanning-tree mst max-age` configuration commands.

---

### Procedure

<table>
<thead>
<tr>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Step 1</strong> switch# configure terminal</td>
<td>Enters global configuration mode.</td>
</tr>
<tr>
<td><strong>Step 2</strong> switch(config)# spanning-tree vlan vlan-range root primary [diameter dia [hello-time hello-time]]</td>
<td>Configures a software switch as the primary root bridge. The <code>vlan-range</code> value can be 2 through 4094 (except reserved VLAN values.) The <code>dia</code> default is 7. The <code>hello-time</code> can be from 1 to 10 seconds, and the default value is 2 seconds.</td>
</tr>
</tbody>
</table>

This example shows how to configure the switch as the root bridge for a VLAN:

```
switch# configure terminal
switch(config)# spanning-tree vlan 5 root primary diameter 4
```

### Configuring a Secondary Root Bridge

When you configure a software switch as the secondary root, the STP bridge priority is modified from the default value (32768) so that the switch is likely to become the root bridge for the specified VLANs if the primary root bridge fails (assuming the other switches in the network use the default bridge priority of 32768). STP sets the bridge priority to 28672.

Enter the `diameter` keyword to specify the network diameter (that is, the maximum number of bridge hops between any two end stations in the network). When you specify the network diameter, the software automatically selects an optimal hello time, forward delay time, and maximum age time for a network of that diameter, which can significantly reduce the STP convergence time. You can enter the `hello-time` keyword to override the automatically calculated hello time.

You configure more than one switch in this manner to have multiple backup root bridges. Enter the same network diameter and hello time values that you used when configuring the primary root bridge.

**Note**

With the switch configured as the root bridge, do not manually configure the hello time, forward-delay time, and maximum-age time using the `spanning-tree mst hello-time`, `spanning-tree mst forward-time`, and `spanning-tree mst max-age` global configuration commands.
### Configuring the Secondary Root Bridge

#### Procedure

<table>
<thead>
<tr>
<th>Step</th>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td><code>switch# configure terminal</code></td>
<td>Enters global configuration mode.</td>
</tr>
<tr>
<td>2</td>
<td><code>switch(config)# spanning-tree vlan vlan-range root secondary [diameter dia [hello-time hello-time]]</code></td>
<td>Configures a software switch as the secondary root bridge. The <code>vlan-range</code> value can be 2 through 4094 (except reserved VLAN values). The <code>dia</code> default is 7. The <code>hello-time</code> can be from 1 to 10 seconds, and the default value is 2 seconds.</td>
</tr>
</tbody>
</table>

This example shows how to configure the switch as the secondary root bridge for a VLAN:

```
switch# configure terminal
switch(config)# spanning-tree vlan 5 root secondary diameter 4
```

### Configuring the Rapid PVST+ Port Priority

You can assign lower priority values to LAN ports that you want Rapid PVST+ to select first and higher priority values to LAN ports that you want Rapid PVST+ to select last. If all LAN ports have the same priority value, Rapid PVST+ puts the LAN port with the lowest LAN port number in the forwarding state and blocks other LAN ports.

The software uses the port priority value when the LAN port is configured as an access port and uses VLAN port priority values when the LAN port is configured as a trunk port.

#### Procedure

<table>
<thead>
<tr>
<th>Step</th>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td><code>switch# configure terminal</code></td>
<td>Enters global configuration mode.</td>
</tr>
<tr>
<td>2</td>
<td><code>switch(config)# interface type slot/port</code></td>
<td>Specifies the interface to configure, and enters interface configuration mode.</td>
</tr>
<tr>
<td>3</td>
<td><code>switch(config-if)# spanning-tree [vlan vlan-list] port-priority priority</code></td>
<td>Configures the port priority for the LAN interface. The <code>priority</code> value can be from 0 to 224. The lower the value indicates the higher the priority. The priority values are 0, 32, 64, 96, 128, 160, 192, and 224. All other values are rejected. The default value is 128.</td>
</tr>
</tbody>
</table>

This example shows how to configure the access port priority of an Ethernet interface:

```
switch# configure terminal
switch(config)# interface ethernet 1/4
switch(config-if)# spanning-tree port-priority 160
```

You can only apply this command to a physical Ethernet interface.
Configuring the Rapid PVST+ Path-Cost Method and Port Cost

On access ports, you assign port cost by the port. On trunk ports, you assign the port cost by VLAN; you can configure the same port cost on all the VLANs on a trunk.

Note

In Rapid PVST+ mode, you can use either the short or long path-cost method, and you can configure the method in either the interface or configuration submode. The default path-cost method is short.

Procedure

<table>
<thead>
<tr>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Step 1</strong> switch# configure terminal</td>
<td>Enters global configuration mode.</td>
</tr>
<tr>
<td><strong>Step 2</strong> switch(config)# spanning-tree pathcost method {long</td>
<td>short}</td>
</tr>
<tr>
<td><strong>Step 3</strong> switch(config)# interface type slot/port</td>
<td>Specifies the interface to configure, and enters interface configuration mode.</td>
</tr>
<tr>
<td><strong>Step 4</strong> switch(config-if)# spanning-tree [vlan vlan-id] cost [value</td>
<td>auto]</td>
</tr>
<tr>
<td></td>
<td>• short—1 to 65535</td>
</tr>
<tr>
<td></td>
<td>• long—1 to 20000000</td>
</tr>
<tr>
<td>Note</td>
<td>You configure this parameter per interface on access ports and per VLAN on trunk ports. The default is <strong>auto</strong>, which sets the port cost on both the path-cost calculation method and the media speed.</td>
</tr>
</tbody>
</table>

This example shows how to configure the access port cost of an Ethernet interface:

```
switch# configure terminal
switch (config)# spanning-tree pathcost method long
switch (config)# interface ethernet 1/4
switch(config-if)# spanning-tree cost 1000
```

You can only apply this command to a physical Ethernet interface.

Configuring the Rapid PVST+ Bridge Priority of a VLAN

You can configure the Rapid PVST+ bridge priority of a VLAN.
**Configuring the Rapid PVST+ Hello Time for a VLAN**

You can configure the Rapid PVST+ hello time for a VLAN.

**Procedure**

<table>
<thead>
<tr>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Step 1</strong></td>
<td>switch# configure terminal</td>
</tr>
<tr>
<td><strong>Step 2</strong></td>
<td>switch(config)# spanning-tree vlan vlan-range hello-time hello-time</td>
</tr>
</tbody>
</table>

This example shows how to configure the hello time for a VLAN:

```
switch# configure terminal
switch(config)# spanning-tree vlan 5 hello-time 7
```
Configuring the Rapid PVST+ Forward Delay Time for a VLAN

You can configure the forward delay time per VLAN when using Rapid PVST+.

Procedure

<table>
<thead>
<tr>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Step 1</strong> switch# configure terminal</td>
<td>Enters global configuration mode.</td>
</tr>
<tr>
<td><strong>Step 2</strong> switch(config)# spanning-tree vlan vlan-range forward-time forward-time</td>
<td>Configures the forward delay time of a VLAN. The forward delay time value can be from 4 to 30 seconds, and the default is 15 seconds.</td>
</tr>
</tbody>
</table>

This example shows how to configure the forward delay time for a VLAN:

```
switch# configure terminal
switch(config)# spanning-tree vlan 5 forward-time 21
```

Configuring the Rapid PVST+ Maximum Age Time for a VLAN

You can configure the maximum age time per VLAN when using Rapid PVST+.

Procedure

<table>
<thead>
<tr>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Step 1</strong> switch# configure terminal</td>
<td>Enters configuration mode.</td>
</tr>
<tr>
<td><strong>Step 2</strong> switch(config)# spanning-tree vlan vlan-range max-age max-age</td>
<td>Configures the maximum aging time of a VLAN. The maximum aging time value can be from 6 to 40 seconds, and the default is 20 seconds.</td>
</tr>
</tbody>
</table>

This example shows how to configure the maximum aging time for a VLAN:

```
switch# configure terminal
switch(config)# spanning-tree vlan 5 max-age 36
```

Specifying the Link Type

Rapid connectivity (802.1w standard) is established only on point-to-point links. By default, the link type is controlled from the duplex mode of the interface. A full-duplex port is considered to have a point-to-point connection; a half-duplex port is considered to have a shared connection.

If you have a half-duplex link physically connected point-to-point to a single port on a remote switch, you can override the default setting on the link type and enable rapid transitions.
If you set the link to shared, STP moves back to 802.1D.

### Procedure

<table>
<thead>
<tr>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Step 1</strong>&lt;br&gt;switch# configure terminal</td>
<td>Enters global configuration mode.</td>
</tr>
<tr>
<td><strong>Step 2</strong>&lt;br&gt;switch(config)# interface type slot/port</td>
<td>Specifies the interface to configure, and enters the interface configuration mode.</td>
</tr>
<tr>
<td><strong>Step 3</strong>&lt;br&gt;switch(config-if)# spanning-tree link-type {auto</td>
<td>point-to-point</td>
</tr>
</tbody>
</table>

This example shows how to configure the link type as a point-to-point link:

```plaintext
switch# configure terminal
switch (config)# interface ethernet 1/4
switch(config-if)# spanning-tree link-type point-to-point
```

You can only apply this command to a physical Ethernet interface.

### Restarting the Protocol

A bridge running Rapid PVST+ can send 802.1D BPDUs on one of its ports when it is connected to a legacy bridge. However, the STP protocol migration cannot determine whether the legacy switch has been removed from the link unless the legacy switch is the designated switch. You can restart the protocol negotiation (force the renegotiation with neighboring switches) on the entire switch or on specified interfaces.

<table>
<thead>
<tr>
<th>Command</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td>switch# clear spanning-tree detected-protocol [interface interface [interface-num</td>
<td>port-channel]]</td>
</tr>
</tbody>
</table>

This example shows how to restart Rapid PVST+ on an Ethernet interface:

```plaintext
switch# clear spanning-tree detected-protocol interface ethernet 1/8
```

### Verifying the Rapid PVST+ Configuration

Use the following commands to display Rapid PVST+ configuration information.

<table>
<thead>
<tr>
<th>Command</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td>show running-config spanning-tree [all]</td>
<td>Displays the current spanning tree configuration.</td>
</tr>
</tbody>
</table>
### Command

<table>
<thead>
<tr>
<th>Command</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td>show spanning-tree [options]</td>
<td>Displays selected detailed information for the current spanning tree configuration.</td>
</tr>
</tbody>
</table>

This example shows how to display spanning tree status:

```
switch# show spanning-tree brief
```

```
VLAN0001
  Spanning tree enabled protocol rstp
  Root ID    Priority 32768
  Address    001c.b05a.5447
  Cost       2
  Port 131 (Ethernet1/3)
  Hello Time 2 sec Max Age 20 sec Forward Delay 15 sec
  Bridge ID  Priority 32769 (priority 32768 sys-id-ext 1)
  Address    000d.ec6d.7841
  Hello Time 2 sec Max Age 20 sec Forward Delay 15 sec

<table>
<thead>
<tr>
<th>Interface</th>
<th>Role</th>
<th>Sts</th>
<th>Cost</th>
<th>Prio.Nbr</th>
<th>Type</th>
</tr>
</thead>
<tbody>
<tr>
<td>Eth1/3</td>
<td>Root</td>
<td>FWD</td>
<td>2</td>
<td>128.131</td>
<td>P2p Peer(STP)</td>
</tr>
</tbody>
</table>
```

### Triggering the VLAN STP State Consistency Checker

You can manually trigger the VLAN STP State consistency checker to compare the hardware and software configuration of the spanning tree state of a VLAN and display the results. To manually trigger the VLAN STP State consistency checker and display the results, use the following command in any mode:

#### Procedure

<table>
<thead>
<tr>
<th>Step 1</th>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>show consistency-checker stp-state vlan vlan-id</td>
<td>Starts a VLAN STP State consistency check on the specified VLAN and displays the results.</td>
</tr>
</tbody>
</table>

This example shows how to trigger a VLAN STP State consistency check and display the results:

```
switch# show consistency-checker stp-state vlan 250
Checks: Spanning tree state
Consistency Check: PASSED
  Vlan:250, Hardware state consistent for:
  Ethernet1/4
  Ethernet1/5
  Ethernet1/6
  Ethernet1/18
  Ethernet1/20
  Ethernet1/29
  Ethernet1/30
  Ethernet1/31
  Ethernet1/32
  Ethernet1/33
  Ethernet1/34
  Ethernet1/35
  Ethernet1/36
  Ethernet1/37
  Ethernet1/38
```
Configuring Rapid PVST+

Triggering the VLAN STP State Consistency Checker

Ethernet1/39
Ethernet1/40
Ethernet1/41
Ethernet1/42
Ethernet1/43
Ethernet1/44
Ethernet1/45
Ethernet1/46
Ethernet1/47
Ethernet1/48
Information About MST

MST Overview

Note

Spanning tree is used to refer to IEEE 802.1w and IEEE 802.1s. If the text is discussing the IEEE 802.1D Spanning Tree Protocol, 802.1D is stated specifically.

MST maps multiple VLANs into a spanning tree instance with each instance having a spanning tree topology independent of other spanning tree instances. This architecture provides multiple forwarding paths for data traffic, enables load balancing, and reduces the number of STP instances required to support a large number of VLANs.

MST provides rapid convergence through explicit handshaking as each MST instance uses the IEEE 802.1w standard, which eliminates the 802.1D forwarding delay and quickly transitions root bridge ports and designated ports to the forwarding state.

MAC address reduction is always enabled while you are using MST. You cannot disable this feature.

MST improves spanning tree operation and maintains backward compatibility with these STP versions:

• Original 802.1D spanning tree
• Rapid per-VLAN spanning tree (Rapid PVST+)

IEEE 802.1w defined the Rapid Spanning Tree Protocol (RSTP) and was incorporated into IEEE 802.1D.

• IEEE 802.1s defined MST and was incorporated into IEEE 802.1Q.
You must enable MST; Rapid PVST+ is the default spanning tree mode.

**MST Regions**

To allow switches to participate in MST instances, you must consistently configure the switches with the same MST configuration information.

A collection of interconnected switches that have the same MST configuration is an MST region. An MST region is a linked group of MST bridges with the same MST configuration.

The MST configuration controls the MST region to which each switch belongs. The configuration includes the name of the region, the revision number, and the MST VLAN-to-instance assignment map.

A region can have one or multiple members with the same MST configuration. Each member must be capable of processing 802.1w bridge protocol data units (BPDUs). There is no limit to the number of MST regions in a network.

Each region can support up to 65 MST instances (MSTIs). Instances are identified by any number in the range from 1 to 4094. The system reserves Instance 0 for a special instance, which is the IST. You can assign a VLAN to only one MST instance at a time.

The MST region appears as a single bridge to adjacent MST regions and to other Rapid PVST+ regions and 802.1D spanning tree protocols.

**Note**

We recommend that you do not partition the network into a large number of regions.

**MST BPDUs**

Each region has only one MST BPDU, and that BPDU carries an M-record for each MSTI within the region (see the following figure). Only the IST sends BPDUs for the MST region; all M-records are encapsulated in that one BPDU that the IST sends. Because the MST BPDU carries information for all instances, the number of BPDUs that need to be processed to support MSTIs is significantly reduced.

*Figure 13: MST BPDU with M-Records for MSTIs*
MST Configuration Information

The MST configuration that must be identical on all switches within a single MST region is configured by the user.

You can configure the following three parameters of the MST configuration:

- **Name**—32-character string, null padded and null terminated, identifying the MST region
- **Revision number**—Unsigned 16-bit number that identifies the revision of the current MST configuration

**Note**

You must set the revision number when required as part of the MST configuration. The revision number is *not* incremented automatically each time that the MST configuration is committed.

- MST configuration table—4096-element table that associates each of the potential 4094 VLANs supported to a given instance with the first (0) and last element (4095) set to 0. The value of element number X represents the instance to which VLAN X is mapped.

**Caution**

When you change the VLAN-to-MSTI mapping, the system restarts MST.

MST BPDUs contain these three configuration parameters. An MST bridge accepts an MST BPDU into its own region only if these three configuration parameters match exactly. If one configuration attribute differs, the MST bridge considers the BPDU to be from another MST region.

IST, CIST, and CST

**IST, CIST, and CST Overview**

Unlike Rapid PVST+, in which all the STP instances are independent, MST establishes and maintains IST, CIST, and CST spanning trees, as follows:

- An IST is the spanning tree that runs in an MST region.

MST establishes and maintains additional spanning trees within each MST region; these spanning trees are called multiple spanning tree instances (MSTIs).

Instance 0 is a special instance for a region, known as the IST. The IST always exists on all ports; you cannot delete the IST, or instance 0. By default, all VLANs are assigned to the IST. All other MST instances are numbered from 1 to 4094.

The IST is the only STP instance that sends and receives BPDUs. All of the other MSTI information is contained in MST records (M-records), which are encapsulated within MST BPDUs.

All MSTIs within the same region share the same protocol timers, but each MSTI has its own topology parameters, such as the root bridge ID, the root path cost, and so forth.

An MSTI is local to the region; for example, MSTI 9 in region A is independent of MSTI 9 in region B, even if regions A and B are interconnected.
• The CST interconnects the MST regions and any instance of 802.1D and 802.1w STP that may be running on the network. The CST is the one STP instance for the entire bridged network and encompasses all MST regions and 802.1w and 802.1D instances.

• A CIST is a collection of the ISTs in each MST region. The CIST is the same as an IST inside an MST region, and the same as a CST outside an MST region.

The spanning tree computed in an MST region appears as a subtree in the CST that encompasses the entire switched domain. The CIST is formed by the spanning tree algorithm running among switches that support the 802.1w, 802.1s, and 802.1D standards. The CIST inside an MST region is the same as the CST outside a region.

Spanning Tree Operation Within an MST Region

The IST connects all the MST switches in a region. When the IST converges, the root of the IST becomes the CIST regional root. The CIST regional root is also the CIST root if there is only one region in the network. If the CIST root is outside the region, the protocol selects one of the MST switches at the boundary of the region as the CIST regional root.

When an MST switch initializes, it sends BPDUs that identify itself as the root of the CIST and the CIST regional root, with both the path costs to the CIST root and to the CIST regional root set to zero. The switch also initializes all of its MSTIs and claims to be the root for all of them. If the switch receives superior MST root information (lower switch ID, lower path cost, and so forth) than the information that is currently stored for the port, it relinquishes its claim as the CIST regional root.

During initialization, an MST region might have many subregions, each with its own CIST regional root. As switches receive superior IST information from a neighbor in the same region, they leave their old subregions and join the new subregion that contains the true CIST regional root. This action causes all subregions to shrink except for the subregion that contains the true CIST regional root.

All switches in the MST region must agree on the same CIST regional root. Any two switches in the region will only synchronize their port roles for an MSTI if they converge to a common CIST regional root.

Spanning Tree Operations Between MST Regions

If you have multiple regions or 802.1w or 802.1D STP instances within a network, MST establishes and maintains the CST, which includes all MST regions and all 802.1w and 802.1D STP switches in the network. The MSTIs combine with the IST at the boundary of the region to become the CST.

The IST connects all the MST switches in the region and appears as a subtree in the CIST that encompasses the entire switched domain. The root of the subtree is the CIST regional root. The MST region appears as a virtual switch to adjacent STP switches and MST regions.
The following figure shows a network with three MST regions and an 802.1D switch (D). The CIST regional root for region 1 (A) is also the CIST root. The CIST regional root for region 2 (B) and the CIST regional root for region 3 (C) are the roots for their respective subtrees within the CIST.

Figure 14: MST Regions, CIST Regional Roots, and CST Root

Only the CST instance sends and receives BPDUs. MSTIs add their spanning tree information into the BPDUs (as M-records) to interact with neighboring switches and compute the final spanning tree topology. Because of this process, the spanning tree parameters related to the BPDU transmission (for example, hello time, forward time, max-age, and max-hops) are configured only on the CST instance but affect all MSTIs. You can configure the parameters related to the spanning tree topology (for example, the switch priority, the port VLAN cost, and the port VLAN priority) on both the CST instance and the MSTI.

MST switches use Version 3 BPDUs or 802.1D STP BPDUs to communicate with 802.1D-only switches. MST switches use MST BPDUs to communicate with MST switches.

MST Terminology

MST naming conventions include identification of some internal or regional parameters. These parameters are used only within an MST region, compared to external parameters that are used throughout the whole network. Because the CIST is the only spanning tree instance that spans the whole network, only the CIST parameters require the external qualifiers and not the internal or regional qualifiers. The MST terminology is as follows:

- The CIST root is the root bridge for the CIST, which is the unique instance that spans the whole network.
• The CIST external root path cost is the cost to the CIST root. This cost is left unchanged within an MST region. An MST region looks like a single switch to the CIST. The CIST external root path cost is the root path cost calculated between these virtual switches and switches that do not belong to any region.

• If the CIST root is in the region, the CIST regional root is the CIST root. Otherwise, the CIST regional root is the closest switch to the CIST root in the region. The CIST regional root acts as a root bridge for the IST.

• The CIST internal root path cost is the cost to the CIST regional root in a region. This cost is only relevant to the IST, instance 0.

**Hop Count**

MST does not use the message-age and maximum-age information in the configuration BPDU to compute the STP topology inside the MST region. Instead, the protocol uses the path cost to the root and a hop-count mechanism similar to the IP time-to-live (TTL) mechanism.

By using the `spanning-tree mst max-hops` global configuration command, you can configure the maximum hops inside the region and apply it to the IST and all MST instances in that region.

The hop count achieves the same result as the message-age information (triggers a reconfiguration). The root bridge of the instance always sends a BPDU (or M-record) with a cost of 0 and the hop count set to the maximum value. When a switch receives this BPDU, it decrements the received remaining hop count by one and propagates this value as the remaining hop count in the BPDU that it generates. When the count reaches zero, the switch discards the BPDU and ages the information held for the port.

The message-age and maximum-age information in the 802.1w portion of the BPDU remain the same throughout the region (only on the IST), and the same values are propagated by the region-designated ports at the boundary.

You configure a maximum aging time as the number of seconds that a switch waits without receiving spanning tree configuration messages before attempting a reconfiguration.

**Boundary Ports**

A boundary port is a port that connects one region to another. A designated port knows that it is on the boundary if it detects an STP bridge or receives an agreement proposal from an MST bridge with a different configuration or a Rapid PVST+ bridge. This definition allows two ports that are internal to a region to share a segment
with a port that belongs to a different region, creating the possibility of receiving both internal and external messages on a port (see the following figure).

**Figure 15: MST Boundary Ports**

![Image showing MST Boundary Ports]

At the boundary, the roles of MST ports do not matter; the system forces their state to be the same as the IST port state. If the boundary flag is set for the port, the MST port-role selection process assigns a port role to the boundary and assigns the same state as the state of the IST port. The IST port at the boundary can take up any port role except a backup port role.

**Spanning-Tree Dispute Mechanism**

Currently, this feature is not present in the IEEE MST standard, but it is included in the standard-compliant implementation. The software checks the consistency of the port role and state in the received BPDUs to detect unidirectional link failures that could cause bridging loops.

When a designated port detects a conflict, it keeps its role but reverts to a discarding state because disrupting connectivity in case of inconsistency is preferable to opening a bridging loop.

The following figure shows a unidirectional link failure that typically creates a bridging loop. Switch A is the root bridge, and its BPDUs are lost on the link leading to Switch B. Rapid PVST+ (802.1w) and MST BPDUs include the role and state of the sending port. With this information, Switch A can detect that Switch B does not react to the superior BPDUs that it sends and that Switch B is the designated, not root port. As a result, Switch A blocks (or keeps blocking) its port, which prevents the bridging loop. The block is shown as an STP dispute.

**Figure 16: Detecting a Unidirectional Link Failure**

![Image showing Detecting a Unidirectional Link Failure]
Port Cost and Port Priority

Spanning tree uses port costs to break a tie for the designated port. Lower values indicate lower port costs, and spanning tree chooses the least costly path. Default port costs are taken from the bandwidth of the interface, as follows:

- 10 Mbps—2,000,000
- 100 Mbps—200,000
- 1 Gigabit Ethernet—20,000
- 10 Gigabit Ethernet—2,000

You can configure the port costs in order to influence which port is chosen.

**Note**  
MST always uses the long path-cost calculation method, so the range of valid values is between 1 and 200,000,000.

The system uses port priorities to break ties among ports with the same cost. A lower number indicates a higher priority. The default port priority is 128. You can configure the priority to values between 0 and 224, in increments of 32.

Interoperability with IEEE 802.1D

A switch that runs MST supports a built-in protocol migration feature that enables it to interoperate with 802.1D STP switches. If this switch receives an 802.1D configuration BPDU (a BPDU with the protocol version set to 0), it sends only 802.1D BPDUs on that port. In addition, an MST switch can detect that a port is at the boundary of a region when it receives an 802.1D BPDU, an MST BPDU (Version 3) associated with a different region, or an 802.1w BPDU (Version 2).

However, the switch does not automatically revert to the MST mode if it no longer receives 802.1D BPDUs because it cannot detect whether the 802.1D switch has been removed from the link unless the 802.1D switch is the designated switch. A switch might also continue to assign a boundary role to a port when the switch to which this switch is connected has joined the region.

To restart the protocol migration process (force the renegotiation with neighboring switches), enter the `clear spanning-tree detected-protocols` command.

All Rapid PVST+ switches (and all 8021.D STP switches) on the link can process MST BPDUs as if they are 802.1w BPDUs. MST switches can send either Version 0 configuration and topology change notification (TCN) BPDUs or Version 3 MST BPDUs on a boundary port. A boundary port connects to a LAN, the designated switch of which is either a single spanning tree switch or a switch with a different MST configuration.

**Note**  
MST interoperates with the Cisco prestandard Multiple Spanning Tree Protocol (MSTP) whenever it receives prestandard MSTP on an MST port; no explicit configuration is necessary.
Interoperability with Rapid PVST+: Understanding PVST Simulation

MST interoperates with Rapid PVST+ with no need for user configuration. The PVST simulation feature enables this seamless interoperability.

Note
PVST simulation is enabled by default. That is, by default, all interfaces on the switch interoperate between MST and Rapid PVST+.

However, you may want to control the connection between MST and Rapid PVST+ to protect against accidentally connecting an MST-enabled port to a Rapid PVST+-enabled port. Because Rapid PVST+ is the default STP mode, you may encounter many Rapid PVST+-enabled connections.

Disabling Rapid PVST+ simulation, which can be done per port or globally for the entire switch, moves the MST-enabled port to the blocking state once it detects it is connected to a Rapid PVST+-enabled port. This port remains in the inconsistent state until the port stops receiving Rapid PVST+/SSTP BPDUs, and then the port resumes the normal STP transition process.

Configuring MST

MST Configuration Guidelines

When configuring MST, follow these guidelines:

- When you work with private VLANs, enter the `private-vlan synchronize` command to map the secondary VLANs to the same MST instance as the primary VLAN.
- When you are in the MST configuration mode, the following guidelines apply:
  - Each command reference line creates its pending regional configuration.
  - The pending region configuration starts with the current region configuration.
  - To leave the MST configuration mode without committing any changes, enter the `abort` command.
  - To leave the MST configuration mode and commit all the changes that you made before you left the mode, enter the `exit` command.

Enabling MST

You must enable MST; Rapid PVST+ is the default.

Caution
Changing the spanning tree mode disrupts traffic because all spanning tree instances are stopped for the previous mode and started for the new mode.
### Enter MST Configuration Mode

You enter MST configuration mode to configure the MST name, VLAN-to-instance mapping, and MST revision number on the switch.

For two or more switches to be in the same MST region, they must have the identical MST name, VLAN-to-instance mapping, and MST revision number.

#### Note
Each command reference line creates its pending regional configuration in MST configuration mode. In addition, the pending region configuration starts with the current region configuration.

When you are working in MST configuration mode, note the difference between the exit and abort commands.

#### Procedure

<table>
<thead>
<tr>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Step 1</strong></td>
<td>switch# configure terminal</td>
</tr>
<tr>
<td><strong>Step 2</strong></td>
<td>switch(config)# spanning-tree mode mst configuration</td>
</tr>
</tbody>
</table>

- MST name
- Instance-to-VLAN mapping
### Specifying the MST Name

You configure a region name on the bridge. For two or more bridges to be in the same MST region, they must have the identical MST name, VLAN-to-instance mapping, and MST revision number.

#### Procedure

<table>
<thead>
<tr>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Step 1</strong></td>
<td>switch# configure terminal</td>
</tr>
<tr>
<td><strong>Step 2</strong></td>
<td>switch(config)# spanning-tree mst configuration</td>
</tr>
<tr>
<td><strong>Step 3</strong></td>
<td>switch(config-mst)# name name</td>
</tr>
</tbody>
</table>

This example shows how to set the name of the MST region:

```
switch# configure terminal
switch(config)# spanning-tree mst configuration
switch(config-mst)# name accounting
```
Specifying the MST Configuration Revision Number

You configure the revision number on the bridge. For two or more bridges to be in the same MST region, they must have the identical MST name, VLAN-to-instance mapping, and MST revision number.

**Procedure**

<table>
<thead>
<tr>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Step 1</strong>  switch# configure terminal</td>
<td>Enters global configuration mode.</td>
</tr>
<tr>
<td><strong>Step 2</strong>  switch(config)# spanning-tree mst configuration</td>
<td>Enters MST configuration submode.</td>
</tr>
<tr>
<td><strong>Step 3</strong>  switch(config-mst)# revision version</td>
<td>Specifies the revision number for the MST region. The range is from 0 to 65535, and the default value is 0.</td>
</tr>
</tbody>
</table>

This example shows how to configure the revision number of the MSTI region for 5:

```plaintext
switch# configure terminal
switch(config)# spanning-tree mst configuration
switch(config-mst)# revision 5
```

Specifying the Configuration on an MST Region

For two or more switches to be in the same MST region, they must have the same VLAN-to-instance mapping, the same configuration revision number, and the same MST name.

A region can have one member or multiple members with the same MST configuration; each member must be capable of processing IEEE 802.1w RSTP BPDUs. There is no limit to the number of MST regions in a network, but each region can support only up to 65 MST instances. You can assign a VLAN to only one MST instance at a time.

**Procedure**

<table>
<thead>
<tr>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Step 1</strong>  switch# configure terminal</td>
<td>Enters global configuration mode.</td>
</tr>
<tr>
<td><strong>Step 2</strong>  switch(config)# spanning-tree mst configuration</td>
<td>Enters MST configuration submode.</td>
</tr>
</tbody>
</table>
| **Step 3**  switch(config-mst)# instance instance-id vlan vlan-range | Maps VLANs to an MST instance as follows:
  - For `instance-id`, the range is from 1 to 4094.
  - For `vlan vlan-range`, the range is from 1 to 4094. |
When you map VLANs to an MST instance, the mapping is incremental, and the VLANs specified in the command are added to or removed from the VLANs that were previously mapped.

To specify a VLAN range, enter a hyphen; for example, enter the `instance 1 vlan 1-63` command to map VLANs 1 through 63 to MST instance 1.

To specify a VLAN series, enter a comma; for example, enter the `instance 1 vlan 10, 20, 30` command to map VLANs 10, 20, and 30 to MST instance 1.

**Step 4**

```
switch(config-mst)# name name
```

Specifies the instance name. The `name` string has a maximum length of 32 case-sensitive characters.

**Step 5**

```
switch(config-mst)# revision version
```

Specifies the configuration revision number. The range is from 0 to 65535.

To return to defaults, do the following:

- To return to the default MST region configuration settings, enter the `no spanning-tree mst configuration` configuration command.
- To return to the default VLAN-to-instance map, enter the `no instance instance-id vlan vlan-range` MST configuration command.
- To return to the default name, enter the `no name` MST configuration command.
- To return to the default revision number, enter the `no revision` MST configuration command.
- To reenable Rapid PVST+, enter the `no spanning-tree mode` or the `spanning-tree mode rapid-pvst` global configuration command.

This example shows how to enter MST configuration mode, map VLANs 10 to 20 to MST instance 1, name the region region1, set the configuration revision to 1, display the pending configuration, apply the changes, and return to global configuration mode:

```
switch(config)# spanning-tree mst configuration
switch(config-mst)# instance 1 vlan 10-20
switch(config-mst)# name region1
switch(config-mst)# revision 1
switch(config-mst)# show pending
Pending MST configuration
Name [region1]
Revision 1
Instances configured 2
Instance Vlans Mapped
---------- ---------------------
0 1-9,21-4094
1 10-20
----------------------------------
```
Mapping and Unmapping VLANs to MST Instances

Caution
When you change the VLAN-to-MSTI mapping, the system restarts MST.

Note
You cannot disable an MSTI.

For two or more bridges to be in the same MST region, they must have the identical MST name, VLAN-to-instance mapping, and MST revision number.

Procedure

<table>
<thead>
<tr>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td>Step 1 switch# configure terminal</td>
<td>Enters global configuration mode.</td>
</tr>
<tr>
<td>Step 2 switch(config)# spanning-tree mst configuration</td>
<td>Enters MST configuration submode.</td>
</tr>
</tbody>
</table>
| Step 3 switch(config-mst)# instance instance-id vlan vlan-range | Maps VLANs to an MST instance, as follows:  
  - For instance-id the range is from 1 to 4094.  
  Instance 0 is reserved for the IST for each MST region.  
  - For vlan-range the range is from 1 to 4094.  
  When you map VLANs to an MSTI, the mapping is incremental, and the VLANs specified in the command are added to or removed from the VLANs that were previously mapped. |
| Step 4 switch(config-mst)# no instance instance-id vlan vlan-range | Deletes the specified instance and returns the VLANs to the default MSTI, which is the CIST. |

This example shows how to map VLAN 200 to MSTI 3:

```
switch# configure terminal
switch(config)# spanning-tree mst configuration
switch(config-mst)# instance 3 vlan 200
```

Mapping Secondary VLANs to Same MSTI as Primary VLANs for Private VLANs

When you are working with private VLANs on the system, all secondary VLANs must be in the same MSTI and their associated primary VLAN.
Configuring Multiple Spanning Tree

### Configuring the Root Bridge

You can configure the switch to become the root bridge.

**Note**

The root bridge for each MSTI should be a backbone or distribution switch. Do not configure an access switch as the spanning tree primary root bridge.

Enter the `diameter` keyword, which is available only for MSTI 0 (or the IST), to specify the network diameter (that is, the maximum number of hops between any two end stations in the network). When you specify the network diameter, the switch automatically sets an optimal hello time, forward-delay time, and maximum-age time for a network of that diameter, which can significantly reduce the convergence time. You can enter the `hello` keyword to override the automatically calculated hello time.

**Note**

With the switch configured as the root bridge, do not manually configure the hello time, forward-delay time, and maximum-age time using the `spanning-tree mst hello-time`, `spanning-tree mst forward-time`, and `spanning-tree mst max-age` global configuration commands.

### Procedure

<table>
<thead>
<tr>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Step 1</strong> switch# configure terminal</td>
<td>Enters global configuration mode.</td>
</tr>
<tr>
<td><strong>Step 2</strong> switch(config)# spanning-tree mst configuration</td>
<td>Enters MST configuration submode.</td>
</tr>
<tr>
<td><strong>Step 3</strong> switch(config-mst)# private-vlan synchronize</td>
<td>Automatically maps all secondary VLANs to the same MSTI as their associated primary VLAN in all private VLANs.</td>
</tr>
</tbody>
</table>
Configuring a Secondary Root Bridge

You can execute this command on more than one switch to configure multiple backup root bridges. Enter the same network diameter and hello-time values that you used when you configured the primary root bridge with the `spanning-tree mst root primary` configuration command.

Procedure

<table>
<thead>
<tr>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Step 1</strong></td>
<td>switch# configure terminal</td>
</tr>
<tr>
<td></td>
<td>Enters global configuration mode.</td>
</tr>
<tr>
<td><strong>Step 2</strong></td>
<td>switch(config)# spanning-tree mst instance-id root {primary</td>
</tr>
<tr>
<td></td>
<td>Configures a switch as the secondary root bridge as follows:</td>
</tr>
<tr>
<td></td>
<td>• For <code>instance-id</code>, you can specify a single instance, a range of instances separated by a hyphen, or a series of instances separated by a comma. The range is from 1 to 4094.</td>
</tr>
<tr>
<td></td>
<td>• For diameter <code>net-diameter</code>, specify the maximum number of hops between any two end stations. The default is 7. This keyword is available only for MST instance 0.</td>
</tr>
<tr>
<td></td>
<td>• For hello-time <code>seconds</code>, specify the interval in seconds between the generation of configuration messages by the root bridge. The range is from 1 to 10 seconds; the default is 2 seconds.</td>
</tr>
</tbody>
</table>
### Configuring the Port Priority

If a loop occurs, MST uses the port priority when selecting an interface to put into the forwarding state. You can assign lower priority values to interfaces that you want selected first and higher priority values to the interface that you want selected last. If all interfaces have the same priority value, MST puts the interface with the lowest interface number in the forwarding state and blocks the other interfaces.

#### Procedure

<table>
<thead>
<tr>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
</table>
| **Step 1** | switch# configure terminal
Enters global configuration mode. |
| **Step 2** | switch(config)# interface {{type slot/port} | {port-channel number}}
Specifies an interface to configure, and enters interface configuration mode. |
| **Step 3** | switch(config-if)# spanning-tree mst instance-id port-priority priority
Configures the port priority as follows:
- For instance-id, you can specify a single MSTI, a range of MSTIs separated by a hyphen, or a series of MSTIs separated by a comma. The range is from 1 to 4094.
- For priority, the range is 0 to 224 in increments of 32. The default is 128. A lower number indicates a higher priority.
The priority values are 0, 32, 64, 96, 128, 160, 192, and 224. The system rejects all other values. |

This example shows how to set the MST interface port priority for MSTI 3 on Ethernet port 3/1 to 64:

```
switch# configure terminal
switch(config)# interface ethernet 3/1
switch(config-if)# spanning-tree mst 3 port-priority 64
```

You can only apply this command to a physical Ethernet interface.
Configuring the Port Cost

The MST path-cost default value is derived from the media speed of an interface. If a loop occurs, MST uses the cost when selecting an interface to put in the forwarding state. You can assign lower cost values to interfaces that you want selected first and higher cost to interfaces values that you want selected last. If all interfaces have the same cost value, MST puts the interface with the lowest interface number in the forwarding state and blocks the other interfaces.

Note

MST uses the long path-cost calculation method.

Procedure

<table>
<thead>
<tr>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td>Step 1 switch# configure terminal</td>
<td>Enters global configuration mode.</td>
</tr>
<tr>
<td>Step 2 switch(config)# interface {{type slot/port}</td>
<td>{port-channel number}}</td>
</tr>
<tr>
<td>Step 3 switch(config-if)# spanning-tree mst instance-id cost [cost</td>
<td>auto]</td>
</tr>
<tr>
<td></td>
<td>• For instance-id, you can specify a single instance, a range of instances separated by a hyphen, or a series of instances separated by a comma. The range is from 1 to 4094.</td>
</tr>
<tr>
<td></td>
<td>• For cost, the range is from 1 to 200000000. The default value is auto, which is derived from the media speed of the interface.</td>
</tr>
</tbody>
</table>

This example shows how to set the MST interface port cost on Ethernet 3/1 for MSTI 4:

```
switch# configure terminal
switch(config)# interface ethernet 3/1
switch(config-if)# spanning-tree mst 4 cost 17031970
```

Configuring the Switch Priority

You can configure the switch priority for an MST instance so that it is more likely that the specified switch is chosen as the root bridge.
Exercise care when using this command. For most situations, we recommend that you enter the `spanning-tree mst root primary` and the `spanning-tree mst root secondary` global configuration commands to modify the switch priority.

### Procedure

<table>
<thead>
<tr>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Step 1</strong> switch# configure terminal</td>
<td>Enters global configuration mode.</td>
</tr>
<tr>
<td><strong>Step 2</strong> switch(config)# spanning-tree mst</td>
<td>Configures a switch priority as follows:</td>
</tr>
<tr>
<td>instance-id priority priority-value</td>
<td>• For <code>instance-id</code>, you can specify a single instance, a range of</td>
</tr>
<tr>
<td></td>
<td>instances separated by a hyphen, or a series of instances separated</td>
</tr>
<tr>
<td></td>
<td>by a comma. The range is from 1 to 4094.</td>
</tr>
<tr>
<td></td>
<td>• For priority, the range is from 0 to 61440 in increments of 4096; the</td>
</tr>
<tr>
<td></td>
<td>default is 32768. A lower number indicates that the switch will</td>
</tr>
<tr>
<td></td>
<td>most likely be chosen as the root bridge.</td>
</tr>
</tbody>
</table>

Priority values are 0, 4096, 8192, 12288, 16384, 20480, 24576, 28672, 32768, 36864, 40960, 45056, 49152, 53248, 57344, and 61440. The system rejects all other values.

This example shows how to configure the priority of the bridge to 4096 for MSTI 5:

```plaintext
switch# configure terminal
switch(config)# spanning-tree mst 5 priority 4096
```

### Configuring the Hello Time

You can configure the interval between the generation of configuration messages by the root bridge for all instances on the switch by changing the hello time.

Exercise care when using this command. For most situations, we recommend that you enter the `spanning-tree mst instance-id root primary` and the `spanning-tree mst instance-id root secondary` configuration commands to modify the hello time.

### Procedure

<table>
<thead>
<tr>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Step 1</strong> switch# configure terminal</td>
<td>Enters global configuration mode.</td>
</tr>
<tr>
<td><strong>Step 2</strong> switch(config)# spanning-tree mst</td>
<td>Configures the hello time for all MST instances. The hello time is</td>
</tr>
<tr>
<td>hello-time seconds</td>
<td>the interval between the generation of configuration</td>
</tr>
</tbody>
</table>

```plaintext
Configuring the Hello Time

You can configure the interval between the generation of configuration messages by the root bridge for all instances on the switch by changing the hello time.

Exercise care when using this command. For most situations, we recommend that you enter the `spanning-tree mst instance-id root primary` and the `spanning-tree mst instance-id root secondary` configuration commands to modify the hello time.

### Procedure

<table>
<thead>
<tr>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Step 1</strong> switch# configure terminal</td>
<td>Enters global configuration mode.</td>
</tr>
<tr>
<td><strong>Step 2</strong> switch(config)# spanning-tree mst</td>
<td>Configures the hello time for all MST instances. The hello</td>
</tr>
<tr>
<td>hello-time seconds</td>
<td>time is the interval between the generation of configuration</td>
</tr>
</tbody>
</table>

```
messages by the root bridge. These messages mean that the switch is alive. For seconds, the range is from 1 to 10, and the default is 2 seconds.

This example shows how to configure the hello time of the switch to 1 second:

```
switch# configure terminal
switch(config)# spanning-tree mst hello-time 1
```

### Configuring the Forwarding-Delay Time

You can set the forward delay timer for all MST instances on the switch with one command.

**Procedure**

<table>
<thead>
<tr>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Step 1</strong> switch# configure terminal</td>
<td>Enters global configuration mode.</td>
</tr>
<tr>
<td><strong>Step 2</strong> switch(config)# spanning-tree mst forward-time seconds</td>
<td>Configures the forward time for all MST instances. The forward delay is the number of seconds that a port waits before changing from its spanning tree blocking and learning states to the forwarding state. For seconds, the range is from 4 to 30, and the default is 15 seconds.</td>
</tr>
</tbody>
</table>

This example shows how to configure the forward-delay time of the switch to 10 seconds:

```
switch# configure terminal
switch(config)# spanning-tree mst forward-time 10
```

### Configuring the Maximum-Aging Time

The maximum-aging timer is the number of seconds that a switch waits without receiving spanning tree configuration messages before attempting a reconfiguration.

You set the maximum-aging timer for all MST instances on the switch with one command (the maximum age time only applies to the IST).

**Procedure**

<table>
<thead>
<tr>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Step 1</strong> switch# configure terminal</td>
<td>Enters global configuration mode.</td>
</tr>
</tbody>
</table>
Configuring the Maximum-Hop Count

MST uses the path cost to the IST regional root and a hop-count mechanism similar to the IP time-to-live (TTL) mechanism. You configure the maximum hops inside the region and apply it to the IST and all MST instances in that region. The hop count achieves the same result as the message-age information (triggers a reconfiguration).

**Procedure**

<table>
<thead>
<tr>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Step 1</strong></td>
<td><code>switch# configure terminal</code></td>
</tr>
<tr>
<td></td>
<td>Enters global configuration mode.</td>
</tr>
<tr>
<td><strong>Step 2</strong></td>
<td><code>switch(config)# spanning-tree mst max-hops hop-count</code></td>
</tr>
<tr>
<td></td>
<td>Specifies the number of hops in a region before the BPDU is discarded, and the information held for a port is aged. For <code>hop-count</code>, the range is from 1 to 255, and the default value is 20 hops.</td>
</tr>
</tbody>
</table>

This example shows how to set the maximum hops to 40:

```
switch# configure terminal
switch(config)# spanning-tree mst max-hops 40
```

Configuring PVST Simulation Globally

You can block this automatic feature either globally or per port. You can enter the global command and change the PVST simulation setting for the entire switch while you are in interface command mode.
### Configuring PVST Simulation Per Port

MST interoperates seamlessly with Rapid PVST+. However, to prevent an accidental connection to a switch that does not run MST as the default STP mode, you may want to disable this automatic feature. If you disable PVST simulation, the MST-enabled port moves to the blocking state once it detects it is connected to a Rapid PVST+-enabled port. This port remains in the inconsistent state until the port stops receiving BPDUs, and then the port resumes the normal STP transition process.

You can block this automatic feature either globally or per port.

#### Procedure

<table>
<thead>
<tr>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Step 1</strong> switch# configure terminal</td>
<td>Enters global configuration mode.</td>
</tr>
<tr>
<td><strong>Step 2</strong> switch(config)# no spanning-tree mst simulate pvst global</td>
<td>Disables all interfaces on the switch from automatically interoperating with a connected switch that is running in Rapid PVST+ mode. By default, all interfaces on the switch operate seamlessly between Rapid PVST+ and MST.</td>
</tr>
</tbody>
</table>

This example shows how to prevent the switch from automatically interoperating with a connecting switch that is running Rapid PVST+:

```bash
switch# configure terminal
switch(config)# no spanning-tree mst simulate pvst global
```
This example shows how to prevent the specified interfaces from automatically interoperating with a connecting switch that is not running MST:

```
switch# configure terminal
switch(config)# interface ethernet 1/4
switch(config-if)# spanning-tree mst simulate pvst disable
```

### Specifying the Link Type

Rapid connectivity (802.1w standard) is established only on point-to-point links. By default, the link type is controlled from the duplex mode of the interface. A full-duplex port is considered to have a point-to-point connection; a half-duplex port is considered to have a shared connection.

If you have a half-duplex link physically connected point-to-point to a single port on a remote switch, you can override the default setting on the link type and enable rapid transitions.

If you set the link to shared, STP reverts to 802.1D.

**Procedure**

<table>
<thead>
<tr>
<th>Step</th>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>switch# configure terminal</td>
<td>Enters configuration mode.</td>
</tr>
<tr>
<td>2</td>
<td>switch(config)# interface type slot/port</td>
<td>Specifies the interface to configure, and enters interface configuration mode.</td>
</tr>
<tr>
<td>3</td>
<td>switch(config-if)# spanning-tree link-type {auto</td>
<td>point-to-point</td>
</tr>
</tbody>
</table>

This example shows how to configure the link type as point to point:

```
switch# configure terminal
switch (config)# interface ethernet 1/4
switch(config-if)# spanning-tree link-type point-to-point
```

### Restarting the Protocol

An MST bridge can detect that a port is at the boundary of a region when it receives a legacy BPDU or an MST BPDU that is associated with a different region. However, the STP protocol migration cannot determine whether the legacy switch, which is a switch that runs only IEEE 802.1D, has been removed from the link unless the legacy switch is the designated switch. Enter this command to restart the protocol negotiation (force the renegotiation with neighboring switches) on the entire switch or on specified interfaces.
Procedure

<table>
<thead>
<tr>
<th>Step 1</th>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>switch# clear spanning-tree detected-protocol [interface interface [interface-num</td>
<td>port-channel]]</td>
</tr>
</tbody>
</table>

This example shows how to restart MST on the Ethernet interface on slot 2, port 8:

```
switch# clear spanning-tree detected-protocol interface ethernet 2/8
```

Verifying the MST Configuration

Use the following commands to display MST configuration information.

<table>
<thead>
<tr>
<th>Command</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td>show running-config spanning-tree [all]</td>
<td>Displays the current spanning tree configuration.</td>
</tr>
<tr>
<td>show spanning-tree mst [options]</td>
<td>Displays detailed information for the current MST configuration.</td>
</tr>
</tbody>
</table>

This example shows how to display the current MST configuration:

```
switch# show spanning-tree mst configuration
% Switch is not in mst mode
Name [mist-attempt]
Revision 1 Instances configured 2
Instance Vlans mapped
-------- ---------------------------------------------------------------
0 1-12,14-41,43-4094
1 13,42
```
CHAPTER 9

Configuring STP Extensions

This chapter contains the following sections:

- Information About STP Extensions, page 103
- Configuring STP Extensions, page 107
- Verifying the STP Extension Configuration, page 116
- Troubleshooting Loop Detection Error Message, page 116
- Generating Syslog Error Messages, page 117

Information About STP Extensions

Overview

Cisco has added extensions to Spanning Tree Protocol (STP) that make convergence more efficient. In some cases, even though similar functionality may be incorporated into the IEEE 802.1w Rapid Spanning Tree Protocol (RSTP) standard, we recommend using these extensions. All of these extensions can be used with both RPVST+ and Multiple Spanning Tree Protocol (MST).

The available extensions are spanning tree port types, Bridge Assurance, BPDU Guard, BPDU Filtering, Loop Guard, and Root Guard. Many of these features can be applied either globally or on specified interfaces.

Note

Spanning tree is used to refer to IEEE 802.1w and IEEE 802.1s. If the text is discussing the IEEE 802.1D Spanning Tree Protocol, 802.1D is stated specifically.

Understanding STP Port Types

You can configure a spanning tree port as an edge port, a network port, or a normal port. A port can be in only one of these states at a given time. The default spanning tree port type is normal. Depending on the type of device to which the interface is connected, you can configure a spanning tree port as one of these port types.
Spanning Tree Edge Ports

Edge ports, which are connected to hosts, can be either an access port or a trunk port. The edge port interface immediately transitions to the forwarding state, without moving through the blocking or learning states. (This immediate transition was previously configured as the Cisco-proprietary feature PortFast.) Interfaces that are connected to hosts should not receive STP bridge protocol data units (BPDUs).

Note
If you configure a port connected to another switch as an edge port, you might create a bridging loop.

Spanning Tree Network Ports

Network ports are connected only to switches or bridges. Configuring a port as a network port while Bridge Assurance is enabled globally, enables Bridge Assurance on that port.

Note
If you mistakenly configure ports that are connected to hosts or other edge devices as spanning tree network ports, those ports automatically move into the blocking state.

Spanning Tree Normal Ports

Normal ports can be connected to either hosts, switches, or bridges. These ports function as normal spanning tree ports.
The default spanning tree interface is a normal port.

Understanding Bridge Assurance

You can use Bridge Assurance to protect against certain problems that can cause bridging loops in the network. Specifically, you use Bridge Assurance to protect against a unidirectional link failure and a device that continues to forward data traffic when it is no longer running the spanning tree algorithm.

Note
Bridge Assurance is supported only by Rapid PVST+ and MST. Legacy 802.1D spanning tree does not support Bridge Assurance.

Bridge Assurance is enabled by default and can only be disabled globally. Also, Bridge Assurance can be enabled only on spanning tree network ports that are point-to-point links. Finally, both ends of the link must have Bridge Assurance enabled.

With Bridge Assurance enabled, BPDUs are sent out on all operational network ports, including alternate and backup ports, for each hello time period. If the port does not receive a BPDU for a specified period, the port moves into the blocking state and is not used in the root port calculation. Once that port receives a BPDU, it resumes the normal spanning tree transitions.
Understanding BPDU Guard

Enabling BPDU Guard shuts down that interface if a BPDU is received.

You can configure BPDU Guard at the interface level. When configured at the interface level, BPDU Guard shuts the port down as soon as the port receives a BPDU, regardless of the port type configuration.

When you configure BPDU Guard globally, it is effective only on operational spanning tree edge ports. In a valid configuration, LAN edge interfaces do not receive BPDUs. A BPDU that is received by an edge LAN interface signals an invalid configuration, such as the connection of an unauthorized host or switch. BPDU Guard, when enabled globally, shuts down all spanning tree edge ports when they receive a BPDU.

Note
On the edge trunk interface level, if the remote side of the disabled VLAN is configured as an access port then the BPDUs will be ignored.

BPDU Guard provides a secure response to invalid configurations, because you must manually put the LAN interface back in service after an invalid configuration.

Note
When enabled globally, BPDU Guard applies to all operational spanning tree edge interfaces.

Understanding BPDU Filtering

You can use BPDU Filtering to prevent the switch from sending or even receiving BPDUs on specified ports.

When configured globally, BPDU Filtering applies to all operational spanning tree edge ports. You should connect edge ports only to hosts, which typically drop BPDUs. If an operational spanning tree edge port receives a BPDU, it immediately returns to a normal spanning tree port type and moves through the regular transitions. In that case, BPDU Filtering is disabled on this port, and spanning tree resumes sending BPDUs on this port.

In addition, you can configure BPDU Filtering by the individual interface. When you explicitly configure BPDU Filtering on a port, that port does not send any BPDUs and drops all BPDUs that it receives. You can effectively override the global BPDU Filtering setting on individual ports by configuring the specific interface. This BPDU Filtering command on the interface applies to the entire interface, whether the interface is trunking or not.

Caution
Use care when configuring BPDU Filtering per interface. If you explicitly configuring BPDU Filtering on a port that is not connected to a host, it can result in bridging loops because the port ignores any BPDU that it receives and goes to forwarding.

If the port configuration is not set to default BPDU Filtering, the edge configuration does not affect BPDU Filtering. The following table lists all the BPDU Filtering combinations.
Table 7: BPDU Filtering Configurations

<table>
<thead>
<tr>
<th>BPDU Filtering Per Port Configuration</th>
<th>BPDU Filtering Global Configuration</th>
<th>STP Edge Port Configuration</th>
<th>BPDU Filtering State</th>
</tr>
</thead>
<tbody>
<tr>
<td>Default</td>
<td>Enabled</td>
<td>Enabled</td>
<td>Enabled/The port transmits at least 10 BPDUs. If this port receives any BPDUs, the port returns to the spanning tree normal port state and BPDU Filtering is disabled.</td>
</tr>
<tr>
<td>Default</td>
<td>Enabled</td>
<td>Disabled</td>
<td>Disabled</td>
</tr>
<tr>
<td>Default</td>
<td>Disabled</td>
<td>Enabled/Disabled</td>
<td>Disabled</td>
</tr>
<tr>
<td>Disable</td>
<td>Enabled/Disabled</td>
<td>Enabled/Disabled</td>
<td>Disabled</td>
</tr>
<tr>
<td>Enabled</td>
<td>Enabled/Disabled</td>
<td>Enabled/Disabled</td>
<td>Enabled Caution/BPDUs are never sent and if received, they do not trigger the regular STP behavior - use with caution.</td>
</tr>
</tbody>
</table>

Understanding Loop Guard

Loop Guard protects networks from loops that are caused by the following:

- Network interfaces that malfunction
- Busy CPUs
- Anything that prevents the normal forwarding of BPDUs

An STP loop occurs when a blocking port in a redundant topology erroneously transitions to the forwarding state. This transition usually happens because one of the ports in a physically redundant topology (not necessarily the blocking port) stops receiving BPDUs.

Loop Guard is useful only in switched networks where devices are connected by point-to-point links. On a point-to-point link, a designated bridge cannot disappear unless it sends an inferior BPU or brings the link down.
Loop Guard can be enabled only on network and normal spanning tree port types.

You can use Loop Guard to determine if a root port or an alternate/backup root port receives BPDUs. If the port does not receive BPDUs, Loop Guard puts the port into an inconsistent state (blocking) until the port starts to receive BPDUs again. A port in the inconsistent state does not transmit BPDUs. If the port receives BPDUs again, the protocol removes its loop-inconsistent condition, and the STP determines the port state because such recovery is automatic.

Loop Guard isolates the failure and allows STP to converge to a stable topology without the failed link or bridge. Disabling Loop Guard moves all loop-inconsistent ports to the listening state.

You can enable Loop Guard on a per-port basis. When you enable Loop Guard on a port, it is automatically applied to all of the active instances or VLANs to which that port belongs. When you disable Loop Guard, it is disabled for the specified ports.

Understanding Root Guard

When you enable Root Guard on a port, Root Guard does not allow that port to become a root port. If a received BPDU triggers an STP convergence that makes that designated port become a root port, that port is put into a root-inconsistent (blocked) state. After the port stops sending superior BPDUs, the port is unblocked again. Through STP, the port moves to the forwarding state. Recovery is automatic.

Root Guard enabled on an interface applies this functionality to all VLANs to which that interface belongs.

You can use Root Guard to enforce the root bridge placement in the network. Root Guard ensures that the port on which Root Guard is enabled is the designated port. Normally, root bridge ports are all designated ports, unless two or more of the ports of the root bridge are connected. If the bridge receives superior BPDUs on a Root Guard-enabled port, the bridge moves this port to a root-inconsistent STP state. In this way, Root Guard enforces the position of the root bridge.

You cannot configure Root Guard globally.

You can enable Root Guard on all spanning tree port types: normal, edge, and network ports.

Configuring STP Extensions

STP Extensions Configuration Guidelines

When configuring STP extensions, follow these guidelines:

- Configure all access and trunk ports connected to hosts as edge ports.
- Bridge Assurance runs only on point-to-point spanning tree network ports. You must configure each side of the link for this feature.
- Loop Guard does not run on spanning tree edge ports.
- Enabling Loop Guard on ports that are not connected to a point-to-point link will not work.
• You cannot enable Loop Guard if Root Guard is enabled.

### Configuring Spanning Tree Port Types Globally

The spanning tree port type designation depends on the type of device the port is connected to, as follows:

- **Edge**—Edge ports are connected to hosts and can be either an access port or a trunk port.
- **Network**—Network ports are connected only to switches or bridges.
- **Normal**—Normal ports are neither edge ports nor network ports; they are normal spanning tree ports. These ports can be connected to any type of device.

You can configure the port type either globally or per interface. By default, the spanning tree port type is normal.

#### Before You Begin

Ensure that STP is configured.

Ensure that you are configuring the ports correctly for the type of device to which the interface is connected.

#### Procedure

<table>
<thead>
<tr>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Step 1</strong></td>
<td>switch# configure terminal</td>
</tr>
<tr>
<td></td>
<td>Enters global configuration mode.</td>
</tr>
<tr>
<td><strong>Step 2</strong></td>
<td>switch(config)# spanning-tree port type edge default</td>
</tr>
<tr>
<td></td>
<td>Configures all interfaces as edge ports. Using this command assumes all ports are connected to hosts/servers. Edge ports immediately transition to the forwarding state without passing through the blocking or learning state at linkup. By default, spanning tree ports are normal port types.</td>
</tr>
<tr>
<td><strong>Step 3</strong></td>
<td>switch(config)# spanning-tree port type network default</td>
</tr>
<tr>
<td></td>
<td>Configures all interfaces as spanning tree network ports. Using this command assumes all ports are connected to switches and bridges. If you enable Bridge Assurance, it automatically runs on network ports. By default, spanning tree ports are normal port types.</td>
</tr>
</tbody>
</table>

**Note**

If you configure interfaces connected to hosts as network ports, those ports automatically move into the blocking state.

This example shows how to configure all access and trunk ports connected to hosts as spanning tree edge ports:

```
switch# configure terminal
switch(config)# spanning-tree port type edge default
```
This example shows how to configure all ports connected to switches or bridges as spanning tree network ports:

```
switch# configure terminal
switch(config)# spanning-tree port type network default
```

### Configuring Spanning Tree Edge Ports on Specified Interfaces

You can configure spanning tree edge ports on specified interfaces. Interfaces configured as spanning tree edge ports immediately transition to the forwarding state without passing through the blocking or learning states on linkup.

This command has four states:

- **spanning-tree port type edge**—This command explicitly enables edge behavior on the access port.
- **spanning-tree port type edge trunk**—This command explicitly enables edge behavior on the trunk port.

```
Note
If you enter the `spanning-tree port type edge trunk` command, the port is configured as an edge port even in the access mode.
```

- **spanning-tree port type normal**—This command explicitly configures the port as a normal spanning tree port and the immediate transition to the forwarding state is not enabled.
- **no spanning-tree port type**—This command implicitly enables edge behavior if you define the `spanning-tree port type edge default` command in global configuration mode. If you do not configure the edge ports globally, the `no spanning-tree port type` command is equivalent to the `spanning-tree port type disable` command.

### Before You Begin

Ensure that STP is configured.

Ensure that the interface is connected to hosts.

### Procedure

<table>
<thead>
<tr>
<th>Step</th>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td>Step 1</td>
<td>switch# configure terminal</td>
<td>Enters global configuration mode.</td>
</tr>
<tr>
<td>Step 2</td>
<td>switch(config)# interface type slot/port</td>
<td>Specifies the interface to configure, and enters the interface configuration mode.</td>
</tr>
<tr>
<td>Step 3</td>
<td>switch(config-if)# spanning-tree port type edge</td>
<td>Configures the specified access interfaces to be spanning edge ports. Edge ports immediately transition to the forwarding state without passing through the blocking or learning state at linkup. By default, spanning tree ports are normal port types.</td>
</tr>
</tbody>
</table>
This example shows how to configure the Ethernet access interface 1/4 to be a spanning tree edge port:

```
switch# configure terminal
switch(config)# interface ethernet 1/4
switch(config-if)# spanning-tree port type edge
```

### Configuring Spanning Tree Network Ports on Specified Interfaces

You can configure spanning tree network ports on specified interfaces.

Bridge Assurance runs only on spanning tree network ports.

This command has three states:

- **spanning-tree port type network**—This command explicitly configures the port as a network port. If you enable Bridge Assurance globally, it automatically runs on a spanning tree network port.

- **spanning-tree port type normal**—This command explicitly configures the port as a normal spanning tree port and Bridge Assurance cannot run on this interface.

- **no spanning-tree port type**—This command implicitly enables the port as a spanning tree network port if you define the `spanning-tree port type network default` command in global configuration mode. If you enable Bridge Assurance globally, it automatically runs on this port.

**Note**

A port connected to a host that is configured as a network port automatically moves into the blocking state.

**Before You Begin**

Ensure that STP is configured.

Ensure that the interface is connected to switches or routers.

**Procedure**

<table>
<thead>
<tr>
<th>Step</th>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>switch# configure terminal</td>
<td>Enters global configuration mode.</td>
</tr>
<tr>
<td>2</td>
<td>switch(config)# interface type slot/port</td>
<td>Specifies the interface to configure, and enters the interface configuration mode. The interface can be a physical Ethernet port.</td>
</tr>
<tr>
<td>3</td>
<td>switch(config-if)# spanning-tree port type network</td>
<td>Configures the specified interfaces to be spanning network ports. If you enable Bridge Assurance, it automatically runs on network ports. By default, spanning tree ports are normal port types.</td>
</tr>
</tbody>
</table>
This example shows how to configure the Ethernet interface 1/4 to be a spanning tree network port:

```
switch# configure terminal
switch(config)# interface ethernet 1/4
switch(config-if)# spanning-tree port type network
```

### Enabling BPDU Guard Globally

You can enable BPDU Guard globally by default. In this condition, the system shuts down an edge port that receives a BPDU.

**Note**

We recommend that you enable BPDU Guard on all edge ports.

**Before You Begin**

Ensure that STP is configured.

Ensure that you have configured some spanning tree edge ports.

**Procedure**

<table>
<thead>
<tr>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Step 1</strong></td>
<td>switch# configure terminal</td>
</tr>
<tr>
<td><strong>Step 2</strong></td>
<td>switch(config)# spanning-tree port type edge bpdu guard default</td>
</tr>
</tbody>
</table>

This example shows how to enable BPDU Guard on all spanning tree edge ports:

```
switch# configure terminal
switch(config)# spanning-tree port type edge bpdu guard default
```

### Enabling BPDU Guard on Specified Interfaces

You can enable BPDU Guard on specified interfaces. Enabling BPDU Guard shuts down the port if it receives a BPDU.

You can configure BPDU Guard on specified interfaces as follows:

- **spanning-tree bpdu guard enable**—Unconditionally enables BPDU Guard on the interface.
- **spanning-tree bpdu guard disable**—Unconditionally disables BPDU Guard on the interface.
- **no spanning-tree bpdu guard**—Enables BPDU Guard on the interface if it is an operational edge port and if the `spanning-tree port type edge bpdu guard default` command is configured.
Before You Begin

Ensure that STP is configured.

Procedure

<table>
<thead>
<tr>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Step 1</strong> switch# configure terminal</td>
<td>Enters global configuration mode.</td>
</tr>
<tr>
<td><strong>Step 2</strong> switch(config)# interface type slot/port</td>
<td>Specifies the interface to configure, and enters the interface configuration mode.</td>
</tr>
<tr>
<td><strong>Step 3</strong> switch(config-if)# spanning-tree bpduguard {enable</td>
<td>disable}</td>
</tr>
<tr>
<td><strong>Step 4</strong> switch(config-if)# no spanning-tree bpduguard</td>
<td>(Optional) Disables BPDU Guard on the interface.</td>
</tr>
<tr>
<td>Note</td>
<td>Enables BPDU Guard on the interface if it is an operational edge port and if you enter the spanning-tree port type edge bpduguard default command.</td>
</tr>
</tbody>
</table>

This example shows how to explicitly enable BPDU Guard on the Ethernet edge port 1/4:

```bash
switch# configure terminal
switch (config)# interface ethernet 1/4
switch(config-if)# spanning-tree bpduguard enable
switch(config-if)# no spanning-tree bpduguard
```

Enabling BPDU Filtering Globally

You can enable BPDU Filtering globally by default on spanning tree edge ports.

If an edge port with BPDU Filtering enabled receives a BPDU, it loses its operation status and as edge port and resumes the regular STP transitions. However, this port maintains it configuration as an edge port.

⚠️ Caution

Be careful when using this command: using it incorrectly can cause bridging loops.

خفف من الاستخدام غير المقصود: يسبب عدم التحليق في حالة استخدامه بشكل خاطئ.

Note

When enabled globally, BPDU Filtering is applied only on ports that are operational edge ports. Ports send a few BPDUs at linkup before they effectively filter outbound BPDUs. If a BPDU is received on an edge port, it immediately loses its operational edge port status and BPDU Filtering is disabled.

⚠️ Caution

Be careful when using this command: using it incorrectly can cause bridging loops.

Caution

When enabled globally, BPDU Filtering is applied only on ports that are operational edge ports. Ports send a few BPDUs at linkup before they effectively filter outbound BPDUs. If a BPDU is received on an edge port, it immediately loses its operational edge port status and BPDU Filtering is disabled.

Before You Begin

Ensure that STP is configured.

Ensure that you have configured some spanning tree edge ports.
Enabling BPDU Filtering on Specified Interfaces

You can apply BPDU Filtering to specified interfaces. When enabled on an interface, that interface does not send any BPDUs and drops all BPDUs that it receives. This BPDU Filtering functionality applies to the entire interface, whether trunking or not.

Be careful when you enter the `spanning-tree bpdufilter enable` command on specified interfaces. Explicitly configuring BPDU Filtering on a port that is not connected to a host can result in bridging loops because the port ignores any BPDU it receives and goes to forwarding.

You can enter this command to override the port configuration on specified interfaces.

This command has three states:

- `spanning-tree bpdufilter enable`—Unconditionally enables BPDU Filtering on the interface.
- `spanning-tree bpdufilter disable`—Unconditionally disables BPDU Filtering on the interface.
- `no spanning-tree bpdufilter`—Enables BPDU Filtering on the interface if the interface is an operational edge port and if you configure the `spanning-tree port type edge bpdufilter default` command.

When you enable BPDU Filtering locally on a port, this feature prevents the device from receiving or sending BPDUs on this port.

**Before You Begin**

Ensure that STP is configured.
**Enabling Loop Guard Globally**

You can enable Loop Guard globally by default on all point-to-point spanning tree normal and network ports. Loop Guard does not run on edge ports.

Loop Guard provides additional security in the bridge network. Loop Guard prevents alternate or root ports from becoming the designated port because of a failure that could lead to a unidirectional link.

**Procedure**

<table>
<thead>
<tr>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td>Step 1 switch# configure terminal</td>
<td>Enters global configuration mode.</td>
</tr>
</tbody>
</table>

**Before You Begin**

Ensure that STP is configured.

Ensure that you have spanning tree normal ports or have configured some network ports.

**Procedure**

<table>
<thead>
<tr>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td>Step 1 switch# configure terminal</td>
<td>Enters global configuration mode.</td>
</tr>
</tbody>
</table>
Enabling Loop Guard or Root Guard on Specified Interfaces

You can enable either Loop Guard or Root Guard on specified interfaces. Enabling Root Guard on a port means that port cannot become a root port, and LoopGuard prevents alternate or root ports from becoming the designated port because of a failure that could lead to a unidirectional link. Both Loop Guard and Root Guard enabled on an interface apply to all VLANs to which that interface belongs.

Note
Entering the Loop Guard command for the specified interface overrides the global Loop Guard command.

Before You Begin
Ensure that STP is configured.
Ensure that you are configuring Loop Guard on spanning tree normal or network ports.

Procedure

<table>
<thead>
<tr>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Step 1</strong></td>
<td>switch# configure terminal</td>
</tr>
<tr>
<td><strong>Step 2</strong></td>
<td>switch(config)# interface type slot/port</td>
</tr>
<tr>
<td><strong>Step 3</strong></td>
<td>switch(config-if)# spanning-tree guard {loop</td>
</tr>
</tbody>
</table>

Note
Loop Guard runs only on spanning tree normal and network interfaces.
This example shows how to enable Root Guard on Ethernet port 1/4:

```
switch# configure terminal
switch (config)# interface ethernet 1/4
switch(config-if)# spanning-tree guard root
```

## Verifying the STP Extension Configuration

Use the following commands to display the configuration information for the STP extensions.

<table>
<thead>
<tr>
<th>Command</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>show running-config spanning-tree [all]</code></td>
<td>Displays the current status of spanning tree on the switch.</td>
</tr>
<tr>
<td><code>show spanning-tree [options]</code></td>
<td>Displays selected detailed information for the current spanning tree configuration.</td>
</tr>
</tbody>
</table>

## Troubleshooting Loop Detection Error Message

This section describes how to address the FWM-2-STM_LOOP_DETECT error message in the log of Cisco Nexus 3000 Series switches.

If the Cisco Nexus 3000 Series switch displays the following message, it indicates that the switch receives frames with the same source MAC address on these two interfaces and that the switch learns the same MAC address on these interfaces at a very high speed. The switch detects this condition as a loop. The switch disables the MAC address learning in order to protect its control plane. It is implemented on all VLANs even if the loop occurs on only one VLAN.

```
2016 Apr 11 18:00:18 N3k-4-3229 %FWM-2-STM_LOOP_DETECT: Loops detected in the network for mac 0000.0602.0602 among ports Eth1/48 and Eth1/50/3 on vlan 4 - Disabling dynamic learning notifications for a period between 120 and 240 seconds on vlan 4
```

The possible causes of the error message are:

- MAC addresses move because of incorrect Spanning Tree Protocol (STP)-port state convergence.
- MAC addresses move because the source of the data is physically moved across all switches while STP states are converged and in correct states.

### Detecting the Loop

The Forwarding Manager (FWM) has a mechanism to count the number of moved MAC addresses and weigh them based on the number of times the MAC address moves. It determines the total count (switch-wide across all VLANs, MACs, and interfaces) of the moved MAC addresses, declares the %FWM-2-STM_LOOP_DETECT condition, and disables learning to protect FWM in loop conditions.

### Note

MAC learning is disabled per VLAN and not per system.
The logic for MAC-move notifications should be noted. It is possible to notify MAC-moves when the MAC-address-table notification for MAC-moves is enabled. This adds notification logs on the console but no action is taken. A move is declared when a given MAC address has moved three times back and forth across a given pair of ports on a VLAN within an aging scan period of 10 seconds.

Note

The MAC address has to be detected 50 times on each of the 2 ports.

You can enable MAC-move notification on the switch to find out which MAC addresses move.

**Procedure**

<table>
<thead>
<tr>
<th>Step</th>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td>Step 1</td>
<td>switch# conf t</td>
<td>Enters the configuration mode.</td>
</tr>
<tr>
<td>Step 2</td>
<td>switch# mac address-table notification mac-move</td>
<td>Enables MAC-move notification on the switch.</td>
</tr>
</tbody>
</table>

**Generating Syslog Error Messages**

It is not always sufficient to enable the MAC-move notification in order to generate a syslog message about MAC-move notification. In order to ensure syslog message generation, enter these commands in conjunction with the previous command: `mac address-table notification mac-move`.

**Procedure**

<table>
<thead>
<tr>
<th>Step</th>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td>Step 1</td>
<td>conf t</td>
<td>Enters the configuration mode.</td>
</tr>
<tr>
<td>Step 2</td>
<td>logging level spanning-tree 6</td>
<td>Enables logging of all spanning-tree events from level 6 up to the highest severity events.</td>
</tr>
<tr>
<td>Step 3</td>
<td>logging level fwm 6</td>
<td>Enables logging of all FWM events from level 6 up to the highest severity events.</td>
</tr>
<tr>
<td>Step 4</td>
<td>logging monitor 6</td>
<td>Enables the device to log messages to the monitor based on severity level 6 or higher.</td>
</tr>
</tbody>
</table>

The addition of these commands ensures that the syslog for FWM detect displays when there is a MAC address move. In order to verify the STP port state across VLANs on the switches, enter the following commands.

```
switch# show spanning-tree
switch# show spanning-tree vlan <id>
switch# show spanning-tree internal interaction
```

In order to check if the MAC addresses move, enter the command:

```
# show mac address-table notification mac-move
MAC Move Notify Triggers: 1206
```
Number of MAC Addresses added: 944088
Number of MAC Addresses moved: 265
Number of MAC Addresses removed: 943920

The MAC address moves are also logged with a minimum logging level of six that is required to display which MAC addresses move:

2016 Jun 12 16:05:31.564 switch %FWM-6-MAC_MOVE_NOTIFICATION:
Host 0000.0000.fe00 in vlan 85 is flapping between
port Eth104/1/8 and port Eth104/1/9

What to Do Next

Check for correct STP convergence and for STP port-states across all the switches in the topology. Confirm that there are no disputes or incorrect port states.

If the source of the data frames that are physically moving is identified, control the source in order to halt the rapid and continuous moves.

By default, dynamic learning is reenabled after 180 seconds. At that point, any STP disputes or inconsistencies should be resolved. If not, the dynamic learning is disabled again.
CHAPTER 10

Configuring LLDP

This chapter contains the following sections:

- Information About Global LLDP Commands, page 119
- Configuring LLDP, page 120
- Information About LLDP Management TLV IP Addresses, page 121
- Configuring LLDP Management TLV IP Addresses on an Interface, page 123
- Configuring Interface LLDP, page 124
- MIBs for LLDP, page 125

Information About Global LLDP Commands

You can set global LLDP settings. These settings include the length of time before discarding LLDP information received from peers, the length of time to wait before performing LLDP initialization on any interface, the rate at which LLDP packets are sent, the port description, system capabilities, system description, and system name.

LLDP supports a set of attributes that it uses to discover neighbor devices. These attributes contain type, length, and value descriptions and are referred to as TLVs. LLDP supported devices can use TLVs to receive and send information to their neighbors. Details such as configuration information, device capabilities, and device identity can be advertised using this protocol.

The switch supports the following required management LLDP TLVs:

- Data Center Ethernet Parameter Exchange (DCBXP) TLV
- Management address TLV
- Port description TLV
- Port VLAN ID TLV (IEEE 802.1 organizationally specific TLVs)
- System capabilities TLV
- System description TLV
- System name TLV
The Data Center Bridging Exchange Protocol (DCBXP) is an extension of LLDP. It is used to announce, exchange, and negotiate node parameters between peers. DCBXP parameters are packaged into a specific DCBXP TLV. This TLV is designed to provide an acknowledgment to the received LLDP packet.

DCBXP is enabled by default, if you enable LLDP. When LLDP is enabled, DCBXP can be enabled or disabled using the `no lldp tlv-select dcbxp` command. DCBXP is disabled on ports where LLDP transmit or receive is disabled.

## Configuring LLDP

### Before You Begin

Ensure that the Link Layer Discovery Protocol (LLDP) feature is enabled on the switch.

### Procedure

<table>
<thead>
<tr>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Step 1</strong> switch# configure terminal</td>
<td>Enters global configuration mode.</td>
</tr>
<tr>
<td><strong>Step 2</strong> switch(config)# lldp {holdtime seconds</td>
<td>reinit seconds</td>
</tr>
<tr>
<td>Command or Action</td>
<td>Purpose</td>
</tr>
<tr>
<td>-------------------</td>
<td>---------</td>
</tr>
<tr>
<td>Use the <code>system-description</code> option to specify the system description TLV messages. Use the <code>system-name</code> option to specify the system name TLV messages.</td>
<td></td>
</tr>
</tbody>
</table>

**Step 3**

```
switch(config)# no lldp
{holdtime | reinit | timer}
```

Resets the LLDP values to their defaults.

**Step 4**

(Optional) `switch# show lldp`

Displays LLDP configurations.

This example shows how to configure the global LLDP hold time to 200 seconds:

```
switch# configure terminal
switch(config)# lldp holdtime 200
```

This example shows how to enable LLDP to send or receive the management address TLVs:

```
switch# configure terminal
switch(config)# lldp tlv-select management-address
```

This example shows how to enable LLDP to send or receive IPv4 management address TLVs:

```
switch# configure terminal
switch(config)# lldp tlv-select management-address v4
```

This example shows how to enable LLDP to send or receive IPv6 management address TLVs:

```
switch# configure terminal
switch(config)# lldp tlv-select management-address v6
```

---

**Information About LLDP Management TLV IP Addresses**

You can use the LLDP management TLV to convey the system information of network devices to their neighbors. The LLDP management TLV includes the management address, which can be used by remote managers to obtain information related to the local device. Currently, by default, the IPv4 and IPv6 address of the management port mgmt0 are sent in the management TLV.

Cisco NX-OS Release 6.0(2)IU4(1) introduces support for two TLVs, IPv4 and IPv6.

You can explicitly specify the management IPv4 or IPv6 address to be sent in the LLDP management TLV. This address can be one of the following:

- IPv4 or IPv6 address of a port
- IPv4 or IPv6 address of a VLAN (SVI)

The following rules are applied while selecting a management address to be sent in the LLDP management TLV for IPv4:

- If the LLDP management v4 TLV is configured to be sent, and if the LLDP management IPv4 address of a port is configured, the LLDP management IPv4 address configured on the port is used in the management TLV of the LLDP protocol data unit (PDU) to be sent.
• If the LLDP management v4 TLV is configured to be sent, and if the LLDP VLAN is configured:
  • If the VLAN ID is specified and the SVI on it is operationally enabled, the SVI IPv4 address of the VLAN ID is used in the management v4 TLV of the LLDP PDU to be sent.
  • If the native VLAN is available and the SVI on it is operationally enabled, the SVI IPv4 address of the native VLAN is used in the management v4 TLV of the LLDP PDU to be sent.

• If the LLDP management v4 TLV is configured to be sent and if neither the LLDP management IPv4 address nor the LLDP VLAN is configured, the IPv4 address of the management port mgmt0 is used in the management v4 TLV of the LLDP PDU to be sent.

• If the LLDP management v4 TLV has no IPv4 address configured, the interface port's MAC address is sent in one TLV.

• If the LLDP management v4 TLV is not configured to be sent, no management TLV IPv4 address is sent.

The following rules are applied while selecting a management address to be sent in the LLDP management TLV for IPv6:

• If the LLDP management v6 TLV is configured to be sent, and if the LLDP management IPv6 address of a port is configured, the LLDP management IPv6 configured on the port is used in the management TLV of the LLDP protocol data unit (PDU) to be sent.

• If the LLDP management v6 TLV is configured to be sent, and if the LLDP VLAN is configured:
  • If the VLAN ID is specified and the SVI on it is operationally enabled, the SVI IPv6 address of the VLAN ID is used in the management v6 TLV of the LLDP PDU to be sent.
  • If the native VLAN is available and the SVI on it is operationally enabled, the SVI IPv6 address of the native VLAN is used in the management v6 TLV of the LLDP PDU to be sent.

• If the LLDP management v6 TLV is configured to be sent and if neither the LLDP management IPv6 address nor the LLDP VLAN is configured, the IPv6 address of the management port mgmt0 is used in the management v6 TLV of the LLDP PDU to be sent.

• If the LLDP management v6 TLV has no IPv6 address configured, the interface port's MAC address is sent in one TLV.

• If the LLDP management v6 TLV is not configured to be sent, no management TLV IPv6 address is sent.

The following are the TLV selection processes that are followed based on the IPv4 or IPv6 address configured:

• No IP address configured—The interface port's MAC address is sent in one TLV.

• Only IPv4 address configured—Two TLVs are sent. One has the IPv4 address and the other has the interface port's MAC address. This process follows the rules applied while selecting a management address to be sent in the LLDP management TLV for IPv4.

• Only IPv6 address configured—The IPv6 address is sent in one TLV. This process follows the rules applied while selecting a management address to be sent in the LLDP management TLV for IPv6.

• Both IPv4 and IPv6 addresses configured—Two TLVs are sent. One has the IPv4 address and the other has the IPv6 address. This process follows the rules applied while selecting a management address to be sent in the LLDP management TLV for IPv4 and for IPv6.
If you configure both TLVs and no IPv4 address is configured, no interface port's MAC address is sent in the v4 TLV. Only one TLV is sent.

If only one TLV is sent that has the interface port's MAC address, this address is displayed in both the IPv4 and IPv6 address column of the peer.

Configuring LLDP Management TLV IP Addresses on an Interface

Before You Begin

Ensure that the LLDP management TLV option is configured.

Procedure

<table>
<thead>
<tr>
<th>Step</th>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td>Step 1</td>
<td>switch# configure terminal</td>
<td>Enters global configuration mode.</td>
</tr>
<tr>
<td>Step 2</td>
<td>switch(config)# interface ethernet slot/port</td>
<td>Specifies an interface to configure and enters interface configuration mode.</td>
</tr>
<tr>
<td>Step 3</td>
<td>switch(config-if)# [no] lldp tlv-set { management-address ip-address [ipv6]</td>
<td>vlan [vlan-id] }</td>
</tr>
</tbody>
</table>

This example shows how to specify the management IPv4 address in the management TLV:

```
switch# configure terminal
switch(config)# interface ethernet 1/8
switch(config-if)# lldp tlv-set management-address 1.1.1.20
```

This example shows how to specify the management IPv6 address in the management TLV:

```
switch# configure terminal
switch(config)# interface ethernet 1/8
switch(config-if)# lldp tlv-set management-address 0dc3:0dc3:0000:0000:0218:baff:fed8:239d ipv6
```

This example shows how to specify the VLAN ID in the management TLV:

```
switch# configure terminal
switch(config)# interface ethernet 1/8
switch(config-if)# lldp tlv-set vlan 10
```
Configuring Interface LLDP

Procedure

<table>
<thead>
<tr>
<th>Step</th>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td>Step 1</td>
<td><code>switch# configure terminal</code></td>
<td>Enters global configuration mode.</td>
</tr>
<tr>
<td>Step 2</td>
<td><code>switch(config)# interface type slot/port</code></td>
<td>Selects the interface to change.</td>
</tr>
<tr>
<td>Step 3</td>
<td>`switch(config-if)# [no] lldp {receive</td>
<td>transmit}`</td>
</tr>
<tr>
<td>Step 4</td>
<td>`switch# show lldp {interface</td>
<td>neighbors</td>
</tr>
</tbody>
</table>

This example shows how to set an interface to transmit LLDP packets:

```
switch# configure terminal
switch(config)# interface ethernet 1/2
switch(config-if)# lldp transmit
```

This example shows how to configure an interface to disable LLDP:

```
switch# configure terminal
switch(config)# interface ethernet 1/2
switch(config-if)# no lldp transmit
switch(config-if)# no lldp receive
```

This example shows how to display LLDP interface information:

```
switch# show lldp interface ethernet 1/2
tx_enabled: TRUE
rx_enabled: TRUE
dcbx_enabled: TRUE
Port MAC address: 00:0d:ec:a3:5f:48
Remote Peers Information
No remote peers exist
```

This example shows how to display LLDP neighbor information:

```
switch# show lldp neighbors
Capability codes:
    (R) Router, (B) Bridge, (T) Telephone, (C) DOCSIS Cable Device
    (W) WLAN Access Point, (P) Repeater, (S) Station, (O) Other
Device ID    Local Intf   Hold-time Capability   Port ID
BLR-VPC2-QS8 Eth1/25   120    BR     Ethernet1/25
BLR-VPC2-QS8 Eth1/26   120    BR     Ethernet1/26
BLR-VPC2-QS8 Eth1/27   120    BR     Ethernet1/27
BLR-VPC2-QS8 Eth1/28   120    BR     Ethernet1/28
Total entries displayed: 4
```

This example shows how to display the interface details about LLDP neighbors:

```
switch(config-if)# show lldp neighbor interface ethernet 1/4 detail
Capability codes:
    (R) Router, (B) Bridge, (T) Telephone, (C) DOCSIS Cable Device
```

This example shows how to display the interface details about LLDP neighbors:
This example shows how to display the system details about LLDP neighbors:

```
switch(config-if)#
```
Information About MAC Addresses

To switch frames between LAN ports, the switch maintains an address table. When the switch receives a frame, it associates the media access control (MAC) address of the sending network device with the LAN port on which it was received.

The switch dynamically builds the address table by using the MAC source address of the frames received. When the switch receives a frame for a MAC destination address not listed in its address table, it floods the frame to all LAN ports of the same VLAN except the port that received the frame. When the destination station replies, the switch adds its relevant MAC source address and port ID to the address table. The switch then forwards subsequent frames to a single LAN port without flooding all LAN ports.

You can also enter a MAC address, which is termed a static MAC address, into the table. These static MAC entries are retained across a reboot of the switch.

In addition, you can enter a non-IP multicast address as a statically configured MAC address. A non-IP multicast address can accept more than one interface as its destination.

The address table can store a number of unicast and non-IP multicast address entries without flooding any frames. The switch uses an aging mechanism, defined by a configurable aging timer, so if an address remains inactive for a specified number of seconds, it is removed from the address table.
Guidelines for Configuring the MAC Address Tables

See the following guidelines and limitations for configuring the MAC address tables:

- Beginning with Cisco NX-OS Release 7.0(3)I7(2), Cisco Nexus 3500 Series switches support configuring MAC move loop detection.

- Starting with Release 7.0(3)I2(1), the aging of the mac-address is not incrementing in the output of the `show mac address-table` CLI command. Therefore, the proper age of the mac-address cannot be determined.

- Starting with Release 7.0(3)I2(1), the `show mac address-table` CLI command does not display the multicast MAC entries. Use the `show mac address-table multicast` CLI command to check the Layer 2 entries.

MAC Address Movement

You can detect and limit the number of times that a MAC address moves from one port to another. This movement of MAC addresses between ports can cause loops. Until Cisco NX-OS Release 6.0(2)U3(1), when a loop was detected between two ports, MAC learning was disabled for 180 seconds. You can now configure the action of bringing down the port with the lower interface index when such a loop is detected by using the `mac address-table loop-detect port-down` command. To revert to the default action of disabling MAC learning, use the `no` form of this command.

Configuring MAC Addresses

Configuring Static MAC Addresses

You can configure static MAC addresses for the switch. These addresses can be configured in interface configuration mode or in VLAN configuration mode.

**Procedure**

<table>
<thead>
<tr>
<th>Step</th>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td>Step 1</td>
<td><code>switch# configure terminal</code></td>
<td>Enters global configuration mode.</td>
</tr>
<tr>
<td>Step 2</td>
<td>`switch(config)# mac-address-table static mac_address vlan vlan-id [drop</td>
<td>interface {type slot/port}</td>
</tr>
<tr>
<td>Step 3</td>
<td><code>switch(config)# no mac address-table static mac_address vlan vlan-id</code></td>
<td>(Optional) Deletes the static entry from the MAC address table.</td>
</tr>
</tbody>
</table>
This example shows how to put a static entry in the MAC address table:

```
switch# configure terminal
switch(config) # mac address-table static 12ab.47dd.ff89 vlan 3 interface ethernet 1/4
switch(config) #
```

### Configuring the Aging Time for the MAC Table

You can configure the amount of time that an entry (the packet source MAC address and port that packet ingresses) remains in the MAC table. MAC aging time can be configured in either interface configuration mode or in VLAN configuration mode.

#### Note

If the Cisco Nexus device is used as a Layer 2 or Layer 3 termination switch, Cisco recommends that you set the `mac-address-table aging-time` to 1800 (higher than the default ARP aging time of 1500 seconds) on all VLANs.

The Cisco Nexus 3000 series switches do not support per-VLAN CAM aging timers.

#### Procedure

<table>
<thead>
<tr>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Step 1</strong></td>
<td>switch# configure terminal</td>
</tr>
</tbody>
</table>

**Step 2**

```
switch(config)# mac-address-table aging-time seconds
```

Specifies the time before an entry ages out and is discarded from the MAC address table.

**Note**

Starting with Release 7.0(3)I2(1), the aging of the mac-address is not incrementing in the output of the `show mac address-table` CLI command. Therefore, the proper age of the mac-address cannot be determined.

The `seconds` range is from 0 to 1000000. The default is 300 seconds for Cisco NX-OS 5500 and 1800 for Cisco NX-OS 5600 and 6000 series. Entering the value 0 disables the MAC aging.

This example shows how to set the aging time for entries in the MAC address table to 300 seconds:

```
switch# configure terminal
switch(config)# mac-address-table aging-time 300
switch(config)# show mac address-table
```

Legend:

- * - primary entry, G - Gateway MAC, (R) - Routed MAC, O - Overlay MAC
- age - seconds since last seen, + - primary entry using vPC Peer-Link,
- (T) - True, (F) - False

<table>
<thead>
<tr>
<th>VLAN</th>
<th>MAC Address</th>
<th>Type</th>
<th>age</th>
<th>Secure NTFY Ports</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
Configuring MAC Move Loop Detection

When the number of MAC address moves between two ports exceeds a threshold, it forms a loop. Until Cisco NX-OS Release 6.0(2)U3(1), when a loop was detected between two ports, MAC learning was disabled for 180 seconds. You can now configure the action of bringing down the port with the lower interface index when such a loop is detected by using the `mac address-table loop-detect port-down` command. To revert to the default action of disabling MAC learning, use the `no` form of this command.

**Procedure**

<table>
<thead>
<tr>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td>Step 1: <code>switch# configure terminal</code></td>
<td>Enters global configuration mode.</td>
</tr>
<tr>
<td>Step 2: <code>switch(config)# [no] mac address-table loop-detect port-down</code></td>
<td>Specifies the port-down action for MAC move loop detection.</td>
</tr>
<tr>
<td></td>
<td>The <code>no</code> form of this command reverts to the default action of disabling MAC learning for 180 seconds.</td>
</tr>
<tr>
<td>Step 3: <code>switch(config)# mac address-table loop-detect port-down edge-port</code></td>
<td>Enables the err-disabled detection for the edge-port on the MAC move loop detection.</td>
</tr>
</tbody>
</table>

Disabling MAC Address Learning on Layer 2 Interfaces

You can now disable and re-enable MAC address learning on Layer 2 interfaces.

**Procedure**

<table>
<thead>
<tr>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td>Step 1: <code>switch# configure terminal</code></td>
<td>Enters global configuration mode.</td>
</tr>
<tr>
<td>Step 2: <code>switch(config)# interface type slot/port</code></td>
<td>Enters the interface configuration mode for the specified interface.</td>
</tr>
<tr>
<td>Step 3: <code>switch(config-if)# [no] switchport mac-learn disable</code></td>
<td>Disables MAC address learning on Layer 2 interfaces.</td>
</tr>
<tr>
<td></td>
<td>The <code>no</code> form of this command re-enables MAC address learning on Layer 2 interfaces.</td>
</tr>
<tr>
<td>Step 4: <code>switch(config-if)# clear mac address-table dynamic interface type slot/port</code></td>
<td>Clears the MAC address table for the specified interface.</td>
</tr>
</tbody>
</table>
### Configuring MAC Address Tables

#### Clearing Dynamic Addresses from the MAC Table

You can clear all dynamic entries in the MAC address table.

<table>
<thead>
<tr>
<th>Command</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td>`switch(config)# clear mac-address-table dynamic {address mac-addr} {interface [type slot/port</td>
<td>port-channel number] {vlan vlan-id}}`</td>
</tr>
</tbody>
</table>

This example shows how to clear the dynamic entries in the MAC address table:

```
switch# clear mac-address-table dynamic
```

#### Verifying the MAC Address Configuration

**Note**

On Cisco Nexus 3000 and Cisco Nexus 3548 Series platforms, the self router MAC or HSRP VMAC are dynamically learned by the switch under the following condition:

- When there is a transient loop in the network due to which the switch receives its own packets.

This behavior is different from other Cisco Nexus platforms. However, there is no operational impact due to these self MAC entries that are present in the MAC table. Any packet that is destined to the router MAC or HSRP MAC is routed. There is no Layer 2 lookup on these packets.

Use one of the following commands to verify the configuration:
Table 8: MAC Address Configuration Verification Commands

<table>
<thead>
<tr>
<th>Command</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>show mac-address-table aging-time</code></td>
<td>Displays the MAC address aging time for all VLANs defined in the switch.</td>
</tr>
<tr>
<td><code>show mac-address-table</code></td>
<td>Displays the contents of the MAC address table.</td>
</tr>
<tr>
<td><code>show mac address-table loop-detect</code></td>
<td>Displays the currently configured action.</td>
</tr>
</tbody>
</table>

This example shows how to display the MAC address table:

```bash
switch# show mac-address-table
VLAN   MAC Address     Type   Age  Port
------------------+---------------------+-------+---------+-------------------------------
 1   0018.b967.3cd0   dynamic 10  Eth1/3
 1   001c.b05a.5380   dynamic 200 Eth1/3
Total MAC Addresses: 2
```

This example shows how to display the current aging time:

```bash
switch# show mac-address-table aging-time
Vlan  Aging Time
-----  ----------
 1    300
 13   300
 42   300
```

This example shows how to display the currently configured action:

```bash
switch# configure terminal
switch(config)# show mac address-table loop-detect
Port Down Action Mac Loop Detect : enabled
```

```bash
switch# configure terminal
switch(config)# no mac address-table loop-detect port-down
switch(config)# show mac address-table loop-detect
Port Down Action Mac Loop Detect : disabled
```

## Triggering the Layer 2 Consistency Checker

You can manually trigger the Layer 2 consistency checker to compare the hardware and software configuration of MAC addresses and display the results. It displays MAC addresses that are configured in the software, but not configured in the hardware, as well as MAC addresses that are configured in the hardware, but not in the software. To manually trigger the Layer 2 consistency checker and display the results, use the following command in any mode:
Procedure

<table>
<thead>
<tr>
<th>Step 1</th>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>switch# show consistency-checker l2</td>
<td>Starts a Layer 2 consistency check on MAC addresses and displays the results.</td>
</tr>
</tbody>
</table>

This example shows how to trigger a Layer 2 consistency check and display the results:

```
switch# show consistency-checker l2
Consistency Check: FAILED
Legend:   * - primary entry, G - Gateway MAC, (R) - Routed MAC       age - seconds since last seen

Missing entries in the MAC Table
  VLAN | MAC Address | Type      | age | Secure | NTFY | Ports
  -----+------------+-----------+-----+--------+-----+---------
     1 | 0100.0100.0106 | dynamic   |    | F      | F   | 0
     1 | 0200.0100.0125 | static    |    | F      | F   | 0

Extra and Discrepant entries in the MAC Table
  VLAN | MAC Address | Type      | age | Secure | NTFY | Ports
  -----+------------+-----------+-----+--------+-----+---------
      *| 0000.0100.0109 | dynamic   | 370 | F      | F   | Eth1/41
```
Triggering the Layer 2 Consistency Checker
CHAPTER 12

Configuring IGMP Snooping

This chapter contains the following sections:

- Information About IGMP Snooping, page 135
- Configuring IGMP Snooping Parameters, page 138
- Verifying the IGMP Snooping Configuration, page 140

Information About IGMP Snooping

The IGMP snooping software examines IGMP protocol messages within a VLAN to discover which interfaces are connected to hosts or other devices interested in receiving this traffic. Using the interface information, IGMP snooping can reduce bandwidth consumption in a multiaccess LAN environment to avoid flooding the entire VLAN. The IGMP snooping feature tracks which ports are attached to multicast-capable routers to help it manage the forwarding of IGMP membership reports. The IGMP snooping software responds to topology change notifications.

Note

IGMP snooping is supported on all Ethernet interfaces. The term snooping is used because Layer 3 control plane packets are intercepted and influence Layer 2 forwarding decisions.

Cisco NX-OS supports IGMPv2 and IGMPv3. IGMPv2 supports IGMPv1, and IGMPv3 supports IGMPv2. Although not all features of an earlier version of IGMP are supported, the features related to membership query and membership report messages are supported for all IGMP versions.
The following figure shows an IGMP snooping switch that is located between the host and the IGMP router. The IGMP snooping switch snoops the IGMP membership reports and leave messages and forwards them only when necessary to the connected IGMP routers.

**Figure 17: IGMP Snooping Switch**

![IGMP Snooping Switch Diagram]

**Note**

The switch supports IGMPv3 snooping based only on the destination multicast MAC address. It does not support snooping based on the source MAC address or on proxy reports.

The Cisco NX-OS IGMP snooping software supports optimized multicast flooding (OMF) that forwards unknown traffic to routers only and performs no data driven state creation. For more information about IGMP snooping, see [http://tools.ietf.org/wg/magma/draft-ietf-magma-snoop/rfc4541.txt](http://tools.ietf.org/wg/magma/draft-ietf-magma-snoop/rfc4541.txt).

### IGMPv1 and IGMPv2

Both IGMPv1 and IGMPv2 support membership report suppression, which means that if two hosts on the same subnet want to receive multicast data for the same group, the host that receives a member report from the other host suppresses sending its report. Membership report suppression occurs for hosts that share a port.

If no more than one host is attached to each VLAN switch port, you can configure the fast leave feature in IGMPv2. The fast leave feature does not send last member query messages to hosts. As soon as the software receives an IGMP leave message, the software stops forwarding multicast data to that port.

IGMPv1 does not provide an explicit IGMP leave message, so the software must rely on the membership message timeout to indicate that no hosts remain that want to receive multicast data for a particular group.

**Note**

Cisco NX-OS ignores the configuration of the last member query interval when you enable the fast leave feature because it does not check for remaining hosts.
IGMPv3

The IGMPv3 snooping implementation on the switch forwards IGMPv3 reports to allow the upstream multicast router to do source-based filtering.

By default, the software tracks hosts on each VLAN port. The explicit tracking feature provides a fast leave mechanism. Because every IGMPv3 host sends membership reports, a report suppression feature limits the amount of traffic the switch sends to other multicast-capable routers. When report suppression is enabled, and no IGMPv1 or IGMPv2 hosts request the same group, the software provides proxy reporting. The proxy feature builds the group state from membership reports from the downstream hosts and generates membership reports in response to queries from upstream queriers.

Even though the IGMPv3 membership reports provide a full accounting of group members on a LAN segment, when the last host leaves, the software sends a membership query. You can configure the parameter last member query interval. If no host responds before the timeout, the software removes the group state.

IGMP Snooping Querier

When there is no multicast router in the VLAN to originate the queries, you must configure an IGMP snooping querier to send membership queries.

When an IGMP snooping querier is enabled, it sends out periodic IGMP queries that trigger IGMP report messages from hosts that want to receive IP multicast traffic. IGMP snooping listens to these IGMP reports to establish appropriate forwarding.

Currently, you can configure the same SVI IP address for the switch querier and the IGMP snooping querier. Both queriers will then be active at the same time, and both queriers will send general queries to the VLAN periodically. To prevent this from happening, ensure that you use different IP addresses for the IGMP snooping querier and the switch querier.

IGMP Forwarding

The control plane of the Cisco Nexus device is able to detect IP addresses but forwarding occurs using the MAC address only.

When a host connected to the switch wants to join an IP multicast group, it sends an unsolicited IGMP join message, specifying the IP multicast group to join. Alternatively, when the switch receives a general query from a connected router, it forwards the query to all interfaces, physical and virtual, in the VLAN. Hosts wanting to join the multicast group respond by sending a join message to the switch. The switch CPU creates a multicast forwarding table entry for the group if it is not already present. The CPU also adds the interface where the join message was received to the forwarding table entry. The host associated with that interface receives multicast traffic for that multicast group.

The router sends periodic multicast general queries and the switch forwards these queries through all ports in the VLAN. Interested hosts respond to the queries. If at least one host in the VLAN wants to receive multicast traffic, the router continues forwarding the multicast traffic to the VLAN. The switch forwards multicast group traffic to only those hosts listed in the forwarding table for that multicast group.

When hosts want to leave a multicast group, they can either silently leave, or they can send a leave message. When the switch receives a leave message from a host, it sends a group-specific query to determine if any other devices connected to that interface are interested in traffic for the specific multicast group. The switch then updates the forwarding table for that MAC group so that only those hosts interested in receiving multicast traffic continue to receive traffic for that group.
traffic for the group are listed in the forwarding table. If the router receives no reports from a VLAN, it removes
the group for the VLAN from its IGMP cache.

## Configuring IGMP Snooping Parameters

To manage the operation of the IGMP snooping process, you can configure the optional IGMP snooping
parameters described in the following table.

**Table 9: IGMP Snooping Parameters**

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>IGMP snooping</td>
<td>Enables IGMP snooping on a per-VLAN basis. The default is enabled.</td>
</tr>
<tr>
<td></td>
<td><strong>Note</strong> If the global setting is disabled, all VLANs are treated as disabled, whether they are enabled or not.</td>
</tr>
<tr>
<td>Explicit tracking</td>
<td>Tracks IGMPv3 membership reports from individual hosts for each port on a per-VLAN basis. The default is enabled.</td>
</tr>
<tr>
<td>Fast leave</td>
<td>Enables the software to remove the group state when it receives an IGMP Leave report without sending an IGMP query message. This parameter is used for IGMPv2 hosts when no more than one host is present on each VLAN port. The default is disabled.</td>
</tr>
<tr>
<td>Last member query interval</td>
<td>Sets the interval that the software waits after sending an IGMP query to verify that no hosts that want to receive a particular multicast group remain on a network segment. If no hosts respond before the last member query interval expires, the software removes the group from the associated VLAN port. Values range from 1 to 25 seconds. The default is 1 second.</td>
</tr>
<tr>
<td>Snooping querier</td>
<td>Configures a snooping querier on an interface when there is no multicast router in the VLAN to generate queries. The default is disabled.</td>
</tr>
<tr>
<td>Report suppression</td>
<td>Limits the membership report traffic sent to multicast-capable routers. When you disable report suppression, all IGMP reports are sent as is to multicast-capable routers. The default is enabled.</td>
</tr>
<tr>
<td>Multicast router</td>
<td>Configures a static connection to a multicast router. The interface to the router must be in the selected VLAN.</td>
</tr>
</tbody>
</table>
### Configuring IGMP Snooping Parameters

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Multicast router vpc-peer-link</td>
<td>Configures a static connection to a virtual port channel (vPC) peer link. By default, the vPC peer link is considered a multicast router port and the multicast packet is sent to the peer link for each receiver VLAN. To send the multicast traffic over a vPC peer link to each receiver VLAN that has orphan ports, use the <code>no ip igmp snooping mrouter vpc-peer-link</code> command. If you use the <code>no ip igmp snooping mrouter vpc-peer-link</code> command, the multicast traffic is not sent over to a peer link for the source VLAN and receiver VLAN unless there is an orphan port in the VLAN. The IGMP snooping mrouter VPC peer link should also be globally disabled on the peer VPC switch.</td>
</tr>
<tr>
<td>Static group</td>
<td>Configures an interface that belongs to a VLAN as a static member of a multicast group.</td>
</tr>
</tbody>
</table>

You can disable IGMP snooping either globally or for a specific VLAN.

**Procedure**

<table>
<thead>
<tr>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Step 1</strong> switch# configure terminal</td>
<td>Enters global configuration mode.</td>
</tr>
<tr>
<td><strong>Step 2</strong> switch(config)# ip igmp snooping</td>
<td>Globally enables IGMP snooping. The default is enabled.</td>
</tr>
<tr>
<td><strong>Note</strong></td>
<td>If the global setting is disabled, all VLANs are treated as disabled, whether they are enabled or not.</td>
</tr>
<tr>
<td><strong>Step 3</strong> switch(config)# vlan configuration vlan-id</td>
<td>Enters VLAN configuration mode.</td>
</tr>
<tr>
<td><strong>Step 4</strong> switch(config-vlan)# ip igmp snooping</td>
<td>Enables IGMP snooping for the current VLAN. The default is enabled.</td>
</tr>
<tr>
<td><strong>Note</strong></td>
<td>If IGMP snooping is enabled globally, this command is not required.</td>
</tr>
<tr>
<td><strong>Step 5</strong> switch(config-vlan)# ip igmp snooping explicit-tracking</td>
<td>Tracks IGMPv3 membership reports from individual hosts for each port on a per-VLAN basis. The default is enabled on all VLANs.</td>
</tr>
</tbody>
</table>
| **Step 6** switch(config-vlan)# ip igmp snooping fast-leave | Supports IGMPv2 hosts that cannot be explicitly tracked because of the host report suppression mechanism of the IGMPv2 protocol. When you enable fast leave, the IGMP }
Verifying the IGMP Snooping Configuration

Use the following commands to verify the IGMP snooping configuration.

<table>
<thead>
<tr>
<th>Command</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>show ip igmp snooping [vlan vlan-id]</td>
<td>Displays IGMP snooping configuration by VLAN.</td>
</tr>
<tr>
<td>show ip igmp snooping groups [vlan vlan-id] [detail]</td>
<td>Displays IGMP snooping information about groups by VLAN.</td>
</tr>
</tbody>
</table>

This example shows how to configure IGMP snooping parameters for a VLAN:

```
switch# configure terminal
switch(config)# vlan configuration 5
switch(config-vlan)# ip igmp snooping last-member-query-interval 3
switch(config-vlan)# ip igmp snooping querier 172.20.52.106
switch(config-vlan)# ip igmp snooping explicit-tracking
switch(config-vlan)# ip igmp snooping fast-leave
switch(config-vlan)# ip igmp snooping report-suppression
switch(config-vlan)# ip igmp snooping mrouter interface ethernet 1/10
switch(config-vlan)# ip igmp snooping static-group 230.0.0.1 interface ethernet 1/10
switch(config-vlan)# end
```
<table>
<thead>
<tr>
<th>Command</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>show ip igmp snooping querier [vlan] vlan-id</code></td>
<td>Displays IGMP snooping queriers by VLAN.</td>
</tr>
<tr>
<td><code>show ip igmp snooping mrouter [vlan] vlan-id</code></td>
<td>Displays multicast router ports by VLAN.</td>
</tr>
<tr>
<td><code>show ip igmp snooping explicit-tracking vlan vlan-id</code></td>
<td>Displays IGMP snooping explicit tracking information by VLAN.</td>
</tr>
</tbody>
</table>

This example shows how to verify the IGMP snooping parameters:

```
switch# show ip igmp snooping
Global IGMP Snooping Information:
  IGMP Snooping enabled
IGMP Snooping information for vlan 1
  IGMP snooping enabled
  IGMP querier none
  Switch-querier disabled
  Explicit tracking enabled
  Fast leave disabled
  Report suppression enabled
  Router port detection using PIM Hellos, IGMP Queries
  Number of router-ports: 0
  Number of groups: 0
IGMP Snooping information for vlan 5
  IGMP snooping enabled
  IGMP querier present, address: 192.0.2.1, version: 3
  Querier interval: 125 secs
  Querier last member query interval: 10 secs
  Querier robustness: 2
  Switch-querier enabled, address 192.0.2.1, currently running
  Explicit tracking enabled
  Fast leave enabled
  Report suppression enabled
  Router port detection using PIM Hellos, IGMP Queries
  Number of router-ports: 1
  Number of groups: 1
```
CHAPTER 13

Configuring Traffic Storm Control

This chapter contains the following sections:

- Information About Traffic Storm Control, page 143
- Guidelines and Limitations for Traffic Storm Control, page 145
- Default Settings for Traffic Storm Control, page 146
- Configuring Traffic Storm Control, page 146
- Traffic Storm Control Example Configuration, page 148

Information About Traffic Storm Control

A traffic storm occurs when packets flood the LAN, creating excessive traffic and degrading network performance. You can use the traffic storm control feature to prevent disruptions on Ethernet interfaces by a broadcast, multicast, or unknown unicast traffic storm.

Traffic storm control (also called traffic suppression) allows you to monitor the levels of the incoming broadcast, multicast, or unknown unicast traffic over a 10-microsecond interval. During this interval, the traffic level, which is a percentage of the total available bandwidth of the port, is compared with the traffic storm control level that you configured. When the ingress traffic reaches the traffic storm control level that is configured on the port, traffic storm control drops the traffic until the interval ends.
The following figure shows the broadcast traffic patterns on an Ethernet interface during a specified time interval. In this example, traffic storm control occurs between times T1 and T2 and between T4 and T5. During those intervals, the amount of broadcast traffic exceeded the configured threshold.

![Figure 18: Broadcast Suppression](image)

The traffic storm control threshold numbers and the time interval allow the traffic storm control algorithm to work with different levels of packet granularity. For example, a higher threshold allows more packets to pass through.

Traffic storm control is implemented in the hardware. The traffic storm control circuitry monitors packets that pass from an Ethernet interface to the switching bus. Using the Individual/Group bit in the packet destination address, the circuitry determines if the packet is unicast or broadcast, tracks the current count of packets within the 10-microsecond interval, and filters out subsequent packets when a threshold is reached.

Traffic storm control uses a bandwidth-based method to measure traffic. You set the percentage of total available bandwidth that the controlled traffic can use. Because packets do not arrive at uniform intervals, the 10-microsecond interval can affect the operation of traffic storm control.

The following are examples of how traffic storm control operation is affected:

- If you enable broadcast traffic storm control, and broadcast traffic exceeds the level within the 10-microsecond interval, traffic storm control drops all exceeding broadcast traffic until the end of the interval.

- If you enable multicast traffic storm control, and the multicast traffic exceeds the level within the 10-microsecond interval, traffic storm control drops all exceeding multicast traffic until the end of the interval.

- If you enable broadcast and multicast traffic storm control, and broadcast traffic exceeds the level within the 10-microsecond interval, traffic storm control drops all exceeding broadcast traffic until the end of the interval.

- If you enable broadcast and multicast traffic storm control, and multicast traffic exceeds the level within the 10-microsecond interval, traffic storm control drops all exceeding multicast traffic until the end of the interval.

You can configure traffic storm control to perform the following optional corrective actions when traffic exceeds the configured level:

- Shut down—When ingress traffic exceeds the traffic storm control level that is configured on a port, traffic storm control puts the port into the error-disabled state. To reenable this port, you can use either
the shutdown and no shutdown options on the configured interface, or the error-disable detection and recovery feature. You are recommended to use the errdisable recovery cause storm-control command for error-disable detection and recovery along with the errdisable recovery interval command for defining the recovery interval. The interval can range between 30 and 65535 seconds.

- Trap—You can configure traffic storm control to generate an SNMP trap when ingress traffic exceeds the configured traffic storm control level. The SNMP trap action is enabled by default. However, storm control traps are not rate-limited by default. You can control the number of traps generated per minute by using the snmp-server enable traps storm-control trap-rate command.

By default, Cisco NX-OS takes no corrective action when traffic exceeds the configured level.

Guidelines and Limitations for Traffic Storm Control

When configuring the traffic storm control level, follow these guidelines and limitations:

- You can configure traffic storm control on a port-channel interface.
- Specify the level as a percentage of the total interface bandwidth:
  - The level can be from 0 to 100.
  - The optional fraction of a level can be from 0 to 99.
  - 100 percent means no traffic storm control.
  - 0.0 percent suppresses all traffic.
- There are local link and hardware limitations that prevent storm-control drops from being counted separately. Instead, storm-control drops are counted with other drops in the discards counter.
- Because of hardware limitations and the method by which packets of different sizes are counted, the level percentage is an approximation. Depending on the sizes of the frames that make up the incoming traffic, the actual enforced level might differ from the configured level by several percentage points.
- Due to a hardware limitation, the output for the show interface counters storm-control command does not show ARP suppressions when storm control is configured and the interface is actually suppressing ARP broadcast traffic. This limitation can lead to the configured action not being triggered, but the incoming ARP broadcast being correctly storm suppressed.
- Due to a hardware limitation, the packet drop counter cannot distinguish between packet drops caused by a traffic storm and certain other discarded input frames. This limitation can lead to the configured action being triggered even in the absence of a traffic storm.
- Storm control is only for ingress traffic, specifically for unknown unicast, unknown multicast, and broadcast traffic.
- The link-level control protocols (LACP, LLDP and so on) are not affected in case of a traffic storm. The storm control is applied to data plane traffic only.
- The burst size values are:
  - For a 10G port, 48.68 Mbytes/390Mbits
  - For a 1G port, 25 Mbytes/200Mbits
Default Settings for Traffic Storm Control

The following table lists the default settings for traffic storm control parameters.

**Table 10: Default Traffic Storm Control Parameters**

<table>
<thead>
<tr>
<th>Parameters</th>
<th>Default</th>
</tr>
</thead>
<tbody>
<tr>
<td>Traffic storm control</td>
<td>Disabled</td>
</tr>
<tr>
<td>Threshold percentage</td>
<td>100</td>
</tr>
</tbody>
</table>

Configuring Traffic Storm Control

You can set the percentage of total available bandwidth that the controlled traffic can use.

**Note**
Traffic storm control uses a 10-microsecond interval that can affect the operation of traffic storm control.

**Procedure**

<table>
<thead>
<tr>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td>Step 1 switch# configure terminal</td>
<td>Enters global configuration mode.</td>
</tr>
<tr>
<td>Step 2 switch(config)# interface {ethernet slot/port</td>
<td>Enters interface configuration mode.</td>
</tr>
<tr>
<td></td>
<td>port-channel number}</td>
</tr>
<tr>
<td>Step 3 switch(config-if)# [no] storm-control [broadcast</td>
<td>Configures traffic storm control for traffic on the interface.</td>
</tr>
<tr>
<td></td>
<td>multicast</td>
</tr>
<tr>
<td></td>
<td>{shutdown</td>
</tr>
<tr>
<td></td>
<td>• shutdown—Shuts down the port or puts it in the error-disable state.</td>
</tr>
<tr>
<td></td>
<td>• trap—Generates an SNMP trap.</td>
</tr>
<tr>
<td></td>
<td>The no form of this command removes the configured level or action. Although shutdown and trap actions can be specified independent of each other, the no form of storm-control action removes both actions.</td>
</tr>
</tbody>
</table>

This example shows how to configure traffic storm control for port channels 122 and 123:

```
switch# configure terminal
switch(config)# interface port-channel 122, port-channel 123
```
Verifying the Traffic Storm Control Configuration

Use the following commands to display traffic storm control configuration information:

<table>
<thead>
<tr>
<th>Command</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td>show interface [ethernet slot/port</td>
<td>port-channel number] counters storm-control</td>
</tr>
<tr>
<td>show running-config interface</td>
<td>Displays the traffic storm control configuration.</td>
</tr>
</tbody>
</table>

Note

When a storm event occurs and either a shutdown or a trap is triggered, a syslog message is generated.

This example shows how to display the storm control configuration:

switch(config-if-range)# show interface port-channel 122 counters storm-control

[Action] S - Shut (Err Disable), T - Trap
-------------------------------------------------------------------------------------------------------------------------
Port        UcastSupp %  McastSupp %  BcastSupp %  TotalSuppDiscards  Action
-------------------------------------------------------------------------------------------------------------------------
Po122       100.00       100.00       100.00       0          [-T]

This example shows how to display the running configuration of an interface with its storm control configuration:

switch(config-if-range)# show running-config interface

interface Ethernet1/15
description IXIA
switchport mode trunk
spanning-tree port type edge trunk
spanning-tree bpdufilter enable
storm-control broadcast level 5.23
storm-control multicast level 0.50
storm-control unicast level 1.23
storm-control action shutdown
storm-control action trap
Traffic Storm Control Example Configuration

This example shows how to configure traffic storm control:

```
switch# configure terminal
switch(config)# interface ethernet 1/4
switch(config-if)# storm-control action trap
switch(config-if)# storm-control broadcast level 40
switch(config-if)# storm-control multicast level 40
switch(config-if)# storm-control unicast level 40
```

This example shows how to specify the number of Storm Control traps per minute:

```
switch# configure terminal
switch(config)# snmp-server enable traps storm-control trap-rate 100
switch(config)#
```
INDEX

802.1Q VLANs 46
  configuring 46

A
  access VLANs 41
  understanding 41
  aging time, configuring 129
  MAC table 129

B
  blocking state, STP 63
  BPDU guard 105
  bridge ID 54
  broadcast storms 143
  broadcast traffic 7
  VLANs 7

C
  changed information 1
    description 1
  CIST regional root 82
  CIST root 83
  community ports 20
  community VLANs 20, 21
  config t 29, 33
  configuring 12
  VLANs 12
  Configuring LLDP management TLV IP address 123
  cut-through mode 51
    licensing 51
  cut-through switching 49, 51, 52
    default settings 51
    enabling 52
    information about 49

D
  default settings 51
    cut-through switching 51
    store-and-forward switching 51
    switching modes 51
  disabling 52
    store-and-forward switching 52

E
  enabling 51
    store-and-forward switching 51
  extended system IDs 8
    VLANs 8

G
  guidelines and limitations 50
    store-and-forward switching 50

H
  host ports 20
    kinds of 20

I
  ICMPv2 136
  IEEE 802.1w 79
  IGMP 138
    snooping parameters, configuring 138
  IGMP forwarding 137
  IGMP snooping 137
    queries 137
  IGMPv1 136
  IGMPv3 137
interface 29, 33
isolated port 20
isolated VLANs 20, 21

L
LAN interface 43
  Ethernet access port 43
Layer 2 switching 3
  Ethernet switching 3
licensing 51
  cut-through mode 51
  store-and-forward mode 51
  switching modes 51
Link Failure 66, 84
  detecting unidirectional 66
LLDP 125
  MIBs 125
loop-detect port-down 130
loop-detect, MAC table 130
loop-detection, MAC table 130

M
MAC address configuration 131
  verifying 131
MAC addresses 128
  static, configuring 128
MAC table 129
  aging time, configuring 129
MIBs 125
  LLDP 125
MST 82, 90
  CIST regional root 82
  setting to default values 90
MSTP 79, 80, 81, 82, 83, 84, 90
  boundary ports 84
    described 84
  CIST regional root 82
  CIST root 83
  CIST, described 81
  CST 81, 82
    defined 81
    operations between regions 82
  IEEE 802.1s 82
    terminology 82
  IST 81, 82
    operations within a region 81
  mapping VLANs to MST instance 90
  MST region 79, 80, 81, 83
    CIST 81
MSTP (continued)
  MST region (continued)
    described 79
    hop-count mechanism 83
    supported spanning-tree instances 80
multicast storms 143
multicast traffic 7
  VLANs 7

N
native 802.1Q VLANs 46
  configuring 46
new information 1
  description 1

P
PortFast BPDU filtering 105
  ports 13
    adding to VLANs 13
  primary VLANs 20
  private VLANs 20, 21, 23
    community VLANs 20, 21
    end station access to 23
    isolated VLANs 20, 21
    ports 20
      community 20
      isolated 20
      promiscuous 20
  primary VLANs 20
  secondary VLANs 20
  promiscuous ports 20
rapid PVST priority 73
Rapid PVST+ 68
  configuring 68
rapid PVST+ configurations 76
  verifying 76
Rapid Spanning Tree Protocol 79
  reduced MAC address 54
  reenabling 52
  cut-through switching 52
  root guard 107
RSTP 58, 61, 65, 79
  active topology 61
  BPDU 65
  processing 65
RSTP (continued)
designated port, defined 61
designated switch, defined 61
proposal-agreement handshake process 58
rapid convergence 58
   point-to-point links 58
   root ports 58
root port, defined 61

S
secondary VLANs 20
show interface switchport 32, 34
snooping parameters, configuring 138
   IGMP 138
static MAC addresses, configuring 128
store-and-forward mode 51
   licensing 51
store-and-forward switching 49, 50, 51
   default settings 51
   enabling 51
   guidelines and limitations 50
   information about 49
STP 58, 63, 64, 103, 104
   edge ports 58, 104
   network ports 104
   normal ports 104
   port types 103
   PortFast 58, 104
   understanding 63, 64
      Blocking State 63
      disabled state 64
      forwarding state 63
      learning state 63
STP bridge ID 54
STP overview 53
STP root guard 107
SVIs 7, 14, 15
   management 15
   routed 14
   VLAN interfaces 7
switching mode 51
   licensing 51
switchport 29, 33
switchport mode private-vlan trunk allowed vlan 34
switchport mode private-vlan trunk promiscuous 33
switchport mode private-vlan trunk secondary 29
switchport private-vlan association trunk 31
switchport private-vlan trunk allowed vlan 29
switchport private-vlan trunk native vlan 29, 33

T
traffic storms 145
   control 145

U
understanding 41
   access VLANs 41
unicast storms 143

V
verifying 17, 76
   rapid PVST+ configurations 76
   VLAN configurations 17
VLAN configurations 17
   verifying 17
VLAN interfaces 7
   communicating between VLANs 7
VLAN numbers 8
   allowed numbers 8
   reserved range 8
VLAN ranges 8
   description 8
VLAN traffic 7
   and routing 7
   distribution 7
VLANs 7, 8, 12, 13, 14, 15
   adding ports to 13
   configuring 12
   configuring as management SVIs 15
   configuring as routed SVIs 14
   description 7
   extended system ID 8
   reserved ranges 8
   SVIs 7
   usable VLANs 8
VTP 7
   mode 7