



Layer 2 Configuration Guide, Cisco IOS XE 17.13.x (Catalyst 9600 Switches)

First Published: 2023-12-08

Americas Headquarters

Cisco Systems, Inc.
170 West Tasman Drive
San Jose, CA 95134-1706
USA
<http://www.cisco.com>
Tel: 408 526-4000
800 553-NETS (6387)
Fax: 408 527-0883

THE SPECIFICATIONS AND INFORMATION REGARDING THE PRODUCTS IN THIS MANUAL ARE SUBJECT TO CHANGE WITHOUT NOTICE. ALL STATEMENTS, INFORMATION, AND RECOMMENDATIONS IN THIS MANUAL ARE BELIEVED TO BE ACCURATE BUT ARE PRESENTED WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED. USERS MUST TAKE FULL RESPONSIBILITY FOR THEIR APPLICATION OF ANY PRODUCTS.

THE SOFTWARE LICENSE AND LIMITED WARRANTY FOR THE ACCOMPANYING PRODUCT ARE SET FORTH IN THE INFORMATION PACKET THAT SHIPPED WITH THE PRODUCT AND ARE INCORPORATED HEREIN BY THIS REFERENCE. IF YOU ARE UNABLE TO LOCATE THE SOFTWARE LICENSE OR LIMITED WARRANTY, CONTACT YOUR CISCO REPRESENTATIVE FOR A COPY.

The Cisco implementation of TCP header compression is an adaptation of a program developed by the University of California, Berkeley (UCB) as part of UCB's public domain version of the UNIX operating system. All rights reserved. Copyright © 1981, Regents of the University of California.

NOTWITHSTANDING ANY OTHER WARRANTY HEREIN, ALL DOCUMENT FILES AND SOFTWARE OF THESE SUPPLIERS ARE PROVIDED "AS IS" WITH ALL FAULTS. CISCO AND THE ABOVE-NAMED SUPPLIERS DISCLAIM ALL WARRANTIES, EXPRESSED OR IMPLIED, INCLUDING, WITHOUT LIMITATION, THOSE OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NON-INFRINGEMENT OR ARISING FROM A COURSE OF DEALING, USAGE, OR TRADE PRACTICE.

IN NO EVENT SHALL CISCO OR ITS SUPPLIERS BE LIABLE FOR ANY INDIRECT, SPECIAL, CONSEQUENTIAL, OR INCIDENTAL DAMAGES, INCLUDING, WITHOUT LIMITATION, LOST PROFITS OR LOSS OR DAMAGE TO DATA ARISING OUT OF THE USE OR INABILITY TO USE THIS MANUAL, EVEN IF CISCO OR ITS SUPPLIERS HAVE BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES.

Any Internet Protocol (IP) addresses and phone numbers used in this document are not intended to be actual addresses and phone numbers. Any examples, command display output, network topology diagrams, and other figures included in the document are shown for illustrative purposes only. Any use of actual IP addresses or phone numbers in illustrative content is unintentional and coincidental.

All printed copies and duplicate soft copies of this document are considered uncontrolled. See the current online version for the latest version.

Cisco has more than 200 offices worldwide. Addresses and phone numbers are listed on the Cisco website at www.cisco.com/go/offices.

Cisco and the Cisco logo are trademarks or registered trademarks of Cisco and/or its affiliates in the U.S. and other countries. To view a list of Cisco trademarks, go to this URL: <https://www.cisco.com/c/en/us/about/legal/trademarks.html>. Third-party trademarks mentioned are the property of their respective owners. The use of the word partner does not imply a partnership relationship between Cisco and any other company. (1721R)

© 2023 Cisco Systems, Inc. All rights reserved.



CONTENTS

CHAPTER 1

Configuring Spanning Tree Protocol 1

- Restrictions for Spanning Tree Protocol 1
- Information About Spanning Tree Protocol 1
 - Spanning Tree Protocol 1
 - Spanning-Tree Topology and Bridge Protocol Data Units 2
 - Bridge ID, Device Priority, and Extended System ID 3
 - Port Priority Versus Path Cost 4
 - Spanning-Tree Interface States 4
 - How a Device or Port Becomes the Root Device or Root Port 7
 - Spanning Tree and Redundant Connectivity 7
 - Spanning-Tree Address Management 8
 - Accelerated Aging to Retain Connectivity 8
 - Spanning-Tree Modes and Protocols 8
 - Supported Spanning-Tree Instances 9
 - Spanning-Tree Interoperability and Backward Compatibility 9
 - Spanning Tree Protocols and IEEE 802.1Q Trunks 10
 - Spanning Tree and Switch Stacks 10
 - Default Spanning-Tree Configuration 11
- How to Configure Spanning Tree Protocol 12
 - Changing the Spanning-Tree Mode 12
 - (Optional) Disabling Spanning Tree 13
 - (Optional) Configuring the Root Device 13
 - (Optional) Configuring a Secondary Root Device 15
 - (Optional) Configuring Port Priority 15
 - (Optional) Configuring Path Cost 16
 - (Optional) Configuring the Device Priority of a VLAN 18

(Optional) Configuring the Hello Time	19
(Optional) Configuring the Forwarding-Delay Time for a VLAN	19
(Optional) Configuring the Maximum-Aging Time for a VLAN	20
(Optional) Configuring the Transmit Hold-Count	21
Monitoring Spanning Tree Protocol Configuration Status	21
Additional References for Spanning Tree Protocol	22
Feature History for Spanning Tree Protocol	22

CHAPTER 2**Configuring Loop Detection Guard 23**

Restrictions for Loop Detection Guard	23
Information About Loop Detection Guard	23
Interaction of Loop Detection Guard with Other Features	25
Spanning Tree Protocol and Loop Detection Guard	25
VLANs and Loop Detection Guard	25
Enabling Loop Detection Guard and Error-Disabling the Required Port	26
Additional References for Configuring Loop Detection Guard	27
Feature History for Loop Detection Guard	28

CHAPTER 3**Configuring Multiple Spanning-Tree Protocol 29**

Prerequisites for Multiple Spanning Tree Protocol	29
Restrictions for Multiple Spanning-Tree Protocol	29
Information About Multiple Spanning Tree Protocol	30
Multiple Spanning Tree Protocol Configuration	30
Multiple Spanning Tree Protocol Configuration Guidelines	30
Root Switch Configuration	31
Multiple Spanning-Tree Regions	31
Internal Spanning Tree, Common and Internal Spanning Tree, and Common Spanning Tree	32
Operations Within an Multiple Spanning Tree Region	32
Operations Between Multiple Spanning Tree Regions	33
IEEE 802.1s Terminology	33
Illustration of Multiple Spanning Tree Regions	34
Hop Count	34
Boundary Ports	35
IEEE 802.1s Implementation	35

Port Role Naming Change	35
Interoperation Between Legacy and Standard Devices	36
Detecting Unidirectional Link Failure	37
Multiple Spanning Tree Protocol and Switch Stacks	37
Interoperability with IEEE 802.1D Spanning Tree Protocol	37
Rapid Spanning Tree Protocol Overview	38
Port Roles and the Active Topology	38
Rapid Convergence	39
Synchronization of Port Roles	40
Bridge Protocol Data Unit Format and Processing	41
Topology Changes	42
Protocol Migration Process	43
Default Multiple Spanning Tree Protocol Configuration	43
How to Configure Multiple Spanning Tree Protocol and Parameters	44
Specifying the Multiple Spanning Tree Region Configuration and Enabling Multiple Spanning Tree Protocol	44
(Optional) Configuring the Root Device	45
(Optional) Configuring a Secondary Root Device	46
(Optional) Configuring Port Priority	47
(Optional) Configuring Path Cost	48
(Optional) Configuring the Device Priority	49
(Optional) Configuring the Hello Time	51
Configuring the Forwarding-Delay Time	51
Configuring the Maximum-Aging Time	52
(Optional) Configuring the Maximum-Hop Count	53
(Optional) Specifying the Link Type to Ensure Rapid Transitions	53
(Optional) Designating the Neighbor Type	54
Restarting the Protocol Migration Process	55
Feature History for Multiple Spanning Tree Protocol	56
CHAPTER 4	Configuring Optional Spanning-Tree Features
	57
Information About Optional Spanning-Tree Features	57
PortFast	57
Spanning Tree Protocol PortFast Port Types	58

Bridge Protocol Data Unit Guard	59
Bridge Protocol Data Unit Filtering	59
Bridge Assurance	60
Guidelines for Configuring Bridge Assurance	62
UplinkFast	63
Cross-Stack UplinkFast	65
How Cross-Stack UplinkFast Works	65
Events That Cause Fast Convergence	66
BackboneFast	67
EtherChannel Guard	70
Root Guard	70
Loop Guard	71
How to Configure Optional Spanning-Tree Features	72
(Optional) Enabling PortFast	72
Enabling PortFast Port Types	73
Configuring the PortFast Default State Globally	73
Configuring a PortFast Edge Port on a Specified Interface	74
Configuring a PortFast Network Port on a Specified Interface	76
(Optional) Enabling Bridge Protocol Data Unit Guard	77
(Optional) Enabling Bridge Protocol Data Unit Filtering	78
Configuring Bridge Assurance	79
(Optional) Enabling UplinkFast for Use with Redundant Links	81
(Optional) Disabling UplinkFast	82
(Optional) Enabling BackboneFast	83
(Optional) Enabling EtherChannel Guard	84
(Optional) Enabling Root Guard	85
(Optional) Enabling Loop Guard	86
Monitoring the Spanning-Tree Status	86
Additional References for Optional Spanning Tree Features	87
Feature History for Optional Spanning Tree Features	87

CHAPTER 5

Configuring EtherChannels	89
Restrictions for EtherChannels	89
Information About EtherChannels	90

EtherChannel Overview	90
Channel Groups and Port-Channel Interfaces	91
Port Aggregation Protocol	93
Port Aggregation Protocol Modes	93
Port Aggregation Protocol Learn Method and Priority	94
Port Aggregation Protocol Interaction with Other Features	95
Link Aggregation Control Protocol	95
Link Aggregation Control Protocol Modes	95
Link Aggregation Control Protocol Standalone Mode on Ethernet Channel	96
Link Aggregation Control Protocol and Link Redundancy	96
Link Aggregation Control Protocol Interaction with Other Features	97
Link Aggregation Control Protocol Interaction with Other Features 1:1 Redundancy	97
EtherChannel On Mode	97
Load-Balancing and Forwarding Methods	97
MAC Address Forwarding	98
IP Address Forwarding	98
VLAN ID based Forwarding	99
Load-Balancing Advantages	99
EtherChannel and Switch Stacks	100
Switch Stack and Port Aggregation Protocol	100
Switch Stacks and Link Aggregation Control Protocol	100
Default EtherChannel Configuration	100
EtherChannel Configuration Guidelines	101
Layer 2 EtherChannel Configuration Guidelines	102
Layer 3 EtherChannel Configuration Guidelines	102
Auto-LAG	102
Auto-LAG Configuration Guidelines	103
How to Configure EtherChannels	103
Configuring Layer 2 EtherChannels	104
Configuring Layer 3 EtherChannels	105
(Optional) Configuring EtherChannel Load-Balancing	107
(Optional) Configuring EtherChannel Extended Load-Balancing	109
(Optional) Configuring the Port Aggregation Protocol Learn Method and Priority	110
Configuring Link Aggregation Control Protocol Hot-Standby Ports	111

(Optional) Configuring the Link Aggregation Control Protocol Max Bundle	112
Configuring Link Aggregation Control Protocol Port-Channel Standalone Disable	112
Configuring Link Aggregation Control Protocol Standalone Mode on EtherChannel	113
Configuring the Link Aggregation Control Protocol Port Channel Min-Links	114
(Optional) Configuring the Link Aggregation Control Protocol System Priority	115
(Optional) Configuring the Link Aggregation Control Protocol Port Priority	116
Configuring Link Aggregation Control Protocol 1:1 Redundancy	117
Configuring Link Aggregation Control Protocol 1:1 Redundancy Fast Rate Timer	118
Configuring Auto-LAG Globally	118
Configuring Auto-LAG on a Port Interface	119
Configuring Persistence with Auto-LAG	120
Monitoring EtherChannel, Port Aggregation Protocol, and Link Aggregation Control Protocol Status	121
Configuration Examples for EtherChannels	121
Example: Configuring Layer 2 EtherChannels	121
Example: Configuring Layer 3 EtherChannels	122
Example: Configuring Link Aggregation Control Protocol Hot-Standby Ports	123
Example: Configuring Link Aggregation Control Protocol 1:1 Redundancy	123
Example: Configuring Standalone Mode on EtherChannel	123
Example: Configuring Auto LAG	124
Additional References for EtherChannels	125
Feature History for EtherChannels	125

CHAPTER 6**Configuring Precision Time Protocol (PTP) 127**

Restrictions and Limitations for Precision Time Protocol	127
Information About Precision Time Protocol	128
Ethernet Switches and Delays	128
Message-Based Synchronisation	128
Precision Time Protocol Version 2 Message Types	130
Precision Time Protocol and Software Defined Architecture Overlay	130
Precision Time Protocol Event Message Sequences	131
End-to-End Delay Mechanism	131
Peer-to-Peer Delay Mechanism	132
Synchronizing the Local Clock	133
Best Master Clock Algorithm	134

PTP Ports	134
Precision Time Protocol Clocks	135
Precision Time Protocol Profiles	137
Default Profile	137
Precision Time Protocol on an EtherChannel Interface	137
AES67 Media Profile	138
G8275.1 Telecom Profile	140
PTP Clocks Supported	140
PTP Messages Supported	141
How to Configure Precision Time Protocol	141
Configuring Precision Time Protocol Default Profile	141
Configuring Precision Time Protocol on Layer 2 interface	142
Configuring Precision Time Protocol on EtherChannel Member Interface	143
Configuring Precision Time Protocol on SVI or Layer 3 Interface	145
Configuring the Source IP on Precision Time Protocol	145
Configuring Precision Time Protocol Timers	146
Configuring the Values of Precision Time Protocol Clocks	147
Configuring Precision Time Protocol Using AES67 Media Profile	148
Configuring Precision Time Protocol Using G8275.1 Telecom Profile	149
Configuration Examples for Precision Time Protocol	150
Example: Configuring Precision Time Protocol Using AES67 Media Profile	150
Verifying Precision Time Protocol Configurations	151
Feature History for Precision Time Protocol	157

CHAPTER 7
Configuring UniDirectional Link Detection 159

Restrictions for Configuring UniDirectional Link Detection	159
Information About UniDirectional Link Detection	159
Fast UniDirectional Link Detection	160
Modes of Operation	160
Normal Mode	160
Aggressive Mode	160
Methods to Detect Unidirectional Links	161
Neighbor Database Maintenance	161
Event-Driven Detection and Echoing	161

UniDirectional Link Detection Reset Options	162
Default UniDirectional Link Detection Configuration	162
How to Configure UniDirectional Link Detection	162
Enabling UniDirectional Link Detection Globally	162
Enabling UniDirectional Link Detection on an Interface	163
Enabling Fast UniDirectional Link Detection on an Interface	164
Enabling Fast UniDirectional Link Detection Error Reporting	165
Disabling UniDirectional Link Detection on Fiber-Optic LAN Interfaces	165
Monitoring and Maintaining UniDirectional Link Detection	166
Console Error Messages For Fast UniDirectional Link Detection	166
Additional References for UniDirectional Link Detection	167
Feature History for UniDirectional Link Detection	167

CHAPTER 8**Configuring Layer 2 Protocol Tunneling 169**

Prerequisites for Layer 2 Protocol Tunneling	169
Restrictions for Layer 2 Protocol Tunneling	169
Information About Layer 2 Protocol Tunneling	169
Layer 2 Protocol Tunneling Overview	169
Layer 2 Protocol Tunneling on Ports	171
Layer 2 Protocol Tunneling for EtherChannels	172
Default Layer 2 Protocol Tunneling Configuration	173
How to Configure Layer 2 Protocol Tunneling	173
Configuring Layer 2 Protocol Tunneling	173
How to Configure Layer 2 Protocol Tunneling for EtherChannels	176
Configuring the SP Edge Switch	176
Configuring the Customer Device	179
Configuration Examples for Layer 2 Protocol Tunneling	181
Example: Configuring Layer 2 Protocol Tunneling	181
Examples: Configuring the SP Edge and Customer Switches	181
Monitoring Tunneling Status	183
Feature History for Layer 2 Protocol Tunneling	183

CHAPTER 9**Configuring IEEE 802.1Q Tunneling 185**

Information About IEEE 802.1Q Tunneling	185
-----------------------------------------	-----

IEEE 802.1Q Tunnel Ports in a Service Provider Network	185
Native VLANs	188
System MTU	189
IEEE 802.1Q Tunneling and Other Features	189
Default IEEE 802.1Q Tunneling Configuration	190
Information About Custom EtherType for Cisco Catalyst 9600 Series Supervisor 2 Module	190
Restrictions for Custom Ethertypes	191
How to Configure IEEE 802.1Q Tunneling	191
How to Configure Custom EtherTypes	193
Monitoring Tunneling Status	194
Example: Configuring an IEEE 802.1Q Tunneling Port	195
Example: Configuring a Custom Ethertype	195
Feature History for IEEE 802.1Q Tunneling	198

CHAPTER 10**Configuring VLAN Mapping 199**

Prerequisites for VLAN Mapping	199
Prerequisites for One to One VLAN Mapping	199
Restrictions for VLAN Mapping	200
Restrictions for One to One VLAN Mapping	200
About VLAN Mapping	200
One-to-One VLAN Mapping	202
Selective Q-in-Q	202
Q-in-Q on a Trunk Port	202
Configuration Guidelines for VLAN Mapping	202
Configuration Guidelines for One-to-One VLAN Mapping	203
Configuration Guidelines for Selective Q-in-Q	203
Configuration Guidelines for Q-in-Q on a Trunk Port	204
How to Configure VLAN Mapping	204
One-to-One VLAN Mapping	204
Selective Q-in-Q on a Trunk Port	206
Q-in-Q on a Trunk Port	209
Feature History for VLAN Mapping	210

CHAPTER 11**Troubleshooting Layer 2 211**

[Overview](#) 211

[Support Articles](#) 211

[Feedback Request](#) 212

[Disclaimer and Caution](#) 212



CHAPTER 1

Configuring Spanning Tree Protocol

This chapter describes how to configure the Spanning Tree Protocol (STP) on port-based VLANs on the Catalyst devices. The device can use either the per-VLAN spanning-tree plus (PVST+) protocol based on the IEEE 802.1D standard and Cisco proprietary extensions, or the rapid per-VLAN spanning-tree plus (rapid-PVST+) protocol based on the IEEE 802.1w standard. A device stack appears as a single spanning-tree node to the rest of the network, and all stack members use the same bridge ID.

- [Restrictions for Spanning Tree Protocol, on page 1](#)
- [Information About Spanning Tree Protocol, on page 1](#)
- [How to Configure Spanning Tree Protocol, on page 12](#)
- [Monitoring Spanning Tree Protocol Configuration Status, on page 21](#)
- [Additional References for Spanning Tree Protocol, on page 22](#)
- [Feature History for Spanning Tree Protocol, on page 22](#)

Restrictions for Spanning Tree Protocol

- An attempt to configure a device as the root device fails if the value necessary to be the root device is less than 1.
- If your network consists of devices that support and do not support the extended system ID, it is unlikely that the device with the extended system ID support will become the root device. The extended system ID increases the device priority value every time the VLAN number is greater than the priority of the connected devices running older software.
- The root device for each spanning tree instance should be a backbone or distribution device. Do not configure an access device as the spanning tree primary root.

Information About Spanning Tree Protocol

The following sections provide information about spanning tree protocol:

Spanning Tree Protocol

Spanning Tree Protocol (STP) is a Layer 2 link management protocol that provides path redundancy while preventing loops in the network. For a Layer 2 Ethernet network to function properly, only one active path

can exist between any two stations. Multiple active paths among end stations cause loops in the network. If a loop exists in the network, end stations might receive duplicate messages. Devices might also learn end-station MAC addresses on multiple Layer 2 interfaces. These conditions result in an unstable network. Spanning-tree operation is transparent to end stations, which cannot detect whether they are connected to a single LAN segment or a switched LAN of multiple segments.

The STP uses a spanning-tree algorithm to select one device of a redundantly connected network as the root of the spanning tree. The algorithm calculates the best loop-free path through a switched Layer 2 network by assigning a role to each port based on the role of the port in the active topology:

- Root—A forwarding port elected for the spanning-tree topology
- Designated—A forwarding port elected for every switched LAN segment
- Alternate—A blocked port providing an alternate path to the root bridge in the spanning tree
- Backup—A blocked port in a loopback configuration

The device that has *all* of its ports as the designated role or as the backup role is the root device. The device that has at least *one* of its ports in the designated role is called the designated device.

Spanning tree forces redundant data paths into a standby (blocked) state. If a network segment in the spanning tree fails and a redundant path exists, the spanning-tree algorithm recalculates the spanning-tree topology and activates the standby path. Devices send and receive spanning-tree frames, called bridge protocol data units (BPDUs), at regular intervals. The devices do not forward these frames but use them to construct a loop-free path. BPDUs contain information about the sending device and its ports, including device and MAC addresses, device priority, port priority, and path cost. Spanning tree uses this information to elect the root device and root port for the switched network and the root port and designated port for each switched segment.

When two ports on a device are part of a loop, the spanning-tree and path cost settings control which port is put in the forwarding state and which is put in the blocking state. The spanning-tree port priority value represents the location of a port in the network topology and how well it is located to pass traffic. The path cost value represents the media speed.



Note The long path cost method is the default STP path cost method used.



Note In addition to STP, the device uses keepalive messages to detect loops. By default, keepalive is enabled on Layer 2 ports. To disable keepalive, use the **no keepalive** command in interface configuration mode.

Spanning-Tree Topology and Bridge Protocol Data Units

The stable, active spanning-tree topology of a switched network is controlled by these elements:

- The unique bridge ID (device priority and MAC address) associated with each VLAN on each device. In a switch stack, all switches use the same bridge ID for a given spanning-tree instance.
- The spanning-tree path cost to the root device.
- The port identifier (port priority and MAC address) associated with each Layer 2 interface.

When the devices in a network are powered up, each functions as the root device. Each device sends a configuration BPDU through all its ports. The BPDUs communicate and compute the spanning-tree topology. Each configuration BPDU contains this information:

- The unique bridge ID of the device that the sending device identifies as the root device.
- The spanning-tree path cost to the root
- The bridge ID of the sending device
- Message age
- The identifier of the sending interface
- Values for the hello, forward delay, and max-age protocol timers

When a device receives a configuration BPDU that contains *superior* information (lower bridge ID, lower path cost, and so forth), it stores the information for that port. If this BPDU is received on the root port of the device, the device also forwards it with an updated message to all attached LANs for which it is the designated device.

If a device receives a configuration BPDU that contains *inferior* information to that currently stored for that port, it discards the BPDU. If the device is a designated device for the LAN from which the inferior BPDU was received, it sends that LAN a BPDU containing the up-to-date information stored for that port. In this way, inferior information is discarded, and superior information is propagated on the network.

A BPDU exchange results in these actions:

- One device in the network is elected as the root switch (the logical center of the spanning-tree topology in a switched network). See the figure following the bullets.

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

- A root port is selected for each device (except the root switch). This port provides the best path (lowest cost) when the device forwards packets to the root switch.
- The shortest distance to the root switch is calculated for each device based on the path cost.
- A designated device for each LAN segment is selected. The designated device incurs the lowest path cost when forwarding packets from that LAN to the root switch. The port through which the designated device is attached to the LAN is called the designated port.

All paths that are not needed to reach the root switch from anywhere in the switched network are placed in the spanning-tree blocking mode.

Bridge ID, Device Priority, and Extended System ID

The IEEE 802.1D standard requires that each device has a unique bridge identifier (bridge ID), which controls the selection of the root switch. Because each VLAN is considered as a different *logical bridge* with PVST+ and Rapid PVST+, the same device must have a different bridge ID for each configured VLAN. Each VLAN on the device has a unique 8-byte bridge ID. The 2 most-significant bytes are used for the device priority, and the remaining 6 bytes are derived from the device MAC address.

The 2 bytes previously used for the device priority are reallocated into a 4-bit priority value and a 12-bit extended system ID value equal to the VLAN ID.

Table 1: Device Priority Value and Extended System ID

Priority Value				Extended System ID (Set Equal to the VLAN ID)										
Bit 16	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
32768	16384	8192	4096	2048	1024	512	256	128	64	32	16	8	4	2

Spanning tree uses the extended system ID, the device priority, and the allocated spanning-tree MAC address to make the bridge ID unique for each VLAN.

Support for the extended system ID affects how you manually configure the root switch, the secondary root switch, and the switch priority of a VLAN. For example, when you change the switch priority value, you change the probability that the switch will be elected as the root switch. Configuring a higher value decreases the probability; a lower value increases the probability.

Port Priority Versus Path Cost

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

The spanning-tree path cost default value is derived from the media speed of an interface. If a loop occurs, spanning tree uses 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 values that you want selected last. If all interfaces have the same cost value, spanning tree puts the interface with the lowest interface number in the forwarding state and blocks the other interfaces.

If your device is a member of a switch stack, you must assign lower cost values to interfaces that you want selected first and higher cost values that you want selected last instead of adjusting its port priority.

Spanning-Tree Interface States

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 an interface transitions directly from nonparticipation in the spanning-tree topology to the forwarding state, it can create temporary data loops. Interfaces must wait for new topology information to propagate through the switched LAN before starting to forward frames. They must allow the frame lifetime to expire for forwarded frames that have used the old topology.

Each Layer 2 interface on a device using spanning tree exists in one of these states:

- **Blocking**—The interface does not participate in frame forwarding.
- **Listening**—The first transitional state after the blocking state when the spanning tree decides that the interface should participate in frame forwarding.
- **Learning**—The interface prepares to participate in frame forwarding.
- **Forwarding**—The interface forwards frames.

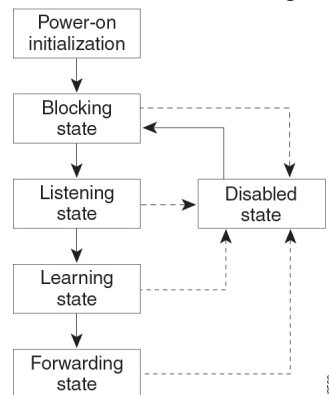
- Disabled—The interface is not participating in spanning tree because of a shutdown port, no link on the port, or no spanning-tree instance running on the port.

An interface moves through these states:

- From initialization to blocking
- From blocking to listening or to disabled
- From listening to learning or to disabled
- From learning to forwarding or to disabled
- From forwarding to disabled

Figure 1: Spanning-Tree Interface States

An interface moves through the states.



When you power up the device, spanning tree is enabled by default, and every interface in the device, VLAN, or network goes through the blocking state and the transitory states of listening and learning. Spanning tree stabilizes each interface at the forwarding or blocking state.

When the spanning-tree algorithm places a Layer 2 interface in the forwarding state, this process occurs:

1. The interface is in the listening state while spanning tree waits for protocol information to move the interface to the blocking state.
2. While spanning tree waits for the forward-delay timer to expire, it moves the interface to the learning state and resets the forward-delay timer.
3. In the learning state, the interface continues to block frame forwarding as the device learns end-station location information for the forwarding database.
4. When the forward-delay timer expires, spanning tree moves the interface to the forwarding state, where both learning and frame forwarding are enabled.

Blocking State

A Layer 2 interface in the blocking state does not participate in frame forwarding. After initialization, a BPDU is sent to each device interface. A device initially functions as the root until it exchanges BPDUs with other devices. This exchange establishes which device in the network is the root or root device. If there is only one device in the network, no exchange occurs, the forward-delay timer expires, and the interface moves to the listening state. An interface always enters the blocking state after device initialization.

An interface in the blocking state performs these functions:

- Discards frames received on the interface
- Discards frames that are switched from another interface for forwarding
- Does not learn addresses
- Receives BPDUs

Listening State

The listening state is the first state a Layer 2 interface enters after the blocking state. The interface enters this state when the spanning tree decides that the interface should participate in frame forwarding.

An interface in the listening state performs these functions:

- Discards frames received on the interface
- Discards frames that are switched from another interface for forwarding
- Does not learn addresses
- Receives BPDUs

Learning State

A Layer 2 interface in the learning state prepares to participate in frame forwarding. The interface enters the learning state from the listening state.

An interface in the learning state performs these functions:

- Discards frames received on the interface
- Discards frames that are switched from another interface for forwarding
- Learns addresses
- Receives BPDUs

Forwarding State

A Layer 2 interface in the forwarding state forwards frames. The interface enters the forwarding state from the learning state.

An interface in the forwarding state performs these functions:

- Receives and forwards frames that are received on the interface.
- Forwards frames that are switched from another interface
- Learns addresses
- Receives BPDUs

Disabled State

A Layer 2 interface in the disabled state does not participate in frame forwarding or in the spanning tree. An interface in the disabled state is nonoperational.

A disabled interface performs these functions:

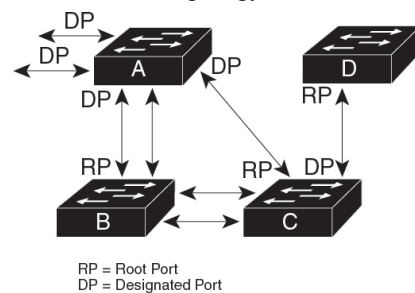
- Discards frames received on the interface
- Discards frames that are switched from another interface for forwarding
- Does not learn addresses
- Does not receive BPDUs

How a Device or Port Becomes the Root Device or Root Port

If all devices in a network are enabled with default spanning-tree settings, the device with the lowest MAC address becomes the root device.

Figure 2: Spanning-Tree Topology

Switch A is elected as the root device because the device priority of all the devices is set to the default (32768) and Switch A has the lowest MAC address. However, because of traffic patterns, number of forwarding interfaces, or link types, Switch A might not be the ideal root device. By increasing the priority (lowering the numerical value) of the ideal device so that it becomes the root device, you force a spanning-tree recalculation to form a new topology with the ideal device as the root.



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 an interface that has a higher number than the root port can cause a root-port change. The goal is to make the fastest link the root port.

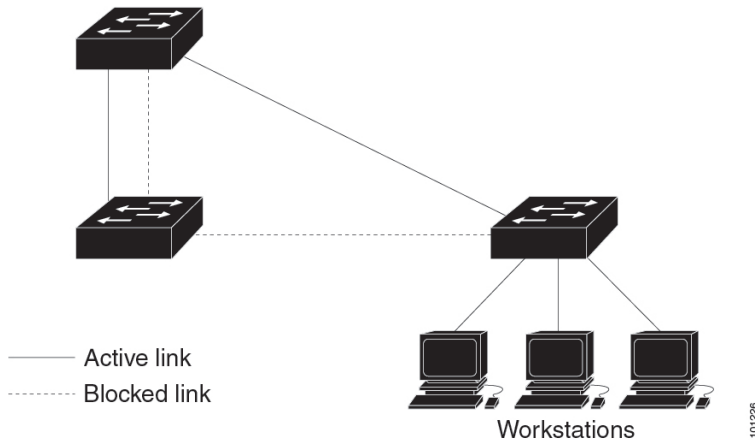
For example, assume that one port on Switch B is a Gigabit Ethernet link and that another port on Switch B (a 10/100 link) is the root port. Network traffic might be more efficient over the Gigabit Ethernet link. By changing the spanning-tree port priority on the Gigabit Ethernet port to a higher priority (lower numerical value) than the root port, the Gigabit Ethernet port becomes the new root port.

Spanning Tree and Redundant Connectivity

You can create a redundant backbone with spanning tree by connecting two switch interfaces to another device or to two different devices. Spanning tree automatically disables one interface but enables it if the other one fails. If one link is high-speed and the other is low-speed, the low-speed link is always disabled. If the speeds are the same, the port priority and port ID are added together, and spanning tree disables the link with the highest value.

[Figure 3: Spanning Tree and Redundant Connectivity, on page 8](#) shows redundant connectivity on a spanning tree topology.

Figure 3: Spanning Tree and Redundant Connectivity



You can also create redundant links between devices by using EtherChannel groups.

Spanning-Tree Address Management

IEEE 802.1D specifies 17 multicast addresses, ranging from 0x00180C200000 to 0x0180C2000010, to be used by different bridge protocols. These addresses are static addresses that cannot be removed.

Regardless of the spanning-tree state, each device in the stack receives but does not forward packets that are destined for addresses between 0x0180C2000000 and 0x0180C200000F.

If spanning tree is enabled, the CPU on the switch or on each switch in the stack receives packets that are destined for 0x0180C2000000 and 0x0180C2000010. If spanning tree is disabled, the switch or each switch in the stack forwards those packets as unknown multicast addresses.

Accelerated Aging to Retain Connectivity

The default for aging dynamic addresses is 5 minutes, the default setting of the **mac address-table aging-time** global configuration command. However, a spanning-tree reconfiguration can cause many station locations to change. Because these stations could be unreachable for 5 minutes or more during a reconfiguration, the address-aging time is accelerated so that station addresses can be dropped from the address table and then relearned. The accelerated aging is the same as the forward-delay parameter value (**spanning-tree vlan *vlan-id* forward-time seconds** global configuration command) when the spanning tree reconfigures.

Because each VLAN is a separate spanning-tree instance, the switch accelerates aging on a per-VLAN basis. A spanning-tree reconfiguration on one VLAN can cause the dynamic addresses that are learned on that VLAN to be subject to accelerated aging. Dynamic addresses on other VLANs can be unaffected and remain subject to the aging interval entered for the switch.

Spanning-Tree Modes and Protocols

The device supports these spanning-tree modes and protocols:

- **PVST+**—This spanning-tree mode is based on the IEEE 802.1D standard and Cisco proprietary extensions. The PVST+ runs on each VLAN on the device up to the maximum supported, ensuring that each has a loop-free path through the network.

The PVST+ provides Layer 2 load-balancing for the VLAN on which it runs. You can create different logical topologies by using the VLANs on your network to ensure that all of your links are used but that

no one link is oversubscribed. Each instance of PVST+ on a VLAN has a single root switch. This root switch propagates the spanning-tree information that is associated with that VLAN to all other devices in the network. Because each device has the same information about the network, this process ensures that the network topology is maintained.

- **Rapid PVST+—**Rapid PVST+ is the default STP mode on your device. This spanning-tree mode is the same as PVST+ except that it uses a rapid convergence based on the IEEE 802.1w standard. To provide rapid convergence, the Rapid PVST+ immediately deletes dynamically learned MAC address entries on a per-port basis upon receiving a topology change. By contrast, PVST+ uses a short aging time for dynamically learned MAC address entries.

Rapid PVST+ uses the same configuration as PVST+ (except where noted), and the device needs only minimal extra configuration. The benefit of Rapid PVST+ is that you can migrate a large PVST+ install base to Rapid PVST+ without having to learn the complexities of the Multiple Spanning Tree Protocol (MSTP) configuration and without having to reprovision your network. In Rapid PVST+ mode, each VLAN runs its own spanning-tree instance up to the maximum supported.

- **MSTP—**This spanning-tree mode is based on the IEEE 802.1s standard. You can map multiple VLANs to the same spanning-tree instance, which reduces the number of spanning-tree instances that are required to support many VLANs. The MSTP runs on top of the RSTP (based on IEEE 802.1w), which provides for rapid convergence of the spanning tree by eliminating the forward delay and by quickly transitioning root ports and designated ports to the forwarding state. In a switch stack, the cross-stack rapid transition (CSRT) feature performs the same function as RSTP. You cannot run MSTP without RSTP or CSRT.

Supported Spanning-Tree Instances

In PVST+ or Rapid PVST+ mode, the device or device stack supports up to 1000 spanning-tree instances. Starting with the Cisco IOS XE Bengaluru 17.5.1 release or later, the device or device stack support up to 4000 spanning-tree instances with the Customizable SDM template for 4K VLAN.

In MSTP mode, the device or device stack supports up to 64 MST instances. The number of VLANs that can be mapped to a particular MST instance is 1000. Starting with the Cisco IOS XE Bengaluru 17.5.1 release or later, the number of VLANs that can be mapped to a particular MST instance is 4000 with the Customizable SDM template for 4K VLAN.



Note On the Cisco Catalyst 9600 Series Supervisor 2 Module, starting with the Cisco IOS XE Cupertino 17.7.1 release, in the PVST+ or Rapid PVST+ mode the device or device stack support up to 4000 spanning-tree instances with the Customizable SDM template for 4K VLAN, by default.

In the MSTP mode, the number of VLANs that can be mapped to a particular MST instance is 4000 with the Customizable SDM template for 4K VLAN, by default.

Spanning-Tree Interoperability and Backward Compatibility

In a mixed MSTP and PVST+ network, the common spanning-tree (CST) root must be inside the MST backbone, and a PVST+ device cannot connect to multiple MST regions.

When a network contains devices running Rapid PVST+ and devices running PVST+, we recommend that the Rapid PVST+ devices and PVST+ devices be configured for different spanning-tree instances. In the Rapid PVST+ spanning-tree instances, the root switch must be a Rapid PVST+ device. In the PVST+ instances, the root switch must be a PVST+ device. The PVST+ devices should be at the edge of the network.

All stack members run the same version of spanning tree (all PVST+, all Rapid PVST+, or all MSTP).

Table 2: PVST+, MSTP, and Rapid-PVST+ Interoperability and Compatibility

	PVST+	MSTP	Rapid PVST+
PVST+	Yes	Yes (with restrictions)	Yes (reverts to PVST+)
MSTP	Yes (with restrictions)	Yes	Yes (reverts to PVST+)
Rapid PVST+	Yes (reverts to PVST+)	Yes (reverts to PVST+)	Yes

Spanning Tree Protocols and IEEE 802.1Q Trunks

The IEEE 802.1Q standard for VLAN trunks imposes some limitations on the spanning-tree strategy for a network. The standard requires only one spanning-tree instance for *all* VLANs allowed on the trunks. However, in a network of Cisco devices that are connected through IEEE 802.1Q trunks, the devices maintain one spanning-tree instance for *each* VLAN allowed on the trunks.

When you connect a Cisco device to a non-Cisco device through an IEEE 802.1Q trunk, the Cisco device uses PVST+ to provide spanning-tree interoperability. If Rapid PVST+ is enabled, the device uses it instead of PVST+. The device combines the spanning-tree instance of the IEEE 802.1Q VLAN of the trunk with the spanning-tree instance of the non-Cisco IEEE 802.1Q device.

However, all PVST+ or Rapid PVST+ information is maintained by Cisco devices that are separated by a cloud of non-Cisco IEEE 802.1Q devices. The non-Cisco IEEE 802.1Q cloud separating the Cisco devices is treated as a single trunk link between the devices.

Rapid PVST+ is automatically enabled on IEEE 802.1Q trunks, and no user configuration is required. The external spanning-tree behavior on access ports and Inter-Switch Link (ISL) trunk ports is not affected by PVST+.

Spanning Tree and Switch Stacks

When the switch stack is operating in PVST+ or Rapid PVST+ mode:

- A switch stack appears as a single spanning-tree node to the rest of the network, and all stack members use the same bridge ID for a given spanning tree. The bridge ID is derived from the MAC address of the active switch.
- When a new device joins the stack, it sets its bridge ID to the active switch bridge ID. If the newly added device has the lowest ID and if the root path cost is the same among all stack members, the newly added device becomes the stack root.
- When a stack member leaves the stack, spanning-tree reconvergence occurs within the stack (and possibly outside the stack). The remaining stack member with the lowest stack port ID becomes the stack root.
- If the switch stack is the spanning-tree root and the active switch fails or leaves the stack, the standby switch becomes the new active switch, bridge IDs remain the same, and a spanning-tree reconvergence might occur.
- If a neighboring device external to the switch stack fails or is powered down, normal spanning-tree processing occurs. Spanning-tree reconvergence might occur as a result of losing a device in the active topology.

- If a new device external to the switch stack is added to the network, normal spanning-tree processing occurs. Spanning-tree reconvergence might occur as a result of adding a device in the network.

Default Spanning-Tree Configuration

Table 3: Default Spanning-Tree Configuration

Feature	Default Setting
Enable state	Enabled on VLAN 1.
Spanning-tree mode	Rapid PVST+ (PVST+ and MST disabled.)
Device priority	32768
Spanning-tree port priority (configurable on a per-interface basis)	128
Spanning-tree port cost (configurable on a per-interface basis)	10 Mbps: 2000000 100 Mbps: 200000 1 Gbps: 20000 10 Gbps: 2000 40 Gbps: 500 100 Gbps: 200 1 Tbps: 20 10 Tbps: 2
Spanning-tree VLAN port priority (configurable on a per-VLAN basis)	128
Spanning-tree VLAN port cost (configurable on a per-VLAN basis)	10 Mbps: 2000000 100 Mbps: 200000 1 Gbps: 20000 10 Gbps: 2000 40 Gbps: 500 100 Gbps: 200 1 Tbps: 20 10 Tbps: 2
Spanning-tree timers	Hello time: 2 seconds Forward-delay time: 15 seconds Maximum-aging time: 20 seconds Transmit hold count: 6 BPDUs

How to Configure Spanning Tree Protocol

The following sections provide information about configuring spanning tree protocol:

Changing the Spanning-Tree Mode

The switch supports three spanning-tree modes: per-VLAN spanning tree plus (PVST+), Rapid PVST+, or Multiple Spanning Tree Protocol (MSTP). By default, the device runs the Rapid PVST+ protocol.

If you want to enable a mode that is different from the default mode, this procedure is required.

Procedure

	Command or Action	Purpose
Step 1	enable Example: Device> enable	Enables privileged EXEC mode. Enter your password if prompted.
Step 2	configure terminal Example: Device# configure terminal	Enters global configuration mode.
Step 3	spanning-tree mode {pvst mst rapid-pvst} Example: Device(config)# spanning-tree mode pvst	Configures a spanning-tree mode. All stack members run the same version of spanning tree. <ul style="list-style-type: none"> • Select pvst to enable PVST+. • Select mst to enable MSTP. • Select rapid-pvst to enable rapid PVST+.
Step 4	interface interface-id Example: Device(config)# interface GigabitEthernet1/0/1	Specifies an interface to configure, and enters interface configuration mode. Valid interfaces include physical ports, VLANs, and port channels. The VLAN ID range is 1 to 4094. The port-channel range is 1 to 128.
Step 5	spanning-tree link-type point-to-point Example: Device(config-if)# spanning-tree link-type point-to-point	Specifies that the link type for this port is point-to-point. If you connect this port (local port) to a remote port through a point-to-point link and the local port becomes a designated port, the device negotiates with the remote port and rapidly changes the local port to the forwarding state.
Step 6	end Example:	Returns to privileged EXEC mode.

	Command or Action	Purpose
	Device(config-if)# end	
Step 7	clear spanning-tree detected-protocols Example: Device# clear spanning-tree detected-protocols	If any port on the device is connected to a port on a legacy IEEE 802.1D device, this command restarts the protocol migration process on the entire device. This step is optional if the designated device detects that this device is running rapid PVST+.

(Optional) Disabling Spanning Tree

Spanning tree is enabled by default on VLAN 1 and on all newly created VLANs up to the spanning-tree limit. Disable spanning tree only if you are sure that there are no loops in the network topology.



Caution When spanning tree is disabled and loops are present in the topology, excessive traffic and indefinite packet duplication can drastically reduce network performance.

To disable spanning tree, perform this procedure:

Procedure

	Command or Action	Purpose
Step 1	enable Example: Device> enable	Enables privileged EXEC mode. Enter your password if prompted.
Step 2	configure terminal Example: Device# configure terminal	Enters global configuration mode.
Step 3	no spanning-tree vlan <i>vlan-id</i> Example: Device(config)# no spanning-tree vlan 300	For <i>vlan-id</i> , the range is 1 to 4094.
Step 4	end Example: Device(config)# end	Returns to privileged EXEC mode.

(Optional) Configuring the Root Device

To configure a device as the root for the specified VLAN, use the **spanning-tree vlan *vlan-id* root** global configuration command to modify the device priority from the default value (32768) to a significantly lower

value. When you enter this command, the software checks the switch priority of the root switches for each VLAN. Because of the extended system ID support, the switch sets its own priority for the specified VLAN to 24576 if this value causes this switch to become the root for the specified VLAN.

Use the **diameter** keyword to specify the Layer 2 network diameter (that is, the maximum number of device hops between any two end stations in the Layer 2 network). When you specify the network diameter, the device 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 use the **hello** keyword to override the automatically calculated hello time.

To configure the root device, perform this procedure:

Procedure

	Command or Action	Purpose
Step 1	enable Example: Device> enable	Enables privileged EXEC mode. Enter your password if prompted.
Step 2	configure terminal Example: Device# configure terminal	Enters global configuration mode.
Step 3	spanning-tree vlan <i>vlan-id</i> root primary [<i>diameter net-diameter</i>] Example: Device(config)# spanning-tree vlan 20-24 root primary diameter 4	Configures a device to become the root for the specified VLAN. <ul style="list-style-type: none"> • For <i>vlan-id</i>, you can specify a single VLAN identified by VLAN ID number, a range of VLANs separated by a hyphen, or a series of VLANs separated by a comma. The range is 1 to 4094. • (Optional) For diameter <i>net-diameter</i>, specify the maximum number of devices between any two end stations. The range is 2 to 7.
Step 4	end Example: Device(config)# end	Returns to privileged EXEC mode.

What to do next

After configuring the switch as the root switch, we recommend that you avoid manually configuring the hello time, forward-delay time, and maximum-age time through the **spanning-tree vlan *vlan-id* hello-time**, **spanning-tree vlan *vlan-id* forward-time**, and the **spanning-tree vlan *vlan-id* max-age** global configuration commands.

(Optional) Configuring a Secondary Root Device

When you configure a switch as the secondary root, the switch priority is modified from the default value (32768) to 28672. With this priority, the switch is likely to become the root switch for the specified VLAN if the primary root switch fails. This is assuming that the other network switches use the default switch priority of 32768, and therefore, are unlikely to become the root switch.

You can execute this command on more than one switch to configure multiple backup root switches. Use the same network diameter and hello-time values that you used when you configured the primary root switch with the **spanning-tree vlan *vlan-id* root primary** global configuration command.

To configure a secondary root device, perform this procedure:

Procedure

	Command or Action	Purpose
Step 1	enable Example: Device> enable	Enables privileged EXEC mode. Enter your password if prompted.
Step 2	configure terminal Example: Device# configure terminal	Enters global configuration mode.
Step 3	spanning-tree vlan <i>vlan-id</i> root secondary [diameter <i>net-diameter</i>] Example: Device(config)# spanning-tree vlan 20-24 root secondary diameter 4	Configures a device to become the secondary root for the specified VLAN. <ul style="list-style-type: none"> For <i>vlan-id</i>, you can specify a single VLAN identified by VLAN ID number, a range of VLANs separated by a hyphen, or a series of VLANs separated by a comma. The range is 1 to 4094. (Optional) For diameter <i>net-diameter</i>, specify the maximum number of devices between any two end stations. The range is 2 to 7. <p>Use the same network diameter value that you used when configuring the primary root switch.</p>
Step 4	end Example: Device(config)# end	Returns to privileged EXEC mode.

(Optional) Configuring Port Priority

To configure port priority, perform this procedure:

Procedure

	Command or Action	Purpose
Step 1	enable Example: Device> enable	Enables privileged EXEC mode. Enter your password if prompted.
Step 2	configure terminal Example: Device# configure terminal	Enters global configuration mode.
Step 3	interface <i>interface-id</i> Example: Device (config) # interface gigabitethernet 1/0/2	Specifies an interface to configure, and enters interface configuration mode. Valid interfaces include physical ports and port-channel logical interfaces (port-channel <i>port-channel-number</i>).
Step 4	spanning-tree port-priority <i>priority</i> Example: Device (config-if) # spanning-tree port-priority 0	Configures the port priority for an interface. For <i>priority</i> , the range is 0 to 240, in increments of 16; the default is 128. Valid values are 0, 16, 32, 48, 64, 80, 96, 112, 128, 144, 160, 176, 192, 208, 224, and 240. All other values are rejected. The lower the number, the higher the priority.
Step 5	spanning-tree vlan <i>vlan-id</i> port-priority <i>priority</i> Example: Device (config-if) # spanning-tree vlan 20-25 port-priority 0	Configures the port priority for a VLAN. <ul style="list-style-type: none"> • For <i>vlan-id</i>, you can specify a single VLAN identified by VLAN ID number, a range of VLANs separated by a hyphen, or a series of VLANs separated by a comma. The range is 1 to 4094. • For <i>priority</i>, the range is 0 to 240, in increments of 16; the default is 128. Valid values are 0, 16, 32, 48, 64, 80, 96, 112, 128, 144, 160, 176, 192, 208, 224, and 240. All other values are rejected. The lower the number, the higher the priority.
Step 6	end Example: Device (config-if) # end	Returns to privileged EXEC mode.

(Optional) Configuring Path Cost

To configure path cost, perform this procedure:

Procedure

	Command or Action	Purpose
Step 1	enable Example: Device> enable	Enables privileged EXEC mode. Enter your password if prompted.
Step 2	configure terminal Example: Device# configure terminal	Enters global configuration mode.
Step 3	interface interface-id Example: Device(config)# interface gigabitethernet 1/0/1	Specifies an interface to configure, and enters interface configuration mode. Valid interfaces include physical ports and port-channel logical interfaces (port-channel port-channel-number).
Step 4	spanning-tree cost cost Example: Device(config-if)# spanning-tree cost 250	Configures the cost for an interface. If a loop occurs, spanning tree uses the path cost when selecting an interface to place into the forwarding state. A lower path cost represents higher-speed transmission. For <i>cost</i> , the range is 1 to 200000000; the default value is derived from the media speed of the interface.
Step 5	spanning-tree vlan vlan-id cost cost Example: Device(config-if)# spanning-tree vlan 10,12-15,20 cost 300	Configures the cost for a VLAN. If a loop occurs, spanning tree uses the path cost when selecting an interface to place into the forwarding state. A lower path cost represents higher-speed transmission. <ul style="list-style-type: none"> • For <i>vlan-id</i>, you can specify a single VLAN identified by VLAN ID number, a range of VLANs separated by a hyphen, or a series of VLANs separated by a comma. The range is 1 to 4094. • For <i>cost</i>, the range is 1 to 200000000; the default value is derived from the media speed of the interface.
Step 6	end Example: Device(config-if)# end	Returns to privileged EXEC mode.

The **show spanning-tree interface interface-id** privileged EXEC command displays information only for ports that are in a link-up operative state. Otherwise, you can use the **show running-config** privileged EXEC command to confirm the configuration.

(Optional) Configuring the Device Priority of a VLAN

You can configure the switch priority and make it more likely that a standalone switch or a switch in the stack will be chosen as the root switch.



Note Exercise care when using this command. For most situations, we recommend that you use the **spanning-tree vlan *vlan-id* root primary** and the **spanning-tree vlan *vlan-id* root secondary** global configuration commands to modify the switch priority.

To configure device priority of a VLAN, perform this procedure:

Procedure

	Command or Action	Purpose
Step 1	enable Example: Device> enable	Enables privileged EXEC mode. Enter your password if prompted.
Step 2	configure terminal Example: Device# configure terminal	Enters global configuration mode.
Step 3	spanning-tree vlan <i>vlan-id</i> priority <i>priority</i> Example: Device(config)# spanning-tree vlan 20 priority 8192	Configures the device priority of a VLAN. <ul style="list-style-type: none"> For <i>vlan-id</i>, you can specify a single VLAN identified by VLAN ID number, a range of VLANs separated by a hyphen, or a series of VLANs separated by a comma. The range is 1 to 4094. For <i>priority</i>, the range is 0 to 61440 in increments of 4096; the default is 32768. The lower the number, the more likely the switch will be chosen as the root switch. Valid priority values are 4096, 8192, 12288, 16384, 20480, 24576, 28672, 32768, 36864, 40960, 45056, 49152, 53248, 57344, and 61440. All other values are rejected.
Step 4	end Example: Device(config-if)# end	Returns to privileged EXEC mode.

(Optional) Configuring the Hello Time

The hello time is the time interval between configuration messages that are generated and sent by the root switch.

To configure the hello time, perform this procedure:

Procedure

	Command or Action	Purpose
Step 1	enable Example: Device> enable	Enables privileged EXEC mode. Enter your password if prompted.
Step 2	spanning-tree vlan <i>vlan-id</i> hello-time <i>seconds</i> Example: Device(config)# spanning-tree vlan 20-24 hello-time 3	Configures the hello time of a VLAN. The hello time is the time interval between configuration messages that are generated and sent by the root switch. These messages mean that the switch is alive. <ul style="list-style-type: none"> • For <i>vlan-id</i>, you can specify a single VLAN identified by VLAN ID number, a range of VLANs separated by a hyphen, or a series of VLANs separated by a comma. The range is 1 to 4094. • For <i>seconds</i>, the range is 1 to 10; the default is 2.
Step 3	end Example: Device(config-if)# end	Returns to privileged EXEC mode.

(Optional) Configuring the Forwarding-Delay Time for a VLAN

To configure the forwarding-delay time for a VLAN, perform this procedure:

Procedure

	Command or Action	Purpose
Step 1	enable Example: Device> enable	Enables privileged EXEC mode. Enter your password if prompted.
Step 2	configure terminal Example: Device# configure terminal	Enters global configuration mode.

	Command or Action	Purpose
Step 3	spanning-tree vlan <i>vlan-id</i> forward-time <i>seconds</i> Example: Device(config)# spanning-tree vlan 20,25 forward-time 18	Configures the forward time of a VLAN. The forwarding delay is the number of seconds an interface waits before changing from its spanning-tree learning and listening states to the forwarding state. <ul style="list-style-type: none"> • For <i>vlan-id</i>, you can specify a single VLAN identified by VLAN ID number, a range of VLANs separated by a hyphen, or a series of VLANs separated by a comma. The range is 1 to 4094. • For <i>seconds</i>, the range is 4 to 30; the default is 15.
Step 4	end Example: Device(config)# end	Returns to privileged EXEC mode.

(Optional) Configuring the Maximum-Aging Time for a VLAN

To configure the maximum-aging time for a VLAN, perform this procedure:

Procedure

	Command or Action	Purpose
Step 1	enable Example: Device> enable	Enables privileged EXEC mode. Enter your password if prompted.
Step 2	configure terminal Example: Device# configure terminal	Enters global configuration mode.
Step 3	spanning-tree vlan <i>vlan-id</i> max-age <i>seconds</i> Example: Device(config)# spanning-tree vlan 20 max-age 30	Configures the maximum-aging time of a VLAN. The maximum-aging time is the number of seconds a switch waits without receiving spanning-tree configuration messages before attempting a reconfiguration. <ul style="list-style-type: none"> • For <i>vlan-id</i>, you can specify a single VLAN identified by VLAN ID number, a range of VLANs separated by a hyphen, or a series of VLANs separated by a comma. The range is 1 to 4094. • For <i>seconds</i>, the range is 6 to 40; the default is 20.

	Command or Action	Purpose
Step 4	end Example: Device(config-if) # end	Returns to privileged EXEC mode.

(Optional) Configuring the Transmit Hold-Count

You can configure the BPDU burst size by changing the transmit hold count value.



Note Changing this parameter to a higher value can have a significant impact on CPU utilization, especially in Rapid PVST+ mode. Lowering this value can slow down convergence in certain scenarios. We recommend that you maintain the default setting.

To configure the transmit hold-count, perform this procedure:

Procedure

	Command or Action	Purpose
Step 1	enable Example: Device> enable	Enables privileged EXEC mode. Enter your password if prompted.
Step 2	configure terminal Example: Device# configure terminal	Enters global configuration mode.
Step 3	spanning-tree transmit hold-count <i>value</i> Example: Device(config) # spanning-tree transmit hold-count 6	Configures the number of BPDUs that can be sent before pausing for 1 second. For <i>value</i> , the range is 1 to 20; the default is 6.
Step 4	end Example: Device(config) # end	Returns to privileged EXEC mode.

Monitoring Spanning Tree Protocol Configuration Status

Table 4: Commands for Displaying STP Configuration Status

show spanning-tree active	Displays STP configuration information on active interfaces only.
show spanning-tree detail	Displays a detailed summary of interface information.

show spanning-tree vlan <i>vlan-id</i>	Displays STP configuration information for the specified VLAN.
show spanning-tree interface <i>interface-id</i>	Displays STP configuration information for the specified interface.
show spanning-tree interface <i>interface-id</i> portfast	Displays STP portfast information for the specified interface.
show spanning-tree summary [totals]	Displays a summary of interface states or displays the total lines of the state section.

To clear STP counters, use the **clear spanning-tree** [**interface** *interface-id*] privileged EXEC command.

Additional References for Spanning Tree Protocol

Related Documents

Related Topic	Document Title
For complete syntax and usage information for the commands used in this chapter.	See the Layer 2/3 Commands section of the <i>Command Reference (Catalyst 9600 Series Switches)</i>

Feature History for Spanning Tree Protocol

This table provides release and related information for features explained in this module.

These features are available on all releases subsequent to the one they were introduced in, unless noted otherwise.

Table 5: New Feature History

Release	Feature	Feature Information
Cisco IOS XE Gibraltar 16.11.1	Spanning Tree Protocol	STP is a Layer 2 link management protocol that provides path redundancy while preventing loops in the network.

Use Cisco Feature Navigator to find information about platform and software image support. To access Cisco Feature Navigator, go to <http://www.cisco.com/go/cfn>.



CHAPTER 2

Configuring Loop Detection Guard

- [Restrictions for Loop Detection Guard, on page 23](#)
- [Information About Loop Detection Guard, on page 26](#)
- [Enabling Loop Detection Guard and Error-Disabling the Required Port, on page 26](#)
- [Additional References for Configuring Loop Detection Guard, on page 27](#)
- [Feature History for Loop Detection Guard, on page 28](#)

Restrictions for Loop Detection Guard

Loop detection guard can be configured only on Layer 2 physical interfaces. Layer 3 ports and virtual interfaces, such as port channels, switch virtual interfaces (SVIs), and tunnels, are not supported.

Information About Loop Detection Guard

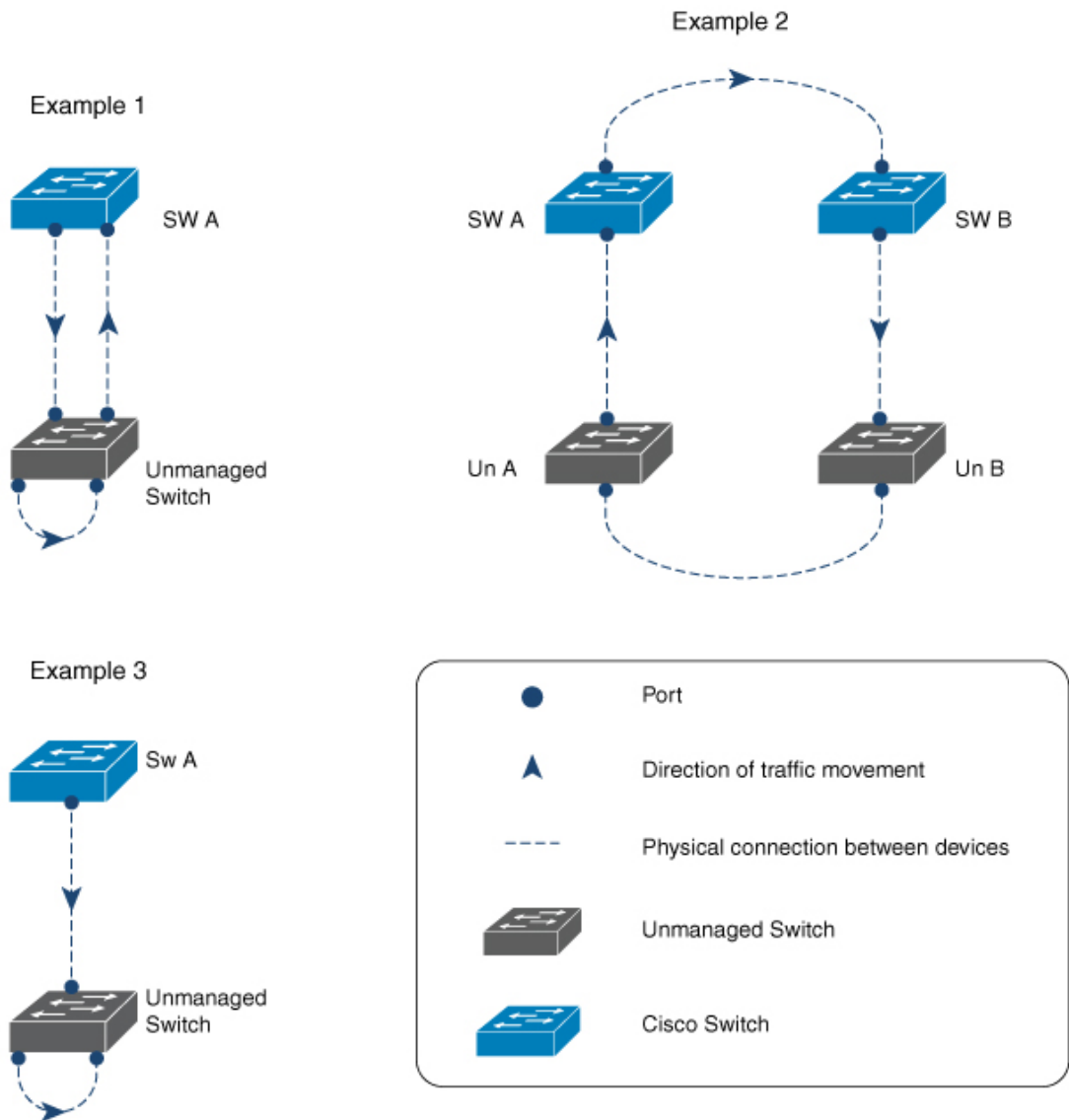
A computer network can experience a network loop where there is more than one Layer 2 path between two endpoints. This is possible when there are multiple connections between two switches in a network or two ports on the same switch are connected to each other. The following figure shows a few examples of a network loop:

Example 1: Switch SW A, which is within the network, is sending traffic to an unmanaged switch on one port and receiving traffic from the same unmanaged switch, on another port. On the unmanaged switch, the port receiving traffic is connected to the port sending traffic back to the SW A in the network, resulting in a network loop.

Example 2: This example shows a network loop involving four switches, two within the network (SW A and SW B) and two unmanaged switches (Un A and Un B). Traffic is moving in the following direction SW A to SW B to Un B to Un A and back to SW A, resulting in a network loop.

Example 3: Two ports on the unmanaged switch are connected to each other, resulting in a network loop.

Figure 4: Examples of Network Loop Between Managed and Unmanaged Switches



3566545

While Spanning Tree Protocol (STP) is normally the protocol that is configured for this purpose (to prevent network loops), loop detection guard is suited to situations where there may be unmanaged switches in a network that do not understand STP, or where STP is not configured on the network.

Loop detection guard is enabled at the interface level. To detect loops, the system sends loop-detect frames from the interface, at preconfigured intervals. When a loop is detected, the configured action is taken.

Loop detection guard is disabled by default. When you enable the feature, you can configure one of these actions:

- Error-disable the port sending traffic.
- Error-disable the port receiving traffic (default).
- Display an error message and not disable any port.

When a port is error-disabled, no traffic is sent or received on that port.

Interaction of Loop Detection Guard with Other Features

The following sections provide information about how loop detection guard interacts with other feature:

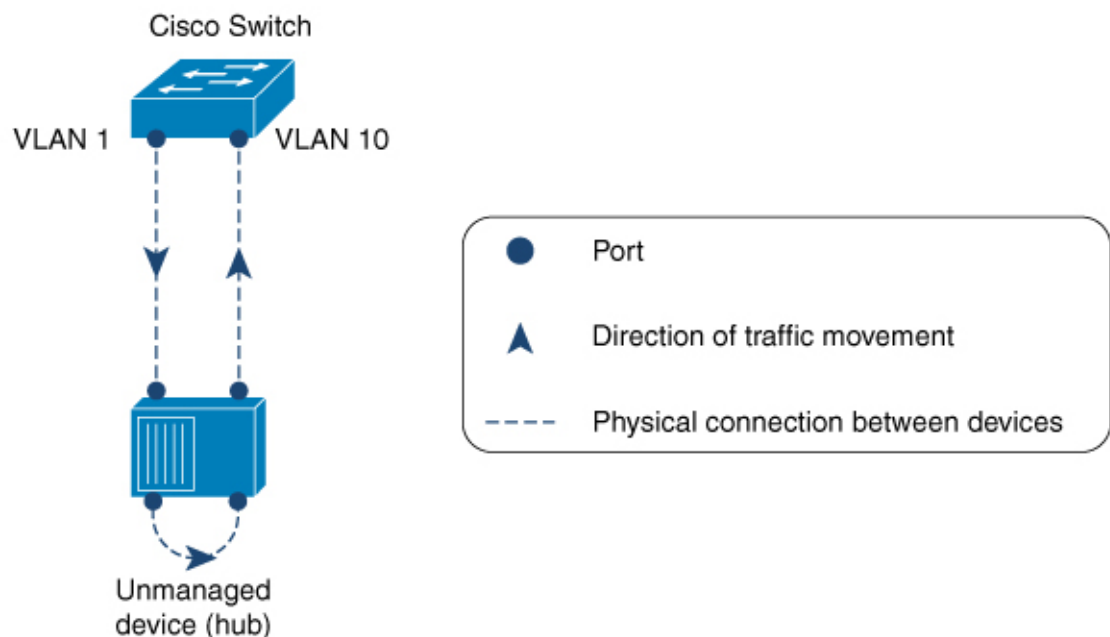
Spanning Tree Protocol and Loop Detection Guard

When both loop detection guard and STP are enabled on a device, STP takes over monitoring the network for loops. In this case loop-detect packets are neither received nor processed in the network.

VLANs and Loop Detection Guard

We do not recommend configuring this feature on a switch that is connected to a hub for these reasons: The hub floods traffic to all of its interfaces. If the switch in your network is receiving traffic from the same hub, but on a port in a different VLAN, you may be inadvertently error-disabling those destination ports. The figure below illustrates such a situation. The port in VLAN 1 is sending traffic to the hub. The switch is also receiving traffic from the same hub, but on a port in a different VLAN, that is, VLAN 10. If you configure loop detection guard (and you have configured the default action of error-disabling the destination port), then the port in VLAN 10 is blocked. Configuring the option to display a message (instead of error-disabling a port) is not recommended either, because the system displays as many messages as the number of interfaces configured in the hub, resulting in a CPU overload.

Figure 5: A Switch Connected to an Unmanaged Network Hub



356546

Enabling Loop Detection Guard and Error-Disabling the Required Port

The feature is disabled by default. Complete the following steps to enable loop detection guard and configure the action that you want the system to take when a loop is detected:

Procedure

	Command or Action	Purpose
Step 1	enable Example: Device> enable	Enables privileged EXEC mode. Enter your password if prompted.
Step 2	configure terminal Example: Device# configure terminal	Enters global configuration mode.
Step 3	interface <i>{ interface-id subinterface-id vlan-id }</i> Example: Device(config)# interface tenGigabitEthernet 1/0/20 Device(config-if)#	Enters interface configuration mode. Specify only a physical interface to configure loop detection guard on the device. Layer 3 ports and virtual interfaces like PortChannels, switch virtual interfaces (SVIs), and tunnels are not supported.
Step 4	[no] loopdetect Example: Device(config-if)# loopdetect	Enables loop detection guard on the device. Loopdetect frames are sent from the configured interface. Use the loopdetect command without any keyword to enable loop detection guard. Use the no form of this command to disable this feature. Note You can enable the feature on trunk ports, but a warning message is displayed, for the following reason: A trunk port carries traffic for several VLANs, simultaneously. A loop that is detected in one VLAN can result in the error-disabling of all VLAN traffic that is associated with the trunk port.
Step 5	[no] loopdetect { time action syslog source-port } Example: Device(config-if)# loopdetect 7	Specifies the frequency at which loop-detect frames are sent and the action the system takes when a loop is detected. If you do not specify an action, the destination port is error-disabled by default.

	Command or Action	Purpose
		<p>You can configure the following:</p> <ul style="list-style-type: none"> • time—Time interval to send loop-detect frame, in seconds. The range is from 1 to 10. The default is 5. • action syslog—Displays a system message and does not error-disable any port. If you use the no form of this command, the system reverts to the last configured option. • source-port—Error-disables the source port. If you use the no form of this command, the destination port is error-disabled. <p>In the example configuration on the left (<code>Device(config-if)# loopdetect 7</code>), the interface is configured to send loop-detect frames every 7 seconds, and to error-disable the destination port if a loop is detected (The default applies, because neither the action syslog option nor the source-port option has been configured).</p>
Step 6	<p>end</p> <p>Example:</p> <pre>Device(config-if)# end</pre>	Returns to privileged EXEC mode.
Step 7	<p>show loopdetect</p> <p>Example:</p> <pre>Device# show loopdetect</pre>	Displays all the interfaces where loop detection guard is enabled, the frequency at which loop-detect packets are sent, and the status of the physical interface.

Additional References for Configuring Loop Detection Guard

Related Documents

Related Topic	Document Title
For complete syntax and usage information for the commands used in this chapter.	See the Layer 2/3 Commands section of the <i>Command Reference (Catalyst 9600 Series Switches)</i>

Feature History for Loop Detection Guard

This table provides release and related information for features explained in this module.

These features are available on all releases subsequent to the one they were introduced in, unless noted otherwise.

Release	Feature	Feature Information
Cisco IOS XE Amsterdam 17.2.1	Loop Detection Guard	Loop detection guard prevents network loops in either networks that are not configured with STP or unmanaged devices in networks that are configured with STP.

Use Cisco Feature Navigator to find information about platform and software image support. To access Cisco Feature Navigator, go to <http://www.cisco.com/go/cfn>.



CHAPTER 3

Configuring Multiple Spanning-Tree Protocol

- [Prerequisites for Multiple Spanning Tree Protocol, on page 29](#)
- [Restrictions for Multiple Spanning-Tree Protocol, on page 29](#)
- [Information About Multiple Spanning Tree Protocol, on page 30](#)
- [How to Configure Multiple Spanning Tree Protocol and Parameters, on page 44](#)
- [Feature History for Multiple Spanning Tree Protocol, on page 56](#)

Prerequisites for Multiple Spanning Tree Protocol

- For two or more devices to be in the same multiple spanning tree (MST) region, they must have the same VLAN-to-instance map, the same configuration revision number, and the same name.
- For load-balancing across redundant paths in the network to work, all VLAN-to-instance mapping assignments must match; otherwise, all traffic flows on a single link.
- For load-balancing between a per-VLAN spanning tree plus (PVST+) and an MST cloud or between a rapid-PVST+ and an MST cloud to work, all MST boundary ports must be forwarding. MST boundary ports are forwarding when the root of the internal spanning tree (IST) of the MST cloud is the root of the common spanning tree (CST). If the MST cloud consists of multiple MST regions, one of the MST regions must contain the CST root, and all of the other MST regions must have a better path to the root contained within the MST cloud than a path through the PVST+ or rapid-PVST+ cloud. You might have to manually configure the devices in the clouds.

Restrictions for Multiple Spanning-Tree Protocol

- The switch stack supports up to 65 MST instances. The number of VLANs that can be mapped to a particular MST instance is the maximum active VLAN supported by a given switch.
- PVST+, Rapid PVST+, and MSTP are supported, but only one version can be active at any time. (For example, all VLANs run PVST+, all VLANs run Rapid PVST+, or all VLANs run MSTP.)
- VLAN Trunking Protocol (VTP) propagation of the MST configuration is not supported. However, you can manually configure the MST configuration (region name, revision number, and VLAN-to-instance mapping) on each device within the MST region by using the command-line interface (CLI) or through the Simple Network Management Protocol (SNMP) support.

- Partitioning the network into a large number of regions is not recommended. However, if this situation is unavoidable, we recommend that you partition the switched LAN into smaller LANs interconnected by routers or non-Layer 2 devices.
- A region can have one member or multiple members with the same MST configuration; each member must be capable of processing rapid spanning tree protocol (RSTP) Bridge Protocol Data Units (BPDUs). There is no limit to the number of MST regions in a network, but each region can only support up to 65 spanning-tree instances. You can assign a VLAN to only one spanning-tree instance at a time.

Information About Multiple Spanning Tree Protocol

The following sections provide information about Multiple Spanning-Tree Protocol (MSTP):

Multiple Spanning Tree Protocol Configuration

Multiple Spanning-Tree Protocol (MSTP), which uses Rapid Spanning-Tree Protocol (RSTP) for rapid convergence, enables multiple VLANs to be grouped into and mapped to the same spanning-tree instance, reducing the number of spanning-tree instances that are needed to support many VLANs. The MSTP provides for multiple forwarding paths for data traffic, enables load balancing, and reduces the number of spanning-tree instances that are required to support many VLANs. It improves the fault tolerance of the network because a failure in one instance (forwarding path) does not affect other instances (forwarding paths).



Note The multiple spanning-tree (MST) implementation is based on the IEEE 802.1s standard.

The most common initial deployment of MSTP is in the backbone and distribution layers of a Layer 2 switched network. This deployment provides the highly available network that is required in a service-provider environment.

When the device is in the MST mode, the RSTP, which is based on IEEE 802.1w, is automatically enabled. The RSTP provides rapid convergence of the spanning tree through explicit handshaking that eliminates the IEEE 802.1D forwarding delay and quickly transitions root ports and designated ports to the forwarding state.

Both MSTP and RSTP improve the spanning-tree operation and maintain backward compatibility with equipment that is based on the (original) IEEE 802.1D spanning tree, with existing Cisco-proprietary Multiple Instance STP (MISTP), and with existing Cisco PVST+ and rapid per-VLAN spanning-tree plus (Rapid PVST+).

A device stack appears as a single spanning-tree node to the rest of the network, and all stack members use the same device ID.

Multiple Spanning Tree Protocol Configuration Guidelines

- When you enable MST by using the **spanning-tree mode mst** global configuration command, RSTP is automatically enabled.
- For configuration guidelines about UplinkFast, BackboneFast, and cross-stack UplinkFast, see the relevant sections in the Related Topics section.

- When the device is in MST mode, it uses the long path-cost calculation method (32 bits) to compute the path cost values. With the long path-cost calculation method, the following path cost values are supported:

Speed	Path Cost Value
10 Mb/s	2,000,000
100 Mb/s	200,000
1 Gb/s	20,000
10 Gb/s	2,000
100 Gb/s	200

Root Switch Configuration

The switch maintains a spanning-tree instance for the group of VLANs mapped to it. A device ID, consisting of the switch priority and the switch MAC address, is associated with each instance. For a group of VLANs, the switch with the lowest device ID becomes the root switch.

When you configure a switch as the root, you modify the switch priority from the default value (32768) to a significantly lower value so that the switch becomes the root switch for the specified spanning-tree instance. When you enter this command, the switch checks the switch priorities of the root switches. Because of the extended system ID support, the switch sets its own priority for the specified instance to 24576 if this value will cause this switches to become the root for the specified spanning-tree instance.

If any root switch for the specified instance has a switch priority lower than 24576, the switch sets its own priority to 4096 less than the lowest switch priority. (4096 is the value of the least-significant bit of a 4-bit switch priority value. For more information, see [Bridge ID, Device Priority, and Extended System ID](#).)

If your network consists of switches that support and do not support the extended system ID, it is unlikely that the switch with the extended system ID support will become the root switch. The extended system ID increases the switch priority value every time the VLAN number is greater than the priority of the connected switches running older software.

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

Use the **diameter** keyword, which is available only for MST instance 0, to specify the Layer 2 network diameter (that is, the maximum number of switch hops between any two end stations in the Layer 2 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 use the **hello** keyword to override the automatically calculated hello time.

Multiple Spanning-Tree Regions

For switches to participate in multiple spanning-tree (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 comprises an MST region.

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

the device for a region by specifying the MST region configuration on it. You can map VLANs to an MST instance, specify the region name, and set the revision number. For instructions and an example, select the "Specifying the MST Region Configuration and Enabling MSTP" link in Related Topics.

A region can have one or multiple members with the same MST configuration. Each member must be capable of processing RSTP bridge protocol data units (BPDUs). There is no limit to the number of MST regions in a network, but each region can support up to 64 spanning-tree instances. Instances can be identified by any number in the range from 0 to 4094. You can assign a VLAN to only one spanning-tree instance at a time.

Internal Spanning Tree, Common and Internal Spanning Tree, and Common Spanning Tree

Unlike PVST+ and Rapid PVST+ in which all the spanning-tree instances are independent, the MSTP establishes and maintains two types of spanning trees:

- An internal spanning tree (IST), which is the spanning tree that runs in an MST region.

Within each MST region, the MSTP maintains multiple spanning-tree instances. Instance 0 is a special instance for a region, known as the internal spanning tree (IST). All other MST instances are numbered from 1 to 4094.

The IST is the only spanning-tree instance that sends and receives BPDUs. All of the other spanning-tree instance information is contained in M-records, which are encapsulated within MSTP BPDUs. Because the MSTP BPDU carries information for all instances, the number of BPDUs that need to be processed to support multiple spanning-tree instances is significantly reduced.

All MST instances within the same region share the same protocol timers, but each MST instance has its own topology parameters, such as root switch ID, root path cost, and so forth. By default, all VLANs are assigned to the IST.

An MST instance is local to the region; for example, MST instance 1 in region A is independent of MST instance 1 in region B, even if regions A and B are interconnected.

- A common and internal spanning tree (CIST), which is a collection of the ISTs in each MST region, and the common spanning tree (CST) that interconnects the MST regions and single spanning trees.

The spanning tree that is computed in a 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 IEEE 802.1w, IEEE 802.1s, and IEEE 802.1D standards. The CIST inside an MST region is the same as the CST outside a region.

Operations Within an Multiple Spanning Tree Region

The IST connects all the MSTP switches in a region. When the IST converges, the root of the IST becomes the CIST regional root. It is the switch within the region with the lowest device ID and path cost to the CIST 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, one of the MSTP switches at the boundary of the region is selected as the CIST regional root.

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

During initialization, a region might have many subregions, each with its own CIST regional root. As switches receive superior IST information, they leave their old subregions and join the new subregion that contains the true CIST regional root. All subregions shrink except for the one that contains the true CIST regional root.

For correct operation, all switches in the MST region must agree on the same CIST regional root. Therefore, any two switches in the region only synchronize their port roles for an MST instance if they converge to a common CIST regional root.

Operations Between Multiple Spanning Tree Regions

If there are multiple regions or legacy IEEE 802.1D switches within the network, MSTP establishes and maintains the CST, which includes all MST regions and all legacy STP switches in the network. The MST instances combine with the IST at the boundary of the region to become the CST.

The IST connects all the MSTP 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.

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

MSTP switches use Version 3 RSTP BPDUs or IEEE 802.1D STP BPDUs to communicate with legacy IEEE 802.1D devices. MSTP switches use MSTP BPDUs to communicate with MSTP devices.

IEEE 802.1s Terminology

Some MST naming conventions that are used in Cisco's prestandard implementation have been changed to identify some *internal* or *regional* parameters. These parameters are significant only within an MST region, as opposed to external parameters that are relevant to the whole network. Because the CIST is the only spanning-tree instance that spans the whole network, only the CIST parameters require the external rather than the internal or regional qualifiers.

- The CIST root is the root switch for the unique instance that spans the whole network, the CIST.
- The CIST external root path cost is the cost to the CIST root. This cost is left unchanged within an MST region. Remember that an MST region looks like a single switch for the CIST. The CIST external root path cost is the root path cost that is calculated between these virtual devices and devices 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 switch 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.

Illustration of Multiple Spanning Tree Regions

This figure displays three MST regions and a legacy IEEE 802.1D device (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. The RSTP runs in all regions.

Figure 6: MST Regions, CIST Regional Root, and CST Root



Hop Count

The IST and MST instances do not use the message-age and maximum-age information in the configuration BPDU to compute the spanning-tree topology. Instead, they use 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 switch 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 BPDUs it generates. When the count reaches zero, the switch discards the BPDU and ages the information that is held for the port.

The message-age and maximum-age information in the RSTP portion of the BPDU remain the same throughout the region, and the same values are propagated by the region designated ports at the boundary.

Boundary Ports

In the Cisco prestandard implementation, a boundary port connects an MST region to a single spanning-tree region running RSTP, to a single spanning-tree region running PVST+ or rapid PVST+, or to another MST region with a different MST configuration. A boundary port also connects to a LAN, the designated device of which is either a single spanning-tree switch or a switch with a different MST configuration.

There is no definition of a boundary port in the IEEE 802.1s standard. The IEEE 802.1Q-2002 standard identifies two kinds of messages that a port can receive:

- internal (coming from the same region)
- external (coming from another region)

When a message is internal, the CIST part is received by the CIST, and each MST instance receives its respective M-record.

When a message is external, it is received only by the CIST. If the CIST role is root or alternate, or if the external BPDU is a topology change, it could have an impact on the MST instances.

An MST region includes both devices and LANs. A segment belongs to the region of its designated port. Therefore, a port in a different region than the designated port for a segment is a boundary port. This definition allows two ports internal to a region to share a segment with a port belonging to a different region, creating the possibility of a port receiving both internal and external messages.

The primary change from the Cisco prestandard implementation is that a designated port is not defined as boundary, unless it is running in an STP-compatible mode.



Note If there is a legacy STP device on the segment, messages are always considered external.

The other change from the Cisco prestandard implementation is that the CIST regional root device ID field is now inserted where an RSTP or legacy IEEE 802.1Q device has the sender device ID. The whole region performs like a single virtual device by sending a consistent sender device ID to neighboring devices. In this example, Switch C would receive a BPDU with the same consistent sender device ID of root, whether or not A or B is designated for the segment.

IEEE 802.1s Implementation

The Cisco implementation of the IEEE MST standard includes features required to meet the standard, as well as some of the desirable prestandard functionality that is not yet incorporated into the published standard.

Port Role Naming Change

The boundary role is no longer in the final MST standard, but this boundary concept is maintained in Cisco's implementation. However, an MST instance port at a boundary of the region might not follow the state of the corresponding CIST port. Two boundary roles currently exist:

- The boundary port is the root port of the CIST regional root—When the CIST instance port is proposed and is in sync, it can send back an agreement and move to the forwarding state only after all the corresponding MSTI ports are in sync (and thus forwarding). The MSTI ports now have a special *primary* role.

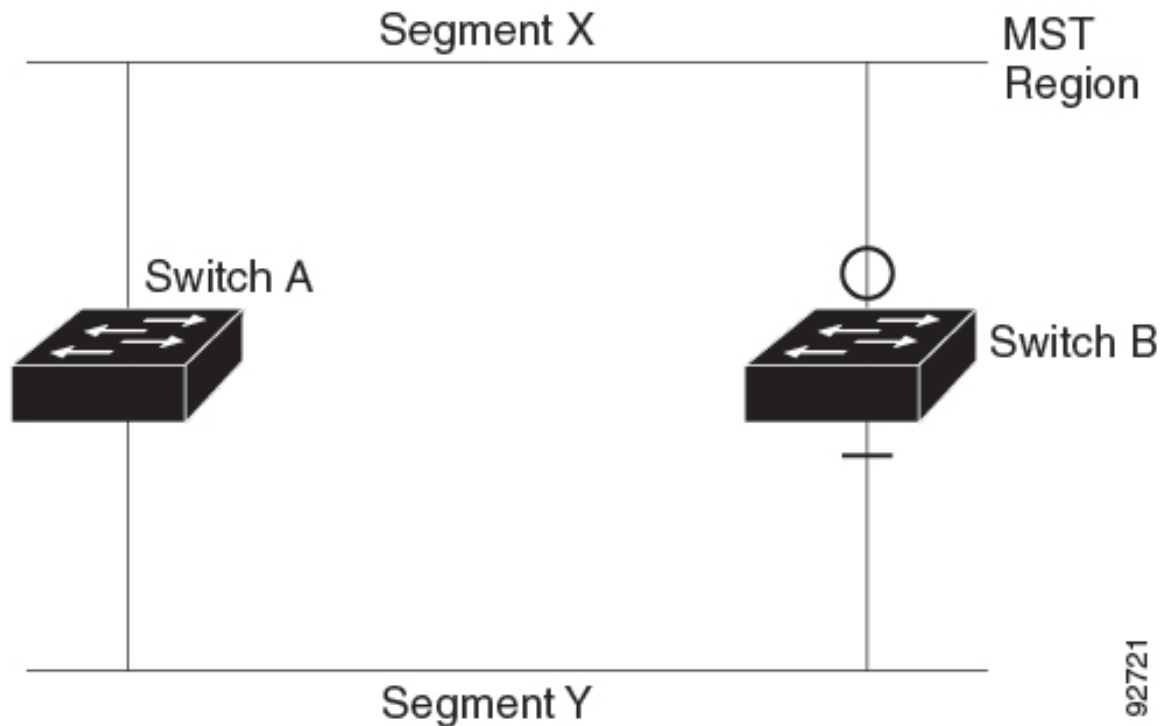
- The boundary port is not the root port of the CIST regional root—The MSTI ports follow the state and role of the CIST port. The standard provides less information, and it might be difficult to understand why an MSTI port can be alternately blocking when it receives no BPDUs (MRecords). In this case, although the boundary role no longer exists, the **show** commands identify a port as boundary in the *type* column of the output.

Interoperation Between Legacy and Standard Devices

Because automatic detection of prestandard devices can fail, you can use an interface configuration command to identify prestandard ports. A region cannot be formed between a standard and a prestandard device, but they can interoperate by using the CIST. Only the capability of load-balancing over different instances is lost in that particular case. The CLI displays different flags depending on the port configuration when a port receives prestandard BPDUs. A syslog message also appears the first time a device receives a prestandard BPDU on a port that has not been configured for prestandard BPDU transmission.

Figure 7: Standard and Prestandard Device Interoperation

Assume that A is a standard switch and B a prestandard switch, both configured to be in the same region. A is the root switch for the CIST, and B has a root port (BX) on segment X and an alternate port (BY) on segment Y. If segment Y flaps, and the port on BY becomes the alternate before sending out a single prestandard BPDU, AY cannot detect that a prestandard switch is connected to Y and continues to send standard BPDUs. The port BY is fixed in a boundary, and no load balancing is possible between A and B. The same problem exists on segment X, but B might transmit topology changes.



Note We recommend that you minimize the interaction between standard and prestandard MST implementations.

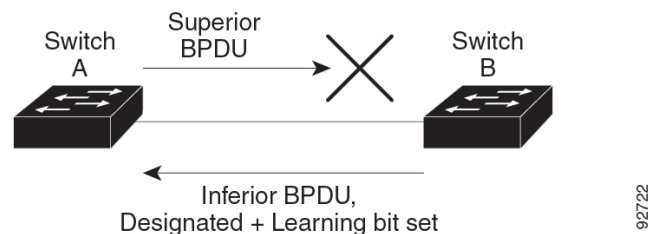
Detecting Unidirectional Link Failure

This feature is not yet present in the IEEE MST standard, but it is included in this Cisco IOS release. 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 the discarding state because disrupting connectivity in case of inconsistency is preferable to opening a bridging loop.

Figure 8: Detecting Unidirectional Link Failure

This figure illustrates a unidirectional link failure that typically creates a bridging loop. Switch A is the root device, and its BPDUs are lost on the link leading to Switch B. RSTP 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 it sends and that Switch B is the designated, not root switch. As a result, Switch A blocks (or keeps blocking) its port, which prevents the bridging loop.



Multiple Spanning Tree Protocol and Switch Stacks

A switch stack appears as a single spanning-tree node to the rest of the network, and all stack members use the same bridge ID for a given spanning tree. The bridge ID is derived from the MAC address of the device.

The active switch is the stack root when the stack is the root of the network and no root selection has been made within the stack.

If the switch stack is the spanning-tree root and the active switch fails or leaves the stack, the standby switch becomes the new active switch, bridge IDs remain the same, and a spanning-tree reconvergence might occur.

If a device that does not support MSTP is added to a switch stack that does support MSTP or the reverse, the device is put into a version mismatch state. If possible, the device is automatically upgraded or downgraded to the same version of software that is running on the switch stack.

Interoperability with IEEE 802.1D Spanning Tree Protocol

A device running MSTP supports a built-in protocol migration mechanism that enables it to interoperate with legacy IEEE 802.1D devices. If this device receives a legacy IEEE 802.1D configuration BPDU (a BPDU with the protocol version set to 0), it sends only IEEE 802.1D BPDUs on that port. An MSTP device also can detect that a port is at the boundary of a region when it receives a legacy BPDU, an MSTP BPDU (Version 3) associated with a different region, or an RSTP BPDU (Version 2).

However, the device does not automatically revert to the MSTP mode if it no longer receives IEEE 802.1D BPDUs because it cannot detect whether the legacy switch has been removed from the link unless the legacy switch is the designated device. A device might also continue to assign a boundary role to a port when the device to which this device is connected has joined the region. To restart the protocol migration process (force the renegotiation with neighboring devices), use the **clear spanning-tree detected-protocols** privileged EXEC command.

If all the legacy switches on the link are RSTP devices, they can process MSTP BPDUs as if they are RSTP BPDUs. Therefore, MSTP devices send either a Version 0 configuration and TCN BPDUs or Version 3 MSTP BPDUs on a boundary port. A boundary port connects to a LAN, the designated device of which is either a single spanning-tree switch or a switch with a different MST configuration.

Rapid Spanning Tree Protocol Overview

The RSTP takes advantage of point-to-point wiring and provides rapid convergence of the spanning tree. Reconfiguration of the spanning tree can occur in less than 1 second (in contrast to 50 seconds with the default settings in the IEEE 802.1D spanning tree).

Port Roles and the Active Topology

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

- Root port—Provides the best path (lowest cost) when the device forwards packets to the root switch.
- Designated port—Connects to the designated device, which incurs the lowest path cost when forwarding packets from that LAN to the root switch. The port through which the designated device is attached to the LAN is called the designated port.
- Alternate port—Offers an alternate path toward the root switch to that provided by the current root port.
- Backup port—Acts as a backup for the path that is 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 device has two or more connections to a shared LAN segment.
- Disabled port—Has no role within the operation of the spanning tree.

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.

In a stable topology with consistent port roles throughout the network, the RSTP ensures that every root port and designated port immediately transition to the forwarding state while all alternate and backup ports are always in the discarding state (equivalent to blocking in IEEE 802.1D). The port state controls the operation of the forwarding and learning processes.

Table 6: Port State Comparison

Operational Status	STP Port State (IEEE 802.1D)	RSTP Port State	Is Port Included in the Active Topology?
Enabled	Blocking	Discarding	No
Enabled	Listening	Discarding	No
Enabled	Learning	Learning	Yes
Enabled	Forwarding	Forwarding	Yes
Disabled	Disabled	Discarding	No

To be consistent with Cisco STP implementations, this guide defines the port state as *blocking* instead of *discarding*. Designated ports start in the listening state.

Rapid Convergence

The RSTP provides for rapid recovery of connectivity following the failure of a device, a device 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—If you configure a port as an edge port on an RSTP device by using the **spanning-tree portfast** interface configuration command, the edge port immediately transitions to the forwarding state. An edge port is the same as a Port Fast-enabled port, and you should enable it only on ports that connect to a single end station.
- Root ports—If the RSTP 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.

Figure 9: Proposal and Agreement Handshaking for Rapid Convergence

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

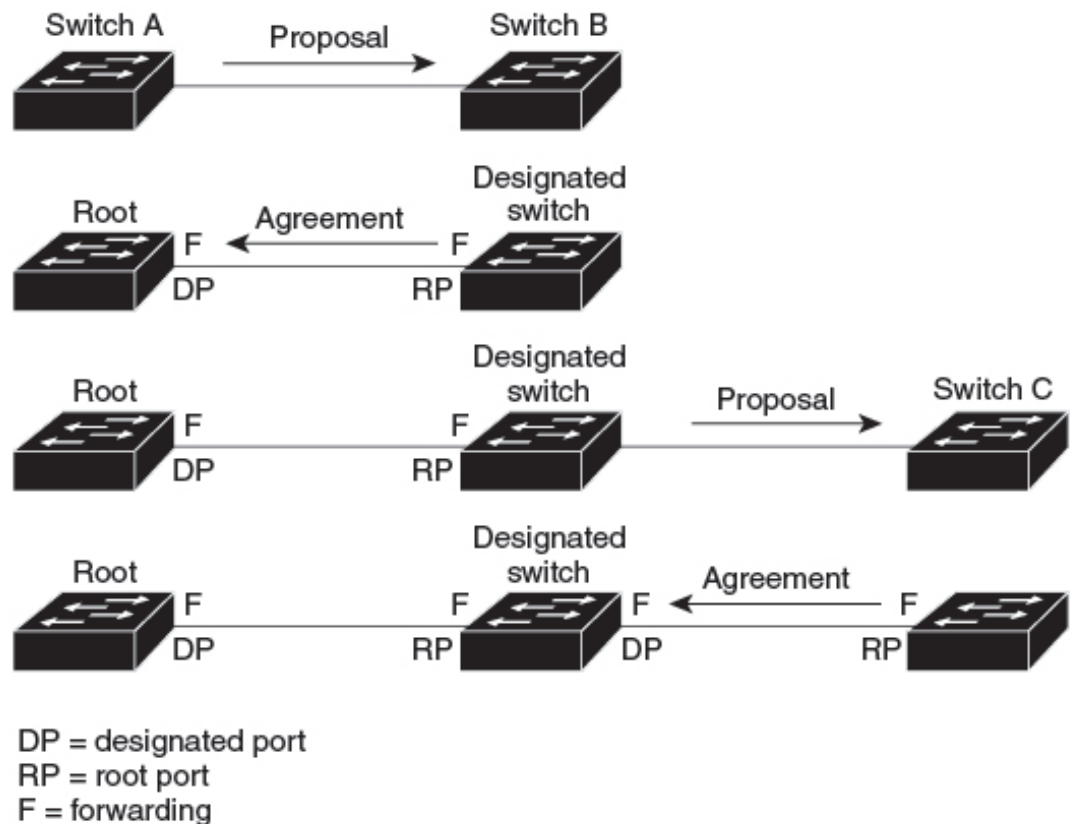
After receiving the proposal message, Switch B selects as its new root port the port from which the proposal message was received, forces all nonedge ports to the blocking state, and sends an agreement message (a BPDU with the agreement flag set) through its new root port.

After receiving Switch B's agreement message, Switch A also immediately transitions its designated port to the forwarding state. No loops in the network are formed because Switch B blocked all of its nonedge ports and because there is a point-to-point link between Switch A and B.

When Switch C is connected 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 immediately transition to the forwarding state. With each iteration of this handshaking process, one more device joins the active topology. As the network converges, this proposal-agreement handshaking progresses from the root toward the leaves of the spanning tree.

In a switch stack, the cross-stack rapid transition (CSRT) feature ensures that a stack member receives acknowledgments from all stack members during the proposal-agreement handshaking before moving the port to the forwarding state. CSRT is automatically enabled when the device is in MST mode.

The device learns the link type from the port duplex mode: a full-duplex port is considered to have a point-to-point connection; 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 using the **spanning-tree link-type** interface configuration command.



88760

Synchronization of Port Roles

When the device receives a proposal message on one of its ports and that port is selected as the new root port, the RSTP forces all other ports to synchronize with the new root information.

The device is synchronized with superior root information that is received on the root port if all other ports are synchronized. An individual port on the device is synchronized if:

- That port is in the blocking state.
- It is an edge port (a port that is 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 RSTP forces it to synchronize with new root information. In general, when the RSTP 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.

Figure 10: Sequence of Events During Rapid Convergence

After ensuring that all of the ports are synchronized, the device sends an agreement message to the designated device corresponding to its root port. When the devices that are connected by a point-to-point link are in agreement about their port roles, the RSTP immediately transitions the port states to forwarding.



88761

Bridge Protocol Data Unit Format and Processing

The RSTP BPDUs format is the same as the IEEE 802.1D BPDUs format except that the protocol version is set to 2. A new 1-byte Version 1 Length field is set to zero, which means that no version 1 protocol information is present.

Table 7: RSTP BPDUs Flags

Bit	Function
0	Topology change (TC)
1	Proposal
2–3:	Port role:
00	Unknown
01	Alternate port
10	Root port
11	Designated port
4	Learning
5	Forwarding
6	Agreement
7	Topology change acknowledgement (TCA)

The sending device sets the proposal flag in the RSTP BPDU to propose itself as the designated device on that LAN. The port role in the proposal message is always set to the designated port.

The sending device sets the agreement flag in the RSTP BPDU to accept the previous proposal. The port role in the agreement message is always set to the root port.

The RSTP does not have a separate topology change notification (TCN) BPDU. It uses the topology change (TC) flag to show the topology changes. However, for interoperability with IEEE 802.1D devices, the RSTP device processes and generates TCN BPDUs.

The learning and forwarding flags are set according to the state of the sending port.

Processing Superior Bridge Protocol Data Unit Information

If a port receives superior root information (lower device ID, lower path cost, and so forth) than currently stored for the port, the RSTP triggers a reconfiguration. If the port is proposed and is selected as the new root port, RSTP forces all the other ports to synchronize.

If the BPDU received is an RSTP BPDU with the proposal flag set, the device sends an agreement message after all of the other ports are synchronized. If the BPDU is an IEEE 802.1D BPDU, the device 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.

If the superior information that is received on the port causes the port to become a backup or alternate port, RSTP sets the port to the blocking state but does not send the agreement message. The designated port continues sending BPDUs with the proposal flag set until the forward-delay timer expires, at which time the port transitions to the forwarding state.

Processing Inferior Bridge Protocol Data Unit Information

If a designated port receives an inferior BPDU (such as a higher device ID or a higher path cost than currently stored for the port) with a designated port role, it immediately replies with its own information.

Topology Changes

This section describes the differences between the RSTP and the IEEE 802.1D in handling spanning-tree topology changes.

- **Detection**—Unlike IEEE 802.1D in which *any* transition between the blocking and the forwarding state causes a topology change, *only* transitions from the blocking to the forwarding state cause a topology change with RSTP (only an increase in connectivity is considered a topology change). State changes on an edge port do not cause a topology change. When an RSTP device detects a topology change, it deletes the learned information on all of its nonedge ports except on those from which it received the TC notification.
- **Notification**—Unlike IEEE 802.1D, which uses TCN BPDUs, the RSTP does not use them. However, for IEEE 802.1D interoperability, an RSTP device processes and generates TCN BPDUs.
- **Acknowledgement**—When an RSTP device receives a TCN message on a designated port from an IEEE 802.1D device, it replies with an IEEE 802.1D configuration BPDU with the TCA bit set. However, if the TC-while timer (the same as the topology-change timer in IEEE 802.1D) is active on a root port that is connected to an IEEE 802.1D device and a configuration BPDU with the TCA bit set is received, the TC-while timer is reset.

This behavior is only required to support IEEE 802.1D devices. The RSTP BPDUs never have the TCA bit set.

- Propagation—When an RSTP device receives a TC message from another device through a designated or root port, it propagates the change to all of its nonedge, designated ports and to the root port (excluding the port on which it is received). The device starts the TC-while timer for all such ports and flushes the information learned on them.
- Protocol migration—For backward compatibility with IEEE 802.1D devices, RSTP selectively sends IEEE 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 RSTP BPDUs are sent), and RSTP BPDUs are sent. While this timer is active, the device processes all BPDUs received on that port and ignores the protocol type.

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

Protocol Migration Process

A device running MSTP supports a built-in protocol migration mechanism that enables it to interoperate with legacy IEEE 802.1D devices. If this device receives a legacy IEEE 802.1D configuration BPDU (a BPDU with the protocol version set to 0), it sends only IEEE 802.1D BPDUs on that port. An MSTP device also can detect that a port is at the boundary of a region when it receives a legacy BPDU, an MST BPDU (Version 3) associated with a different region, or an RST BPDU (Version 2).

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

Default Multiple Spanning Tree Protocol Configuration

Table 8: Default MSTP Configuration

Feature	Default Setting
Spanning-tree mode	
Device priority (configurable on a per-CIST port basis)	32768
Spanning-tree port priority (configurable on a per-CIST port basis)	128
Spanning-tree port cost (configurable on a per-CIST port basis)	
Hello time	
Forward-delay time	
Maximum-aging time	20 seconds
Maximum hop count	20 hops

How to Configure Multiple Spanning Tree Protocol and Parameters

The following sections provide information about configuring MSTP and MSTP parameters:

Specifying the Multiple Spanning Tree Region Configuration and Enabling Multiple Spanning Tree Protocol

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 name.

A region can have one member or multiple members with the same MST configuration; each member must be capable of processing RSTP BPDUs. There is no limit to the number of MST regions in a network, but each region can only support up to 64 spanning-tree instances. You can assign a VLAN to only one spanning-tree instance at a time.

Procedure

	Command or Action	Purpose
Step 1	enable Example: Device> enable	Enables privileged EXEC mode. Enter your password if prompted.
Step 2	configure terminal Example: Device# configure terminal	Enters global configuration mode.
Step 3	spanning-tree mst configuration Example: Device(config)# spanning-tree mst configuration	Enters MST configuration mode.
Step 4	instance instance-id vlan vlan-range Example: Device(config-mst)# instance 1 vlan 10-20	Maps VLANs to an MST instance. <ul style="list-style-type: none"> • For <i>instance-id</i>, the range is 0 to 4094. • For vlan <i>vlan-range</i>, the range is 1 to 4094. <p>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.</p>

	Command or Action	Purpose
		<p>To specify a VLAN range, use a hyphen; for example, instance 1 vlan 1-63 maps VLANs 1 through 63 to MST instance 1.</p> <p>To specify a VLAN series, use a comma; for example, instance 1 vlan 10, 20, 30 maps VLANs 10, 20, and 30 to MST instance 1.</p>
Step 5	name <i>name</i> Example: Device(config-mst)# name region1	Specifies the configuration name. The <i>name</i> string has a maximum length of 32 characters and is case sensitive.
Step 6	revision <i>version</i> Example: Device(config-mst)# revision 1	Specifies the configuration revision number. The range is 0 to 65535.
Step 7	show pending Example: Device(config-mst)# show pending	Verifies your configuration by displaying the pending configuration.
Step 8	exit Example: Device(config-mst)# exit	Applies all changes, and returns to global configuration mode.
Step 9	spanning-tree mode mst Example: Device(config)# spanning-tree mode mst	<p>Enables MSTP. RSTP is also enabled.</p> <p>Changing spanning-tree modes can disrupt traffic because all spanning-tree instances are stopped for the previous mode and restarted in the new mode.</p> <p>You cannot run both MSTP and PVST+ or both MSTP and Rapid PVST+ at the same time.</p>
Step 10	end Example: Device(config)# end	Returns to privileged EXEC mode.

(Optional) Configuring the Root Device

To configure the root device, perform this procedure:

Before you begin

- An MST must be specified and enabled on the device.
- You must also know the specified MST instance ID.

Procedure

	Command or Action	Purpose
Step 1	enable Example: Device> enable	Enables privileged EXEC mode. Enter your password if prompted.
Step 2	configure terminal Example: Device# configure terminal	Enters global configuration mode.
Step 3	spanning-tree mst <i>instance-id</i> root primary Example: Device(config)# spanning-tree mst 0 root primary	Configures a device as the root device. For <i>instance-id</i> , 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 0 to 4094.
Step 4	end Example: Device(config)# end	Returns to privileged EXEC mode.

(Optional) Configuring a Secondary Root Device

When you configure a device with the extended system ID support as the secondary root, the device priority is modified from the default value (32768) to 28672. The device is then likely to become the root device for the specified instance if the primary root device fails. This is assuming that the other network devices use the default device priority of 32768 and therefore are unlikely to become the root device.

You can execute this command on more than one device to configure multiple backup root devices. Use the same network diameter and hello-time values that you used when you configured the primary root device with the **spanning-tree mst *instance-id* root primary** global configuration command.

To configure a secondary root device, perform this procedure:

Before you begin

- An MST must be specified and enabled on the device.
- You must also know the specified MST instance ID.

Procedure

	Command or Action	Purpose
Step 1	enable Example: Device> enable	Enables privileged EXEC mode. Enter your password if prompted.

	Command or Action	Purpose
Step 2	configure terminal Example: Device# <code>configure terminal</code>	Enters global configuration mode.
Step 3	spanning-tree mst <i>instance-id</i> root secondary Example: Device(config)# <code>spanning-tree mst 0 root secondary</code>	Configures a devices as the secondary root device. For <i>instance-id</i> , 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 0 to 4094.
Step 4	end Example: Device(config)# <code>end</code>	Returns to privileged EXEC mode.

(Optional) Configuring Port Priority

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



Note If the device is a member of a switch stack, you must use the **spanning-tree mst [*instance-id*] cost *cost*** interface configuration command instead of the **spanning-tree mst [*instance-id*] port-priority *priority*** interface configuration command to select a port to put in the forwarding state. Assign lower cost values to ports that you want selected first and higher cost values to ports that you want selected last.

To configure port priority, perform this procedure:

Before you begin

- An MST must be specified and enabled on the device.
- You must also know the specified MST instance ID and the interface used.

Procedure

	Command or Action	Purpose
Step 1	enable Example: Device> <code>enable</code>	Enables privileged EXEC mode. Enter your password if prompted.

	Command or Action	Purpose
Step 2	configure terminal Example: Device# configure terminal	Enters global configuration mode.
Step 3	interface <i>interface-id</i> Example: Device (config)# interface gigabitethernet 1/0/1	Specifies an interface to configure, and enters interface configuration mode.
Step 4	spanning-tree mst <i>instance-id</i> port-priority <i>priority</i> Example: Device (config-if)# spanning-tree mst 0 port-priority 64	Configures port priority. <ul style="list-style-type: none"> • For <i>instance-id</i>, 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 0 to 4094. • For <i>priority</i>, the range is 0 to 240 in increments of 16. The default is 128. The lower the number, the higher the priority. <p>The priority values are 0, 16, 32, 48, 64, 80, 96, 112, 128, 144, 160, 176, 192, 208, 224, and 240. All other values are rejected.</p>
Step 5	end Example: Device (config-if)# end	Returns to privileged EXEC mode.

The **show spanning-tree mst interface *interface-id*** privileged EXEC command displays information only if the port is in a link-up operative state. Otherwise, you can use the **show running-config interface** privileged EXEC command to confirm the configuration.

(Optional) Configuring Path Cost

The MSTP path cost default value is derived from the media speed of an interface. If a loop occurs, the MSTP uses 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 values that you want selected last. If all interfaces have the same cost value, the MSTP puts the interface with the lowest interface number in the forwarding state and blocks the other interfaces.

To configure path cost, perform this procedure:

Before you begin

- An MST must be specified and enabled on the device.
- You must also know the specified MST instance ID and the interface used.

Procedure

	Command or Action	Purpose
Step 1	enable Example: Device> enable	Enables privileged EXEC mode. Enter your password if prompted.
Step 2	configure terminal Example: Device# configure terminal	Enters global configuration mode.
Step 3	interface <i>interface-id</i> Example: Device(config)# interface gigabitethernet 1/0/1	Specifies an interface to configure, and enters interface configuration mode. Valid interfaces include physical ports and port-channel logical interfaces. The port-channel range is 1 to 48.
Step 4	spanning-tree mst <i>instance-id</i> cost <i>cost</i> Example: Device(config-if)# spanning-tree mst 0 cost 17031970	Configures the cost. If a loop occurs, the MSTP uses the path cost when selecting an interface to place into the forwarding state. A lower path cost represents higher-speed transmission. <ul style="list-style-type: none">• For <i>instance-id</i>, 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 0 to 4094.• For <i>cost</i>, the range is 1 to 200000000; the default value is derived from the media speed of the interface.
Step 5	end Example: Device(config-if)# end	Returns to privileged EXEC mode.

The **show spanning-tree mst interface *interface-id*** privileged EXEC command displays information only for ports that are in a link-up operative state. Otherwise, you can use the **show running-config** privileged EXEC command to confirm the configuration.

(Optional) Configuring the Device Priority

Changing the priority of a device makes it more likely to be chosen as the root switch whether it is a standalone switch or a switch in the stack.



Note Exercise care when using this command. For normal network configurations, we recommend that you use the **spanning-tree mst *instance-id* root primary** and the **spanning-tree mst *instance-id* root secondary** global configuration commands to specify a device as the root or secondary root device. You should modify the device priority only in circumstances where these commands do not work.

To configure the device priority, perform this procedure:

Before you begin

- An MST must be specified and enabled on the device.
- You must also know the specified MST instance ID used.

Procedure

	Command or Action	Purpose
Step 1	enable Example: Device> enable	Enables privileged EXEC mode. Enter your password if prompted.
Step 2	configure terminal Example: Device# configure terminal	Enters global configuration mode.
Step 3	spanning-tree mst <i>instance-id</i> priority <i>priority</i> Example: Device(config)# spanning-tree mst 0 priority 40960	Configures the device priority. <ul style="list-style-type: none"> • For <i>instance-id</i>, 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 0 to 4094. • For <i>priority</i>, the range is 0 to 61440 in increments of 4096; the default is 32768. The lower the number, the more likely the device will be chosen as the root switch. <p>Priority values are 0, 4096, 8192, 12288, 16384, 20480, 24576, 28672, 32768, 36864, 40960, 45056, 49152, 53248, 57344, and 61440. These are the only acceptable values.</p>
Step 4	end Example: Device(config-if)# end	Returns to privileged EXEC mode.

(Optional) Configuring the Hello Time

The hello time is the time interval between configuration messages that are generated and sent by the root device.

To configure the hello time, perform this procedure:

Before you begin

An MST must be specified and enabled on the device.

Procedure

	Command or Action	Purpose
Step 1	enable Example: Device> enable	Enables privileged EXEC mode. Enter your password if prompted.
Step 2	configure terminal Example: Device# configure terminal	Enters global configuration mode.
Step 3	spanning-tree mst hello-time <i>seconds</i> Example: Device(config)# spanning-tree mst hello-time 4	Configures the hello time for all MST instances. The hello time is the time interval between configuration messages that are generated and sent by the root device. These messages indicate that the device is alive. For <i>seconds</i> , the range is 1 to 10; the default is 3.
Step 4	end Example: Device(config)# end	Returns to privileged EXEC mode.

Configuring the Forwarding-Delay Time

To configure the forwarding-delay time, perform this procedure:

Before you begin

An MST must be specified and enabled on the device.

Procedure

	Command or Action	Purpose
Step 1	enable Example:	Enables privileged EXEC mode. Enter your password if prompted.

	Command or Action	Purpose
	Device> enable	
Step 2	configure terminal Example: Device# configure terminal	Enters global configuration mode.
Step 3	spanning-tree mst forward-time seconds Example: Device(config)# spanning-tree mst forward-time 25	Configures the forward time for all MST instances. The forwarding delay is the number of seconds a port waits before changing from its spanning-tree learning and listening states to the forwarding state. For <i>seconds</i> , the range is 4 to 30; the default is 20.
Step 4	end Example: Device(config)# end	Returns to privileged EXEC mode.

Configuring the Maximum-Aging Time

To configure the maximum-aging time, perform this procedure:

Before you begin

An MST must be specified and enabled on the device.

Procedure

	Command or Action	Purpose
Step 1	enable Example: Device> enable	Enables privileged EXEC mode. Enter your password if prompted.
Step 2	configure terminal Example: Device# configure terminal	Enters global configuration mode.
Step 3	spanning-tree mst max-age seconds Example: Device(config)# spanning-tree mst max-age 40	Configures the maximum-aging time for all MST instances. The maximum-aging time is the number of seconds a device waits without receiving spanning-tree configuration messages before attempting a reconfiguration. For <i>seconds</i> , the range is 6 to 40; the default is 20.

	Command or Action	Purpose
Step 4	end Example: Device(config) # end	Returns to privileged EXEC mode.

(Optional) Configuring the Maximum-Hop Count

To configure the maximum-hop count, perform this procedure:

Before you begin

An MST must be specified and enabled on the device.

Procedure

	Command or Action	Purpose
Step 1	enable Example: Device> enable	Enables privileged EXEC mode. Enter your password if prompted.
Step 2	configure terminal Example: Device# configure terminal	Enters global configuration mode.
Step 3	spanning-tree mst max-hops <i>hop-count</i> Example: Device(config) # spanning-tree mst max-hops 25	Specifies the number of hops in a region before the BPDU is discarded, and the information that is held for a port is aged. For <i>hop-count</i> , the range is 1 to 255; the default is 20.
Step 4	end Example: Device(config) # end	Returns to privileged EXEC mode.

(Optional) Specifying the Link Type to Ensure Rapid Transitions

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

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 device running MSTP, you can override the default setting of the link type and enable rapid transitions to the forwarding state.

To specify the link type to ensure rapid transitions, perform this procedure:

(Optional) Designating the Neighbor Type**Before you begin**

- An MST must be specified and enabled on the device.
- You must also know the specified MST instance ID and the interface used.

Procedure

	Command or Action	Purpose
Step 1	enable Example: Device> enable	Enables privileged EXEC mode. Enter your password if prompted.
Step 2	configure terminal Example: Device# configure terminal	Enters global configuration mode.
Step 3	interface interface-id Example: Device(config)# interface gigabitethernet 1/0/1	Specifies an interface to configure, and enters interface configuration mode. Valid interfaces include physical ports, VLANs, and port-channel logical interfaces. The VLAN ID range is 1 to 4094. The port-channel range is 1 to 48.
Step 4	spanning-tree link-type point-to-point Example: Device(config-if)# spanning-tree link-type point-to-point	Specifies that the link type of a port is point-to-point.
Step 5	end Example: Device(config-if)# end	Returns to privileged EXEC mode.

(Optional) Designating the Neighbor Type

A topology could contain both prestandard and IEEE 802.1s standard compliant devices. By default, ports can automatically detect prestandard devices, but they can still receive both standard and prestandard BPDUs. When there is a mismatch between a device and its neighbor, only the CIST runs on the interface.

You can choose to set a port to send only prestandard BPDUs. The prestandard flag appears in all the **show** commands, even if the port is in STP compatibility mode.

To designate the neighbor type, perform this procedure:

Before you begin

An MST must be specified and enabled on the device.

Procedure

	Command or Action	Purpose
Step 1	enable Example: Device> enable	Enables privileged EXEC mode. Enter your password if prompted.
Step 2	configure terminal Example: Device# configure terminal	Enters global configuration mode.
Step 3	interface interface-id Example: Device(config)# interface gigabitethernet 1/0/1	Specifies an interface to configure, and enters interface configuration mode. Valid interfaces include physical ports.
Step 4	spanning-tree mst pre-standard Example: Device(config-if)# spanning-tree mst pre-standard	Specifies that the port can send only prestandard BPDUs.
Step 5	end Example: Device(config-if)# end	Returns to privileged EXEC mode.

Restarting the Protocol Migration Process

This procedure restarts the protocol migration process and forces renegotiation with neighboring devices. It reverts the device to MST mode. It is needed when the device no longer receives IEEE 802.1D BPDUs after it has been receiving them.

Follow these steps to restart the protocol migration process (force the renegotiation with neighboring devices) on the device.

Before you begin

- An MST must be specified and enabled on the device.
- If you want to use the interface version of the command, you must also know the MST interface used.

Procedure

	Command or Action	Purpose
Step 1	enable Example: Device> enable	Enables privileged EXEC mode. Enter your password if prompted.

	Command or Action	Purpose
Step 2	Enter one of the following commands: <ul style="list-style-type: none"> • clear spanning-tree detected-protocols • clear spanning-tree detected-protocols interface <i>interface-id</i> Example: Device# clear spanning-tree detected-protocols or Device# clear spanning-tree detected-protocols interface gigabitethernet 1/0/1	The device reverts to the MSTP mode, and the protocol migration process restarts.

What to do next

This procedure may need to be repeated if the device receives more legacy IEEE 802.1D configuration BPDUs (BPDUs with the protocol version set to 0).

Feature History for Multiple Spanning Tree Protocol

This table provides release and related information for features explained in this module.

These features are available on all releases subsequent to the one they were introduced in, unless noted otherwise.

Release	Feature	Feature Information
Cisco IOS XE Gibraltar 16.11.1	Multiple Spanning-Tree Protocol	MSTP, which uses RSTP for rapid convergence, enables multiple VLANs to be grouped into and mapped to the same spanning-tree instance, reducing the number of spanning-tree instances needed to support a large number of VLANs.

Use Cisco Feature Navigator to find information about platform and software image support. To access Cisco Feature Navigator, go to <http://www.cisco.com/go/cfn>.



CHAPTER 4

Configuring Optional Spanning-Tree Features

- [Information About Optional Spanning-Tree Features, on page 57](#)
- [How to Configure Optional Spanning-Tree Features, on page 72](#)
- [Monitoring the Spanning-Tree Status, on page 86](#)
- [Additional References for Optional Spanning Tree Features, on page 87](#)
- [Feature History for Optional Spanning Tree Features, on page 87](#)

Information About Optional Spanning-Tree Features

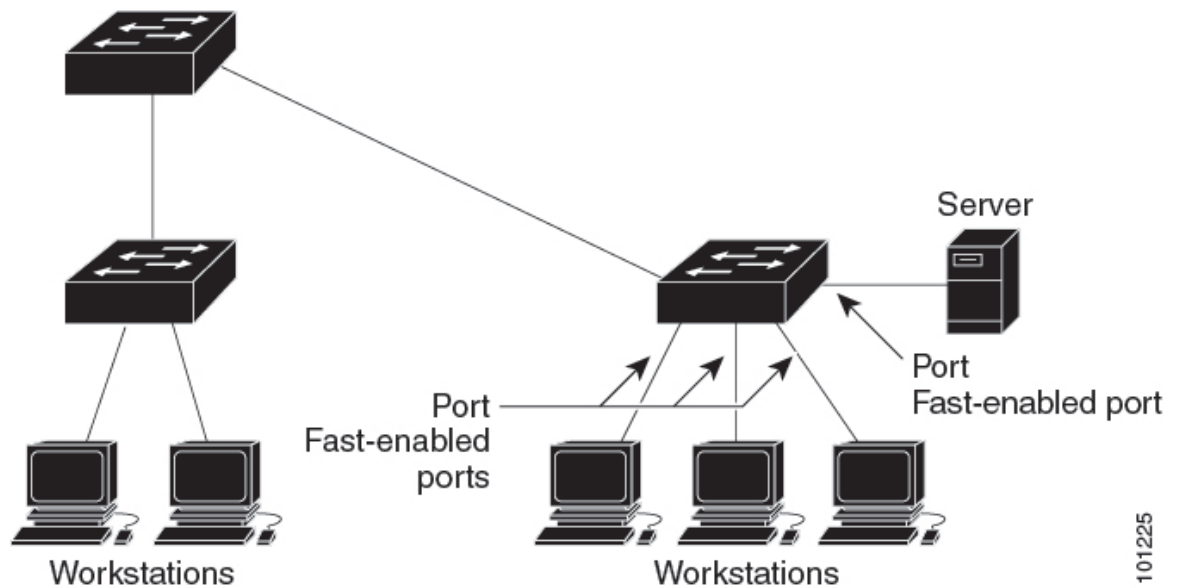
The following sections provide information about Optional Spanning-Tree features:

PortFast

PortFast immediately brings an interface that is configured as an access or trunk port to the forwarding state from a blocking state, bypassing the listening and learning states.

Figure 11: PortFast-Enabled Interfaces

You can use PortFast on interfaces that are connected to a single workstation or server to allow those devices to immediately connect to the network, rather than waiting for the spanning tree to converge.



Interfaces connected to a single workstation or server should not receive bridge protocol data units (BPDUs). An interface with PortFast enabled goes through the normal cycle of spanning-tree status changes when the switch is restarted.

You can enable this feature by enabling it on either the interface or on all nontrunking ports.

Spanning Tree Protocol PortFast Port Types

You can configure a spanning tree port (STP) 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. You can configure the port type either globally or per interface.

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:

- A PortFast edge port: It is connected to a Layer 2 host. This can be either an access port or an edge trunk port (**portfast edge trunk**). This type of port interface immediately transitions to the forwarding state, bypassing the listening and learning states. Use PortFast edge on Layer 2 access ports that are connected to a single workstation or server to allow those devices to connect to the network immediately, rather than waiting for spanning tree to converge.

Even if the interface receives a bridge protocol data unit (BPDU), spanning tree does not place the port into the blocking state. Spanning tree sets the operating state of the port to nonport fast even if the configured state remains port fast edge and starts participating in the topology change.



Note If you configure a port that is connected to a Layer 2 switch or bridge as an edge port, you might create a bridging loop.

- A PortFast network port: It is connected only to a Layer 2 switch or bridge.

Bridge Assurance is enabled only on PortFast network ports. For more information, see [#unique_99](#).



Note If you configure a port that is connected to a Layer 2 host as a spanning tree network port, the port automatically moves into the blocking state.

- A PortFast normal port: It is the default type of spanning tree port.



Note If you enter the **spanning-tree portfast trunk** command in the global or interface configuration mode, the system automatically saves it as **spanning-tree portfast edge trunk**.

Bridge Protocol Data Unit Guard

The Bridge Protocol Data Unit (BPDU) guard feature can be globally enabled on the switch or can be enabled per port, but the feature operates with some differences.

When you enable BPDU guard at the global level on PortFast edge-enabled ports, spanning tree shuts down ports that are in a PortFast edge-operational state if any BPDU is received on them. In a valid configuration, PortFast edge-enabled ports do not receive BPDUs. Receiving a BPDU on a PortFast edge-enabled port means an invalid configuration, such as the connection of an unauthorized device, and the BPDU guard feature puts the port in the error-disabled state. When this happens, the switch shuts down the entire port on which the violation occurred.

When you enable BPDU guard at the interface level on any port without also enabling the PortFast edge feature, and the port receives a BPDU, it is put in the error-disabled state.

The BPDU guard feature provides a secure response to invalid configurations because you must manually put the interface back in service. Use the BPDU guard feature in a service-provider network to prevent an access port from participating in the spanning tree.

Bridge Protocol Data Unit Filtering

The BPDU filtering feature can be globally enabled on the switch or can be enabled per interface, but the feature operates with some differences.

Enabling BPDU filtering on PortFast edge-enabled interfaces at the global level keeps those interfaces that are in a PortFast edge-operational state from sending or receiving BPDUs. The interfaces still send a few BPDUs at link-up before the switch begins to filter outbound BPDUs. You should globally enable BPDU filtering on a switch so that hosts that are connected to these interfaces do not receive BPDUs. If a BPDU is received on a PortFast edge-enabled interface, the interface loses its PortFast edge-operational status, and BPDU filtering is disabled.

Enabling BPDU filtering on an interface without also enabling the PortFast edge feature keeps the interface from sending or receiving BPDUs.



Caution Enabling BPDU filtering on an interface is the same as disabling spanning tree on it and can result in spanning-tree loops.

You can enable the BPDU filtering feature for the entire switch or for an interface.

Bridge Assurance

You can use Bridge Assurance to help prevent looping conditions that are caused by unidirectional links (one-way traffic on a link or port), or a malfunction in a neighboring switch. Here, a malfunction refers to a switch that is not able to run STP any more, while still forwarding traffic (a brain dead switch).

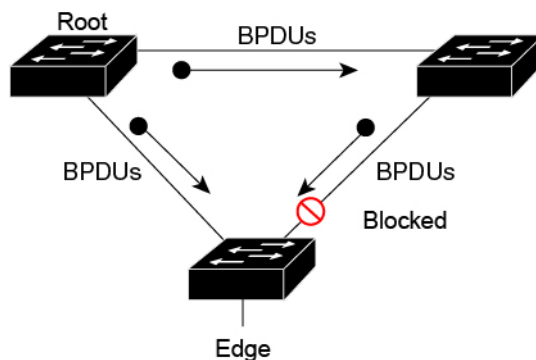
BPDUs are sent out on all operational network ports, including alternate and backup ports, for each hello time period. Bridge Assurance monitors the receipt of BPDUs on point-to-point links on all network ports. When a port does not receive BPDUs within the allotted hello time period, the port is put into a blocked state (the same as a port inconsistent state, which stops forwarding of frames). When the port resumes receipt of BPDUs, the port resumes normal spanning tree operations.



Note Only Rapid PVST+ and MST spanning tree protocols support Bridge Assurance. PVST+ does not support Bridge Assurance.

The following example shows how Bridge Assurance protects your network from bridging loops. Here, [Figure 12: Network with Normal STP Topology](#), on page 60 shows a normal STP topology, and [Figure 13: Network Loop Due to a Malfunctioning Switch](#), on page 61 demonstrates a potential network problem when the device fails (brain dead) and Bridge Assurance is not enabled on the network.

Figure 12: Network with Normal STP Topology



354159

Figure 13: Network Loop Due to a Malfunctioning Switch

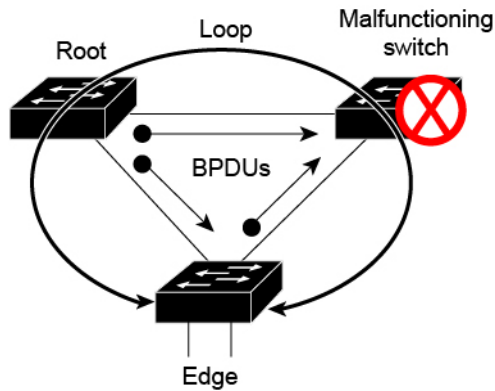


Figure 14: Network with STP Topology Running Bridge Assurance , on page 61 shows that the network with Bridge Assurance enabled, and the STP topology progressing normally with bidirectional BPDUs issuing from every STP network port. Figure 15: Network Problem Averted with Bridge Assurance Enabled, on page 62 shows how the potential network problem shown in Figure 13: Network Loop Due to a Malfunctioning Switch , on page 61 does not occur when you have Bridge Assurance enabled on your network.

Figure 14: Network with STP Topology Running Bridge Assurance

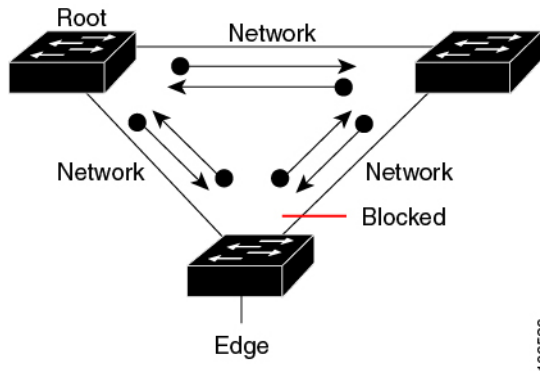
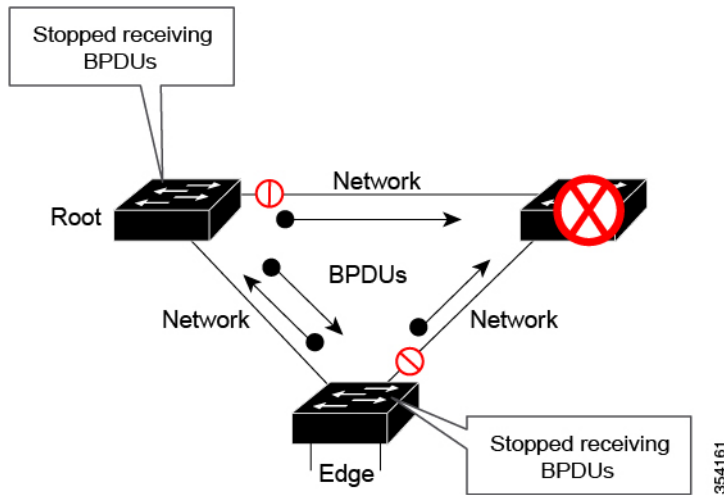


Figure 15: Network Problem Averted with Bridge Assurance Enabled



The system generates syslog messages when a port is blocked or unblocked. The following sample outputs show the log that is generated for each of these states:

Blocked port:

```
Sep 17 09:48:16.249 PDT: %SPANTREE-2-BRIDGE_ASSURANCE_BLOCK: Bridge Assurance blocking port GigabitEthernet5/8 on VLAN0200. (stack-dut-R4-4)
```

Unblocked Port:

```
Sep 17 09:48:58.426 PDT: %SPANTREE-2-BRIDGE_ASSURANCE_UNBLOCK: Bridge Assurance unblocking port GigabitEthernet5/8 on VLAN0200. (stack-dut-R4-4)
```

Guidelines for Configuring Bridge Assurance

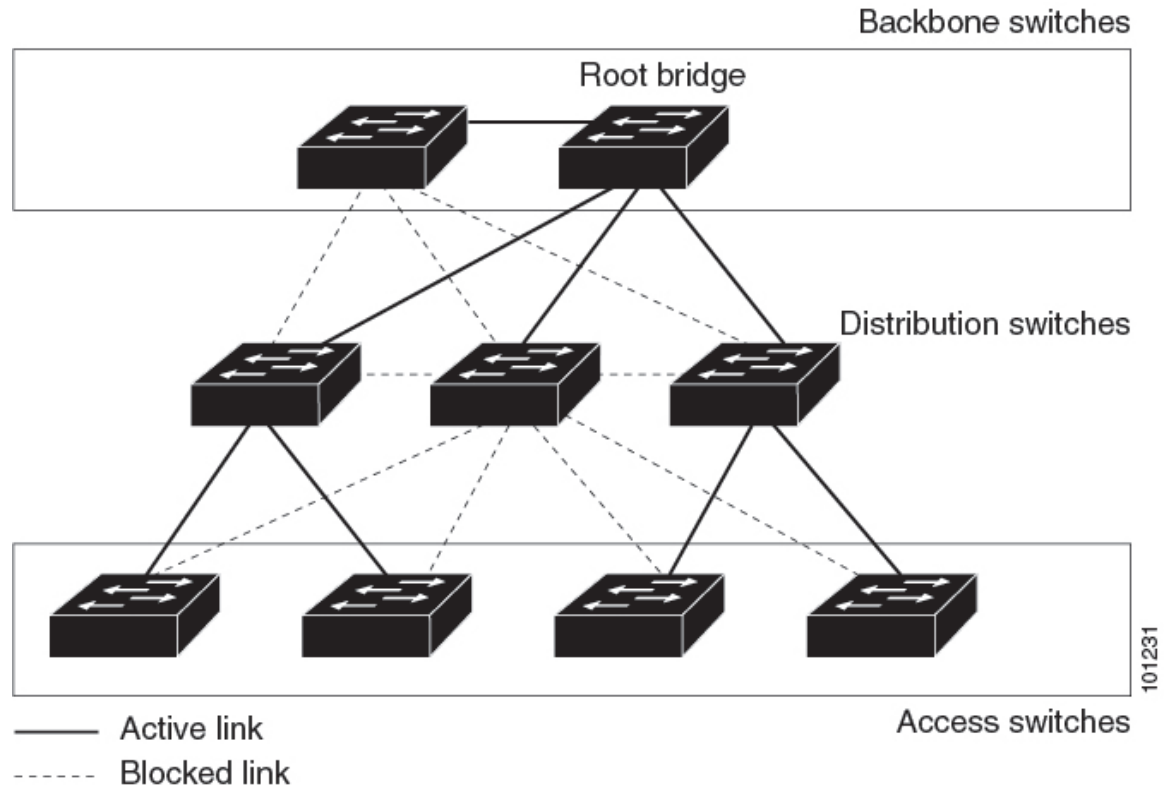
Observe these guidelines when configuring Bridge Assurance:

- Bridge Assurance can be enabled or disabled globally.
- Bridge Assurance applies to all operational network ports, including alternate and backup ports.
- Only Rapid PVST+ and MST spanning tree protocols support Bridge Assurance. PVST+ does not support Bridge Assurance.
- For Bridge Assurance to work properly, it must be supported and configured on both ends of a point-to-point link. If the device on one side of the link has Bridge Assurance enabled and the device on the other side does not, then the connecting port is blocked (a Bridge Assurance inconsistent state). We recommend that you enable Bridge Assurance throughout your network.
- To enable Bridge Assurance on a port, BPDUs filtering and BPDUs Guard must be disabled.
- You can enable Bridge Assurance along with Loop Guard.
- You can enable Bridge Assurance along with Root Guard. The latter is designed to provide a way to enforce the root bridge placement in the network.

UplinkFast

Figure 16: Switches in a Hierarchical Network

Switches in hierarchical networks can be grouped into backbone switches, distribution switches, and access switches. This complex network has distribution switches and access switches that each have at least one redundant link that spanning tree blocks to prevent loops.



If a switch loses connectivity, it begins using the alternate paths when the spanning tree selects a new root port. You can accelerate the choice of a new root port when a link or switch fails or when the spanning tree reconfigures itself by enabling UplinkFast. The root port transitions to the forwarding state immediately without going through the listening and learning states, as it would with the normal spanning-tree procedures.

When the spanning tree reconfigures the new root port, other interfaces flood the network with multicast packets, one for each address that was learned on the interface. You can limit these bursts of multicast traffic by reducing the max-update-rate parameter (the default for this parameter is 150 packets per second). However, if you enter zero, station-learning frames are not generated, so the spanning-tree topology converges more slowly after a loss of connectivity.



Note UplinkFast is most useful in wiring-closet switches at the access or edge of the network. It is not appropriate for backbone devices. This feature might not be useful for other types of applications.

UplinkFast provides fast convergence after a direct link failure and achieves load-balancing between redundant Layer 2 links using uplink groups. An uplink group is a set of Layer 2 interfaces (per VLAN), only one of which is forwarding at any given time. Specifically, an uplink group consists of the root port (which is

forwarding) and a set of blocked ports, except for self-looping ports. The uplink group provides an alternate path in case the currently forwarding link fails.

Figure 17: UplinkFast Example Before Direct Link Failure

This topology has no link failures. Switch A, the root switch, is connected directly to Switch B over link L1 and to Switch C over link L2. The Layer 2 interface on Switch C that is connected directly to Switch B is in a blocking state.

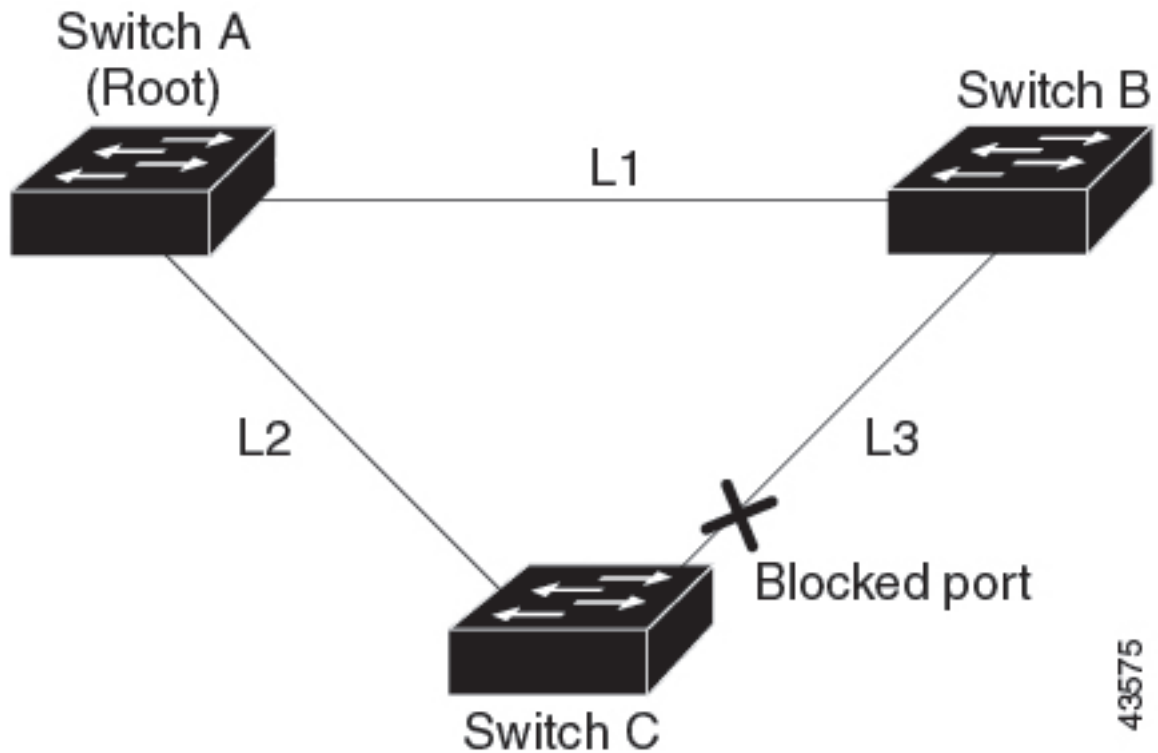
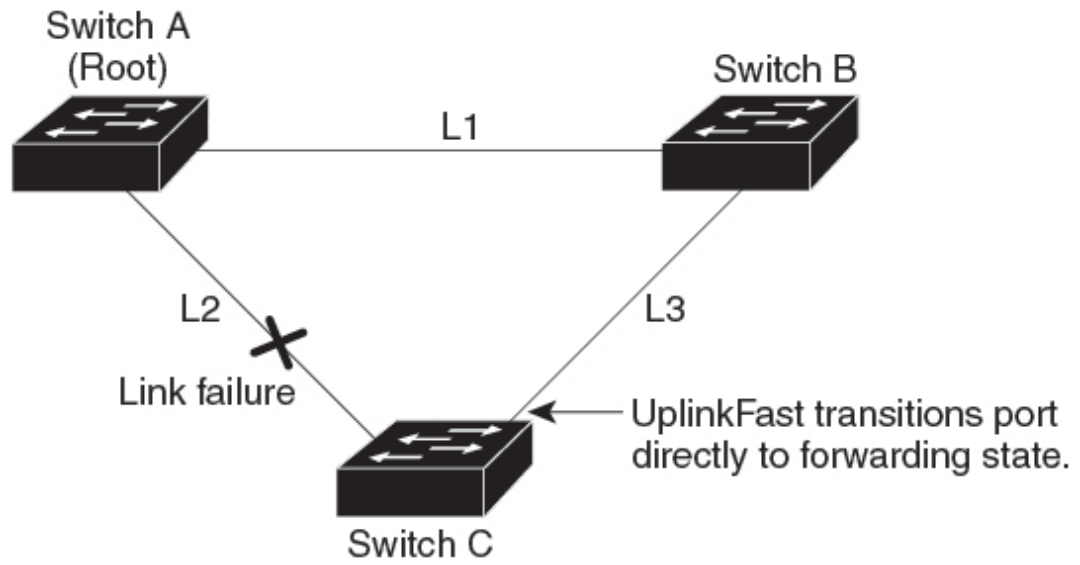


Figure 18: UplinkFast Example After Direct Link Failure

If Switch C detects a link failure on the currently active link L2 on the root port (a direct link failure), UplinkFast unblocks the blocked interface on Switch C and transitions it to the forwarding state without going through the listening and learning states. This change takes approximately 1 to 5 seconds.



43576

Cross-Stack UplinkFast

Cross-Stack UplinkFast (CSUF) provides a fast spanning-tree transition (fast convergence in less than 1 second under normal network conditions) across a switch stack. During the fast transition, an alternate redundant link on the switch stack is placed in the forwarding state without causing temporary spanning-tree loops or loss of connectivity to the backbone. With this feature, you can have a redundant and resilient network in some configurations. CSUF is automatically enabled when you enable the UplinkFast feature.

CSUF might not provide a fast transition all the time; in these cases, the normal spanning-tree transition occurs, completing in 30 to 40 seconds. For more information, see [Related Topics](#).

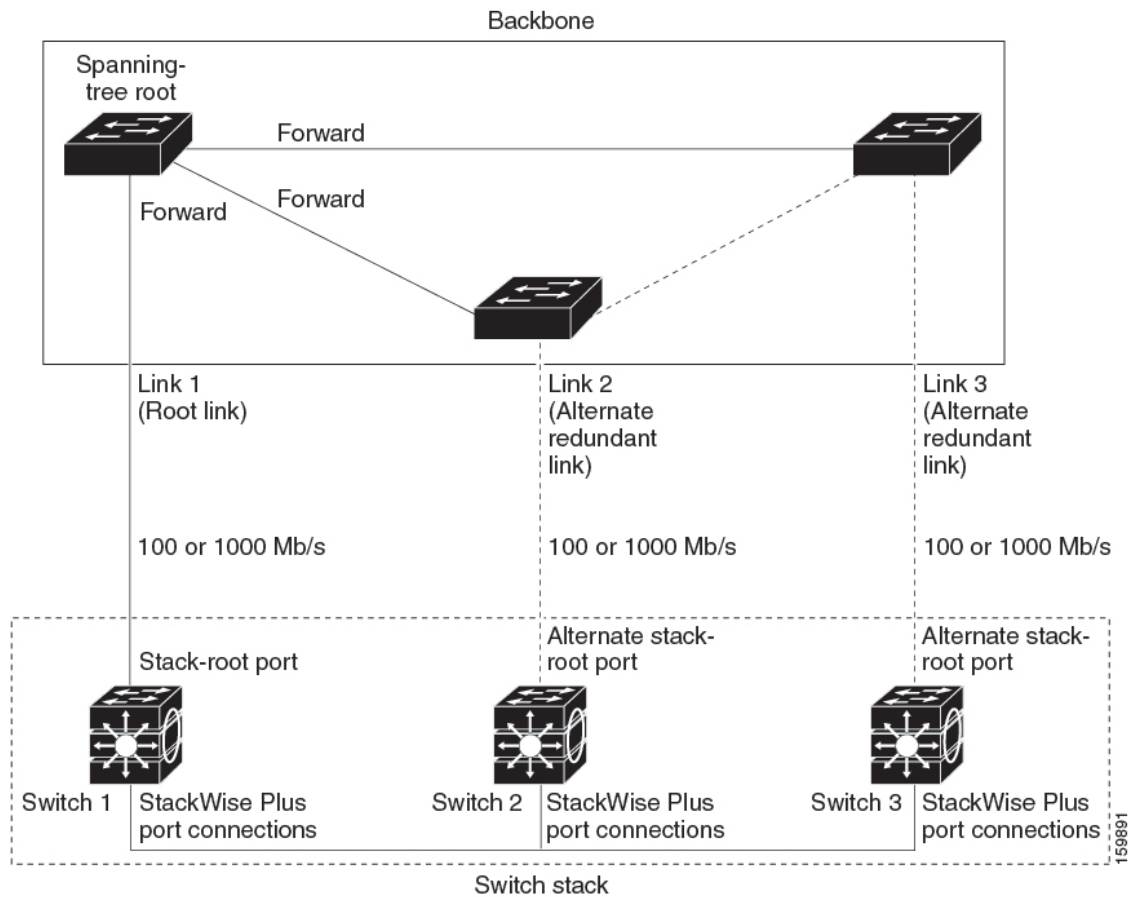
How Cross-Stack UplinkFast Works

Cross-Stack UplinkFast (CSUF) ensures that one link in the stack is elected as the path to the root.

Figure 19: Cross-Stack UplinkFast Topology

The stack-root port on Switch 1 provides the path to the root of the spanning tree. The alternate stack-root ports on Switches 2 and 3 can provide an alternate path to the spanning-tree root if the current stack-root switch fails or if its link to the spanning-tree root fails.

Link 1, the root link, is in the spanning-tree forwarding state. Links 2 and 3 are alternate redundant links that are in the spanning-tree blocking state. If Switch 1 fails, if its stack-root port fails, or if Link 1 fails, CSUF selects either the alternate stack-root port on Switch 2 or Switch 3 and puts it into the forwarding state in less than 1 second.



When certain link loss or spanning-tree events occur (described in the following topic), the Fast Uplink Transition Protocol uses the neighbor list to send fast-transition requests to stack members.

The switch sending the fast-transition request needs to do a fast transition to the forwarding state of a port that it has chosen as the root port, and it must obtain an acknowledgment from each stack switch before performing the fast transition.

Each switch in the stack decides if the sending switch is a better choice than itself to be the stack root of this spanning-tree instance by comparing the root, cost, and bridge ID. If the sending switch is the best choice as the stack root, each switch in the stack returns an acknowledgment; otherwise, it sends a fast-transition request. The sending switch then has not received acknowledgments from all stack switches.

When acknowledgments are received from all stack switches, the Fast Uplink Transition Protocol on the sending switch immediately transitions its alternate stack-root port to the forwarding state. If acknowledgments from all stack switches are not obtained by the sending switch, the normal spanning-tree transitions (blocking, listening, learning, and forwarding) take place, and the spanning-tree topology converges at its normal rate ($2 * \text{forward-delay time} + \text{max-age time}$).

The Fast Uplink Transition Protocol is implemented on a per-VLAN basis and affects only one spanning-tree instance at a time.

Events That Cause Fast Convergence

Depending on the network event or failure, the CSUF fast convergence might or might not occur.

Fast convergence (less than 1 second under normal network conditions) occurs under these circumstances:

- The stack-root port link fails.
If two switches in the stack have alternate paths to the root, only one of the switches performs the fast transition.
- The failed link, which connects the stack root to the spanning-tree root, recovers.
- A network reconfiguration causes a new stack-root switch to be selected.
- A network reconfiguration causes a new port on the current stack-root switch to be chosen as the stack-root port.



Note The fast transition might not occur if multiple events occur simultaneously. For example, if a stack member is powered off, and at the same time, the link connecting the stack root to the spanning-tree root comes back up, the normal spanning-tree convergence occurs.

Normal spanning-tree convergence (30 to 40 seconds) occurs under these conditions:

- The stack-root switch is powered off, or the software failed.
- The stack-root switch, which was powered off or failed, is powered on.
- A new switch, which might become the stack root, is added to the stack.

BackboneFast

BackboneFast detects indirect failures in the core of the backbone. BackboneFast is a complementary technology to the UplinkFast feature, which responds to failures on links that are directly connected to access switches. BackboneFast optimizes the maximum-age timer, which controls the amount of time the switch stores protocol information that is received on an interface. When a switch receives an inferior BPDU from the designated port of another switch, the BPDU is a signal that the other switch might have lost its path to the root, and BackboneFast tries to find an alternate path to the root.

BackboneFast starts when a root port or blocked interface on a switch receives inferior BPDUs from its designated switch. An inferior BPDU identifies a switch that declares itself as both the root bridge and the designated switch. When a switch receives an inferior BPDU, it means that a link to which the switch is not directly connected (an indirect link) has failed (that is, the designated switch has lost its connection to the root switch). Under spanning-tree rules, the switch ignores inferior BPDUs for the maximum aging time (default is 20 seconds).

The switch tries to find if it has an alternate path to the root switch. If the inferior BPDU arrives on a blocked interface, the root port and other blocked interfaces on the switch become alternate paths to the root switch. (Self-looped ports are not considered alternate paths to the root switch.) If the inferior BPDU arrives on the root port, all blocked interfaces become alternate paths to the root switch. If the inferior BPDU arrives on the root port and there are no blocked interfaces, the switch assumes that it has lost connectivity to the root switch, causes the maximum aging time on the root port to expire, and becomes the root switch according to normal spanning-tree rules.

If the switch has alternate paths to the root switch, it uses these alternate paths to send a root link query (RLQ) request. The switch sends the RLQ request on all alternate paths to learn if any stack member has an alternate

root to the root switch and waits for an RLQ reply from other switches in the network and in the stack. The switch sends the RLQ request on all alternate paths and waits for an RLQ reply from other switches in the network.

When a stack member receives an RLQ reply from a nonstack member on a blocked interface and the reply is destined for another nonstacked switch, it forwards the reply packet, regardless of the spanning-tree interface state.

When a stack member receives an RLQ reply from a nonstack member and the response is destined for the stack, the stack member forwards the reply so that all the other stack members receive it.

If the switch discovers that it still has an alternate path to the root, it expires the maximum aging time on the interface that received the inferior BPDU. If all the alternate paths to the root switch indicate that the switch has lost connectivity to the root switch, the switch expires the maximum aging time on the interface that received the RLQ reply. If one or more alternate paths can still connect to the root switch, the switch makes all interfaces on which it received an inferior BPDU its designated ports and moves them from the blocking state (if they were in the blocking state), through the listening and learning states, and into the forwarding state.

Figure 20: BackboneFast Example Before Indirect Link Failure

This is an example topology with no link failures. Switch A, the root switch, connects directly to Switch B over link L1 and to Switch C over link L2. The Layer 2 interface on Switch C that connects directly to Switch B is in the blocking state.

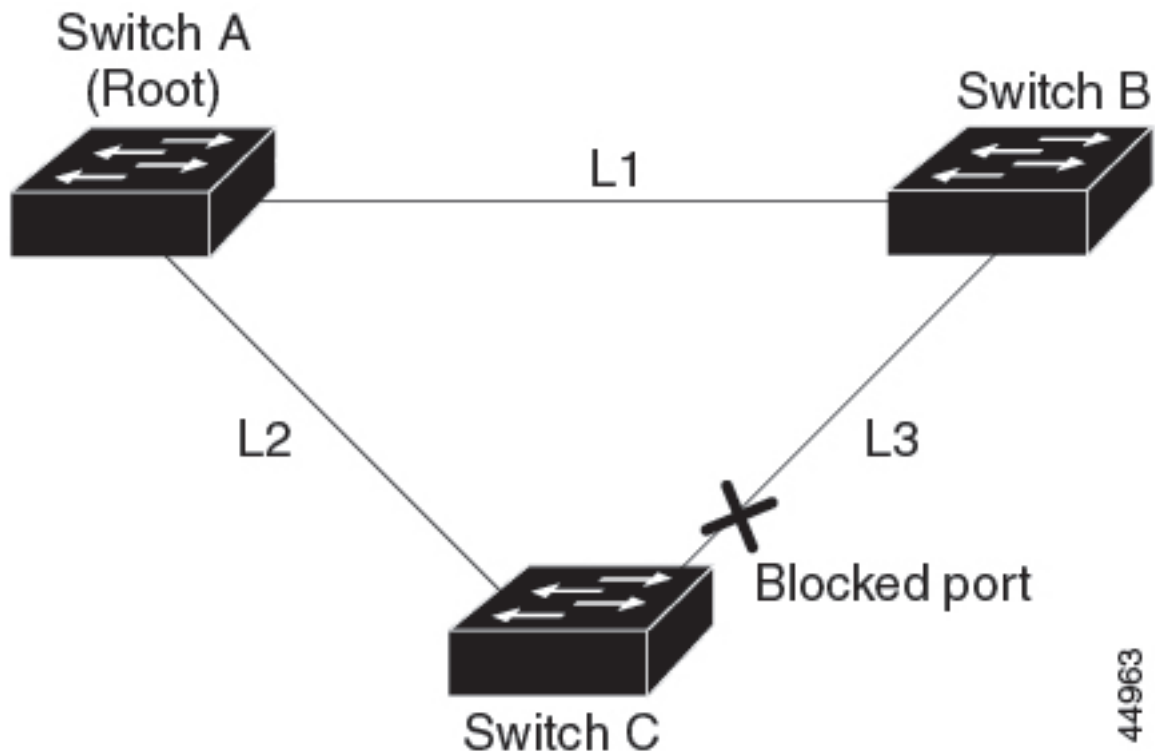
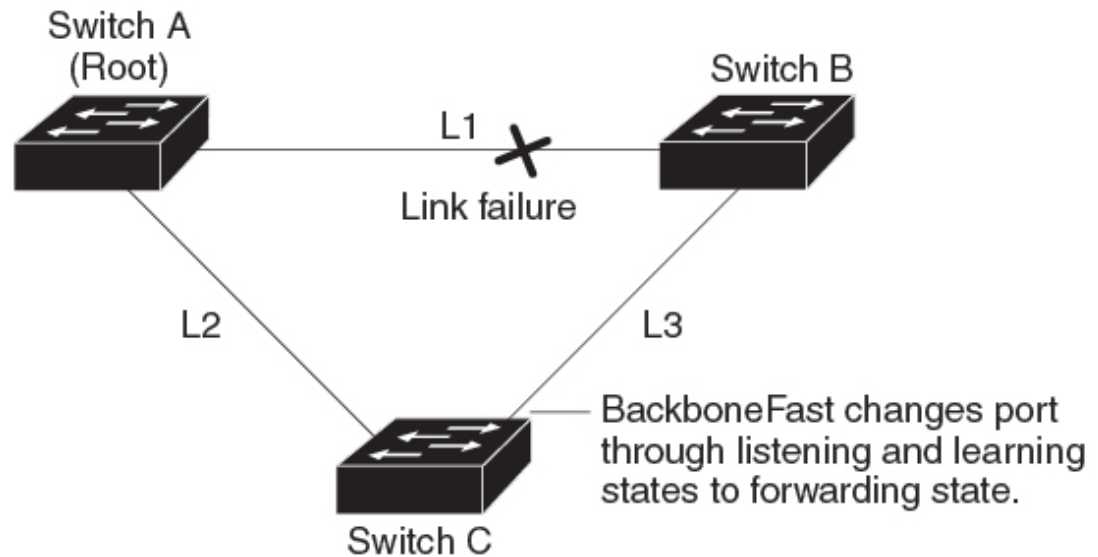


Figure 21: BackboneFast Example After Indirect Link Failure

If link L1 fails, Switch C cannot detect this failure because it is not connected directly to link L1. However, because Switch B is directly connected to the root switch over L1, it detects the failure, elects itself the root, and begins sending BPDUs to Switch C, identifying itself as the root. When Switch C receives the inferior

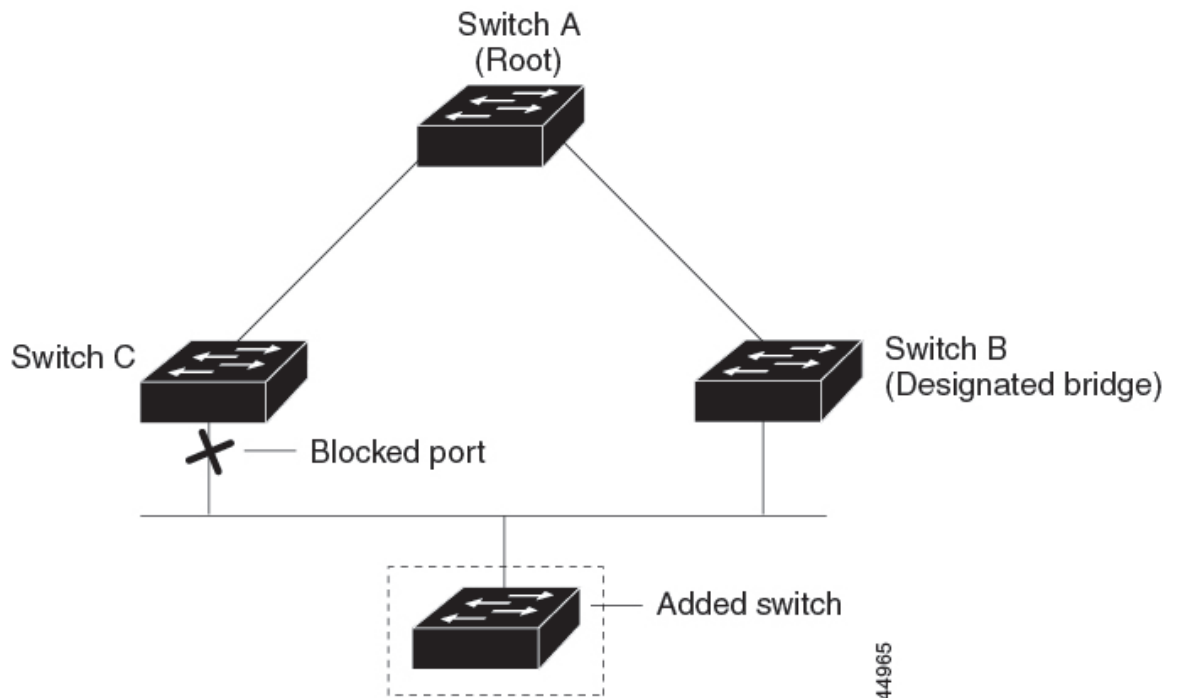
BPDUs from Switch B, Switch C assumes that an indirect failure has occurred. At that point, BackboneFast allows the blocked interface on Switch C to move immediately to the listening state without waiting for the maximum aging time for the interface to expire. BackboneFast then transitions the Layer 2 interface on Switch C to the forwarding state, providing a path from Switch B to Switch A. The root-switch election takes approximately 30 seconds, twice the Forward Delay time if the default Forward Delay time of 15 seconds is set. BackboneFast reconfigures the topology to account for the failure of link L1.



44964

Figure 22: Adding a Switch in a Shared-Medium Topology

If a new switch is introduced into a shared-medium topology, BackboneFast is not activated because the inferior BPDUs did not come from the recognized designated switch (Switch B). The new switch begins sending inferior BPDUs that indicate it is the root switch. However, the other switches ignore these inferior BPDUs, and the new switch learns that Switch B is the designated switch to Switch A, the root switch.



EtherChannel Guard

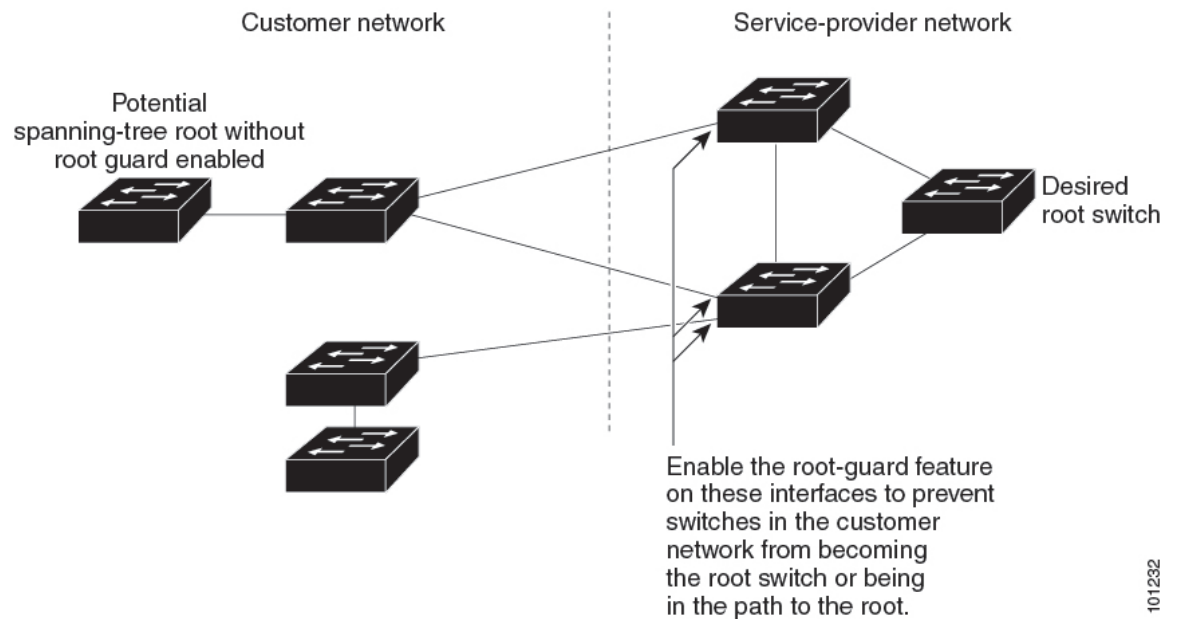
You can use EtherChannel guard to detect an EtherChannel misconfiguration between the switch and a connected device. A misconfiguration can occur if the switch interfaces are configured in an EtherChannel, but the interfaces on the other device are not. A misconfiguration can also occur if the channel parameters are not the same at both ends of the EtherChannel.

If the switch detects a misconfiguration on the other device, EtherChannel guard places the switch interfaces in the error-disabled state, and displays an error message.

Root Guard

Figure 23: Root Guard in a Service-Provider Network

The Layer 2 network of a service provider (SP) can include many connections to switches that are not owned by the SP. In such a topology, the spanning tree can reconfigure itself and select a customer switch as the root switch. You can avoid this situation by enabling root guard on SP switch interfaces that connect to switches in your customer's network. If spanning-tree calculations cause an interface in the customer network to be selected as the root port, root guard then places the interface in the root-inconsistent (blocked) state to prevent the customer's switch from becoming the root switch or being in the path to the root.



If a switch outside the SP network becomes the root switch, the interface is blocked (root-inconsistent state), and spanning tree selects a new root switch. The customer's switch does not become the root switch and is not in the path to the root.

If the switch is operating in MST mode, root guard forces the interface to be a designated port. If a boundary port is blocked in an internal spanning-tree (IST) instance because of root guard, the interface also is blocked in all MST instances. A boundary port is an interface that connects to a LAN, the designated switch of which is either an IEEE 802.1D switch or a switch with a different MST region configuration.

Root guard that is enabled on an interface applies to all the VLANs to which the interface belongs. VLANs can be grouped and mapped to an MST instance.



Caution Misuse of the root guard feature can cause a loss of connectivity.

Loop Guard

You can use loop guard to prevent alternate or root ports from becoming designated ports because of a failure that leads to a unidirectional link. This feature is most effective when it is enabled on the entire switched network. Loop guard prevents alternate and root ports from becoming designated ports, and spanning tree does not send BPDUs on root or alternate ports.

When the switch is operating in PVST+ or rapid-PVST+ mode, loop guard prevents alternate and root ports from becoming designated ports, and spanning tree does not send BPDUs on root or alternate ports.

When the switch is operating in MST mode, BPDUs are not sent on nonboundary ports only if the interface is blocked by loop guard in all MST instances. On a boundary port, loop guard blocks the interface in all MST instances.

How to Configure Optional Spanning-Tree Features

The following sections provide information about configuring Optional Spanning-Tree features:

(Optional) Enabling PortFast

An interface with the PortFast feature enabled is moved directly to the spanning-tree forwarding state without waiting for the standard forward-time delay.

If you enable the voice VLAN feature, the PortFast feature is automatically enabled. When you disable voice VLAN, the PortFast feature is not automatically disabled.

You can enable this feature if your switch is running PVST+, Rapid PVST+, or MSTP.



Caution Use PortFast only when connecting a single end station to an access or trunk port. Enabling this feature on an interface that is connected to a switch or hub could prevent spanning tree from detecting and disabling loops in your network, which could cause broadcast storms and address-learning problems.

To enable PortFast, perform this procedure:

Procedure

	Command or Action	Purpose
Step 1	enable Example: Device> enable	Enables privileged EXEC mode. Enter your password if prompted.
Step 2	configure terminal Example: Device# configure terminal	Enters global configuration mode.
Step 3	interface <i>interface-id</i> Example: Device(config)# interface gigabitethernet 1/0/2	Specifies an interface to configure, and enters interface configuration mode.
Step 4	spanning-tree portfast [trunk] Example: Device(config-if)# spanning-tree portfast trunk	Enables PortFast on an access port that is connected to a single workstation or server. By specifying the trunk keyword, you can enable PortFast on a trunk port.

	Command or Action	Purpose
		<p>Note To enable PortFast on trunk ports, you must use the spanning-tree portfast trunk interface configuration command. The spanning-tree portfast command will not work on trunk ports.</p> <p>Make sure that there are no loops in the network between the trunk port and the workstation or server before you enable PortFast on a trunk port.</p> <p>By default, PortFast is disabled on all interfaces.</p>
Step 5	<p>end</p> <p>Example:</p> <pre>Device(config-if)# end</pre>	Returns to privileged EXEC mode.

What to do next

You can use the **spanning-tree portfast default** global configuration command to globally enable the PortFast feature on all nontrunking ports.

Enabling PortFast Port Types

The following sections provide configurational information about enabling PortFast port types:

Configuring the PortFast Default State Globally

To configure the default PortFast state, perform this task:

Procedure

	Command or Action	Purpose
Step 1	<p>configure terminal</p> <p>Example:</p> <pre>Device# configure terminal</pre>	Enters global configuration mode.
Step 2	<p>spanning-tree portfast [edge network normal] default</p> <p>Example:</p> <pre>Device(config)# spanning-tree portfast edge default</pre>	<p>Configures the default state for all interfaces on the switch. You have these options:</p> <ul style="list-style-type: none"> • (Optional) edge: Configures all interfaces as edge ports. This assumes that all ports are connected to hosts/servers. • (Optional) network: Configures all interfaces as spanning tree network ports. This assumes that all ports are connected

	Command or Action	Purpose
		<p>to switches and bridges. Bridge Assurance is enabled on all network ports by default.</p> <ul style="list-style-type: none"> • (Optional) normal: Configures all interfaces as normal spanning tree ports. These ports can be connected to any type of device. • default: The default port type is normal.
Step 3	end Example: Device(config)# end	Exits configuration mode.

Related Topics

[Configuring a PortFast Edge Port on a Specified Interface](#), on page 74

[Configuring a PortFast Network Port on a Specified Interface](#), on page 76

Configuring a PortFast Edge Port on a Specified Interface

Interfaces configured as edge ports immediately transition to the forwarding state, without passing through the blocking or learning states, on linkup. To configure an edge port on a specified interface, perform this task:



Note Because the purpose of this type of port is to minimize the time that access ports must wait for spanning tree to converge, it is most effective when used on access ports. If you enable PortFast edge on a port connecting to another switch, you risk creating a spanning tree loop.

To configure a PortFast edge port on a specified interface, perform this procedure:

Procedure

	Command or Action	Purpose
Step 1	configure terminal Example: Device# configure terminal	Enters global configuration mode.
Step 2	interface { fastethernet gigabitethernet tengigabitethernet } <i>slot /port</i> } { port-channel <i>port_channel_number</i> } Example: Device(config)# interface gigabitethernet1/1	<p>Enables edge behavior on a Layer 2 access port connected to an end workstation or server.</p> <p>(Optional) trunk: Enables edge behavior on a trunk port. Use this keyword if the link is a trunk. Use this command only on ports that are connected to end host devices that terminate VLANs and from which the port should never receive STP BPDUs. Such end host devices</p>

	Command or Action	Purpose
		include workstations, servers, and ports on routers that are not configured to support bridging. Use the no version of the command to disable PortFast edge.
Step 3	end Example: Device(config-if)# end	Exits configuration mode.
Step 4	show spanning-tree interface { { fastethernet gigabitethernet tengigabitethernet } <i>slot</i> / <i>port</i> } { port-channel <i>port_channel_number</i> } portfast edge Example: Device# show spanning-tree interface	Displays spanning-tree PortFast information for the specified interface.

Example

This example shows how to enable edge behavior on GigabitEthernet interface 5/7 and verify the configuration:

```
Device# configure terminal
Device(config)# interface fastethernet 5/7
Device(config-if)# spanning-tree portfast edge
Device(config-if)# end
Device#
```

```
Device# show running-config interface fastethernet 5/7
Building configuration...
Current configuration:
!
interface GigabitEthernet5/7
no ip address
switchport
switchport access vlan 200
switchport mode access
spanning-tree portfast edge
end
```

This example shows how you can display that port GigabitEthernet 5/8 is currently in the edge state:

```
Device# show spanning-tree vlan 200
VLAN0200
Spanning tree enabled protocol rstp
Root ID Priority 2
Address 001b.2a68.5fc0
Cost 3
Port 125 (GigabitEthernet5/9)
Hello Time 2 sec Max Age 20 sec Forward Delay 15 sec

Bridge ID Priority 2 (priority 0 sys-id-ext 2)
Address 7010.5c9c.5200
Hello Time 2 sec Max Age 20 sec Forward Delay 15 sec
Aging Time 0 sec
```

```
Interface Role Sts Cost Prio.Nbr Type
-----
Gi5/7 Desg FWD 4 128.1 P2p Edge
```

Related Topics

[Configuring the PortFast Default State Globally](#), on page 73

[Configuring a PortFast Network Port on a Specified Interface](#), on page 76

Configuring a PortFast Network Port on a Specified Interface

Ports that are connected to Layer 2 switches and bridges can be configured as network ports.



Note Bridge Assurance is enabled only on PortFast network ports. For more information, see [Bridge Assurance](#), on page 60.

To configure a port as a network port, perform this task:

Procedure

	Command or Action	Purpose
Step 1	configure terminal Example: Device# configure terminal	Enters global configuration mode.
Step 2	interface { fastethernet gigabitethernet tengigabitethernet } <i>slot /port</i> } { port-channel <i>port_channel_number</i> } Example: Device(config)# spanning-tree portfast network	Configures the port as a network port. If you have enabled Bridge Assurance globally, it automatically runs on a spanning tree network port. Use the no keyword to disable PortFast.
Step 3	end Example: Device(config-if)# end	Exits configuration mode.
Step 4	show running interface { fastethernet gigabitethernet tengigabitethernet } <i>slot /port</i> } { port-channel <i>port_channel_number</i> } Example: Device# show running interface gigabitethernet 5/8	Verifies the configuration.

Example

This example shows how to configure GigabitEthernet interface 5/8 as a network port and verify configuration:

```
Device# configure terminal
Device(config)# interface gigabitethernet 5/8
Device(config-if)# spanning-tree portfast network
Device(config-if)# end
Device#

Device# show running-config interface gigabitethernet 5/8
Building configuration...
Current configuration:
!
interface GigabitEthernet5/8
no ip address
switchport
switchport access vlan 200
switchport mode access
spanning-tree portfast network
end
```

Related Topics

[Configuring the PortFast Default State Globally](#), on page 73

[Configuring a PortFast Edge Port on a Specified Interface](#), on page 74

(Optional) Enabling Bridge Protocol Data Unit Guard

You can enable the BPDU guard feature if your switch is running PVST+, Rapid PVST+, or MSTP.



Caution Configure PortFast edge only on ports that connect to end stations; otherwise, an accidental topology loop could cause a data packet loop and disrupt switch and network operation.

To enable BPDU guard, perform this procedure:

Procedure

	Command or Action	Purpose
Step 1	enable Example: Device> enable	Enables privileged EXEC mode. Enter your password if prompted.
Step 2	configure terminal Example: Device# configure terminal	Enters global configuration mode.
Step 3	spanning-tree portfast edge bpduguard default Example:	Enables BPDU guard.

	Command or Action	Purpose
	Device (config)# spanning-tree portfast edge bpduguard default	
Step 4	interface <i>interface-id</i> Example: Device (config)# interface gigabitethernet 1/0/2	Specifies the interface that is connected to an end station, and enters interface configuration mode.
Step 5	spanning-tree portfast edge Example: Device (config-if)# spanning-tree portfast edge	Enables the PortFast edge feature.
Step 6	end Example: Device (config-if)# end	Returns to privileged EXEC mode.

What to do next

To prevent the port from shutting down, you can use the **errdisable detect cause bpduguard shutdown vlan** global configuration command to shut down just the offending VLAN on the port where the violation occurred.

You also can use the **spanning-tree bpduguard enable** interface configuration command to enable BPDU guard on any port without also enabling the PortFast edge feature. When the port receives a BPDU, it is put in the error-disabled state.

(Optional) Enabling Bridge Protocol Data Unit Filtering

You can also use the **spanning-tree bpdupfilter enable** interface configuration command to enable BPDU filtering on any interface without also enabling the PortFast edge feature. This command prevents the interface from sending or receiving BPDUs.



Caution Enabling BPDU filtering on an interface is the same as disabling spanning tree on it and can result in spanning-tree loops.

You can enable the BPDU filtering feature if your switch is running PVST+, Rapid PVST+, or MSTP.



Caution Configure PortFast edge only on interfaces that connect to end stations; otherwise, an accidental topology loop could cause a data packet loop and disrupt switch and network operation.

To enable BPDU filter, perform this procedure:

Procedure

	Command or Action	Purpose
Step 1	enable Example: Device> enable	Enables privileged EXEC mode. Enter your password if prompted.
Step 2	configure terminal Example: Device# configure terminal	Enters global configuration mode.
Step 3	spanning-tree portfast edge bpdufilter default Example: Device(config)# spanning-tree portfast edge bpdufilter default	Globally enables BPDU filtering. By default, BPDU filtering is disabled.
Step 4	interface <i>interface-id</i> Example: Device(config)# interface gigabitethernet 1/0/2	Specifies the interface that is connected to an end station, and enters interface configuration mode.
Step 5	spanning-tree portfast edge Example: Device(config-if)# spanning-tree portfast edge	Enables the PortFast edge feature on the specified interface.
Step 6	end Example: Device(config-if)# end	Returns to privileged EXEC mode.

Configuring Bridge Assurance

Procedure

	Command or Action	Purpose
Step 1	configure terminal Example: Device# configure terminal	Enters the global configuration mode.
Step 2	spanning-tree bridge assurance Example: Device# spanning-tree bridge assurance	Enables Bridge Assurance on all network ports on the switch. Bridge Assurance is enabled by default. Use the no version of the command to disable the feature. Disabling Bridge Assurance causes

	Command or Action	Purpose
		all configured network ports to behave as normal spanning tree ports.
Step 3	end Example: Device# end	Exits configuration mode.
Step 4	show spanning-tree summary Example: Device# show spanning-tree summary	Displays spanning tree information and shows if Bridge Assurance is enabled.

Example

This example shows how to display spanning tree information and verify if Bridge Assurance is enabled. Look for these details in the output:

- Portfast Default—Network
- Bridge Assurance—Enabled

```
Device# show spanning-tree summary
Switch is in rapid-pvst mode
Root bridge for: VLAN0199-VLAN0200, VLAN0128
EtherChannel misconfig guard is enabled
Extended system ID is enabled
Portfast Default is network
Portfast Edge BPDU Guard Default is disabled
Portfast Edge BPDU Filter Default is disabled
Loopguard Default is enabled
PVST Simulation Default is enabled but inactive in rapid-pvst mode
Bridge Assurance is enabled
UplinkFast is disabled
BackboneFast is disabled
Configured Pathcost method used is short
```

```
Name Blocking Listening Learning Forwarding STP Active
-----
VLAN0199 0 0 0 5 5
VLAN0200 0 0 0 4 4
VLAN0128 0 0 0 4 4
-----
3 vlans 0 0 0 13 13
```

This example shows how to verify if GigabitEthernet 5/8 (configured as a network port), is in a normal state.

(From the **show spanning-tree summary** output above, we know that Bridge Assurance is enabled on GigabitEthernet 5/8).

```
Device# show spanning-tree vlan
Sep 17 09:51:36.370 PDT: %SYS-5-CONFIG_I: Configured from console by console2
VLAN0200
Spanning tree enabled protocol rstp
Root ID Priority 2
Address 7010.5c9c.5200
This bridge is the root
```

```

Hello Time 2 sec Max Age 20 sec Forward Delay 15 sec

Bridge ID Priority 2 (priority 0 sys-id-ext 2)
Address 7010.5c9c.5200
Hello Time 2 sec Max Age 20 sec Forward Delay 15 sec
Aging Time 0 sec

Interface Role Sts Cost Prio.Nbr Type
-----
Gi5/7 Desg FWD 4 128.1 P2p Edge
Gi5/8 Desg FWD 3 128.480 P2p Network
Gi5/9 Desg FWD 4 128.169 P2p Edge
Gi5/10 Desg FWD 4 128.215 P2p Network

```

This example shows how port GigabitEthernet 5/8 (configured as a network port), is currently in the Bridge Assurance inconsistent state:



Note The output shows the port type as network and *BA_Inc, indicating that the port is in an inconsistent state.

```

Device# show spanning-tree vlan

VLAN200
Spanning tree enabled protocol rstp
Root ID Priority 32778
Address 0002.172c.f400
This bridge is the root
Hello Time 2 sec Max Age 20 sec Forward Delay 15 sec
Bridge ID Priority 32778 (priority 32768 sys-id-ext 10)
Address 0002.172c.f400
Hello Time 2 sec Max Age 20 sec Forward Delay 15 sec
Aging Time 300
Interface Role Sts Cost Prio.Nbr Type
-----
Gia5/8 Desg BKN*4 128.270 Network, P2p *BA_Inc

```

(Optional) Enabling UplinkFast for Use with Redundant Links



Note When you enable UplinkFast, it affects all VLANs on the switch or switch stack. You cannot configure UplinkFast on an individual VLAN.

You can configure the UplinkFast or the Cross-Stack UplinkFast (CSUF) feature for Rapid PVST+ or for the MSTP, but the feature remains disabled (inactive) until you change the spanning-tree mode to PVST+.

Follow these steps to enable UplinkFast and CSUF.

Before you begin

UplinkFast cannot be enabled on VLANs that have been configured with a switch priority. To enable UplinkFast on a VLAN with switch priority configured, first restore the switch priority on the VLAN to the default value using the **no spanning-tree vlan *vlan-id* priority** global configuration command.

Procedure

	Command or Action	Purpose
Step 1	enable Example: Device> enable	Enables privileged EXEC mode. Enter your password if prompted.
Step 2	configure terminal Example: Device# configure terminal	Enters global configuration mode.
Step 3	spanning-tree uplinkfast [max-update-rate pkts-per-second] Example: Device(config)# spanning-tree uplinkfast max-update-rate 200	Enables UplinkFast. (Optional) For <i>pkts-per-second</i> , the range is 0 to 32000 packets per second; the default is 150. If you set the rate to 0, station-learning frames are not generated, and the spanning-tree topology converges more slowly after a loss of connectivity. When you enter this command, CSUF also is enabled on all nonstack port interfaces.
Step 4	end Example: Device(config)# end	Returns to privileged EXEC mode.

When UplinkFast is enabled, the switch priority of all VLANs is set to 49152. If you change the path cost to a value less than 3000 and you enable UplinkFast or UplinkFast is already enabled, the path cost of all interfaces and VLAN trunks is increased by 3000 (if you change the path cost to 3000 or above, the path cost is not altered). The changes to the switch priority and the path cost reduce the chance that a switch will become the root switch.

When UplinkFast is disabled, the switch priorities of all VLANs and path costs of all interfaces are set to default values if you did not modify them from their defaults.

When you enable the UplinkFast feature using these instructions, CSUF is automatically globally enabled on nonstack port interfaces.

(Optional) Disabling UplinkFast

Follow these steps to disable UplinkFast and Cross-Stack UplinkFast (CSUF).

Before you begin

UplinkFast must be enabled.

Procedure

	Command or Action	Purpose
Step 1	enable Example: Device> enable	Enables privileged EXEC mode. Enter your password if prompted.
Step 2	configure terminal Example: Device# configure terminal	Enters global configuration mode.
Step 3	no spanning-tree uplinkfast Example: Device(config)# no spanning-tree uplinkfast	Disables UplinkFast and CSUF on the switch and all of its VLANs.
Step 4	end Example: Device(config)# end	Returns to privileged EXEC mode.

When UplinkFast is disabled, the switch priorities of all VLANs and path costs of all interfaces are set to default values if you did not modify them from their defaults.

When you disable the UplinkFast feature using these instructions, CSUF is automatically globally disabled on nonstack port interfaces.

(Optional) Enabling BackboneFast

You can enable BackboneFast to detect indirect link failures and to start the spanning-tree reconfiguration sooner.

You can configure the BackboneFast feature for Rapid PVST+ or for the MSTP, but the feature remains disabled (inactive) until you change the spanning-tree mode to PVST+.

Follow these steps to enable BackboneFast on the switch.

Before you begin

If you use BackboneFast, you must enable it on all switches in the network. BackboneFast is not supported on Token Ring VLANs. This feature is supported for use with third-party switches.

Procedure

	Command or Action	Purpose
Step 1	enable Example: Device> enable	Enables privileged EXEC mode. Enter your password if prompted.

	Command or Action	Purpose
Step 2	configure terminal Example: Device# <code>configure terminal</code>	Enters global configuration mode.
Step 3	spanning-tree backbonefast Example: Device(config)# <code>spanning-tree backbonefast</code>	Enables BackboneFast.
Step 4	end Example: Device(config)# <code>end</code>	Returns to privileged EXEC mode.

(Optional) Enabling EtherChannel Guard

You can enable EtherChannel guard to detect an EtherChannel misconfiguration if your device is running PVST+, Rapid PVST+, or MSTP.

Follow these steps to enable EtherChannel Guard on the device.

Procedure

	Command or Action	Purpose
Step 1	enable Example: Device> <code>enable</code>	Enables privileged EXEC mode. Enter your password if prompted.
Step 2	configure terminal Example: Device# <code>configure terminal</code>	Enters global configuration mode.
Step 3	spanning-tree etherchannel guard misconfig Example: Device(config)# <code>spanning-tree etherchannel guard misconfig</code>	Enables EtherChannel guard.
Step 4	end Example: Device(config)# <code>end</code>	Returns to privileged EXEC mode.

What to do next

You can use the **show interfaces status err-disabled** privileged EXEC command to show which device ports are disabled because of an EtherChannel misconfiguration. On the remote device, you can enter the **show etherchannel summary** command in privileged EXEC mode to verify the EtherChannel configuration.

After the configuration is corrected, enter the **shutdown** and **no shutdown** interface configuration commands on the port-channel interfaces that were misconfigured.

(Optional) Enabling Root Guard

Root guard that is enabled on an interface applies to all the VLANs to which the interface belongs. Do not enable the root guard on interfaces to be used by the UplinkFast feature. With UplinkFast, the backup interfaces (in the blocked state) replace the root port in the case of a failure. However, if root guard is also enabled, all the backup interfaces used by the UplinkFast feature are placed in the root-inconsistent state (blocked) and are prevented from reaching the forwarding state.



Note You cannot enable both root guard and loop guard at the same time.

You can enable this feature if your switch is running PVST+, Rapid PVST+, or MSTP.

Follow these steps to enable root guard on the switch.

Procedure

	Command or Action	Purpose
Step 1	enable Example: Device> enable	Enables privileged EXEC mode. Enter your password if prompted.
Step 2	configure terminal Example: Device# configure terminal	Enters global configuration mode.
Step 3	interface <i>interface-id</i> Example: Device(config)# interface gigabitethernet 1/0/2	Specifies an interface to configure, and enters interface configuration mode.
Step 4	spanning-tree guard root Example: Device(config-if)# spanning-tree guard root	Enables root guard on the interface. By default, root guard is disabled on all interfaces.
Step 5	end Example: Device(config-if)# end	Returns to privileged EXEC mode.

(Optional) Enabling Loop Guard

You can use loop guard to prevent alternate or root ports from becoming designated ports because of a failure that leads to a unidirectional link. This feature is most effective when it is configured on the entire switched network. Loop guard operates only on interfaces that are considered point-to-point by the spanning tree.



Note You cannot enable both loop guard and root guard at the same time.

You can enable this feature if your device is running PVST+, Rapid PVST+, or MSTP.

Follow these steps to enable loop guard on the device.

Procedure

	Command or Action	Purpose
Step 1	Enter one of the following commands: <ul style="list-style-type: none"> • <code>show spanning-tree active</code> • <code>show spanning-tree mst</code> Example: Device# <code>show spanning-tree active</code> or Device# <code>show spanning-tree mst</code>	Verifies which interfaces are alternate or root ports.
Step 2	configure terminal Example: Device# <code>configure terminal</code>	Enters global configuration mode.
Step 3	spanning-tree loopguard default Example: Device(config)# <code>spanning-tree loopguard default</code>	Enables loop guard. By default, loop guard is disabled.
Step 4	end Example: Device(config)# <code>end</code>	Returns to privileged EXEC mode.

Monitoring the Spanning-Tree Status

Table 9: Commands for Monitoring the Spanning-Tree Status

Command	Purpose
<code>show spanning-tree active</code>	Displays spanning-tree information on active interfaces only.

Command	Purpose
show spanning-tree detail	Displays a detailed summary of interface information.
show spanning-tree interface <i>interface-id</i>	Displays spanning-tree information for the specified interface.
show spanning-tree mst interface <i>interface-id</i>	Displays MST information for the specified interface.
show spanning-tree summary [totals]	Displays a summary of interface states or displays the total link spanning-tree state section.
show spanning-tree mst interface <i>interface-id</i> portfast edge	Displays spanning-tree PortFast information for the specified interface.

Additional References for Optional Spanning Tree Features

Related Documents

Related Topic	Document Title
For complete syntax and usage information for the commands used in this chapter.	<i>Command Reference (Catalyst 9600 Series Switches)</i>

Feature History for Optional Spanning Tree Features

This table provides release and related information for features explained in this module.

These features are available on all releases subsequent to the one they were introduced in, unless noted otherwise.

Release	Feature	Feature Information
Cisco IOS XE Gibraltar 16.11.1	Optional Spanning Tree Protocol	The optional features of the STP enhance loop prevention, protect against some possible user misconfigurations, and provide better control over the protocol parameters.

Use Cisco Feature Navigator to find information about platform and software image support. To access Cisco Feature Navigator, go to <http://www.cisco.com/go/cfn>.



CHAPTER 5

Configuring EtherChannels

- [Restrictions for EtherChannels, on page 89](#)
- [Information About EtherChannels, on page 90](#)
- [How to Configure EtherChannels, on page 103](#)
- [Monitoring EtherChannel, Port Aggregation Protocol, and Link Aggregation Control Protocol Status, on page 121](#)
- [Configuration Examples for EtherChannels, on page 121](#)
- [Additional References for EtherChannels, on page 125](#)
- [Feature History for EtherChannels, on page 125](#)

Restrictions for EtherChannels

The following are restrictions for EtherChannels:

- All ports in an EtherChannel must be assigned to the same VLAN or they must be configured as trunk port.
- The LACP 1:1 redundancy feature is supported on port channel interfaces only.

Unsupported EtherChannel features:

- You cannot configure a voice VLAN on a port channel or a member interface.
- You cannot convert an interface to an ether channel if dot1ad is configured on the interface.
- You cannot configure `non negotiate` and dynamic commands on a port channel.
- You cannot configure pruning VLAN if MVRP feature is already configured on the device.
- You cannot configure network policy commands on a routed or trunk port and on an ether channel.
- You can configure the **rep segment** command only on switch port mode trunk.
- You cannot configure **switchport priority extend trust** command and **switchport priorit extend cos 3** command on an etherchannel.
- You cannot configure **platform qos low-latency** command on an interface port-channel 10.
- You cannot use Layer 2 configurations on a Layer 3 port.
- When there are any misconfigurations detected in a port mode or VLAN mask, the ports are suspended.

- On Cisco Catalyst 9600 Series Supervisor 2 Module EtherChannel member port selection is software based in Layer 2 and Layer 3 multicast route. This means that all multicast traffic under a group will be routed via the same physical port of the EtherChannel. As a result, the distribution of multicast traffic load balance over etherchannels might not evenly spread across member ports.

Information About EtherChannels

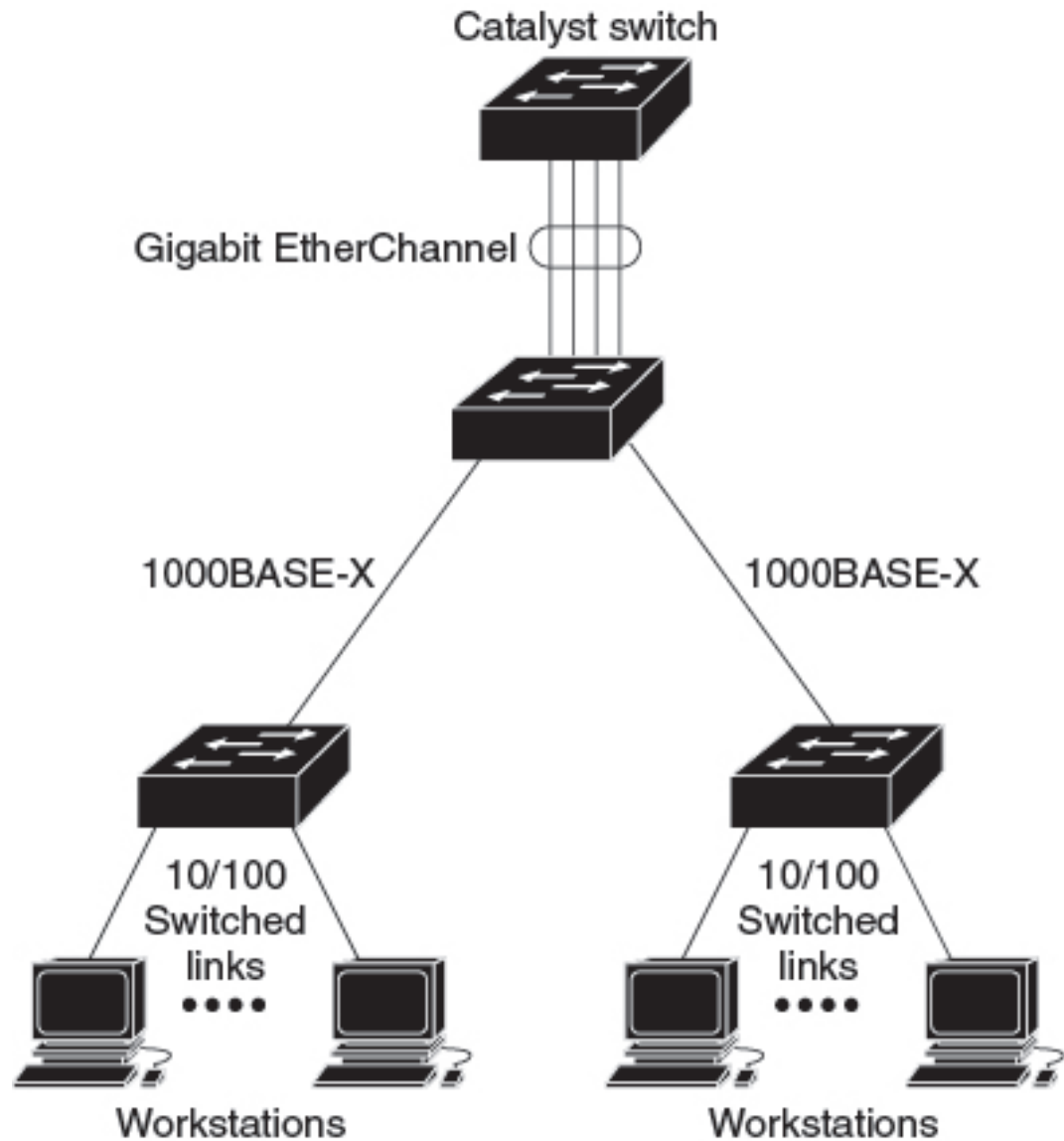
The following sections provide information about EtherChannels and the various modes to configure EtherChannels.

EtherChannel Overview

EtherChannel provides fault-tolerant high-speed links between switches, routers, and servers. You can use the EtherChannel to increase the bandwidth between the wiring closets and the data center, and you can deploy it anywhere in the network where bottlenecks are likely to occur. EtherChannel provides automatic recovery for the loss of a link by redistributing the load across the remaining links. If a link fails, EtherChannel redirects traffic from the failed link to the remaining links in the channel without intervention.

An EtherChannel consists of individual Ethernet links that are bundled into a single logical link, and each EtherChannel can consist of up to eight compatibly configured Ethernet ports.

Figure 24: Typical EtherChannel Configuration

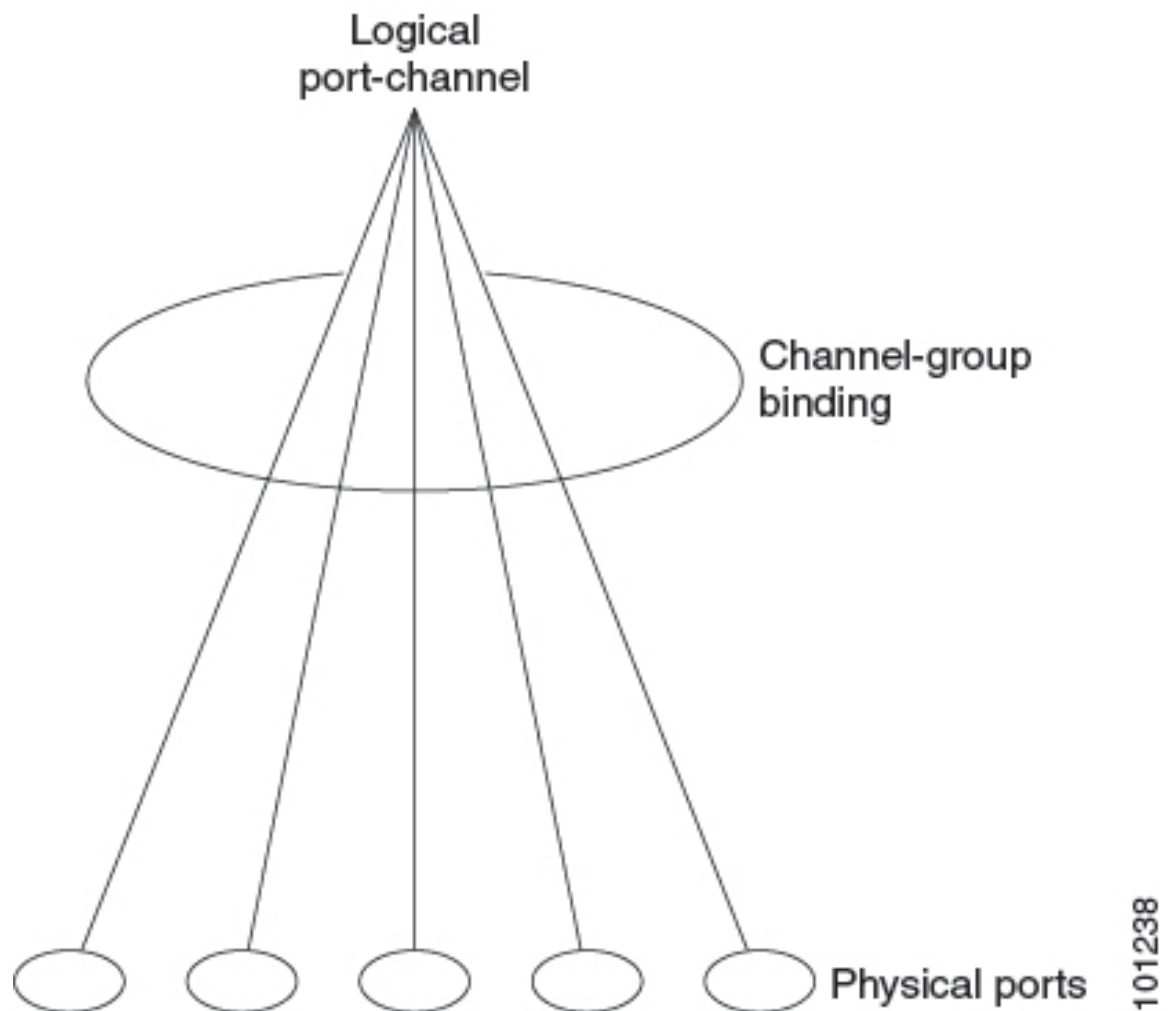


347662

Channel Groups and Port-Channel Interfaces

An EtherChannel comprises a channel group and a port-channel interface. The channel group binds physical ports to the port-channel interface. Configuration changes applied to the port-channel interface apply to all the physical ports bound together in the channel group.

Figure 25: Relationship Between Physical Ports, a Channel Group, and a Port-Channel Interface



The **channel-group** command binds the physical port and the port-channel interface together. Each EtherChannel has a port-channel logical interface that is numbered from 1 to 192. This port-channel interface number corresponds to the one specified with the **channel-group** interface configuration command.

- With Layer 2 ports, use the **channel-group** interface configuration command to dynamically create the port-channel interface.

You also can use the **interface port-channel** *port-channel-number* global configuration command to manually create the port-channel interface, but then you must use the **channel-group** *channel-group-number* command to bind the logical interface to a physical port. The *channel-group-number* can be the same as the *port-channel-number*; or you can use a new number. If you use a new number, the **channel-group** command dynamically creates a new port channel.

- With Layer 3 ports, you should manually create the logical interface by using the **interface port-channel** global configuration command followed by the **no switchport** interface configuration command. You then manually assign an interface to the EtherChannel by using the **channel-group** interface configuration command.

- With Layer 3 ports, use the **no switchport** interface command to configure the interface as a Layer 3 interface, and then use the **channel-group** interface configuration command to dynamically create the port-channel interface.



Note While moving a port from Layer 2 to Layer 3, if the Layer 2 port is a member of a port-channel, first remove the port from the channel group using the **no channel group** command. Then use the **no switchport** command to move the port to Layer 3.

Port Aggregation Protocol

The Port Aggregation Protocol (PAgP) is a Cisco-proprietary protocol that can be run only on Cisco devices and on those devices that are licensed by vendors to support PAgP. PAgP facilitates the automatic creation of EtherChannels by exchanging PAgP packets between Ethernet ports. PAgP can be enabled on cross-stack EtherChannels.

By using PAgP, the switch or switch stack learns the identity of partners capable of supporting PAgP and the capabilities of each port. It then dynamically groups similarly configured ports (on a single device in the stack) into a single logical link (channel or aggregate port). Similarly configured ports are grouped based on hardware, administrative, and port parameter constraints. For example, PAgP groups the ports with the same speed, duplex mode, native VLAN, VLAN range, and trunking status and type. After grouping the links into an EtherChannel, PAgP adds the group to the spanning tree as a single device port.

Port Aggregation Protocol Modes

PAgP modes specify whether a port can send PAgP packets, which start PAgP negotiations, or only respond to PAgP packets received.

Table 10: EtherChannel PAgP Modes

Mode	Description
auto	Places a port into a passive negotiating state, in which the port responds to PAgP packets it receives but does not start PAgP packet negotiation. This setting minimizes the transmission of PAgP packets.
desirable	Places a port into an active negotiating state, in which the port starts negotiations with other ports by sending PAgP packets.

Switch ports exchange PAgP packets only with partner ports that are configured in the **auto** or **desirable** modes. Ports that are configured in the **on** mode do not exchange PAgP packets.

Both the **auto** and **desirable** modes enable ports to negotiate with partner ports to form an EtherChannel based on criteria such as port speed, and for Layer 2 EtherChannels, based on trunk state and VLAN numbers.

Ports can form an EtherChannel when they are in different PAgP modes as long as the modes are compatible. For example:

- A port in the **desirable** mode can form an EtherChannel with another port that is in the **desirable** or **auto** mode.
- A port in the **auto** mode can form an EtherChannel with another port in the **desirable** mode.

A port in the **auto** mode cannot form an EtherChannel with another port that is also in the **auto** mode because neither port starts PAgP negotiation.

Silent Mode

If your switch is connected to a partner that is PAgP-capable, you can configure the switch port for nonsilent operation by using the **non-silent** keyword. If you do not specify **non-silent** with the **auto** or **desirable** mode, silent mode is assumed.

Use the silent mode when the switch is connected to a device that is not PAgP-capable and seldom, if ever, sends packets. An example of a silent partner is a file server or a packet analyzer that is not generating traffic. In this case, running PAgP on a physical port that is connected to a silent partner prevents that switch port from ever becoming operational. However, the silent setting allows PAgP to operate, to attach the port to a channel group, and to use the port for transmission.

Port Aggregation Protocol Learn Method and Priority

Network devices are classified as PAgP physical learners or aggregate-port learners. A device is a physical learner if it learns addresses by physical ports and directs transmissions based on that knowledge. A device is an aggregate-port learner if it learns addresses by aggregate (logical) ports. The learn method must be configured the same at both ends of the link.

When a device and its partner are both aggregate-port learners, they learn the address on the logical port-channel. The device sends packets to the source by using any of the ports in the EtherChannel. With aggregate-port learning, it is not important on which physical port the packet arrives.

PAgP cannot automatically detect when the partner device is a physical learner and when the local device is an aggregate-port learner. Therefore, you must manually set the learning method on the local device to learn addresses by physical ports. You also must set the load-distribution method to source-based distribution, so that any given source MAC address is always sent on the same physical port.

You also can configure a single port within the group for all transmissions and use other ports for hot-standby. The unused ports in the group can be swapped into operation in just a few seconds if the selected single port loses hardware-signal detection. You can configure which port is always selected for packet transmission by changing its priority with the **pagp port-priority** interface configuration command. The higher the priority, the more likely that the port will be selected.



Note The device supports address learning only on aggregate ports even though the **physical-port** keyword is provided in the CLI. The **pagp learn-method** command and the **pagp port-priority** command have no effect on the device hardware, but they are required for PAgP interoperability with devices that only support address learning by physical ports, such as the Catalyst 1900 switch.

When the link partner of the device is a physical learner, we recommend that you configure the device as a physical-port learner by using the **pagp learn-method physical-port** interface configuration command. Set the load-distribution method based on the source MAC address by using the **port-channel load-balance src-mac** global configuration command. The device then sends packets to the physical learner using the same port in the EtherChannel from which it learned the source address. Only use the **pagp learn-method** command in this situation.

Port Aggregation Protocol Interaction with Other Features

The Dynamic Trunking Protocol (DTP) and the Cisco Discovery Protocol (CDP) send and receive packets over the physical ports in the EtherChannel. Trunk ports send and receive PAgP protocol data units (PDUs) on the lowest numbered VLAN.

In Layer 2 EtherChannels, the first port in the channel that comes up provides its MAC address to the EtherChannel. If this port is removed from the bundle, one of the remaining ports in the bundle provides its MAC address to the EtherChannel. For Layer 3 EtherChannels, the MAC address is allocated by the active device as soon as the interface is created (through the **interface port-channel** global configuration command).

PAgP sends and receives PAgP PDUs only from ports that are up and have PAgP enabled for the auto or desirable mode.

Link Aggregation Control Protocol

The LACP is defined in IEEE 802.3ad and enables Cisco devices to manage Ethernet channels between devices that conform to the IEEE 802.3ad protocol. LACP facilitates the automatic creation of EtherChannels by exchanging LACP packets between Ethernet ports.

By using LACP, the switch or switch stack learns the identity of partners capable of supporting LACP and the capabilities of each port. It then dynamically groups similarly configured ports into a single logical link (channel or aggregate port). Similarly configured ports are grouped based on hardware, administrative, and port parameter constraints. For example, LACP groups the ports with the same speed, duplex mode, native VLAN, VLAN range, and trunking status and type. After grouping the links into an EtherChannel, LACP adds the group to the spanning tree as a single device port.

The independent mode behavior of ports in a port channel is changed. With CSCtn96950, by default, standalone mode is enabled. When no response is received from an LACP peer, ports in the port channel are moved to suspended state.

Link Aggregation Control Protocol Modes

LACP modes specify whether a port can send LACP packets or only receive LACP packets.

Table 11: EtherChannel LACP Modes

Mode	Description
active	Places a port into an active negotiating state in which the port starts negotiations with other ports by sending LACP packets.
passive	Places a port into a passive negotiating state in which the port responds to LACP packets that it receives, but does not start LACP packet negotiation. This setting minimizes the transmission of LACP packets.

Both the **active** and **passive LACP** modes enable ports to negotiate with partner ports to an EtherChannel based on criteria such as port speed, and for Layer 2 EtherChannels, based on trunk state and VLAN numbers.

Ports can form an EtherChannel when they are in different LACP modes as long as the modes are compatible. For example:

- A port in the **active** mode can form an EtherChannel with another port that is in the **active** or **passive** mode.

- A port in the **passive** mode cannot form an EtherChannel with another port that is also in the **passive** mode because neither port starts LACP negotiation.

Link Aggregation Control Protocol Standalone Mode on Ethernet Channel

When one end of an EtherChannel has more members than the other, the unmatched ports enter the standalone state. The standalone mode is also called the independent mode. In the standalone mode the port is not bundled in an EtherChannel. The port functions as a standalone data port and it can send and receive BPDUs and data traffic.

In a topology that is not protected from Layer 2 loops by the spanning tree protocol (STP), a port in the standalone state can cause significant network errors. You can enter the **port-channel standalone-disable** command in the interface configuration mode to put ports into the suspended state instead of the standalone state.

The standalone mode is particularly relevant when a port (A) in a Layer 2 LACP EtherChannel is connected to an unresponsive port (B) on the peer. When LACP standalone is disabled on the EtherChannel, all traffic arriving on A is blocked (the default behavior on a switch). In some scenarios, you might want to allow management traffic on such ports. You can do this by enabling LACP standalone (or independent) mode. To enable the standalone mode on a Layer 2 LACP Etherchannel, use the **no port-channel standalone disable** command in the interface configuration mode. To disable the Standalone mode and revert to the default use the **port-channel standalone disable** command in the interface configuration mode.



Note LACP standalone mode is disabled by default.

Starting with the Cisco IOS XE Dublin 17.10.1 release, you can configure the LACP standalone mode on a Layer 3 EtherChannel. To configure the standalone mode use the **no port-channel standalone disable** command in the interface configuration mode. To disable the Standalone mode and revert to the default use the **port-channel standalone disable** command in the interface configuration mode.

Link Aggregation Control Protocol and Link Redundancy

LACP port-channel operation, bandwidth availability, and link redundancy can be further refined with the LACP port-channel min-links and the LACP max-bundle features.

The LACP port-channel min-links feature:

- Configures the minimum number of ports that must be linked up and bundled in the LACP port channel.
- Prevents a low-bandwidth LACP port channel from becoming active.
- Causes an LACP port channel to become inactive if there are too few active members ports to supply the required minimum bandwidth.

The LACP max-bundle feature:

- Defines an upper limit on the number of bundled ports in an LACP port channel.
- Allows hot-standby ports with fewer bundled ports. For example, in an LACP port channel with five ports, you can specify a max-bundle of three, and the two remaining ports are designated as hot-standby ports.

Link Aggregation Control Protocol Interaction with Other Features

The DTP and the CDP send and receive packets over the physical ports in the EtherChannel. Trunk ports send and receive LACP PDUs on the lowest numbered VLAN.

In Layer 2 EtherChannels, the first port in the channel that comes up provides its MAC address to the EtherChannel. If this port is removed from the bundle, one of the remaining ports in the bundle provides its MAC address to the EtherChannel. For Layer 3 EtherChannels, the MAC address is allocated by the active device as soon as the interface is created through the **interface port-channel** global configuration command.

LACP sends and receives LACP PDUs only from ports that are up and have LACP enabled for the active or passive mode.

Link Aggregation Control Protocol Interaction with Other Features 1:1 Redundancy

The LACP 1:1 Redundancy feature supports an EtherChannel configuration with one active link, and fast switchover to a hot-standby link. The link that is connected to the port with the lower port priority number (and therefore, of a higher priority) will be the active link, and the other link will be in a hot-standby state. If the active link goes down, LACP performs a fast switchover to the hot-standby link to keep the EtherChannel up. When the failed link becomes operational again, LACP performs another fast switchover to revert to the original active link.

To allow the higher priority port to stabilize when it becomes active again after a higher-priority to lower-priority switchover, the LACP 1:1 Hot Standby Dampening feature configures a timer that delays switchover back to the higher priority port after higher priority port becomes active.

EtherChannel On Mode

EtherChannel **on** mode can be used to manually configure an EtherChannel. The **on** mode forces a port to join an EtherChannel without negotiations. The **on** mode can be useful if the remote device does not support PAgP or LACP. In the **on** mode, a usable EtherChannel exists only when the devices at both ends of the link are configured in the **on** mode.

Ports that are configured in the **on** mode in the same channel group must have compatible port characteristics, such as speed and duplex. Ports that are not compatible are suspended, even though they are configured in the **on** mode.

**Caution**

You should use care when using the **on** mode. This is a manual configuration, and ports on both ends of the EtherChannel must have the same configuration. If the group is misconfigured, packet loss or spanning-tree loops can occur.

Load-Balancing and Forwarding Methods

**Note**

Configurable load-Balancing is enabled by default in the device.

Fields from Layer 2, Layer 3 and Layer 4 headers are parameters that are used for load-balancing.

EtherChannel balances the traffic load across the links in a channel by reducing part of the binary pattern that is formed from the addresses in the frame to a numerical value that selects one of the links in the channel. You can specify one of several different load-balancing modes, including load distribution based on MAC addresses, IP addresses, VLAN IDs, source addresses, destination addresses, or both source and destination addresses. The selected mode applies to all EtherChannels configured on the device.



Note Layer 3 Equal-cost multi path (ECMP) load balancing is based on source IP address, destination IP address, source port, destination port, and layer 4 protocol. Fragmented packets will be treated on two different links based on the algorithm that is calculated using these parameters. Any changes in one of these parameters result in load balancing.

MAC Address Forwarding

With source-MAC address forwarding, when packets are forwarded to an EtherChannel, they are distributed across the ports in the channel based on the source-MAC address of the incoming packet. Therefore, to provide load-balancing, packets from different hosts use different ports in the channel, but packets from the same host use the same port in the channel.

With destination-MAC address forwarding, when packets are forwarded to an EtherChannel, they are distributed across the ports in the channel based on the destination host's MAC address of the incoming packet. Therefore, packets to the same destination are forwarded over the same port, and packets to a different destination are sent on a different port in the channel.

With source-and-destination MAC address forwarding, when packets are forwarded to an EtherChannel, they are distributed across the ports in the channel based on both the source and destination MAC addresses. This forwarding method, a combination source-MAC and destination-MAC address forwarding methods of load distribution, can be used if it is not clear whether source-MAC or destination-MAC address forwarding is better suited on a particular device. With source-and-destination MAC-address forwarding, packets sent from host A to host B, host A to host C, and host C to host B could all use different ports in the channel.

IP Address Forwarding

With source-IP address-based forwarding, packets are distributed across the ports in the EtherChannel based on the source-IP address of the incoming packet. To provide load balancing, packets from different IP addresses use different ports in the channel, and packets from the same IP address use the same port in the channel.

With destination-IP address-based forwarding, packets are distributed across the ports in the EtherChannel based on the destination-IP address of the incoming packet. To provide load balancing, packets from the same IP source address that is sent to different IP destination addresses could be sent on different ports in the channel. Packets sent from different source IP addresses to the same destination IP address are always sent on the same port in the channel.

With source-and-destination IP address-based forwarding, packets are distributed across the ports in the EtherChannel based on both the source and destination IP addresses of the incoming packet. This forwarding method, a combination of source-IP and destination-IP address-based forwarding, can be used if it is not clear whether source-IP or destination-IP address-based forwarding is better suited on a particular device. In this method, packets sent from the IP address A to IP address B, from IP address A to IP address C, and from IP address C to IP address B could all use different ports in the channel.

VLAN ID based Forwarding

With VLAN ID based forwarding, packets are distributed across the ports in the EtherChannel based on the VLAN ID of the incoming packets and other parameters that are mentioned in the chosen load balance method. Packets with different VLAN IDs will use different ports in the channel and packets with the same VLAN ID will use the same port in the channel. In case of double tagged frames, the outer VLAN ID will be considered. In case of untagged frames, load balancing will be based on the other parameters mentioned. For example, if **vlan-dst-ip** is chosen, then packets without a VLAN tag will be distributed based on the destination IP address. In case of VLAN translation, load balancing will be based on the translated VLAN ID.

Load-Balancing Advantages

Different load-balancing methods have different advantages, and the choice of a particular load-balancing method should be based on the position of the device in the network and the kind of traffic that needs to be load-distributed.

Figure 26: Load Distribution and Forwarding Methods

In the following figure, an EtherChannel of four workstations communicates with a router. Because the router is a single MAC-address device, source-based forwarding on the switch EtherChannel ensures that the switch uses all available bandwidth to the router. The router is configured for destination-based forwarding because the large number of workstations ensures that the traffic is evenly distributed from the router EtherChannel.

Use the option that provides the greatest variety in your configuration. For example, if the traffic on a channel is going only to a single MAC address, using the destination-MAC address always chooses the same link in the channel. Using source addresses or IP addresses might result in better load-balancing.

EtherChannel and Switch Stacks

If a stack member that has ports participating in an EtherChannel fails or leaves the stack, the active switch removes the failed stack member switch ports from the EtherChannel. The remaining ports of the EtherChannel, if any, continue to provide connectivity.

When a switch is added to an existing stack, the new switch receives the running configuration from the active switch and updates itself with the EtherChannel-related stack configuration. The stack member also receives the operational information (the list of ports that are up and are members of a channel).

When two stacks merge that have EtherChannels configured between them, self-looped ports result. Spanning tree detects this condition and acts accordingly. Any PAgP or LACP configuration on a winning switch stack is not affected, but the PAgP or LACP configuration on the losing switch stack is lost after the stack reboots.

Switch Stack and Port Aggregation Protocol

With PAgP, if the active switch fails or leaves the stack, the standby switch becomes the new active switch. The new active switch synchronizes the configuration of the stack members to that of the active switch. The PAgP configuration is not affected after an active switch change unless the EtherChannel has ports residing on the old active switch.

Switch Stacks and Link Aggregation Control Protocol

With LACP, the system ID uses the stack MAC address from the active switch. When an active switch fails or leaves the stack and the standby switch becomes the new active switch, the LACP system ID is unchanged. By default, the LACP configuration is not affected after the active switch changes.

Default EtherChannel Configuration

The default EtherChannel configuration is described in this table.

Table 12: Default EtherChannel Configuration

Feature	Default Setting
Channel groups	None assigned.
Port-channel logical interface	None defined.
PAgP mode	No default.
PAgP learn method	Aggregate-port learning on all ports.
PAgP priority	128 on all ports.
LACP mode	No default.
LACP learn method	Aggregate-port learning on all ports.

Feature	Default Setting
LACP port priority	32768 on all ports.
LACP system priority	32768.
LACP system ID	LACP system priority and the switch or stack MAC address.
Load-balancing	Load distribution on the switch is based on the source-MAC address of the incoming packet. The source-MAC address is src-dst-mixed-ip-port .

EtherChannel Configuration Guidelines

If improperly configured, some EtherChannel ports are automatically disabled to avoid network loops and other problems. Follow these guidelines to avoid configuration problems:

- A maximum of 192 EtherChannels are supported on Cisco Catalyst 9600 Series Supervisor 1 Module, and a maximum of 252 EtherChannels are supported on Cisco Catalyst 9600 Series Supervisor 2 Module.



Note On Cisco Catalyst 9600 Series Supervisor 1 Module, port channels 127 and 128 are reserved by default for StackWise Virtual mode.

- On Cisco Catalyst 9600 Series Supervisor 1 Module, configure a PAgP EtherChannel with up to eight Ethernet ports of the same type for port channels ranging from 1 to 128. You can configure a PAgP EtherChannel with up to four Ethernet ports of the same type for port channels ranging from 129 to 192.

On Cisco Catalyst 9600 Series Supervisor 2 Module, you can configure a PAgP EtherChannel with up to eight Ethernet ports of the same type for all 252 port channels.

- On Cisco Catalyst 9600 Series Supervisor 1 Module, configure a LACP EtherChannel with up to 16 Ethernet ports of the same type for port channels ranging from 1 to 128. Up to eight ports can be active, and up to eight ports can be in hot-standby mode. For port channel range starting from 129 to 192, configure a LACP EtherChannel with up to eight ports of the same type. Four ports can be active, and four ports can be in hot-standby mode.

On Cisco Catalyst 9600 Series Supervisor 2 Module, configure a LACP EtherChannel with up to 16 Ethernet ports of the same type for all 252 port channels. Up to eight ports can be active, and up to eight ports can be in hot-standby mode.

- Configure all ports in an EtherChannel to operate at the same speeds and duplex modes.
- Enable all ports in an EtherChannel. A port in an EtherChannel that is disabled by using the **shutdown** interface configuration command is treated as a link failure, and its traffic is transferred to one of the remaining ports in the EtherChannel.
- When a group is first created, all ports follow the parameters set for the first port to be added to the group. If you change the configuration of one of these parameters, you must also make the changes to all ports in the group:
 - Allowed-VLAN list

- Spanning-tree path cost for each VLAN
- Spanning-tree port priority for each VLAN
- Spanning-tree Port Fast setting
- Do not configure a port to be a member of more than one EtherChannel group.
- Do not configure an EtherChannel in both the PAGP and LACP modes. EtherChannel groups running PAGP and LACP can coexist on the same switch or on different switches in the stack. Individual EtherChannel groups can run either PAGP or LACP, but they cannot interoperate.
- Do not configure a port that is an active or a not-yet-active member of an EtherChannel as an IEEE 802.1x port. If you try to enable IEEE 802.1x on an EtherChannel port, an error message appears, and IEEE 802.1x is not enabled.
- If EtherChannels are configured on device interfaces, remove the EtherChannel configuration from the interfaces before globally enabling IEEE 802.1x on a device by using the **dot1x system-auth-control** global configuration command.

Layer 2 EtherChannel Configuration Guidelines

When configuring Layer 2 EtherChannels, follow these guidelines:

- Assign all ports in the EtherChannel to the same VLAN, or configure them as trunks. Ports with different native VLANs cannot form an EtherChannel.
- An EtherChannel supports the same allowed range of VLANs on all the ports in a trunking Layer 2 EtherChannel. If the allowed range of VLANs is not the same, the ports do not form an EtherChannel even when PAGP is set to the **auto** or **desirable** mode.
- Ports with different spanning-tree path costs can form an EtherChannel if they are otherwise compatibly configured. Setting different spanning-tree path costs does not, by itself, make ports incompatible for the formation of an EtherChannel.

Layer 3 EtherChannel Configuration Guidelines

For Layer 3 EtherChannels, assign the Layer 3 address to the port-channel logical interface, not to the physical ports in the channel.

Auto-LAG

The auto-LAG feature provides the ability to auto create EtherChannels on ports that are connected to a switch. By default, auto-LAG is disabled globally and is enabled on all port interfaces. The auto-LAG applies to a switch only when it is enabled globally.

On enabling auto-LAG globally, the following scenarios are possible:

- All port interfaces participate in creation of auto EtherChannels provided the partner port interfaces have EtherChannel configured on them. For more information, see the *"The supported auto-LAG configurations between the actor and partner devices"* table below.
- Ports that are already part of manual EtherChannels cannot participate in creation of auto EtherChannels.

- When auto-LAG is disabled on a port interface that is already a part of an auto created EtherChannel, the port interface unbundles from the auto EtherChannel.

The following table shows the supported auto-LAG configurations between the actor and partner devices:

Table 13: The supported auto-LAG configurations between the actor and partner devices

Actor/Partner	Active	Passive	Auto
Active	Yes	Yes	Yes
Passive	Yes	No	Yes
Auto	Yes	Yes	Yes

On disabling auto-LAG globally, all auto created Etherchannels become manual EtherChannels.

You cannot add any configurations in an existing auto created EtherChannel. To add, you should first convert it into a manual EtherChannel by executing the **port-channel**<channel-number>**persistent**.



Note Auto-LAG uses the LACP protocol to create auto EtherChannel. Only one EtherChannel can be automatically created with the unique partner devices.

Auto-LAG Configuration Guidelines

Follow these guidelines when configuring the auto-LAG feature.

- When auto-LAG is enabled globally and on the port interface, and if you do not want the port interface to become a member of the auto EtherChannel, disable the auto-LAG on the port interface.
- A port interface will not bundle to an auto EtherChannel when it is already a member of a manual EtherChannel. To allow it to bundle with the auto EtherChannel, first unbundle the manual EtherChannel on the port interface.
- When auto-LAG is enabled and auto EtherChannel is created, you can create multiple EtherChannels manually with the same partner device. But by default, the port tries to create auto EtherChannel with the partner device.
- The auto-LAG is supported only on Layer 2 EtherChannel. It is not supported on Layer 3 interface and Layer 3 EtherChannel.
- The auto-LAG is supported on cross-stack EtherChannel.

How to Configure EtherChannels

After you configure an EtherChannel, configuration changes applied to the port-channel interface apply to all the physical ports assigned to the port-channel interface, and configuration changes that are applied to the physical port affect only the port where you apply the configuration.

The following sections provide various configuration information for EtherChannels:

Configuring Layer 2 EtherChannels

Configure Layer 2 EtherChannels by assigning ports to a channel group with the **channel-group** command in interface configuration mode. This command automatically creates the port-channel logical interface.

Procedure

	Command or Action	Purpose
Step 1	enable Example: Device> enable	Enables privileged EXEC mode. Enter your password if prompted.
Step 2	configure terminal Example: Device# configure terminal	Enters global configuration mode.
Step 3	interface <i>interface-id</i> Example: Device(config)# interface gigabitethernet 1/0/1	Specifies a physical port, and enters interface configuration mode. Valid interfaces are physical ports. For a PAgP EtherChannel, you can configure up to eight ports of the same type and speed for the same group. For a LACP EtherChannel, you can configure up to 16 Ethernet ports of the same type. Up to eight ports can be active, and up to eight ports can be in standby mode.
Step 4	switchport mode {access trunk} Example: Device(config-if)# switchport mode access	Assigns all ports as static-access ports in the same VLAN, or configure them as trunks. If you configure the port as a static-access port, assign it to only one VLAN. The range is 1 to 4094.
Step 5	switchport access vlan <i>vlan-id</i> Example: Device(config-if)# switchport access vlan 22	(Optional) If you configure the port as a static-access port, assign it to only one VLAN. The range is 1 to 4094.
Step 6	channel-group <i>channel-group-number</i> mode {auto [non-silent] desirable [non-silent] on } { active passive} Example: Device(config-if)# channel-group 5 mode auto	Assigns the port to a channel group, and specifies the PAgP or the LACP mode. For mode , select one of these keywords: <ul style="list-style-type: none"> • auto –Enables PAgP only if a PAgP device is detected. It places the port into a passive negotiating state, in which the port responds to PAgP packets it receives but does not start PAgP packet negotiation.

	Command or Action	Purpose
		<ul style="list-style-type: none"> • desirable—Unconditionally enables PAgP. It places the port into an active negotiating state, in which the port starts negotiations with other ports by sending PAgP packets. • on—Forces the port to channel without PAgP or LACP. In the on mode, an EtherChannel exists only when a port group in the on mode is connected to another port group in the on mode. • non-silent—(Optional) If your device is connected to a partner that is PAgP-capable, configures the device port for nonsilent operation when the port is in the auto or desirable mode. If you do not specify non-silent, silent is assumed. The silent setting is for connections to file servers or packet analyzers. This setting allows PAgP to operate, to attach the port to a channel group, and to use the port for transmission. • active—Enables LACP only if a LACP device is detected. It places the port into an active negotiating state in which the port starts negotiations with other ports by sending LACP packets. • passive—Enables LACP on the port and places it into a passive negotiating state in which the port responds to LACP packets that it receives, but does not start LACP packet negotiation.
Step 7	end Example: Device(config-if)# end	Returns to privileged EXEC mode.

Configuring Layer 3 EtherChannels

Follow these steps to assign an Ethernet port to a Layer 3 EtherChannel. This procedure is required.

Procedure

	Command or Action	Purpose
Step 1	enable Example:	Enables privileged EXEC mode. Enter your password if prompted.

	Command or Action	Purpose
	Device> enable	
Step 2	configure terminal Example: Device# configure terminal	Enters global configuration mode.
Step 3	interface <i>interface-id</i> Example: Device(config)# interface gigabitethernet 1/0/2	Specifies a physical port, and enters interface configuration mode. Valid interfaces include physical ports. For a PAgP EtherChannel, you can configure up to eight ports of the same type and speed for the same group. For a LACP EtherChannel, you can configure up to 16 Ethernet ports of the same type. Up to eight ports can be active, and up to eight ports can be in standby mode.
Step 4	no ip address Example: Device(config-if)# no ip address	Ensures that there is no IP address assigned to the physical port.
Step 5	no switchport Example: Device(config-if)# no switchport	Puts the port into Layer 3 mode.
Step 6	channel-group <i>channel-group-number</i> mode { auto [non-silent] desirable [non-silent] on } { active passive } Example: Device(config-if)# channel-group 5 mode auto	Assigns the port to a channel group, and specifies the PAgP or the LACP mode. For mode , select one of these keywords: <ul style="list-style-type: none">• auto—Enables PAgP only if a PAgP device is detected. It places the port into a passive negotiating state, in which the port responds to PAgP packets it receives but does not start PAgP packet negotiation. This keyword is not supported when EtherChannel members are from different switches in the switch stack.• desirable—Unconditionally enables PAgP. It places the port into an active negotiating state, in which the port starts negotiations with other ports by sending PAgP packets. This keyword is not supported when EtherChannel members are from different switches in the switch stack.

	Command or Action	Purpose
		<ul style="list-style-type: none"> • on—Forces the port to channel without PAgP or LACP. In the on mode, an EtherChannel exists only when a port group in the on mode is connected to another port group in the on mode. • non-silent—(Optional) If your device is connected to a partner that is PAgP capable, configures the device port for nonsilent operation when the port is in the auto or desirable mode. If you do not specify non-silent, silent is assumed. The silent setting is for connections to file servers or packet analyzers. This setting allows PAgP to operate, to attach the port to a channel group, and to use the port for transmission. • active—Enables LACP only if a LACP device is detected. It places the port into an active negotiating state in which the port starts negotiations with other ports by sending LACP packets. • passive—Enables LACP on the port and places it into a passive negotiating state in which the port responds to LACP packets that it receives, but does not start LACP packet negotiation.
Step 7	end Example: Device(config-if)# end	Returns to privileged EXEC mode.

(Optional) Configuring EtherChannel Load-Balancing



Note Load-Balancing is enabled by default on the Cisco Catalyst 9600 Series Supervisor 2 Module.

You can configure EtherChannel load-balancing to use one of several different forwarding methods.

To configure EtherChannel Load-balancing, perform this procedure:

Procedure

	Command or Action	Purpose
Step 1	enable	Enables privileged EXEC mode.

	Command or Action	Purpose
	Example: Device> enable	Enter your password if prompted.
Step 2	configure terminal Example: Device# configure terminal	Enters global configuration mode.
Step 3	port-channel load-balance { dst-ip dst-mac dst-mixed-ip-port dst-port extended src-ip src-mac src-port src-dst-ip src-dst-mac src-dst-mixed-ip-port src-dst-port src-ip src-mac src-mixed-ip-port src-port vlan-dst-ip vlan-dst-mixed-ip-port vlan-src-dst-ip vlan-src-dst-mixed-ip-port vlan-src-ip vlan-src-mixed-ip-port } Example: Device (config) # port-channel load-balance src-mac	Configures an EtherChannel load-balancing method. The default is src-dst-mixed-ip-port . Select one of these load-distribution methods: <ul style="list-style-type: none"> • dst-ip—Specifies destination-host IP address. • dst-mac—Specifies the destination-host MAC address of the incoming packet. • dst-mixed-ip-port—Specifies the host IP address and TCP/UDP port. • dst-port—Specifies the destination TCP/UDP port. • src-dst-ip—Specifies the source and destination host IP address. • src-dst-mac—Specifies the source and destination host MAC address. • src-dst-mixed-ip-port—Specifies the source and destination host IP address and TCP/UDP port. • src-dst-port—Specifies the source and destination TCP/UDP port. • src-ip—Specifies the source host IP address. • src-mac—Specifies the source MAC address of the incoming packet. • src-mixed-ip-port—Specifies the source host IP address and TCP/UDP port. • src-port—Specifies the source TCP/UDP port. • vlan-dst-ip—Specifies the VLAN ID and destination IP address.

	Command or Action	Purpose
		<ul style="list-style-type: none"> • vlan-dst-mixed-ip-port—Specifies the VLAN ID, destination IP address, and TCP/UDP port number. • vlan-src-dst-ip—Specifies the VLAN ID, source and destination IP address. • vlan-src-dst-mixed-ip-port—Specifies the VLAN ID, source and destination IP address, and TCP/UDP port number. • vlan-src-ip—Specifies the VLAN ID and source IP address. • vlan-src-mixed-ip-port—Specifies the VLAN ID, source IP address, and TCP/UDP port number.
Step 4	end Example: Device(config)# end	Returns to privileged EXEC mode.

(Optional) Configuring EtherChannel Extended Load-Balancing



Note This feature is enabled by default on the Cisco Catalyst 9600 Series Supervisor 2 Module..

Configure EtherChannel extended load-balancing when you want to use a combination of load-balancing methods.

To configure EtherChannel extended load-balancing, perform this procedure:

Procedure

	Command or Action	Purpose
Step 1	enable Example: Device> enable	Enables privileged EXEC mode. Enter your password if prompted.
Step 2	configure terminal Example: Device# configure terminal	Enters global configuration mode.
Step 3	port-channel load-balance extended { dst-ip dst-mac dst-port ipv6-label l3-proto src-ip src-mac src-port }	Configures an EtherChannel extended load-balancing method. The default is src-dst-mixed-ip-port .

	Command or Action	Purpose
	Example: <pre>Device(config)# port-channel load-balance extended dst-ip dst-mac src-ip</pre>	Select one of these load-distribution methods: <ul style="list-style-type: none"> • dst-ip—Specifies destination-host IP address. • dst-mac—Specifies the destination-host MAC address of the incoming packet. • dst-port—Specifies the destination TCP/UDP port. • ipv6-label—Specifies the IPv6 flow label. • l3-proto—Specifies the Layer 3 protocol. • src-ip—Specifies the source host IP address. • src-mac—Specifies the source MAC address of the incoming packet. • src-port—Specifies the source TCP/UDP port.
Step 4	end Example: <pre>Device(config)# end</pre>	Returns to privileged EXEC mode.

(Optional) Configuring the Port Aggregation Protocol Learn Method and Priority

To configure the PAgP learn method and priority, perform this procedure:

Procedure

	Command or Action	Purpose
Step 1	enable Example: <pre>Device> enable</pre>	Enables privileged EXEC mode. Enter your password if prompted.
Step 2	configure terminal Example: <pre>Device# configure terminal</pre>	Enters global configuration mode.
Step 3	interface <i>interface-id</i> Example: <pre>Device(config)# interface gigabitethernet 1/0/2</pre>	Specifies the port for transmission, and enters interface configuration mode.
Step 4	pagp learn-method physical-port	Selects the PAgP learning method.

	Command or Action	Purpose
	Example: Device(config-if) # pagp learn-method physical port	By default, aggregation-port learning is selected, which means the device sends packets to the source by using any of the ports in the EtherChannel. With aggregate-port learning, it is not important on which physical port the packet arrives. Selects physical-port to connect with another device that is a physical learner. Make sure to configure the port-channel load-balance global configuration command to src-mac . The learning method must be configured the same at both ends of the link.
Step 5	pagp port-priority priority Example: Device(config-if) # pagp port-priority 200	Assigns a priority so that the selected port is chosen for packet transmission. For <i>priority</i> , the range is 0 to 255. The default is 128. The higher the priority, the more likely that the port will be used for PAgP transmission.
Step 6	end Example: Device(config-if) # end	Returns to privileged EXEC mode.

Configuring Link Aggregation Control Protocol Hot-Standby Ports

When LACP is enabled, the software, by default, tries to configure the maximum number of LACP-compatible ports in a channel, up to a maximum of 16 ports. Only eight LACP links can be active at one time; the remaining eight links are placed in hot-standby mode. If one of the active links becomes inactive, a link that is in the hot-standby mode becomes active in its place.

You can override the default behavior by specifying the maximum number of active ports in a channel, in which case, the remaining ports become hot-standby ports. For example, if you specify a maximum of five ports in a channel, up to 11 ports become hot-standby ports.

If you configure more than eight links for an EtherChannel group, the software automatically decides which of the hot-standby ports to make active based on the LACP priority. To every link between systems that operate LACP, the software assigns a unique priority that is made up of these elements (in priority order):

- LACP system priority
- System ID (the device MAC address)
- LACP port priority
- Port number

In priority comparisons, numerically lower values have higher priority. The priority decides which ports should be put in standby mode when there is a hardware limitation that prevents all compatible ports from aggregating.

Determining which ports are active and which are hot standby is a two-step procedure. First the system with a numerically lower system priority and system ID is placed in charge of the decision. Next, that system decides which ports are active and which are hot standby, based on its values for port priority and port number. The port priority and port number values for the other system are not used.

You can change the default values of the LACP system priority and the LACP port priority to affect how the software selects active and standby links.

(Optional) Configuring the Link Aggregation Control Protocol Max Bundle

When you specify the maximum number of bundled LACP ports allowed in a port channel, the remaining ports in the port channel are designated as hot-standby ports.

Beginning in privileged EXEC mode, follow these steps to configure the maximum number of LACP ports in a port channel.

Procedure

	Command or Action	Purpose
Step 1	enable Example: Device> enable	Enables privileged EXEC mode. Enter your password if prompted.
Step 2	configure terminal Example: Device# configure terminal	Enters global configuration mode.
Step 3	interface port-channel <i>channel-number</i> Example: Device(config)# interface port-channel 2	Enters interface configuration mode for a port channel. For <i>channel-number</i> , the range is 1 to 192.
Step 4	lACP max-bundle <i>max-bundle-number</i> Example: Device(config-if)# lACP max-bundle 3	Specifies the maximum number of LACP ports in the port-channel bundle. The range is 1 to 8.
Step 5	end Example: Device(config-if)# end	Returns to privileged EXEC mode.

Configuring Link Aggregation Control Protocol Port-Channel Standalone Disable

To disable the standalone EtherChannel member port state on a port channel, perform this task on the port channel interface:

Procedure

	Command or Action	Purpose
Step 1	enable Example: Device> enable	Enables privileged EXEC mode. Enter your password if prompted.
Step 2	configure terminal Example: Device# configure terminal	Enters global configuration mode.
Step 3	interface port-channel <i>channel-group</i> Example: Device(config)# interface port-channel <i>channel-group</i>	Selects a port channel interface to configure.
Step 4	port-channel standalone-disable Example: Device(config-if)# port-channel standalone-disable	Disables the standalone mode on the port-channel interface.
Step 5	end Example: Device(config-if)# end	Exits configuration mode.
Step 6	show etherchannel Example: Device# show etherchannel <i>channel-group</i> port-channel Device# show etherchannel <i>channel-group</i> detail	Verifies the configuration.

Configuring Link Aggregation Control Protocol Standalone Mode on EtherChannel

To configure LACP Standalone or Independent mode on an EtherChannel, perform the following procedure:

Procedure

	Command or Action	Purpose
Step 1	enable Example: Device> enable	Enables privileged EXEC mode. Enter your password, if prompted.
Step 2	configure terminal Example: Device# configure terminal	Enters global configuration mode.

	Command or Action	Purpose
Step 3	interface port-channel <i>channel-number</i> Example: Device(config)# interface port-channel 1	Enters interface configuration mode for a port channel.
Step 4	no port-channel standalone-disable Example: Device(config-if)# no port-channel standalone-disable	Enables the LACP standalone or independent mode.
Step 5	end Example: Device(config-if)# end	Returns to privileged EXEC mode.
Step 6	show running-configuration (<i>fastethernet gigabitethernet</i>) <i>slot</i> port-channel <i>port-channel number</i> Example: Device# show running-config interface port-channel 1	Verifies the configuration.

Configuring the Link Aggregation Control Protocol Port Channel Min-Links

You can specify the minimum number of active ports that must be in the link-up state and bundled in an EtherChannel for the port channel interface to transition to the link-up state. Using EtherChannel min-links, you can prevent low-bandwidth LACP EtherChannels from becoming active. Port channel min-links also cause LACP EtherChannels to become inactive if they have too few active member ports to supply the required minimum bandwidth.

To configure the minimum number of links that are required for a port channel. Perform the following tasks.

Procedure

	Command or Action	Purpose
Step 1	enable Example: Device> enable	Enables privileged EXEC mode. Enter your password if prompted.
Step 2	configure terminal Example: Device# configure terminal	Enters global configuration mode.
Step 3	interface port-channel <i>channel-number</i> Example: Device(config)# interface port-channel 2	Enters interface configuration mode for a port-channel. For <i>channel-number</i> , the range is 1 to 192.

	Command or Action	Purpose
Step 4	port-channel min-links <i>min-links-number</i> Example: Device(config-if) # port-channel min-links 3	Specifies the minimum number of member ports that must be in the link-up state and bundled in the EtherChannel for the port channel interface to transition to the link-up state. For <i>min-links-number</i> , the range is 2 to 8 if the port channel number is 128 or lesser. For <i>min-links-number</i> , the range is 2 to 4 if the port channel number is 129 or greater.
Step 5	end Example: Device(config) # end	Returns to privileged EXEC mode.

(Optional) Configuring the Link Aggregation Control Protocol System Priority

You can configure the system priority for all the EtherChannels that are enabled for LACP by using the **lACP system-priority** command in global configuration mode. You cannot configure a system priority for each LACP-configured channel. By changing this value from the default, you can affect how the software selects active and standby links.

You can use the **show etherchannel summary** command in privileged EXEC mode to see which ports are in the hot-standby mode (denoted with an H port-state flag).

Follow these steps to configure the LACP system priority.

Procedure

	Command or Action	Purpose
Step 1	enable Example: Device> enable	Enables privileged EXEC mode. Enter your password if prompted.
Step 2	configure terminal Example: Device# configure terminal	Enters global configuration mode.
Step 3	lACP system-priority <i>priority</i> Example: Device(config) # lACP system-priority 32000	Configures the LACP system priority. The range is 1 to 65535. The default is 32768. The lower the value, the higher the system priority.
Step 4	end Example: Device(config) # end	Returns to privileged EXEC mode.

(Optional) Configuring the Link Aggregation Control Protocol Port Priority

By default, all ports use the same port priority. If the local system has a lower value for the system priority and the system ID than the remote system, you can affect which of the hot-standby links become active first by changing the port priority of LACP EtherChannel ports to a lower value than the default. The hot-standby ports that have lower port numbers become active in the channel first. You can use the **show etherchannel summary** privileged EXEC command to see which ports are in the hot-standby mode (denoted with an H port-state flag).



Note If LACP is not able to aggregate all the ports that are compatible (for example, the remote system might have more restrictive hardware limitations), all the ports that cannot be actively included in the EtherChannel are put in the hot-standby state and are used only if one of the channeled ports fails.

Follow these steps to configure the LACP port priority.

Procedure

	Command or Action	Purpose
Step 1	enable Example: Device> enable	Enables privileged EXEC mode. Enter your password if prompted.
Step 2	configure terminal Example: Device# configure terminal	Enters global configuration mode.
Step 3	interface <i>interface-id</i> Example: Device(config)# interface gigabitethernet 1/0/2	Specifies the port to be configured, and enters interface configuration mode.
Step 4	lACP port-priority <i>priority</i> Example: Device(config-if)# lACP port-priority 32000	Configures the LACP port priority. The range is 1 to 65535. The default is 32768. The lower the value, the more likely that the port will be used for LACP transmission.
Step 5	end Example: Device(config-if)# end	Returns to privileged EXEC mode.

Configuring Link Aggregation Control Protocol 1:1 Redundancy



- Note**
- LACP 1:1 redundancy must be enabled at both ends of the LACP EtherChannel.
 - For the LACP 1:1 Redundancy feature to work, the **lACP max-bundle 1** command must be configured along with the **lACP fast-switchover** command.
 - For the LACP 1:1 Hot Standby Dampening feature to work, the **lACP max-bundle 1** and **lACP fast-switchover** commands must be configured before the **lACP fast-switchover dampening** command is configured.

To configure LACP 1:1 redundancy, perform this procedure:

Procedure

	Command or Action	Purpose
Step 1	enable Example: Device> enable	Enables privileged EXEC mode. Enter your password, if prompted.
Step 2	configure terminal Example: Device# configure terminal	Enters global configuration mode.
Step 3	interface port-channel <i>group_number</i> Example: Device(config)# interface port-channel 40	Selects an LACP port channel interface and enters interface configuration mode.
Step 4	lACP fast-switchover Example: Device(config-if)# lACP fast-switchover	Enables the LACP 1:1 Redundancy feature on the EtherChannel.
Step 5	lACP max-bundle 1 Example: Device(config-if)# lACP max-bundle 1	Sets the maximum number of active member ports to be one. The only value that is supported with LACP 1:1 redundancy is 1.
Step 6	lACP fast-switchover dampening <i>seconds</i> Example: Device(config-if)# lACP fast-switchover dampening 60	(Optional) Enables the LACP 1:1 Hot Standby Dampening feature for this EtherChannel. The range for the time parameter is from 30 to 180 seconds.
Step 7	end Example: Device(config-if)# end	Exits interface configuration mode and returns to privileged EXEC mode.

Configuring Link Aggregation Control Protocol 1:1 Redundancy Fast Rate Timer

You can change the LACP timer rate to modify the duration of the LACP timeout. Use the **lACP rate** command to set the rate at which LACP control packets are received by an LACP-supported interface. You can change the timeout rate from the default rate (30 seconds) to the fast rate (1 second). This command is supported only on LACP-enabled interfaces.

To configure LACP 1:1 redundancy fast rate timer, perform this procedure:

Procedure

	Command or Action	Purpose
Step 1	enable Example: Device> enable	Enables privileged EXEC mode. Enter your password if prompted.
Step 2	configure terminal Example: Device# configure terminal	Enters global configuration mode.
Step 3	interface {fastethernet gigabitethernet tengigabitethernet} slot/port Example: Device (config)# interface gigabitEthernet 2/1	Configures an interface and enters interface configuration mode.
Step 4	lACP rate {normal fast} Example: Device (config-if)# lACP rate fast	Configures the rate at which LACP control packets are received by an LACP-supported interface. To reset the timeout rate to its default, use the no lACP rate command.
Step 5	end Example: Device (config)# end	Returns to privileged EXEC mode.
Step 6	show lACP internal Example: Device# show lACP internal Device# show lACP counters	Verifies your configuration.

Configuring Auto-LAG Globally

To configure Auto-LAG globally, perform this procedure:

Procedure

	Command or Action	Purpose
Step 1	enable Example: Device> enable	Enables privileged EXEC mode. Enter your password if prompted.
Step 2	configure terminal Example: Device# configure terminal	Enters global configuration mode.
Step 3	[no] port-channel auto Example: Device(config)# port-channel auto	Enables the auto-LAG feature on a switch globally. Use the no form of this command to disable the auto-LAG feature on the switch globally. Note By default, the auto-LAG feature is enabled on the port.
Step 4	end Example: Device(config)# end	Returns to privileged EXEC mode.
Step 5	show etherchannel auto Example: Device# show etherchannel auto	Displays that EtherChannel is created automatically.

Configuring Auto-LAG on a Port Interface

To configure Auto-LAG on a port interface, perform this procedure:

Procedure

	Command or Action	Purpose
Step 1	enable Example: Device> enable	Enables privileged EXEC mode. Enter your password if prompted.
Step 2	configure terminal Example: Device# configure terminal	Enters global configuration mode.
Step 3	interface <i>interface-id</i> Example: Device(config)# interface gigabitethernet 1/0/1	Specifies the port interface to be enabled for auto-LAG, and enters interface configuration mode.

	Command or Action	Purpose
Step 4	[no] channel-group auto Example: Device(config-if) # channel-group auto	(Optional) Enables auto-LAG feature on individual port interface. Use the no form of this command to disable the auto-LAG feature on individual port interface. Note By default, the auto-LAG feature is enabled on the port.
Step 5	end Example: Device(config-if) # end	Returns to privileged EXEC mode.
Step 6	show etherchannel auto Example: Device# show etherchannel auto	Displays that EtherChannel is created automatically.

Configuring Persistence with Auto-LAG

You use the persistence command to convert the auto created EtherChannel into a manual one and allow you to add configuration on the existing EtherChannel.

To configure persistence with Auto-LAG, perform this procedure:

Procedure

	Command or Action	Purpose
Step 1	enable Example: Device> enable	Enables privileged EXEC mode. Enter your password if prompted.
Step 2	port-channel <i>channel-number</i> persistent Example: Device# port-channel 1 persistent	Converts the auto created EtherChannel into a manual one and allows you to add configuration on the EtherChannel.
Step 3	show etherchannel summary Example: Device# show etherchannel summary	Displays the EtherChannel information.

Monitoring EtherChannel, Port Aggregation Protocol, and Link Aggregation Control Protocol Status

You can display EtherChannel, PAgP, and LACP status using the commands listed in this table.

Table 14: Commands for Monitoring EtherChannel, PAgP, and LACP Status

Command	Description
clear lacp { <i>channel-group-number</i> counters counters }	Clears LACP channel-group information and traffic counters.
clear pagp { <i>channel-group-number</i> counters counters }	Clears PAgP channel-group information and traffic counters.
show etherchannel [<i>channel-group-number</i> { detail load-balance port port-channel protocol summary }] [detail load-balance port port-channel protocol auto summary]	Displays EtherChannel information in a brief, detailed, and one-line summary form. Also displays the load-balance or frame-distribution scheme, port, port-channel, protocol, and Auto-LAG information.
show pagp [<i>channel-group-number</i>] { counters internal neighbor }	Displays PAgP information such as traffic information, the internal PAgP configuration, and neighbor information.
show pagp [<i>channel-group-number</i>] dual-active	Displays the dual-active detection status.
show lacp [<i>channel-group-number</i>] { counters internal neighbor sys-id }	Displays LACP information such as traffic information, the internal LACP configuration, and neighbor information.
show running-config	Verifies your configuration entries.
show etherchannel load-balance	Displays the load balance or frame distribution scheme among ports in the port channel.

Configuration Examples for EtherChannels

The following sections provide various configuration examples for EtherChannels:

Example: Configuring Layer 2 EtherChannels

This example shows how to configure an EtherChannel on a single switch in the stack. It assigns two ports as static-access ports in VLAN 10 to channel 5 with the PAgP mode **desirable**:

```
Device# configure terminal
Device(config)# interface range gigabitethernet2/0/1 -2
Device(config-if-range)# switchport mode access
Device(config-if-range)# switchport access vlan 10
```

```
Device(config-if-range)# channel-group 5 mode desirable non-silent
Device(config-if-range)# end
```

This example shows how to configure an EtherChannel on a single switch in the stack. It assigns two ports as static-access ports in VLAN 10 to channel 5 with the LACP mode **active**:

```
Device# configure terminal
Device(config)# interface range gigabitethernet2/0/1 -2
Device(config-if-range)# switchport mode access
Device(config-if-range)# switchport access vlan 10
Device(config-if-range)# channel-group 5 mode active
Device(config-if-range)# end
```

This example shows how to configure a cross-stack EtherChannel. It uses LACP passive mode and assigns two ports on stack member 1 and one port on stack member 2 as static-access ports in VLAN 10 to channel 5:

```
Device# configure terminal
Device(config)# interface range gigabitethernet2/0/4 -5
Device(config-if-range)# switchport mode access
Device(config-if-range)# switchport access vlan 10
Device(config-if-range)# channel-group 5 mode passive
Device(config-if-range)# exit
Device(config)# interface gigabitethernet3/0/3
Device(config-if)# switchport mode access
Device(config-if)# switchport access vlan 10
Device(config-if)# channel-group 5 mode passive
Device(config-if)# exit
```

PoE or LACP negotiation errors may occur if you configure two ports from switch to the access point (AP). This scenario can be avoided if the port channel configuration is on the switch side. For more details, see the following example:

```
Device(config)# interface Port-channel1
Device(config-if)# switchport access vlan 20
Device(config-if)# switchport mode access
Device(config-if)# switchport nonegotiate
Device(config-if)# no port-channel standalone-disable
Device(config-if)# spanning-tree portfast
```



Note If the port reports LACP errors on port flap, you should include the following command as well: **no errdisable detect cause pagp-flap**

Example: Configuring Layer 3 EtherChannels

This example shows how to configure a Layer 3 EtherChannel. It assigns two ports to channel 5 with the LACP mode **active**:

```
Device# configure terminal
Device(config)# interface range gigabitethernet2/0/1 -2
Device(config-if-range)# no ip address
Device(config-if-range)# no switchport
Device(config-if-range)# channel-group 5 mode active
Device(config-if-range)# end
```

This example shows how to configure a cross-stack Layer 3 EtherChannel. It assigns two ports on stack member 2 and one port on stack member 3 to channel 7 using LACP active mode:

```

Device# configure terminal
Device(config)# interface range gigabitethernet2/0/4 -5
Device(config-if-range)# no ip address
Device(config-if-range)# no switchport
Device(config-if-range)# channel-group 7 mode active
Device(config-if-range)# exit
Device(config)# interface gigabitethernet3/0/3
Device(config-if)# no ip address
Device(config-if)# no switchport
Device(config-if)# channel-group 7 mode active
Device(config-if)# exit

```

Example: Configuring Link Aggregation Control Protocol Hot-Standby Ports

This example shows how to configure an EtherChannel (port channel 2) that will be active when there are at least three active ports, will comprise up to seven active ports and the remaining ports (up to nine) as hot-standby ports:

```

Device# configure terminal
Device(config)# interface port-channel 2
Device(config-if)# port-channel min-links 3
Device(config-if)# lacp max-bundle 7

```

Example: Configuring Link Aggregation Control Protocol 1:1 Redundancy

This example shows how to configure the LACP 1:1 Redundancy feature on the EtherChannel:

```

Device> enable
Device# configure terminal
Device(config)# interface port-channel 40
Device(config-if)# lacp fast-switchover
Device(config-if)# lacp max-bundle 1
Device(config-if)# lacp fast-switchover dampening 60
Device(config-if)# end

```

This is a sample output from the **show lacp internal** command:

```

Device# show lacp 1 internal

Flags: S - Device is requesting Slow LACPDUs
       F - Device is requesting Fast LACPDUs
       A - Device is in Active mode
       P - Device is in Passive mode

Channel group 1, [146 s left to exit dampening state]

Port      Flags   State      LACP port   Admin   Oper   Port      Port
Fal/1     FA      hot-sby    30000*      0x1     0x1    0x103     0x7
Fal/2     SA      bndl       32768       0x1     0x1    0x102     0x3D

```

Example: Configuring Standalone Mode on EtherChannel

This example shows how to configure the Standalone mode or Independent mode on an Port channel:

```

Device(config)# interface port-channel 1
Device(config-if)# no port-channel standalone-disable
Device(config-if)# end

```

This example shows how to verify the configuration of the Standalone mode on a Port Channel interface:

```
Device# show running-config interface port-channel 1
Building configuration...
Current configuration:
!
interface Port-channell
  no ip address
  no switchport
  no port-channel standalone-disable
end
```

Example: Configuring Auto LAG

This example shows how to configure Auto-LAG on a switch

```
Device> enable
Device# configure terminal
Device(config)# port-channel auto
Device(config-if)# end
Device# show etherchannel auto
```

This example shows the summary of EtherChannel that was created automatically.

```
Device# show etherchannel auto
Flags: D - down          P - bundled in port-channel
       I - stand-alone  s - suspended
       H - Hot-standby (LACP only)
       R - Layer3       S - Layer2
       U - in use       f - failed to allocate aggregator
       M - not in use, minimum links not met
       u - unsuitable for bundling
       w - waiting to be aggregated
       d - default port
       A - formed by Auto LAG
```

```
Number of channel-groups in use: 1
Number of aggregators:          1
```

Group	Port-channel	Protocol	Ports
1	Pol(SUA)	LACP	Gi1/0/45(P) Gi2/0/21(P) Gi3/0/21(P)

This example shows the summary of auto EtherChannel after executing the **port-channel 1 persistent** command.

```
Device# port-channel 1 persistent
```

```
Device# show etherchannel summary
Switch# show etherchannel summary
Flags: D - down          P - bundled in port-channel
       I - stand-alone  s - suspended
       H - Hot-standby (LACP only)
       R - Layer3       S - Layer2
       U - in use       f - failed to allocate aggregator
       M - not in use, minimum links not met
       u - unsuitable for bundling
       w - waiting to be aggregated
       d - default port
       A - formed by Auto LAG
```

```
Number of channel-groups in use: 1
Number of aggregators:          1
```


Group	Port-channel	Protocol	Ports
1	Po1 (SU)	LACP	Gi1/0/45 (P) Gi2/0/21 (P) Gi3/0/21 (P)

Additional References for EtherChannels

Related Documents

Related Topic	Document Title
For complete syntax and usage information for the commands used in this chapter.	See the Layer 2/3 Commands section of the <i>Command Reference (Catalyst 9600 Series Switches)</i>

Feature History for EtherChannels

This table provides release and related information for features explained in this module.

These features are available on all releases subsequent to the one they were introduced in, unless noted otherwise.

Release	Feature	Feature Information
Cisco IOS XE Gibraltar 16.11.1	EtherChannels	EtherChannel provides fault-tolerant high-speed links between switches, routers, and servers.
Cisco IOS XE Amsterdam 17.3.1	LACP 1:1 Redundancy and Dampening	The LACP 1:1 Redundancy feature supports an EtherChannel configuration with one active link and fast switchover to a hot-standby link. The LACP 1:1 Hot Standby Dampening feature configures a timer that delays switchover back to the higher priority port after it becomes active.
Cisco IOS XE Bengaluru 17.4.1	EtherChannels Support in non-StackWise Virtual setup and StackWise Virtual setup	A maximum of 192 EtherChannels are supported on a switch or switch stack.
Cisco IOS XE Dublin 17.10.1	Standalone Mode on Layer 3 Etherchannels	Support for configuring Standalone mode or Independent mode on Layer 3 EtherChannels was introduced in this release.

Use Cisco Feature Navigator to find information about platform and software image support. To access Cisco Feature Navigator, go to <http://www.cisco.com/go/cfn>.



CHAPTER 6

Configuring Precision Time Protocol (PTP)

- [Restrictions and Limitations for Precision Time Protocol, on page 127](#)
- [Information About Precision Time Protocol, on page 128](#)
- [How to Configure Precision Time Protocol, on page 141](#)
- [Configuration Examples for Precision Time Protocol, on page 150](#)
- [Verifying Precision Time Protocol Configurations, on page 151](#)
- [Feature History for Precision Time Protocol, on page 157](#)

Restrictions and Limitations for Precision Time Protocol

- Synchronization of PTP clock from system clock and vice versa is not supported.
- Inter-VLAN is not supported in PTP Transparent Clock Mode.
- The switch supports IEEE 802.1AS and IEEE 1588 Default profile and they are both mutually exclusive. Only one profile can be enabled on the switch at a time.
- The Cisco PTP implementation supports only the two-step clock and not the one-step clock. If the switch receives a one-step message from the Grand Master Clock, the message will be dropped.
- We do not recommend having non-PTP enabled devices in the PTP network since it decreases clock synchronization accuracy.
- Signaling messages are not supported in Cisco IOS XE Gibraltar 16.12.1. These messages are dropped in the switch without being processed.
- Management messages with broadcast target id will be forwarded with a reduced hop count when the boundary clock mode is enabled. Management messages will be forwarded without decreasing the boundary hop count when transparent clock mode is enabled.
- Moving directly from one PTP mode to the other is not recommended. Clear the existing mode using **no PTP mode** and then configure a new mode.
- IPv6 and VRF do not support PTP.
- Transparent clock mode is not supported on native Layer 3 ports and EtherChannel interfaces. (boundary clock mode is supported on native Layer 3 ports)
- PTP cannot be configured on supervisor modules.

The line card models that are supported are C9600-LC-24C, C9600-LC-48S, C9600-LC-48YL, and C9600-LC-48TX.

- PTP is not supported on Cisco StackWise Virtual configured devices.

Information About Precision Time Protocol

Precision Time Protocol (PTP) is defined in IEEE 1588 as Precision Clock Synchronization for Networked Measurements and Control Systems, and was developed to synchronize the clocks in packet-based networks that include distributed device clocks of varying precision and stability. PTP is designed specifically for industrial, networked measurement and control systems, and is optimal for use in distributed systems because it requires minimal bandwidth and little processing overhead.

Smart grid power automation applications such as peak-hour billing, virtual power generators, and outage monitoring and management, require extremely precise time accuracy and stability. Timing precision improves network monitoring accuracy and troubleshooting ability.

In addition to providing time accuracy and synchronization, the PTP message-based protocol can be implemented on packet-based networks, such as Ethernet networks. The benefits of using PTP in an Ethernet network include:

- Low cost and easy setup in existing Ethernet networks
- Limited bandwidth required for PTP data packets

Ethernet Switches and Delays

In an Ethernet network, switches provide a full-duplex communication path between network devices. Switches send data packets to packet destinations using address information contained in the packets. When the switch attempts to send multiple packets simultaneously, some of the packets are buffered by the switch so that they are not lost before they are sent. When the buffer is full, the switch delays sending packets. This delay can cause device clocks on the network to lose synchronization with one another.

Additional delays occur when packets entering a switch are stored in local memory while the switch searches the MAC address table to verify packet CRC fields. This process causes variations in packet forwarding time latency, and these variations result in asymmetrical packet delay times.

Adding PTP to a network can compensate for these latency and delay by correctly adjusting device clocks so that they stay synchronized with one another. PTP enables network switches to function as PTP devices, including boundary clocks and transparent clocks.

Message-Based Synchronisation

To ensure clock synchronization, PTP requires an accurate measurement of the communication path delay between the time source (master) and the receiver (slave). PTP sends messages between the master and slave device to determine the delay measurement. Messages are described in detail in [#unique_194](#). Then, PTP measures the exact message transmit and receive times and uses these times to calculate the communication path delay. PTP then adjusts current time information network data for the calculated delay, resulting in more accurate time information.

This delay measurement principle determines path delay between devices on the network, and the local clocks are adjusted for this delay using a series of messages that are sent between masters and slaves. The one-way delay time is calculated by averaging the path delay of the transmit and receive messages. This calculation assumes a symmetrical communication path; switched networks do not necessarily have symmetrical communication paths, due to the buffering process.

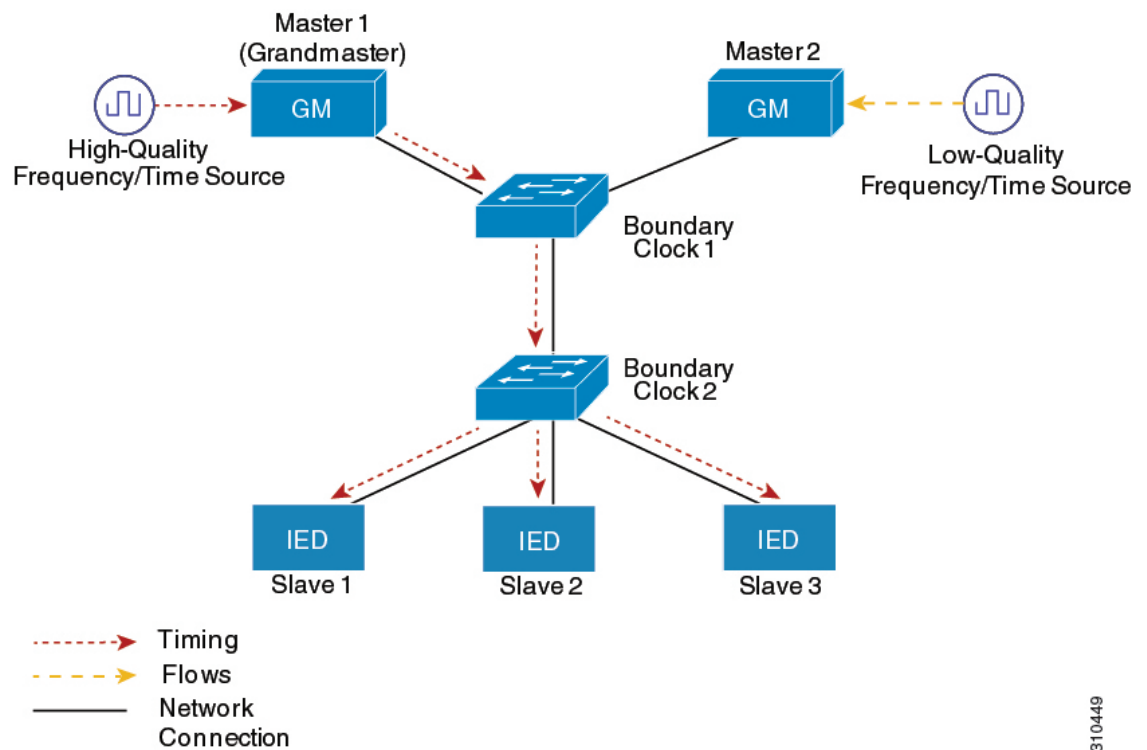
PTP provides a method, using transparent clocks, to measure and account for the delay in a time-interval field in network timing packets, making the switches temporarily transparent to the master and slave nodes on the network. An end-to-end transparent clock forwards all messages on the network the same way that a switch does.



Note Cisco PTP supports multicast PTP messages only.

The following figure shows a typical 1588 PTP network that includes grandmaster clocks, switches in boundary clock mode, and Intelligent Electronic Device (IEDs) such as a digital relays or protection devices. In this diagram, Master 1 is the grandmaster clock. If Master 1 becomes unavailable, the boundary clock slaves switch to Master 2 for synchronization.

Figure 27: A PTP Network with Switches in Boundary Clock Mode



310449



Note PTP with etherchannel interface and MACsec is supported.

Precision Time Protocol Version 2 Message Types

PTP messages are categorized into the following types:

Event Messages are tagged with timestamps when data packets reach or leave a port and are used to calculate the link delay based on the timestamps. Messages:

- Sync
- Delay_Req
- Pdelay_Req
- Pdelay_Resp

General Messages are not tagged with timestamps and are used to establish a master-slave hierarchy. General messages are listed below:

- Announce
- Follow_Up
- Delay_Resp
- Pdelay_Resp_Follow_Up

Announce messages are used to establish the synchronization hierarchy.

Sync, Delay_Req, Follow_Up, and Delay_Resp messages are used to synchronize ordinary and boundary clocks.

Pdelay_Req, Pdelay_Resp, and Pdelay_Resp_Follow_Up messages are used to measure the link delay in transparent clocks.

The [\(Best Master Clock Algorithm \(BMCA\)\)](#) elects the grandmaster clock and assign the ports as master or slave. Following this, all the master ports start sourcing the clock to the downstream slaves using the Sync and Follow_Up messages. The downstream slaves receive the clock and update their clock after computing the delay of the link, time offset, frequency offset and drift error parameters.

The downstream slaves compute the link delay using one of the mechanisms.

- [End-to-End Delay Mechanism, on page 131](#)
- [Peer-to-Peer Delay Mechanism, on page 132](#)

Precision Time Protocol and Software Defined Architecture Overlay

- Software Defined Architecture (SDA) fabric switches do not support PTP messages with hardware timestamp.
- PTP clock accuracy and latency gets affected when overlay PTP messages are forwarded without hardware timestamp. PTP clock accuracy and latency gets affected as the number of hops increases.
- Configure multicast protocols — IGMP and PIM to forward Layer 2, Layer 3 format of PTP messages in overlay.

Precision Time Protocol Event Message Sequences

This section describes the PTP event message sequences that occur during synchronization.

End-to-End Delay Mechanism

The ordinary and boundary clocks that are configured for the delay request-response mechanism use the following event messages to generate and communicate timing information:

- Sync
- Delay_Req
- Follow_Up
- Delay_Resp

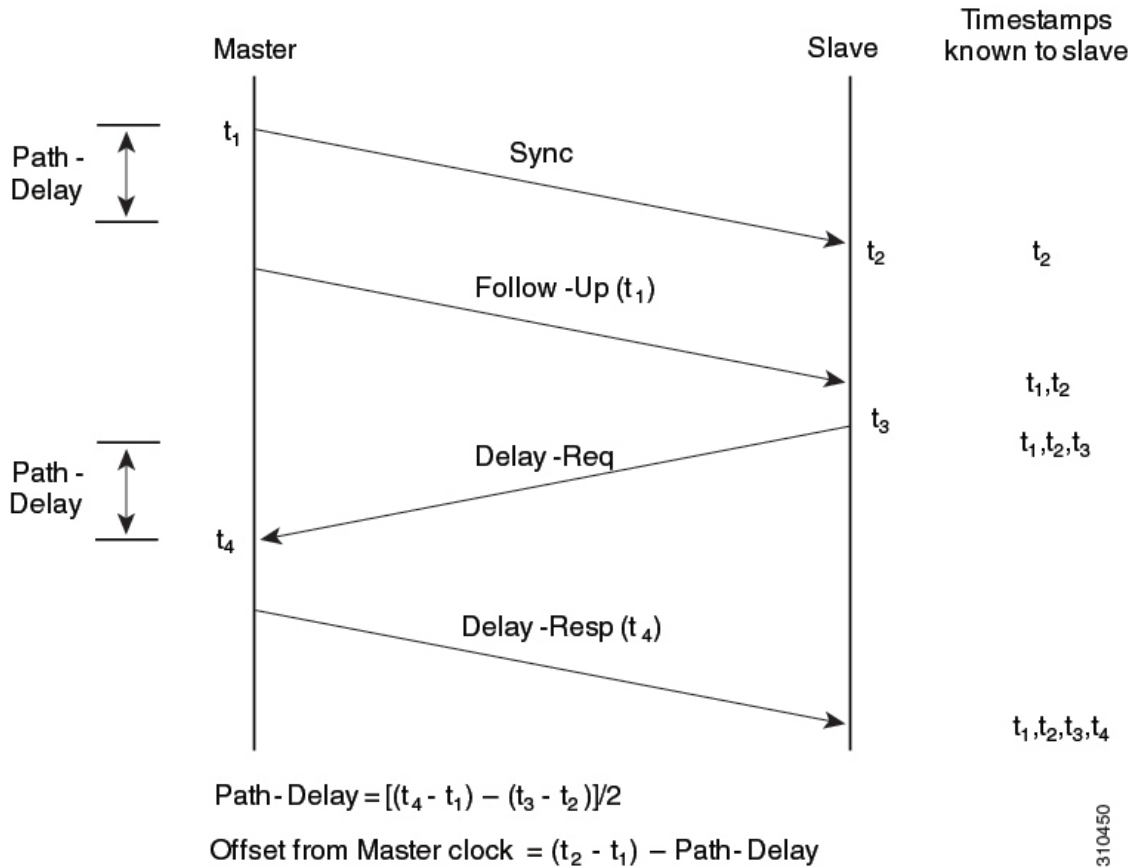
These messages are sent in the following sequence:

1. The master sends a Sync message to the slave and notes the time (t1) at which it was sent.
2. The slave receives the Sync message and notes the time of reception (t2).
3. The master conveys to the slave the timestamp t1 by embedding the timestamp t1 in a Follow_Up message.
4. The slave sends a Delay_Req message to the master and notes the time (t3) at which it was sent.
5. The master receives the Delay_Req message and notes the time of reception (t4).
6. The master conveys to the slave the timestamp t4 by embedding it in a Delay_Resp message.

After this sequence, the slave possesses all four timestamps. These timestamps can be used to compute the offset of the slave clock relative to the master, and the mean propagation time of messages between the two clocks.

The offset calculation is based on the assumption that the time for the message to propagate from master to slave is the same as the time required from slave to master. Assumption is not always valid on an Ethernet network due to asymmetrical packet delay times.

Figure 28: End-to-End Delay Mechanism



Peer-to-Peer Delay Mechanism

When the network includes multiple levels of boundary clocks in the hierarchy, with non-PTP enabled devices between them, synchronization accuracy decreases.

The round-trip time is assumed to be equal to `mean_path_delay/2`, this is not always valid for Ethernet networks. To improve accuracy, the resident time of each intermediary clock is added to the offset in the end-to-end transparent clock. Resident time, however, does not take into consideration the link delay between peers, which is handled by peer-to-peer transparent clocks.

Peer-to-peer transparent clocks measure the link delay between two clock ports implementing the peer delay mechanism. The link delay is used to correct timing information in Sync and Follow_Up messages.

Peer-to-peer transparent clocks use the following event messages:

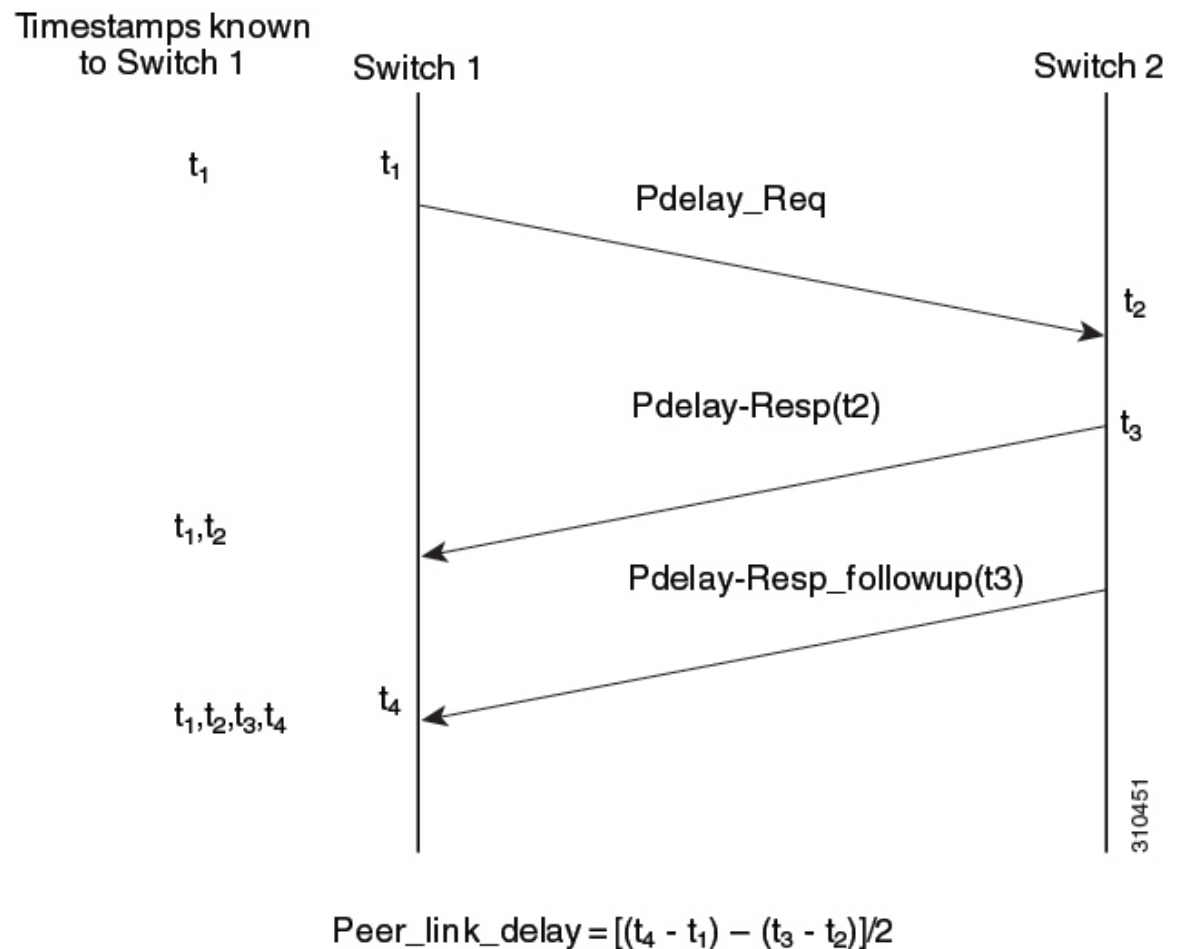
- Pdelay_Req
- Pdelay_Resp
- Pdelay_Resp_Follow_Up

These messages are sent in the following sequence:

1. Port 1 generates timestamp `t1` for a Pdelay_Req message.

2. Port 2 receives and generates timestamp t_2 for this message.
3. Port 2 returns and generates timestamp t_3 for a Pdelay_Resp message.
To minimize errors due to any frequency offset between the two ports, Port 2 returns the Pdelay_Resp message as quickly as possible after the receipt of the Pdelay_Req message.
4. Port 2 returns timestamps t_2 and t_3 in the Pdelay_Resp and Pdelay_Resp_Follow_Up messages respectively.
5. Port 1 generates timestamp t_4 after receiving the Pdelay_Resp message. Port 1 then uses the four timestamps (t_1 , t_2 , t_3 , and t_4) to calculate the mean link delay.

Figure 29: Peer-to-Peer Delay Mechanism



Synchronizing the Local Clock

In an ideal PTP network, the master and slave clock operate at the same frequency. However, drift can occur on the network. Drift is the frequency difference between the master and slave clock. You can compensate for drift by using the timestamp information in the device hardware and follow-up messages (intercepted by the switch) to adjust the frequency of the local clock to match the frequency of the master clock.

Best Master Clock Algorithm

The Best Master Clock Algorithm (BMCA) is the basis of PTP functionality. The BMCA specifies how each clock on the network determines the best master clock in the subdomain of all the clocks that it can see, including itself. The BMCA runs locally on each port in network continuously for every Announce interval and quickly adjusts for changes in network configuration. BMCA based on IEEE 1588-2008 uses Announce messages for advertising clock properties.

The BMCA uses the following criteria to determine the best master clock in the subdomain:

- Clock quality. Example, GPS is considered the highest quality.
- Accuracy of the clock's time base
- Stability of the local oscillator
- Closest clock to the grandmaster

BMCA based on IEEE 1588-2008 uses own data set with the received data set to determine the best clock based on the attributes with the following properties, in the indicated order:

1. Priority1: User-assigned priority to each clock. The range is from 0 to 255. The default value is 128.
2. Class: Class to which a clock belongs, each class has its own priority.
3. Accuracy: Precision between clock and UTC, in nanoseconds
4. Variance: Variability of clock
5. Priority2: Final priority. The range is from 0 to 255. The default value is 128.
6. Unique Identifier: 64-bit Extended Unique Identifier (EUI)

In addition to identifying the best master clock, the BMCA also ensures that clock conflicts do not occur on the PTP network by ensuring that:

- There is no misconfiguration, such as two master clocks or no master clocks, as a result of the master clock identification process.
- Clocks do not have to negotiate with one another



Note Starting Cisco IOS XE Bengaluru 17.6.4 release, the default PTP profile determines its own synchronization tree, and this synchronization tree is different from spanning tree.

PTP Ports

This topic describes about the different PTP port states and their respective functions:

PTP Port State	Description
INITIALIZING	When a port is in the INITIALIZING state, it initiates data sets, hardware and communication facilities. There are no PTP messages placed by the ports of the clock in its communication path. If one of the ports of boundary clock is in INITIALIZING state, then all the ports will be in the INITIALIZING state.

PTP Port State	Description
FAULTY	When a port is in the FAULTY state, there are no PTP messages allowed in the communication path. Ports in a boundary clock is not affected by a FAULTY port with no activity. When the port link status is down, PTP port state moves to FAULTY.
DISABLED	A DISABLED port places no messages on its communication path. This port state affects no activity at any other port of the boundary clock. A DISABLED port state discards all PTP messages received except for peer delay messages.
LISTENING	The LISTENING port state waits for the announceReceiptTimeout to expire or to receive an Announce message from the master clock. This port state essentially allows orderly addition of clocks to a domain. When a port is in LISTENING state, does not accept any PTP messages except for Pdelay_Req, Pdelay_Resp, Pdelay_Resp_Follow_Up messages.
PRE_MASTER	The PRE_MASTER port behaves similar to a MASTER port state. The PRE_MASTER port state does not place any messages on its communication path except for Pdelay_Req, Pdelay_Resp, Pdelay_Resp_Follow_Up, signaling, or management messages.
MASTER	This port state behaves as the MASTER of the time source.
PASSIVE	The PASSIVE port state does not place any messages on its communication path except for Pdelay_Req, Pdelay_Resp, Pdelay_Resp_Follow_Up.
UNCALIBRATED	A port state is UNCALIBATED when there is more than one master port in the domain. This happens when the appropriate master port is selected and the local port is preparing to synchronize with the selected master port. This is a transient state that allows initialization of synchronization servos, updating of data sets when a new master port has been selected, and other implementation-specific activity.
SLAVE	This port state synchronises with the selected MASTER port.

Precision Time Protocol Clocks

A PTP network is made up of PTP-enabled devices. These devices typically consist of the following clock types.

Grandmaster Clock

Within a PTP domain, the grandmaster clock is the primary source of time for clock synchronization using PTP. The grandmaster clock usually has a very precise time source, such as a GPS or atomic clock. When the network does not require any external time reference and only needs to be synchronized internally, the grandmaster clock can have a free run.



Note Use the device as grandmaster clock in the network considering its reduced clock accuracy.

Ordinary Clock

An ordinary clock is a PTP clock with a single PTP port. It functions as a node in a PTP network and can be selected by the BMCA as a master or slave within a subdomain. These are the most common clock type on

a PTP network because they are used as end nodes on a network that is connected to devices requiring synchronization. Ordinary clocks have various interfaces to external devices.

Boundary Clock

A boundary clock in a PTP network operates in place of a standard network switch or router. Boundary clocks have more than one PTP port, and each port provides access to a separate PTP communication path. Boundary clocks provide an interface between PTP domains. They intercept and process all PTP messages, and pass all other network traffic. The BMCA is used to select the best clock seen by any port. The selected port is then set as a slave and the other ports are set as master. The master port synchronizes the clocks that are connected downstream, while the slave port synchronizes with the upstream master clock.

To set a port permanently as primary (master), use the **ptp role primary** command in interface configuration mode. Setting a port permanently as primary (master) ensures that the port remains as a primary (master) even if a clock connected to the port can be elected as a grandmaster clock.



Note The command **ptp role primary** must be used only on ports that are used as end nodes on a network that are connected to devices requiring synchronization.

Use the **show ptp port interface_id** command to verify if the port is set as primary (master).

Transparent Clock

The role of transparent clocks in a PTP network is to update the time-interval field that is part of PTP event message. This update compensates for switch delay and has an accuracy of within one picosecond.

There are two types of transparent clocks:

End-to-end (E2E) transparent clocks measure the PTP event message transit time (also known as resident time) for SYNC and DELAY_REQUEST messages. This measured transit time is added to a data field (correction field) in the corresponding messages:

- The measured transit time of a SYNC message is added to the correction field of the corresponding SYNC or the FOLLOW_UP message.
- The measured transit time of a DELAY_REQUEST message is added to the correction field of the corresponding DELAY_RESPONSE message.

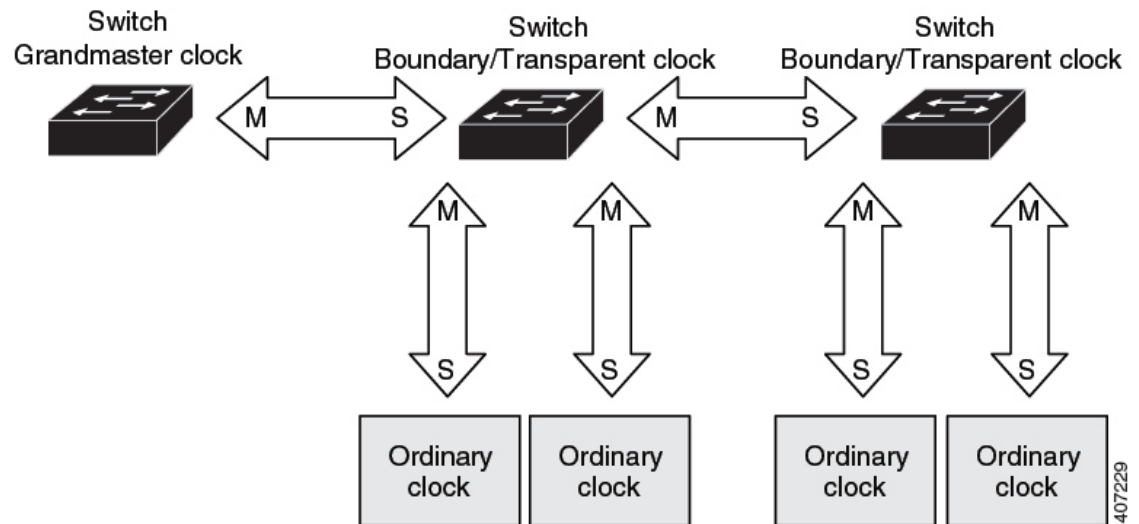
The slave uses this information when determining the offset between the slave's and the master's time. E2E transparent clocks do not provide correction for the propagation delay of the link itself.

Peer-to-peer (P2P) transparent clocks measure PTP event message transit time the same way E2E transparent clocks do. In addition, P2P transparent clocks measure the upstream link delay. The upstream link delay is the estimated packet propagation delay between the upstream neighbor P2P transparent clock and the P2P transparent clock under consideration.

These two times (message transit time and upstream link delay time) are both added to the correction field of the PTP event message, correction field of the message that is received by the slave contains the sum of all link delays. In theory, this is the total end-to-end delay (from master to slave) of the SYNC packet.

The following figure illustrates PTP clocks in a master-slave hierarchy within a PTP network.

Figure 30: PTP Clock Hierarchy



Precision Time Protocol Profiles

The IEEE 1588 definition of a PTP profile is the set of allowed PTP features applicable to a device. A PTP profile is usually specific to a particular type of application or environment and defines the following values:

- Best master clock algorithm options
- Configuration management options
- Path delay mechanisms (peer delay)
- Range and default values of all PTP configurable attributes and data set members
- Closest clock to the grandmaster
- Transport mechanisms that are required, permitted, or prohibited
- Node types that are required, permitted, or prohibited
- Options that are required, permitted, or prohibited

Default Profile

The default PTP profile mode on switch is default profile mode. The PTP mode of transport is Layer 2 and Layer 3.

By default, PTP default profile is disabled globally on these platforms.

Precision Time Protocol on an EtherChannel Interface

An EtherChannel interface allows multiple physical Ethernet links to combine into one logical channel. Configuring EtherChannel interface allows load sharing of traffic among the links in the channel as well as redundancy if one or more links in the EtherChannel fail. This behaviour of an EtherChannel interface does

not change when PTP is configured. The example below illustrates how PTP works when it is configured on an EtherChannel interface.

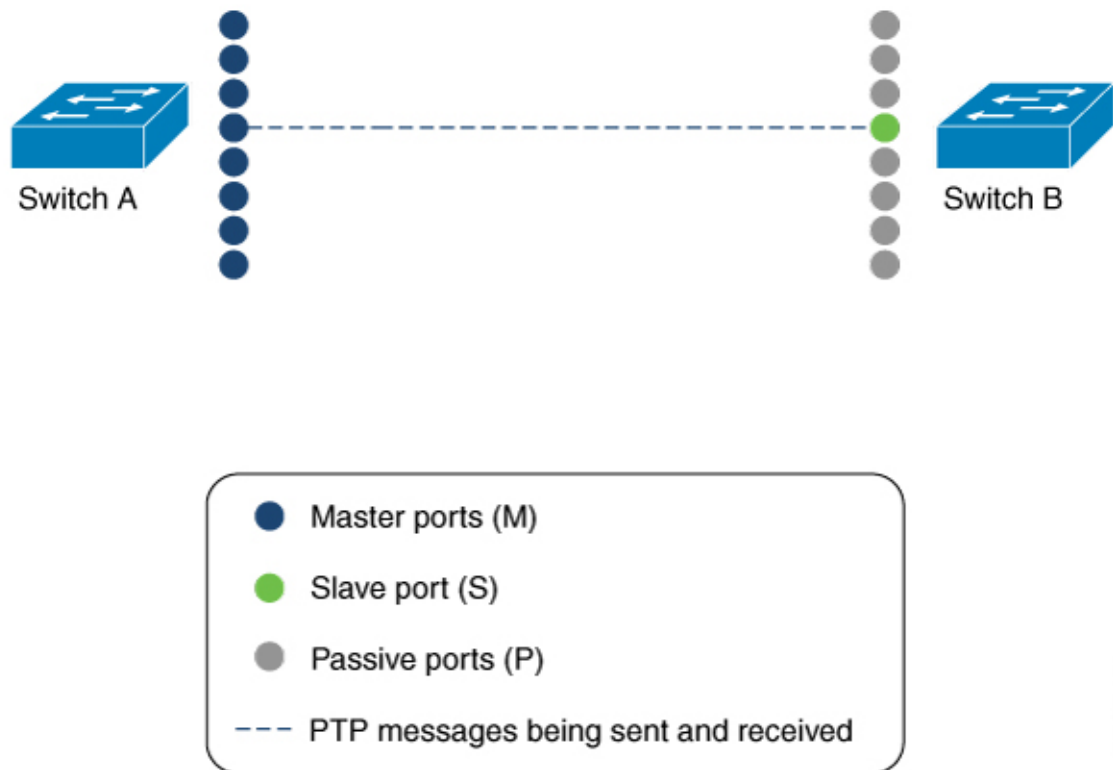


Note PTP configurations can be done only on EtherChannel member interfaces and not on the main EtherChannel or PortChannel interface.

For example, in the figure below there are two switches (Switch A and Switch B) connected through an eight member EtherChannel. If you consider Switch A as the master clock, all the ports part of the EtherChannel are master ports. Similarly, Switch B is the slave clock and one of the ports from the EtherChannel bundle becomes the slave port while all other ports become passive ports. It is always the port with the lowest port number in the Etherchannel bundle that is designated as the slave port. If that slave port is disabled or shut down for any reason, the next port with the lowest port number is designated as the slave port.

The master and slave relationship is established when the feature is configured on an EtherChannel interface as well. The master ports from Switch A send and receives PTP messages. In Switch B only the slave port exchanges PTP messages. There is no exchange of PTP messages in the passive ports.

Figure 31: Precision Time Protocol on an EtherChannel Interface



356549

AES67 Media Profile

The AES67 PTP media profile is based on the AES67 standard that is used for high-performance streaming and audio-over-IP interoperability in audio devices. AES67 uses IEEE 1588-2008 PTP for clock synchronization

of devices and prioritization of control and data traffic. IEEE 1588-2008 also defines various parameters for audio traffic and device discovery protocol that are not required to be implemented in switches or bridges.

The following table defines the various parameters for PTP to be compliant with AES67.

Table 15: PTP Parameters for AES67 Compliance

Attribute	1588 Profile Range (Default Value)	AES67 Profile Range (Default Value)
PTP Domain	0 to 255 (0)	0 to 255 (0)
Priority1/Priority2	0 to 255 (128)	0 to 255 (128)
Announce Interval	0 to 4 (1)	0 to 4 (1)
Announce Timeout	2 to 10 (3)	2 to 10 (3)
Sync Interval	-3 to +1 (0)	-4 to 1 (-3)
Delay Request Minimum Interval	0 to 5 (0)	-3 to 5 (0)

AES67 uses three traffic classes with different traffic types. PTP packets sent by a device must be marked with DSCP EF (46) for expedited forwarding through IP networks. For media and best-effort classes, traffic streams are marked by end devices with DSCP values, as mentioned in the following table. You can also configure QoS policies on the interface for data packets, and re-mark using QoS ingress policies.

Table 16: Traffic Classification

Class Name	Traffic Type	Default DiffServ Class (DSCP Decimal Value)
Clock	IEEE 1588-2008 Announce, Sync, Follow_Up, Delay_Req, Delay_Resp, Pdelay_Req, Pdelay_Resp and Pdelay_Resp_Follow_Up packets	EF (46)
Media	Real-time Transport Protocol (RTP) and Real-Time Transport Control Protocol (RTCP) media stream data	AF41 (34)
Best Effort	IEEE 1588-2008 signalling and management messages. Discovery and connection management messages.	DF (0)

IP IGMPv2 and IGMPv3 snooping enables transmission of multicast audio traffic on specified interfaces. For AES67 deployments using Dante PTPv1 messages, the device enables IGMP snooping by default on the VLAN, and interested endpoints can express the need for PTPv1 groups. This enables multicast snooping entries to be created, which in turn allows forwarding of PTPv1 messages.

**Note**

- SMPTE PTP media profile is not supported. However, because SMPTE uses IEEE 1588 PTP profile, if SMPTE can operate PTP with AES67 PTP definitions, SMPTE can be added to the network.
- The audio-over-IP interoperability in audio devices is supported only if both devices use AES67 media profile.

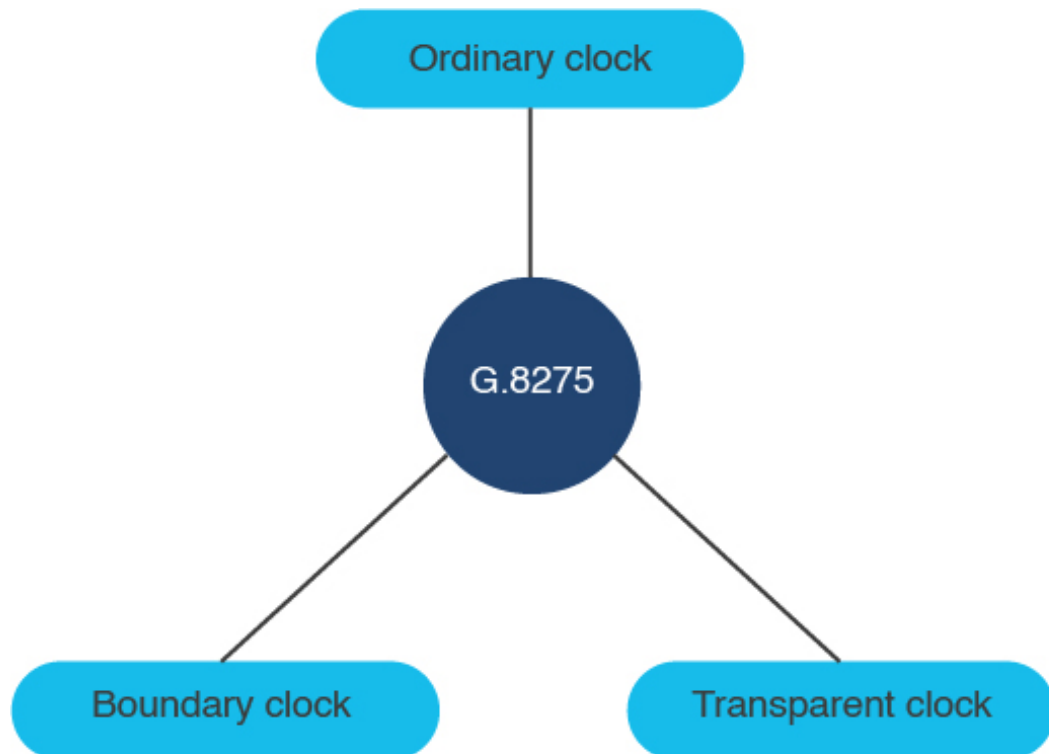
G8275.1 Telecom Profile

The G8275.1 telecom profile is based on the ITU-T standard that is used to ensure network interoperability for accurate delivery of phase and time synchronization. The G8275.1 specifies a profile for telecommunication applications based on IEEE 1588 PTP.

The G8275.1 profile can receive messages from one-step clock and two-step clocks without any specific configuration required. The clock is not required to support one-step or two-step mode to transmit messages. The ingress messages from one-step clock and egress messages from two-step clock is supported after processing

PTP Clocks Supported

The ordinary clock, boundary clock and transparent clock are used in this profile.



357794



Note This profile supports only end-to-end transparent clock. You cannot use peer-to-peer transparent clock in this profile.

PTP Messages Supported

The G8275.1 profile uses Sync, Follow_Up, Announce, Delay_Req, and Delay_Resp messages only.

Figure 32: PTP Messages supported on G8275.1 profile



How to Configure Precision Time Protocol

The following sections provide information about the various tasks that comprise the configuration of PTP.

Configuring Precision Time Protocol Default Profile

To configure Layer 2 PTP globally, follow these steps:

Procedure

	Command or Action	Purpose
Step 1	enable Example: Device> enable	Enables privileged EXEC mode. Enter your password if prompted.
Step 2	configure terminal Example: Device configure terminal	Enters global configuration mode.
Step 3	ptp mode {boundary {delay-req pdelay-req } e2transparent p2ptransparent} Example: Device(config)# ptp mode boundary delay-req Device(config)# ptp mode boundary pdelay-req Device(config)# ptp mode e2transparent Device(config)# ptp mode p2ptransparent	Specifies the synchronization clock mode: • boundary to enable the switch to participate in selecting the best master clock. If no better clocks are detected, the switch becomes the grandmaster clock on the network and the parent clock to all connected devices. If the best master is determined to be a clock that is connected to the switch, the switch synchronizes to that clock as a child to the clock, then acts as a parent clock to devices connected to other ports. After initial synchronization, the switch and the connected devices exchange timing messages to correct time

	Command or Action	Purpose
		<p>skew that is caused by clock offsets and network delays. Use this mode when overload or heavy load conditions produce significant delay jitter</p> <ul style="list-style-type: none"> • e2transparent for the switch to synchronize all switch ports with the grand master clock that is connected to the switch,. This is the default clock mode. The switch corrects for the delay that is incurred by every packet passing through it (referred to residence time). This mode causes less jitter and error accumulation than boundary mode. • p2transparent where the switch does not synchronize its clock with the master clock. A switch in this mode does not participate in master clock selection and uses the default PTP clock mode on all ports. <p>Note Once PTP default profile is enabled globally, PTP is enabled on all the interfaces. To disable PTP selectively on individual interfaces, use no ptp enable command in interface configuration mode.</p>
Step 4	<p>[no]ptp domain <i>value</i></p> <p>Example:</p> <pre>Device(config)# ptp domain 8</pre>	<p>Configures the domain value on PTP.</p> <ul style="list-style-type: none"> • A single domain value can be set. The range is from 4 to 127. The default value is 0. no ptp domain command will set the value to default.

Configuring Precision Time Protocol on Layer 2 interface

Procedure

	Command or Action	Purpose
Step 1	<p>enable</p> <p>Example:</p> <pre>Device> enable</pre>	<p>Enables privileged EXEC mode.</p> <p>Enter your password if prompted.</p>

	Command or Action	Purpose
Step 2	configure terminal Example: Device <code>configure terminal</code>	Enters global configuration mode.
Step 3	interface <i>interface-id</i> Example: Device(config)# <code>interface TenGigabitEthernet1/0/1</code>	Specifies the physical interface to be configured, and enters interface configuration mode. The interface that you specify can be part of an EtherChannel.
Step 4	[no]ptp enable	
Step 5	ptp vlan <i>vlan-id</i> Example: Device(config-if)# <code>ptp vlan 5</code>	Sets the PTP VLAN on a trunk port. The default is the native VLAN of the trunk port. In boundary mode, only PTP packets in PTP VLAN will be processed, PTP packets from other VLANs will be dropped. Before configuring the PTP VLAN on an interface, the PTP VLAN must be created and allowed on the trunk port.
Step 6	end Example: Device(config-if)# <code>end</code>	Returns to privileged EXEC mode.

Configuring Precision Time Protocol on EtherChannel Member Interface

To configure PTP on EtherChannel member interface, follow these steps:

Procedure

	Command or Action	Purpose
Step 1	enable Example: Device> <code>enable</code>	Enables privileged EXEC mode. Enter your password, if prompted.
Step 2	configure terminal Example: Device# <code>configure terminal</code>	Enters global configuration mode.
Step 3	interface port-channel <i>number</i> Example: Device(config-if)# <code>interface port-channel 1</code>	Specifies the port channel interface to be configured, and enters interface configuration mode.

	Command or Action	Purpose
Step 4	switchport mode trunk Example: Device(config-if)# switchport mode trunk	Configures the port channel interface as a trunk port.
Step 5	no shutdown Example: Device(config-if)# no shutdown	Enables the port channel interface.
Step 6	exit Example: Device(config-if)# exit	Exits interface configuration mode and returns to global configuration mode.
Step 7	interface interface-id Example: Device(config)# interface TenGigabitEthernet1/0/42	Specifies the physical interface to be configured, and enters interface configuration mode.
Step 8	switchport mode trunk Example: Device(config-if)# switchport mode trunk	Configures the physical interface as a trunk port.
Step 9	channel-group channel-number mode on Example: Device(config-if)# channel-group 1 mode on	Configures the port in a channel group and sets the mode. The channel-number range is from 1 to 4096.
Step 10	ptp enable Example: Device(config-if)# ptp enable	Enables PTP on the member interface.
Step 11	ptp sync interval value Example: Device(config-if)# ptp sync interval -3	Configures the logarithmic mean interval to send synchronization messages. The range is -3 to 1. The default is 0 (1 second).
Step 12	end Example: Device(config-if)# end	Returns to privileged EXEC mode.

Configuring Precision Time Protocol on SVI or Layer 3 Interface

Procedure

	Command or Action	Purpose
Step 1	enable Example: Device> enable	Enables privileged EXEC mode. Enter your password if prompted.
Step 2	configure terminal Example: Device# configure terminal	Enters global configuration mode.
Step 3	ptp transport ipv4 udp Example: Device(config)# ptp transport ipv4 udp	Configures IPv4 as the PTP transport mode. Note Only IPv4 is supported as the PTP transport method for Layer 3 PTP.

Configuring the Source IP on Precision Time Protocol

To configure the source IP on PTP, perform this procedure:

Procedure

	Command or Action	Purpose
Step 1	enable Example: Device> enable	Enables privileged EXEC mode. Enter your password if prompted.
Step 2	configure terminal Example: Device# configure terminal	Enters global configuration mode.
Step 3	[no]ptp source {source-address loopback vlan} Example: Device(config)# ptp source source address Device(config)# ptp source loopback Device(config)# ptp source vlan	Specifies the synchronization clock mode: <ul style="list-style-type: none"> • source address once configured, PTP messages in all the interfaces will carry this source ip. • loopback PTP messages in all the interfaces will carry the IP that is configured on the loopback interface. • vlan PTP messages will carry the IP configured on the SVI interface corresponding to the port.

	Command or Action	Purpose
		Note You can use no ptp source command as default.
Step 4	end Example: Device(config-if)# end	Returns to privileged EXEC mode.

Configuring Precision Time Protocol Timers

To configure the PTP timer values from default to required values, follow these steps:

Before you begin

Timer inputs are measured in units of log mean message interval value. To determine the *value* in seconds for the **interval** keyword, use a logarithmic scale. The following table shows examples of the *value* keyword that is converted to seconds with a logarithmic scale:

Value Entered	Logarithmic Calculation	Value in Seconds
-1	2^{-1}	1/2
0	2^0	1

0 indicates 1 packet per second and -1 indicates 1 packet per 2 seconds.

Procedure

	Command or Action	Purpose
Step 1	enable Example: Device> enable	Enables privileged EXEC mode. Enter your password if prompted.
Step 2	configure terminal Example: Device# configure terminal	Enters global configuration mode.
Step 3	interface <i>interface-id</i> Example: Device(config)# interface gigabitethernet2/0/1	Specifies the physical port to be configured, and enters interface configuration mode.
Step 4	ptp announce { interval <i>value</i> timeout <i>count</i> } Example: Device(config-if)# ptp announce interval 1	(Optional) Configures the interval between PTP announce messages on an interface or the number of PTP intervals before a timeout occurs on an interface.

	Command or Action	Purpose
		<ul style="list-style-type: none"> • interval value sets the logarithmic mean interval to send announce messages. The range is 0 to 4. The default is 0 (1 second). • timeout count sets the logarithmic mean interval in seconds to announce timeout messages. The range is 2 to 10. The default is 3 (8 seconds).
Step 5	<p>ptp sync {interval value limit offset-value}</p> <p>Example:</p> <pre>Device(config-if)# ptp sync interval 1</pre>	<p>(Optional) Configures the interval between PTP synchronization messages on an interface.</p> <ul style="list-style-type: none"> • interval value sets the logarithmic mean interval to send synchronization messages. The range is -3 to 1. The default is 0 (1 second). • limit offset-value sets the maximum clock offset value before PTP attempts to resynchronize. The range is from 50 to 500000000 nanoseconds. The default is 500000000 nanoseconds.
Step 6	<p>ptp delay-req interval value</p> <p>Example:</p> <pre>Device(config-if)# ptp delay-req interval 1</pre>	<p>(Optional) Configures the logarithmic mean interval allowed between PTP delay request messages when the port is in the master state. The range is 0 to 5. The default is 0 (1 second).</p>
Step 7	<p>ptp pdelay-req interval value</p> <p>Example:</p> <pre>Device(config-if)# ptp pdelay-req interval 1</pre>	<p>(Optional) Configures the logarithmic mean interval allowed between pdelay request messages when the port is in the master state. The range is 0 to 5. The default is 0 (1 second).</p>
Step 8	<p>end</p> <p>Example:</p> <pre>Device(config-if)# end</pre>	<p>Returns to privileged EXEC mode.</p>

Configuring the Values of Precision Time Protocol Clocks

Follow these steps to configure the values of PTP clock priority1 and priority2:

Procedure

	Command or Action	Purpose
Step 1	<p>enable</p> <p>Example:</p> <pre>Device> enable</pre>	<p>Enables privileged EXEC mode.</p> <p>Enter your password, if prompted.</p>

	Command or Action	Purpose
Step 2	configure terminal Example: Device# <code>configure terminal</code>	Enters global configuration mode.
Step 3	ptp priority1 value Example: Device(config)# <code>ptp priority1 120</code>	Sets the value of PTP clock priority1. The range is from 0 to 255. The default value is 128. Note If the value of priority1 is configured as 255, the clock cannot be considered as Grandmaster.
Step 4	ptp priority2 value Example: Device(config)# <code>ptp priority2 120</code>	Sets the value of PTP clock priority2. The range is from 0 to 255. The default value is 128.
Step 5	exit Example: Device(config)# <code>exit</code>	Returns to global configuration mode.

Configuring Precision Time Protocol Using AES67 Media Profile

Procedure

	Command or Action	Purpose
Step 1	enable Example: Device> <code>enable</code>	Enables privileged EXEC mode. Enter your password, if prompted.
Step 2	configure terminal Example: Device <code>configure terminal</code>	Enters global configuration mode.
Step 3	ptp transport-protocol ipv4 udp Example: Device(config)# <code>ptp transport-protocol ipv4 udp</code>	Enables the PTP Layer 3 mode.
Step 4	ptp mode boundary delay-req Example: Device(config)# <code>ptp mode boundary delay-req</code>	Configures the device for boundary clock mode using the peer delay request mechanism.

	Command or Action	Purpose
Step 5	interface range <i>interface-range</i> Example: Device(config)# interface range GigabitEthernet1/0/1-GigabitEthernet1/0/2	Specifies the range of interfaces to be configured, and enters interface-range configuration mode.
Step 6	ptp sync interval <i>value</i> Example: Device(config-if-range)# ptp sync interval -3	Configures the interval between PTP synchronization messages on the interface range. The range is from -4 to 1.
Step 7	ptp delay-req interval <i>value</i> Example: Device(config-if-range)# ptp delay-req interval -3	Configures the logarithmic mean interval that is allowed between PTP delay request messages when the port is in the primary state. The range is from -3 to 5.
Step 8	exit Example: Device(config-if-range)# exit	Exits interface-range configuration mode and enters global configuration mode.
Step 9	ptp ip dscp <i>value</i> message general Example: Device(config)# ptp ip dscp 46 message general	Configures IP DSCP value for PTP general messages. The range is from 0 to 63.
Step 10	ptp ip dscp <i>value</i> message event Example: Device(config)# ptp ip dscp 46 message event	Configures IP DSCP value for PTP event messages. The range is from 0 to 63.
Step 11	end Example: Device(config)# end	Returns to privileged EXEC mode.

Configuring Precision Time Protocol Using G8275.1 Telecom Profile

To configure PTP using G8275.1 Telecom profile, perform this procedure:

Procedure

	Command or Action	Purpose
Step 1	enable Example: Device> enable	Enables privileged EXEC mode. Enter your password, if prompted.

	Command or Action	Purpose
Step 2	configure terminal Example: Device <code>configure terminal</code>	Enters global configuration mode.
Step 3	ptp profile 8275.1 clock-mode {boundary transparent } Example: Device(config)# <code>ptp profile 8275.1</code> <code>clock-mode boundary</code>	Configures the clock mode for the device. <ul style="list-style-type: none"> • boundary mode sets the device to boundary clock mode. • transparent mode sets the device to transparent clock mode.
Step 4	interface range interface-range Example: Device(config)# <code>interface range</code> <code>GigabitEthernet1/0/1-GigabitEthernet1/0/2</code>	Specifies the range of interfaces to be configured, and enters interface-range configuration mode.
Step 5	exit Example: Device(config-if-range)# <code>exit</code>	Exits interface-range configuration mode and enters global configuration mode.
Step 6	ptp ip dscp value message general Example: Device(config)# <code>ptp ip dscp 46 message</code> <code>general</code>	Configures IP DSCP value for PTP general messages. The range is from 0 to 63.
Step 7	ptp ip dscp value message event Example: Device(config)# <code>ptp ip dscp 46 message</code> <code>event</code>	Configures IP DSCP value for PTP event messages. The range is from 0 to 63.
Step 8	end Example: Device(config)# <code>end</code>	Returns to privileged EXEC mode.

Configuration Examples for Precision Time Protocol

The following sections provide PTP configuration examples.

Example: Configuring Precision Time Protocol Using AES67 Media Profile

The following is a sample PTP configuration to set up a device using AES67 media profile:

```
Device# configure terminal
Device(config)# ptp transport-protocol ipv4 udp
```

```

Device(config)# ptp mode boundary delay-req
Device(config)# interface range gigabitethernet1/0/1-gigabitethernet1/0/2
Device(config-if-range)# ptp sync interval -3
Device(config-if-range)# ptp delay-req interval -3
Device(config-if-range)# exit
Device(config)# ptp ip dscp 46 message general
Device(config)# ptp ip dscp 46 message event
Device(config)# end

```

The following is a sample QoS configuration to set up a device such that the expedited forwarding messages face minimal latency and media traffic faces no drops:

```

Device# configure terminal
Device(config)# class-map match-all PTP
Device(config-cmap)# match dscp 46
Device(config-cmap)# class-map match-any VOICE
Device(config-cmap)# match dscp 34
Device(config-cmap)# exit
Device(config)# policy-map GENERAL-QOS
Device(config-pmap)# class PTP
Device(config-pmap-c)# priority level 1 percent 10
Device(config-pmap-c)# class VOICE
Device(config-pmap-c)# bandwidth remaining percent 30
Device(config-pmap-c)# class class-default
Device(config-pmap-c)# bandwidth remaining percent 60
Device(config-pmap-c)# end

```

Verifying Precision Time Protocol Configurations

Verifying Layer 2 and Layer 3 PTP Configurations

show ptp port *interface-name*

To verify PTP port state, use **show ptp port** *interface-name* command.

To verify the PTP port states on all interfaces use **show ptp brief** command.

The following is a sample output for boundary mode configuration with delay request mechanism:

```

Device# show ptp port GigabitEthernet1/0/45
PTP PORT DATASET: GigabitEthernet1/0/45
  Port identity: clock identity: 0xCC:46:D6:FF:FE:C5:24:0
  Port identity: port number: 45
  PTP version: 2
  Port state: SLAVE
  Delay request interval(log mean): 0
  Announce receipt time out: 3
  Announce interval(log mean): 1
  Sync interval(log mean): 0
  Delay Mechanism: End to End
  Peer delay request interval(log mean): 0
  Sync fault limit: 500000000

```

The following is a sample output for boundary mode configuration with pdelay request mechanism:

```

Device# show ptp port GigabitEthernet1/0/45
PTP PORT DATASET: GigabitEthernet1/0/45

```

```

Port identity: clock identity: 0xCC:46:D6:FF:FE:C5:24:0
Port identity: port number: 45
PTP version: 2
Port state: MASTER
Delay request interval(log mean): 0
Announce receipt time out: 3
Announce interval(log mean): 1
Sync interval(log mean): 0
Delay Mechanism: Peer to Peer
Peer delay request interval(log mean): 0
Sync fault limit: 500000000

```

show ptp brief

To verify the PTP port states on all interfaces use **show ptp brief** command.

The following is a sample output for **show ptp brief** command:

```

Device# show ptp brief
Interface                               Domain    PTP State
TenGigabitEthernet1/0/1                 0        MASTER
TenGigabitEthernet1/0/2                 0        SLAVE
TenGigabitEthernet1/0/3                 0        FAULTY

```

show ptp clock

To verify the PTP clock identity details and to verify the configured values of Priority1 and Priority2, use **show ptp clock** command.

The following is a sample output for **show ptp clock** command:

```

Device# show ptp clock
PTP CLOCK INFO
  PTP Device Type: Boundary clock
  PTP Device Profile: Default Profile
  Clock Identity: 0xCC:46:D6:FF:FE:C5:24:0 <<clock identity of this
switch>>
  Clock Domain: 0
  Number of PTP ports: 52
  Priority1: 128
  Priority2: 128
  Clock Quality:
    Class: 248
    Accuracy: Unknown
    Offset (log variance): 16640
  Offset From Master(ns): 0
  Mean Path Delay(ns): 0
  Steps Removed: 1

```

show ptp parent

To identify which Grandmaster Clock identity the device is synced to in boundary mode, use **show ptp parent** command.



Note `show ptp parent` will not display any output if the device is configured in transparent clock mode.

The following is a sample output for `show ptp parent` command:

```
Device# show ptp parent
PTP PARENT PROPERTIES
  Parent Clock:
    Parent Clock Identity: 0x0:11:1:FF:FE:0:0:1
    Parent Port Number: 1
    Observed Parent Offset (log variance): 16640
    Observed Parent Clock Phase Change Rate: N/A

  Grandmaster Clock:
    Grandmaster Clock Identity: 0x0:11:1:FF:FE:0:0:1    <<Grandmaster clock
identity to which the device is synced to>>
    Grandmaster Clock Quality:
      Class: 6
      Accuracy: Within 25ns
      Offset (log variance): 0
      Priority1: 128
      Priority2: 128
```

`show platform software fed active ptp domain 0`

To verify the local servo PTP clock synchronization to Grandmaster clock on a device configured in boundary mode with delay-request mechanism, use `show platform software fed active ptp domain 0` command.

```
Device#
show platform software fed active ptp domain 0
```

```
Displaying data for domain number 0
=====
```

```
Profile Type : DEFAULT
Profile State: enabled
Clock Mode : BOUNDARY CLOCK
Delay mechanism: End-to-End
PTP clock : 2017-6-28 5:58:59
Transport Method: L2 Ethernet
```

By default, local servo PTP clock will be displaying EPOCH time(1970-1-1) when the device is not synced to any PTP Grandmaster Clock.

`show ptp port interface-name`

To verify PTP port state, use `show ptp port interface-name` command.

To verify the PTP port states on all interfaces use `show ptp brief` command.

The following is a sample output for boundary mode configuration with delay request mechanism:

```
Device# show ptp port FortyGigabitEthernet1/0/10
PTP PORT DATASET: FortyGigabitEthernet1/0/10
  Port identity: clock identity: 0x0:A3:D1:FF:FE:5A:12:0
  Port identity: port number: 10
```

```

PTP version: 2
Port state: SLAVE
Delay request interval(log mean): 0
Announce receipt time out: 3
Announce interval(log mean): 1
Sync interval(log mean): 0
Delay Mechanism: End to End
    << PTP mode delay >>
Peer delay request interval(log mean): 0
Sync fault limit: 500000000

```

show ptp parent

To identify which Grandmaster Clock identity the device is synced to in boundary mode, use **show ptp parent** command.



Note **show ptp parent** will not display any output if the device is configured in transparent clock mode.

The following is a sample output for **show ptp parent** command:

```

Device# show ptp parent
PTP PARENT PROPERTIES
  Parent Clock:
    Parent Clock Identity: 0x38:E:4D:FF:FE:81:FE:29
<< Immediate next Master >>
  Parent Port Number: 196
  Observed Parent Offset (log variance): 17258
  Observed Parent Clock Phase Change Rate: N/A

  Grandmaster Clock:
    Grandmaster Clock Identity: 0x0:0:0:5:0:0:0:1
<< GM: External Clock Source acting Grand Master >>
  Grandmaster Clock Quality:
    Class: 6
    Accuracy: Within 1us
    Offset (log variance): 0
    Priority1: 128
    Priority2: 128

```

show platform software fed active ptp domain 0

To verify the local servo PTP clock synchronization to Grandmaster clock on a device configured in boundary mode with delay-request mechanism, use **show platform software fed active ptp domain 0** command.

```

Device#
show platform software fed active ptp domain 0
Displaying data for domain number 0
=====

Profile Type : DEFAULT
Profile State: enabled
Clock Mode : BOUNDARY CLOCK
Delay Mechanism: : END-TO-END

```

```
PTP clock : 2017-12-15 15:27:27
mean_path_delay 214 nanoseconds
Transport Method : udp-ipv4
>> << PTP Transport Method
```

Table 17: Debug Commands

Command	Purpose
debug ptp messages	Enables debugging of PTP messages.
debug ptp error	Enables debugging of PTP errors.
debug ptp bmc	Enables debugging of the PTP Best Master Clock Algorithm.
debug ptp event	Enables debugging of PTP state event.

Verifying PTP Configurations on an EtherChannel Interface

Master Clock

The following command verifies the PTP state on an interface:

```
Device# show ptp port tengigabitethernet 1/0/39
PTP PORT DATASET: TenGigE1/0/39
  Port identity: clock identity: 0x0:A7:42:FF:FE:8A:84:C0
  Port identity: port number: 39
  PTP version: 2
  Port state: MASTER
  Delay request interval(log mean): 0
  Announce receipt time out: 3
  Announce interval(log mean): 0
  Sync interval(log mean): 0
  Delay Mechanism: End to End
  Peer delay request interval(log mean): 0
  Sync fault limit: 500000000
```

```
Device# show ptp port tengigabitethernet 1/0/44
PTP PORT DATASET: TenGigE1/0/44
  Port identity: clock identity: 0x0:A7:42:FF:FE:8A:84:C0
  Port identity: port number: 44
  PTP version: 2
  Port state: MASTER
  Delay request interval(log mean): 0
  Announce receipt time out: 3
  Announce interval(log mean): 0
  Sync interval(log mean): 0
  Delay Mechanism: End to End
  Peer delay request interval(log mean): 0
  Sync fault limit: 500000000
```

```
Device# show ptp port tengigabitethernet 1/0/48
PTP PORT DATASET: TenGigE1/0/48
  Port identity: clock identity: 0x0:A7:42:FF:FE:8A:84:C0
  Port identity: port number: 48
  PTP version: 2
  Port state: MASTER
  Delay request interval(log mean): 0
  Announce receipt time out: 3
```

```

Announce interval(log mean): 0
Sync interval(log mean): 0
Delay Mechanism: End to End
Peer delay request interval(log mean): 0
Sync fault limit: 500000000

```

Slave Clock

The following command can be used to verify the PTP state on the interfaces:

```

Device# show ptp brief | exclude FAULTY
Interface          Domain    PTP State
tenGigE1/0/12     0        SLAVE
TenGigE1/0/20     0        PASSIVE
TenGigE1/0/23     0        PASSIVE

```

The following command verifies if interface configured on each port is an EtherChannel interface:

```

Device# show etherchannel 1 summary
Flags: D - down          P - bundled in port-channel
       I - stand-alone  s - suspended
       H - Hot-standby (LACP only)
       R - Layer3       S - Layer2
       U - in use       f - failed to allocate aggregator

       M - not in use, minimum links not met
       u - unsuitable for bundling
       w - waiting to be aggregated
       d - default port
       A - formed by Auto LAG

```

```

Number of channel-groups in use: 1
Number of aggregators:          1

```

```

Group  Port-channel  Protocol  Ports
-----+-----+-----+-----
1      Pol(SU)      LACP      Hu1/0/12(P)  Hu1/0/20(P)
                               Hu1/0/23(P)

```

The following command verifies port state of each interface:

```

Device# show ptp port tengigabitethernet 1/0/12
PTP PORT DATASET: TenGigE1/0/12
Port identity: clock identity: 0x0:A7:42:FF:FE:9B:DA:E0
Port identity: port number: 12
PTP version: 2
PTP port number: 12
PTP slot number: 0
Port state: SLAVE
Delay request interval(log mean): 0
Announce receipt time out: 3
Announce interval(log mean): 0
Sync interval(log mean): 0
Delay Mechanism: End to End
Peer delay request interval(log mean): 0
Sync fault limit: 500000000

```

```

Device# show ptp port tengigabitethernet 1/0/20
PTP PORT DATASET: TenGigE1/0/20
Port identity: clock identity: 0x0:A7:42:FF:FE:9B:DA:E0
Port identity: port number: 20
PTP version: 2
PTP port number: 20
PTP slot number: 0
Port state: PASSIVE

```



```

Delay request interval(log mean): 0
Announce receipt time out: 3
Announce interval(log mean): 0
Sync interval(log mean): 0
Delay Mechanism: End to End
Peer delay request interval(log mean): 0
Sync fault limit: 500000000

```

```

Device# show ptp port tengigabitethernet 1/0/23
PTP PORT DATASET: TenGigE1/0/23
Port identity: clock identity: 0x0:A7:42:FF:FE:9B:DA:E0
Port identity: port number: 23
PTP version: 2
PTP port number: 23
PTP slot number: 0
Port state: PASSIVE
Delay request interval(log mean): 0
Announce receipt time out: 3
Announce interval(log mean): 0
Sync interval(log mean): 0
Delay Mechanism: End to End
Peer delay request interval(log mean): 0
Sync fault limit: 500000000

```

Feature History for Precision Time Protocol

This table provides release and related information for features explained in this module.

These features are available on all releases subsequent to the one they were introduced in, unless noted otherwise.

Release	Feature	Feature Information
Cisco IOS XE Cupertino 17.7.1	Audio Engineering Society: AES67 Timing Profile	AES67 timing profile support for high-performance streaming and audio-over-IP interoperability in audio devices was introduced.
Cisco IOS XE Cupertino 17.7.1	IEEE 1588v2, Precision Time Protocol (PTP) support	PTP was developed to synchronize the clocks in packet-based networks that include distributed device clocks of varying precision and stability.
Cisco IOS XE Cupertino 17.8.1	G8275.1 Telecom Profile	G8275.1 telecom profile support for interoperability for accurate delivery of phase and time synchronization. The G8275.1 specifies a profile for telecommunication applications based on IEEE 1588 PTP.
Cisco IOS XE Cupertino 17.9.1	Stateful Switchover	Stateful Switchover support is introduced. The PTP protocol does not restart.
Cisco IOS XE Dublin 17.10.1	PTPv2 on Cisco StackWise Virtual	Cisco StackWise Virtual is a network system virtualization technology that pairs two switches into one virtual switch to simplify operational efficiency with a single control and management plane. PTPv2 is supported on Cisco StackWise Virtual.

Use Cisco Feature Navigator to find information about platform and software image support. To access Cisco Feature Navigator, go to <http://www.cisco.com/go/cfn>.



CHAPTER 7

Configuring UniDirectional Link Detection

- [Restrictions for Configuring UniDirectional Link Detection, on page 159](#)
- [Information About UniDirectional Link Detection, on page 159](#)
- [How to Configure UniDirectional Link Detection, on page 162](#)
- [Monitoring and Maintaining UniDirectional Link Detection, on page 166](#)
- [Console Error Messages For Fast UniDirectional Link Detection, on page 166](#)
- [Additional References for UniDirectional Link Detection, on page 167](#)
- [Feature History for UniDirectional Link Detection, on page 167](#)

Restrictions for Configuring UniDirectional Link Detection

The following are restrictions for configuring UniDirectional Link Detection (UDLD):

- A UDLD-capable port can't detect a unidirectional link if it's connected to a UDLD-incapable port of another device.
- When configuring the mode (normal or aggressive), make sure that the same mode is configured on both sides of the link.
- In the initial phase, the number of ports on which Fast UDLD can be enabled is limited to 32. If this number is reached, then Fast UDLD isn't enabled on additional ports and an error message is printed on the console:

```
UDLD: hundredGigE <> not enabled for fast hello, maximum number of fast hello ports (4) reached
```
- If you disable UDLD when Fast UDLD is configured, the entire UDLD configuration is removed.



Caution

Loop guard works only on point-to-point links. We recommend that each end of the link has a directly connected device that is running STP.

Information About UniDirectional Link Detection

UniDirectional Link Detection (UDLD) is a Layer 2 protocol that enables devices that are connected through fiber-optic or twisted-pair Ethernet cables to monitor the physical configuration of the cables and detect when

a unidirectional link exists. All connected devices must support UDLD for the protocol to successfully identify and disable unidirectional links. When UDLD detects a unidirectional link, it disables the affected port and alerts you. Unidirectional links can cause a variety of problems, including spanning-tree topology loops.

Fast UniDirectional Link Detection

Fast UDLD supports timers in the few-hundred milliseconds range, which enables subsecond unidirectional link detection. With Fast UDLD, the time to detect a unidirectional link can vary from less than one second to a few seconds (the detection time also depends on how the timers are configured). Link status messages are exchanged every 200ms.

A transition from slow mode to fast mode occurs on the port when both sides of a link have Fast UDLD configured and have negotiated successfully to move into fast mode. A transition from fast mode to slow mode occurs when one of the Fast UDLD configured ports has its port-level Fast UDLD configuration removed.

Modes of Operation

UDLD and Fast UDLD support two modes of operation: normal (the default) and aggressive. In normal mode, UDLD can detect unidirectional links due to misconnected ports on fiber-optic connections. In aggressive mode, UDLD can also detect unidirectional links due to one-way traffic on fiber-optic and twisted-pair links and to misconnected ports on fiber-optic links.

In normal and aggressive modes, UDLD works with the Layer 1 mechanisms to learn the physical status of a link. At Layer 1, autonegotiation takes care of physical signaling and fault detection. UDLD performs tasks that autonegotiation cannot perform, such as detecting the identities of neighbors and shutting down misconnected ports. When you enable both autonegotiation and UDLD, the Layer 1 and Layer 2 detections work together to prevent physical and logical unidirectional connections and the malfunctioning of other protocols.

A unidirectional link occurs whenever traffic sent by a local device is received by its neighbor but traffic from the neighbor is not received by the local device.

Normal Mode

In normal mode, UDLD detects a unidirectional link when fiber strands in a fiber-optic port are misconnected and the Layer 1 mechanisms do not detect this misconnection. If the ports are connected correctly but the traffic is one way, UDLD does not detect the unidirectional link because the Layer 1 mechanism, which is supposed to detect this condition, does not do so. In this case, the logical link is considered undetermined, and UDLD does not disable the port.

When UDLD is in normal mode, if one of the fiber strands in a pair is disconnected, as long as autonegotiation is active, the link does not stay up because the Layer 1 mechanisms detects a physical problem with the link. In this case, UDLD does not take any action and the logical link is considered undetermined.

Aggressive Mode

In aggressive mode, UDLD detects a unidirectional link by using the previous detection methods. UDLD in aggressive mode can also detect a unidirectional link on a point-to-point link on which no failure between the two devices is allowed. It can also detect a unidirectional link when one of these problems exists:

- On fiber-optic or twisted-pair links, one of the ports cannot send or receive traffic.
- On fiber-optic or twisted-pair links, one of the ports is down while the other is up.

- One of the fiber strands in the cable is disconnected.

In these cases, UDLD disables the affected port.

In a point-to-point link, UDLD hello packets can be considered as a heart beat whose presence guarantees the health of the link. Conversely, the loss of the heart beat means that the link must be shut down if it is not possible to reestablish a bidirectional link.

If both fiber strands in a cable are working normally from a Layer 1 perspective, UDLD in aggressive mode detects whether those fiber strands are connected correctly and whether traffic is flowing bidirectionally between the correct neighbors. This check cannot be performed by autonegotiation because autonegotiation operates at Layer 1.

Methods to Detect Unidirectional Links

UDLD operates by using two methods:

- Neighbor database maintenance
- Event-driven detection and echoing

Neighbor Database Maintenance

UDLD learns about other UDLD-capable neighbors by periodically sending a hello packet (also called an advertisement or probe) on every active port to keep each device informed about its neighbors.

When the device receives a hello message, it caches the information until the age time (hold time or time-to-live) expires. If the device receives a new hello message before an older cache entry ages, the device replaces the older entry with the new one.

Whenever a port is disabled and UDLD is running, whenever UDLD is disabled on a port, or whenever the device is reset, UDLD clears all existing cache entries for the ports that are affected by the configuration change. UDLD sends at least one message to inform the neighbors to flush the part of their caches affected by the status change. The message is intended to keep the caches synchronized.



Note An interface does not support multiple UDLD neighbors. If an ingress UDLD protocol data unit (PDU) has multiple device IDs in echo type, length and value (TLV), the interface enters the error-disabled state.

Event-Driven Detection and Echoing

UDLD relies on echoing as its detection operation. Whenever a UDLD device learns about a new neighbor or receives a resynchronization request from an out-of-sync neighbor, it restarts the detection window on its side of the connection and sends echo messages in reply. Because this behavior is the same on all UDLD neighbors, the sender of the echoes expects to receive an echo in reply.

If the detection window ends and no valid reply message are received, the link might shut down, depending on the UDLD mode. When UDLD is in normal mode, the link might be considered undetermined and might not be shut down. When UDLD is in aggressive mode, the link is considered unidirectional, and the port is disabled.

UniDirectional Link Detection Reset Options

If an interface becomes disabled by UDLD, you can use one of the following options to reset UDLD:

- The **udld reset** interface configuration command.
- The **shutdown** interface configuration command followed by the **no shutdown** interface configuration command restarts the disabled port.
- The **no udld {aggressive | enable}** global configuration command followed by the **udld {aggressive | enable}** global configuration command reenables the disabled ports.
- The **no udld port** interface configuration command followed by the **udld port [aggressive]** interface configuration command reenables the disabled fiber-optic port.
- The **errdisable recovery cause udld** global configuration command enables the timer to automatically recover from the UDLD error-disabled state, and the **errdisable recovery interval interval** global configuration command specifies the time to recover from the UDLD error-disabled state.

The **udld port disable** command disables UDLD on fiber-optic LAN ports.



Note This command is only supported on fiber-optic LAN ports.

Default UniDirectional Link Detection Configuration

Table 18: Default UDLD Configuration

Feature	Default Setting
UDLD global enable state	Globally disabled
UDLD per-port enable state for fiber-optic media	Disabled on all Ethernet fiber-optic ports
UDLD per-port enable state for twisted-pair (copper) media	Disabled on all Ethernet 10/100 and 1000BASE-TX p
UDLD aggressive mode	Disabled
Fast UDLD per-port enable state	Disabled on all ports

How to Configure UniDirectional Link Detection

The following sections provide information about configuring UDLD:

Enabling UniDirectional Link Detection Globally

Follow these steps to enable UDLD in the aggressive or normal mode and to set the configurable message timer on all fiber-optic ports on the device.

Procedure

	Command or Action	Purpose
Step 1	enable Example: Device> enable	Enables privileged EXEC mode. Enter your password if prompted.
Step 2	configure terminal Example: Device# configure terminal	Enters global configuration mode.
Step 3	udld {aggressive enable message time message-timer-interval} Example: Device (config) # udld enable message time 10	Specifies the UDLD mode of operation: <ul style="list-style-type: none"> • aggressive—Enables UDLD in aggressive mode on all fiber-optic ports. • enable—Enables UDLD in normal mode on all fiber-optic ports on the device. UDLD is disabled by default. An individual interface configuration overrides the setting of the udld enable global configuration command. • message time message-timer-interval—Configures the period of time between UDLD probe messages on ports that are in the advertisement phase and are detected to be bidirectional. The range is from 1 to 90 seconds; the default value is 15. <p>Note This command affects fiber-optic ports only. Use the udld interface configuration command to enable UDLD on other port types.</p> <p>Use the no form of this command, to disable UDLD.</p>
Step 4	end Example: Device (config) # end	Returns to privileged EXEC mode.

Enabling UniDirectional Link Detection on an Interface

Follow these steps either to enable UDLD in the aggressive or normal mode or to disable UDLD on a port.

Procedure

	Command or Action	Purpose
Step 1	enable Example: Device> enable	Enables privileged EXEC mode. Enter your password if prompted.
Step 2	configure terminal Example: Device# configure terminal	Enters global configuration mode.
Step 3	interface <i>interface-id</i> Example: Device (config) # interface gigabitethernet 1/0/1	Specifies the port to be enabled for UDLD, and enters interface configuration mode.
Step 4	udld port [aggressive] Example: Device (config-if) # udld port aggressive	UDLD is disabled by default. <ul style="list-style-type: none"> • udld port—Enables UDLD in normal mode on the specified port. • udld port aggressive—(Optional) Enables UDLD in aggressive mode on the specified port. <p>Note Use the no udld port interface configuration command to disable UDLD on a specified fiber-optic port.</p>
Step 5	end Example: Device (config-if) # end	Returns to privileged EXEC mode.

Enabling Fast UniDirectional Link Detection on an Interface

Follow these steps to enable Fast UDLD on a port.

Procedure

	Command or Action	Purpose
Step 1	configure terminal Example: Device# configure terminal	Enters global configuration mode.
Step 2	interface <i>interface-id</i> Example:	Specifies the port to be enabled for Fast UDLD, and enters interface configuration mode.

	Command or Action	Purpose
	Device(config)# interface gigabitethernet 1/0/1	
Step 3	udld fast-hello <i>message time interval</i> Example: Device(config-if)# udld fast-hello 200	Enables Fast UDLD on the specified port. <ul style="list-style-type: none"> • message time <i>message-timer-interval</i>—Configures the period of time between UDLD probe messages on ports that are in the advertisement phase and are detected to be bidirectional. Note Fast UDLD can be enabled only if UDLD is already enabled on the specified port.
Step 4	end Example: Device(config-if)# end	Returns to privileged EXEC mode.

Enabling Fast UniDirectional Link Detection Error Reporting

Follow these steps to enable Fast UDLD error reporting on the switch.

Procedure

	Command or Action	Purpose
Step 1	configure terminal Example: Device# configure terminal	Enters global configuration mode.
Step 2	udld fast-hello error-reporting Example: Device(config)# udld fast-hello error-reporting	Enables the display of console messages to report the error upon detection of a link failure. Note The detected unidirectional link will not be disabled if udld fast-hello error-reporting has been enabled.
Step 3	end Example: Device(config)# end	Returns to privileged EXEC mode.

Disabling UniDirectional Link Detection on Fiber-Optic LAN Interfaces

To disable UDLD on Fiber-optic LAN interfaces, perform this procedure:

Procedure

	Command or Action	Purpose
Step 1	enable Example: Device> enable	Enables privileged EXEC mode. Enter your password if prompted.
Step 2	configure terminal Example: Device# configure terminal	Enters global configuration mode.
Step 3	interface type number Example: Device(config)# interface gigabitethernet 0/1/1	Configures an interface and enters interface configuration mode.
Step 4	udld port disable Example: Device(config-if)# udld port disable	Disables UDLD on a fiber-optic LAN port. <ul style="list-style-type: none"> • The udld port disable command is only supported on fiber-optic LAN ports. • The no udld port disable command reverts to the udld enable global configuration command setting.
Step 5	end Example: Device(config-if)# end	Exits interface configuration mode and returns to privileged EXEC mode.

Monitoring and Maintaining UniDirectional Link Detection

Command	Purpose
show udld [<i>interface-id</i> neighbors]	Displays the UDLD status for the specified port or for all ports.
show udld fast-hello [<i>interface-id</i>]	Displays fast-hello information for the specified port or for all ports.

Console Error Messages For Fast UniDirectional Link Detection

When a link failure is detected by fast UDLD, the unidirectional link is err-disabled by UDLD after displaying the following message on the console:

```
%UDLD-4-UDLD_PORT_DISABLED: UDLD disabled interface Hu1/0/10, unidirectional link detected
```

If the **udld fast-hello error-reporting** is configured, when fast UDLD detects a link failure, it prints the following console message instead of err-disabling the affected port:

```
%UDLD-SP-4-UDLD_PORT_FAILURE: UDLD failure reported per user request, interface Hu1/0/10,
fast udld unidirectional link detected
```

The **udld reset** command can be used to clear the UDLD port state in both the cases.

Additional References for UniDirectional Link Detection

Related Documents

Related Topic	Document Title
For complete syntax and usage information for the commands used in this chapter.	See the Layer 2/3 Commands section of the <i>Command Reference (Catalyst 9600 Series Switches)</i>

Feature History for UniDirectional Link Detection

This table provides release and related information for features explained in this module.

These features are available on all releases subsequent to the one they were introduced in, unless noted otherwise.

Release	Feature	Feature Information
Cisco IOS XE Gibraltar 16.11.1	UniDirectional Link Detection (UDLD)	UDLD is a Layer 2 protocol that enables devices connected through fiber-optic or twisted-pair Ethernet cables to monitor the physical configuration of the cables and detect when a unidirectional link exists.

Use Cisco Feature Navigator to find information about platform and software image support. To access Cisco Feature Navigator, go to <http://www.cisco.com/go/cfn>.



CHAPTER 8

Configuring Layer 2 Protocol Tunneling

- [Prerequisites for Layer 2 Protocol Tunneling, on page 169](#)
- [Restrictions for Layer 2 Protocol Tunneling, on page 169](#)
- [Information About Layer 2 Protocol Tunneling, on page 169](#)
- [How to Configure Layer 2 Protocol Tunneling, on page 173](#)
- [How to Configure Layer 2 Protocol Tunneling for EtherChannels, on page 176](#)
- [Configuration Examples for Layer 2 Protocol Tunneling, on page 181](#)
- [Monitoring Tunneling Status, on page 183](#)
- [Feature History for Layer 2 Protocol Tunneling, on page 183](#)

Prerequisites for Layer 2 Protocol Tunneling

The following sections list prerequisites and considerations for configuring Layer 2 protocol tunneling.

To configure Layer 2 point-to-point tunneling to facilitate the automatic creation of EtherChannels, you need to configure both the SP (service-provider) edge switch and the customer device.

Restrictions for Layer 2 Protocol Tunneling

Layer 2 protocol tunneling and native VLAN tagging are not supported on the same trunk port. If native VLAN tagging is enabled globally on the device and Layer 2 protocol tunneling needs to be enabled on a trunk port, disable native VLAN tagging on the trunk port using the **no switchport trunk native vlan tag** command before configuring Layer 2 protocol tunneling.

Information About Layer 2 Protocol Tunneling

The following sections provide information about Layer 2 protocol tunneling:

Layer 2 Protocol Tunneling Overview

Customers at different sites that are connected across a service-provider network need to use various Layer 2 protocols to scale their topologies to include all remote sites, as well as the local sites. STP must run properly, and every VLAN should build a proper spanning tree that includes the local site and all remote sites across

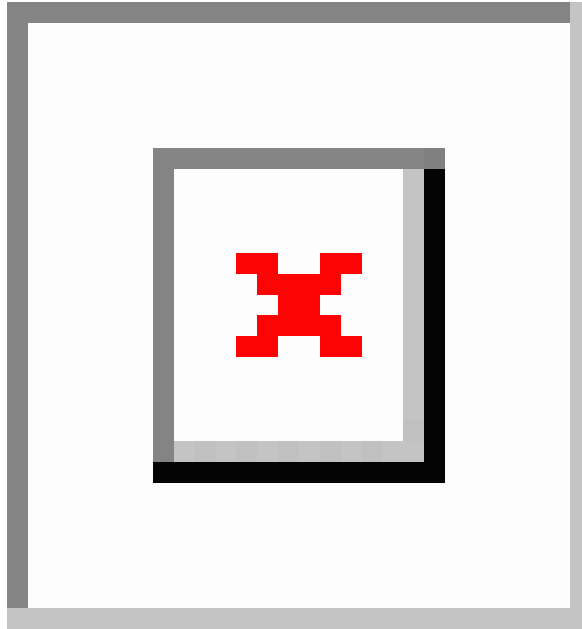
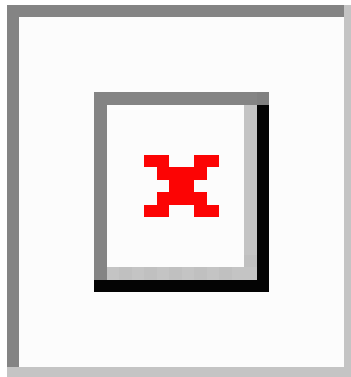
the service-provider network. Cisco Discovery Protocol (CDP) must discover neighboring Cisco devices from local and remote sites. VLAN Trunking Protocol (VTP) must provide consistent VLAN configuration throughout all sites in the customer network.

When protocol tunneling is enabled, edge device on the inbound side of the service-provider network encapsulate Layer 2 protocol packets with a special MAC address and send them across the service-provider network. Core devices in the network do not process these packets but forward them as normal packets. Layer 2 protocol data units (PDUs) for CDP, STP, or VTP cross the service-provider network and are delivered to customer devices on the outbound side of the service-provider network. Identical packets are received by all customer ports on the same VLANs with these results:

- Users on each of a customer's sites can properly run STP, and every VLAN can build a correct spanning tree based on parameters from all sites and not just from the local site.
- CDP discovers and shows information about the other Cisco devices that are connected through the service-provider network.
- VTP provides consistent VLAN configuration throughout the customer network, propagating to all devices through the service provider.

Layer 2 protocol tunneling can be used independently or can enhance IEEE 802.1Q tunneling. If protocol tunneling is not enabled on IEEE 802.1Q tunneling ports, remote devices at the receiving end of the service-provider network do not receive the PDUs and cannot properly run STP, CDP, and VTP. When protocol tunneling is enabled, Layer 2 protocols within each customer's network are totally separate from those running within the service-provider network. Customer devices on different sites that send traffic through the service-provider network with IEEE 802.1Q tunneling achieve complete knowledge of the customer's VLAN. If IEEE 802.1Q tunneling is not used, you can still enable Layer 2 protocol tunneling by connecting to the customer device through access ports and by enabling tunneling on the service-provider access port.

For example, in the following figure (Layer 2 Protocol Tunneling), Customer X has four switches in the same VLAN, that are connected through the service-provider network. If the network does not tunnel PDUs, switches on the far ends of the network cannot properly run STP, CDP, and VTP. For example, STP for a VLAN on a switch in Customer X, Site 1, will build a spanning tree on the switches at that site without considering convergence parameters based on Customer X's switch in Site 2. This could result in the topology that is shown in the Layer 2 Network Topology without Proper Convergence figure.

Figure 33: Layer 2 Protocol Tunneling*Figure 34: Layer 2 Network Topology Without Proper Convergence*

Layer 2 Protocol Tunneling on Ports

You can enable Layer 2 protocol tunneling (by protocol) on the ports that are connected to the customer in the edge devices of the service-provider network. The service-provider edge devices connected to the customer device perform the tunneling process. Edge device tunnel ports are connected to customer IEEE 802.1Q trunk ports. Edge device access ports are connected to customer access ports. The edge devices connected to the customer device perform the tunneling process.

You can enable Layer 2 protocol tunneling on ports that are configured as access ports or tunnel ports or trunk ports. You cannot enable Layer 2 protocol tunneling on ports that are configured in either **switchport mode dynamic auto** mode (the default mode) or **switchport mode dynamic desirable** mode.

The device supports Layer 2 protocol tunneling for CDP, STP, and VTP. For emulated point-to-point network topologies, it also supports PAgP, LACP, LLDP, and UDLD protocols.



Note PAgP, LACP, and UDLD protocol tunneling are only intended to emulate a point-to-point topology. An erroneous configuration that sends tunneled packets to many ports could lead to a network failure.

When the Layer 2 PDUs that entered the service-provider inbound edge device through a Layer 2 protocol-enabled port exit through the trunk port into the service-provider network, the device overwrites the customer PDU-destination MAC address with a well-known Cisco proprietary multicast address (01-00-0c-cd-cd-d0). If IEEE 802.1Q tunneling is enabled, packets are also double-tagged; the outer tag is the customer metro tag, and the inner tag is the customer's VLAN tag. The core devices ignore the inner tags and forward the packet to all trunk ports in the same metro VLAN. The edge devices on the outbound side restore the proper Layer 2 protocol and MAC address information and forward the packets to all tunnel or access ports in the same metro VLAN. Therefore, the Layer 2 PDUs remain intact and are delivered across the service-provider infrastructure to the other side of the customer network.

See the Layer 2 Protocol Tunneling figure in [Layer 2 Protocol Tunneling Overview](#), with Customer X and Customer Y in access VLANs 30 and 40, respectively. Asymmetric links connect the customers in Site 1 to edge switches in the service-provider network. The Layer 2 PDUs (for example, BPDUs) coming into Switch B from Customer Y in Site 1 are forwarded to the infrastructure as double-tagged packets with the well-known MAC address as the destination MAC address. These double-tagged packets have the metro VLAN tag of 40, as well as an inner VLAN tag (for example, VLAN 100). When the double-tagged packets enter Switch D, the outer VLAN tag 40 is removed, the well-known MAC address is replaced with the respective Layer 2 protocol MAC address, and the packet is sent to Customer Y on Site 2 as a single-tagged frame in VLAN 100.

You can also enable Layer 2 protocol tunneling on access ports on the edge switch that is connected to access or trunk ports on the customer switch. In this case, the encapsulation and decapsulation process are the same as described in the previous paragraph, except that the packets are not double-tagged in the service-provider network. The single tag is the customer-specific access VLAN tag.

In switch stacks, Layer 2 protocol tunneling configuration is distributed among all member switches. Each member switch that receives an ingress packet on a local port encapsulates or decapsulates the packet and forwards it to the appropriate destination port. On a single switch, ingress Layer 2 protocol-tunneled traffic is sent across all local ports in the same VLAN on which Layer 2 protocol tunneling is enabled. In a stack, packets received by a Layer 2 protocol-tunneled port are distributed to all ports in the stack that are configured for Layer 2 protocol tunneling and are in the same VLAN. All Layer 2 protocol tunneling configuration is handled by the active switch and distributed to all member switches in the stack.

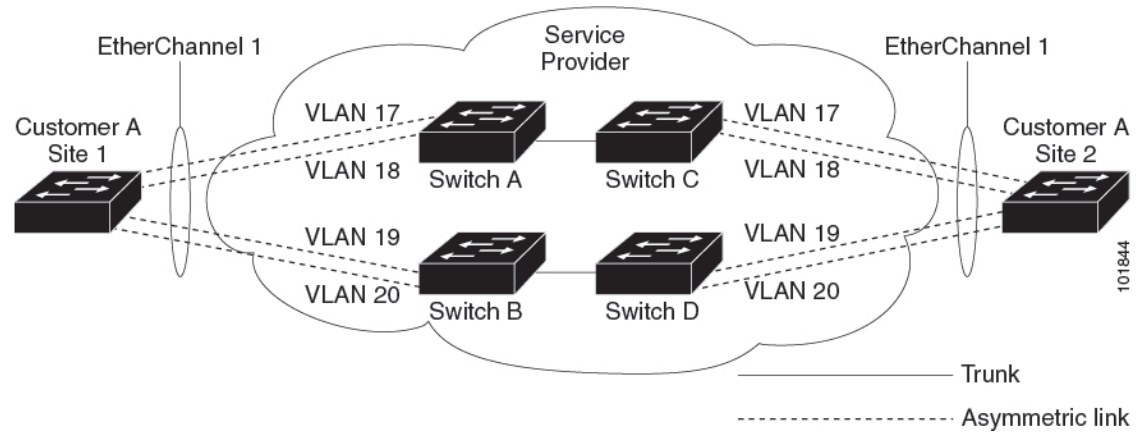
Layer 2 Protocol Tunneling for EtherChannels

In an SP network, you can use Layer 2 protocol tunneling to enhance the creation of EtherChannels by emulating a point-to-point network topology. When you enable protocol tunneling (PAgP or LACP) on the SP switch, remote customer switches receive the PDUs and can negotiate the automatic creation of EtherChannels.

For example, in the following figure (Layer 2 Protocol Tunneling for EtherChannels), Customer A has two switches in the same VLAN that are connected through the SP network. When the network tunnels PDUs, switches on the far ends of the network can negotiate the automatic creation of EtherChannels without needing dedicated lines.

While configuring Layer 2 Protocol Tunneling on trunk ports, both the trunk ports on the SP edge device should be configured with different native VLANs. The native VLAN of one trunk port should not be in the list of allowed VLANs of the other trunk port to avoid loops.

Figure 35: Layer 2 Protocol Tunneling for EtherChannels



Default Layer 2 Protocol Tunneling Configuration

The following table shows the default Layer 2 protocol tunneling configuration.

Table 19: Default Layer 2 Ethernet Interface VLAN Configuration

Feature	Default Setting
Layer 2 protocol tunneling	Disabled.
Shutdown threshold	None set.
Drop threshold	None set.

How to Configure Layer 2 Protocol Tunneling

The following section provides configuration information on how to configure a layer 2 protocol tunnel:

Configuring Layer 2 Protocol Tunneling

Procedure

	Command or Action	Purpose
Step 1	enable Example: Device> enable	Enables privileged EXEC mode. Enter your password if prompted.
Step 2	configure terminal Example: Device# configure terminal	Enters global configuration mode.

	Command or Action	Purpose
Step 3	interface <i>interface-id</i> Example: Device(config)# interface gigabitethernet1/0/1	Specifies the interface that is connected to the phone, and enters interface configuration mode.
Step 4	Use one of the following: <ul style="list-style-type: none"> • switchport mode dot1q-tunnel Example: Device(config-if)# switchport mode dot1q-tunnel	Configures the interface as an IEEE 802.1Q tunnel port or a trunk port.
Step 5	l2protocol-tunnel [cdp lldp point-to-point stp vtp] Example: Device(config-if)# l2protocol-tunnel cdp	Enables protocol tunneling for the desired protocol. If no keyword is entered, tunneling is enabled for all four Layer 2 protocols. Note Use the no l2protocol-tunnel [cdp lldp point-to-point stp vtp] interface configuration command to disable protocol tunneling for one of the Layer 2 protocols or for all three.
Step 6	l2protocol-tunnel shutdown-threshold [<i>packet_second_rate_value</i> cdp lldp point-to-point stp vtp] Example: Device(config-if)# l2protocol-tunnel shutdown-threshold 100 cdp	(Optional) Configures the threshold for packets-per-second that are accepted for encapsulation. The interface is disabled if the configured threshold is exceeded. If no protocol option is specified, the threshold applies to each of the tunneled Layer 2 protocol types. The range is 1 to 4096. The default is to have no threshold configured. Note If you also set a drop threshold on this interface, the shutdown-threshold value must be greater than or equal to the drop-threshold value. Note Use the no l2protocol-tunnel shutdown-threshold [<i>packet_second_rate_value</i> cdp lldp point-to-point stp vtp] and the no l2protocol-tunnel drop-threshold [<i>packet_second_rate_value</i> cdp lldp point-to-point stp vtp] commands to return the shutdown and drop thresholds to the default settings.

	Command or Action	Purpose
Step 7	<p>l2protocol-tunnel drop-threshold [<i>packet_second_rate_value</i> cdp lldp point-to-point stp vtp]</p> <p>Example:</p> <pre>Device(config-if)# l2protocol-tunnel drop-threshold 100 cdp</pre>	<p>(Optional) Configures the threshold for packets-per-second that are accepted for encapsulation. The interface drops packets if the configured threshold is exceeded. If no protocol option is specified, the threshold applies to each of the tunneled Layer 2 protocol types. The range is 1 to 4096. The default is to have no threshold configured.</p> <p>Note If you also set a shutdown threshold on this interface, the drop-threshold value must be less than or equal to the shutdown-threshold value.</p> <p>Note Use the no l2protocol-tunnel shutdown-threshold [cdp lldp point-to-point stp vtp] and the no l2protocol-tunnel drop-threshold [cdp stp vtp] commands to return the shutdown and drop thresholds to the default settings.</p>
Step 8	<p>exit</p> <p>Example:</p> <pre>Device(config-if)# exit</pre>	Returns to global configuration mode.
Step 9	<p>errdisable recovery cause l2ptguard</p> <p>Example:</p> <pre>Device(config)# errdisable recovery cause l2ptguard</pre>	<p>(Optional) Configures the recovery mechanism from a Layer 2 maximum-rate error so that the interface is reenabled and can try again. Errdisable recovery is disabled by default; when enabled, the default time interval is 300 seconds.</p>
Step 10	<p>spanning-tree bpdudfilter enable</p> <p>Example:</p> <pre>Device(config)# spanning-tree bpdudfilter enable</pre>	<p>Inserts a BPDU filter for spanning tree.</p> <p>Note While configuring Layer 2 Protocol Tunneling on a trunk port, you must enable a BPDU filter for spanning tree.</p>
Step 11	<p>end</p> <p>Example:</p> <pre>Device(config)# end</pre>	Returns to privileged EXEC mode.

	Command or Action	Purpose
Step 12	show l2protocol Example: Device# <code>show l2protocol</code>	Displays the Layer 2 tunnel ports on the device, including the protocols configured, the thresholds, and the counters.
Step 13	copy running-config startup-config Example: Device# <code>copy running-config startup-config</code>	(Optional) Saves your entries in the configuration file.

How to Configure Layer 2 Protocol Tunneling for EtherChannels

For EtherChannels, you need to configure both the SP (service-provider) edge devices and the customer devices for Layer 2 protocol tunneling. The following sections provide configuration information on how to configure the SP edge device and how to configure the customer device:

Configuring the SP Edge Switch

Procedure

	Command or Action	Purpose
Step 1	enable Example: Device> <code>enable</code>	Enables privileged EXEC mode. Enter your password if prompted.
Step 2	configure terminal Example: Device# <code>configure terminal</code>	Enters global configuration mode.
Step 3	interface <i>interface-id</i> Example: Device(config)# <code>interface gigabitethernet1/0/1</code>	Specifies the interface that is connected to the phone, and enters interface configuration mode.
Step 4	switchport trunk native vlan <i>vlan-id</i> Example: Device(config-if)# <code>switchport trunk native vlan 2</code>	Configures the native VLAN. Note While configuring Layer 2 Protocol Tunneling for EtherChannels on trunk ports, you must configure different native VLANs on both trunk ports on the SP edge device.
Step 5	switchport trunk allowed vlan <i>vlan-id list</i>	Specifies the list of allowed VLANs.

	Command or Action	Purpose
	<p>Example:</p> <pre>Device(config-if)# switchport trunk allowed vlan 1,2,4-3003,3005-4094</pre>	<p>Note While configuring Layer 2 Protocol Tunneling for EtherChannels on trunk ports, you must ensure that the native VLAN of one trunk port of the SP edge device should not be in the list of allowed VLANs of the other trunk port to avoid loops.</p>
Step 6	<p>Use one of the following:</p> <ul style="list-style-type: none"> • switchport mode dot1q-tunnel • switchport mode trunk <p>Example:</p> <pre>Device(config-if)# switchport mode dot1q-tunnel</pre> <p>or</p> <pre>Device(config-if)# switchport mode trunk</pre>	<p>Configures the interface as an IEEE 802.1Q tunnel port or as a trunk port.</p>
Step 7	<p>l2protocol-tunnel point-to-point [pagp lacp udld]</p> <p>Example:</p> <pre>Device(config-if)# l2protocol-tunnel point-to-point pagp</pre>	<p>(Optional) Enables point-to-point protocol tunneling for the desired protocol. If no keyword is entered, tunneling is enabled for all three protocols.</p> <p>Note To avoid a network failure, make sure that the network is a point-to-point topology before you enable tunneling for PAgP, LACP, or UDLD packets.</p> <p>Note Use the no l2protocol-tunnel [point-to-point [pagp lacp udld]] interface configuration command to disable point-to-point protocol tunneling for one of the Layer 2 protocols or for all three.</p>
Step 8	<p>l2protocol-tunnel shutdown-threshold [point-to-point [pagp lacp udld]] value</p> <p>Example:</p> <pre>Device(config-if)# l2protocol-tunnel shutdown-threshold point-to-point pagp 100</pre>	<p>(Optional) Configures the threshold for packets-per-second that are accepted for encapsulation. The interface is disabled if the configured threshold is exceeded. If no protocol option is specified, the threshold applies to each of the tunneled Layer 2 protocol types. The range is 1 to 4096. The default is to have no threshold configured.</p>

	Command or Action	Purpose
		<p>Note If you also set a drop threshold on this interface, the shutdown-threshold value must be greater than or equal to the drop-threshold value.</p> <p>Note Use the no l2protocol-tunnel shutdown-threshold [point-to-point [pagp lacp udld]] and the no l2protocol-tunnel drop-threshold [[point-to-point [pagp lacp udld]]] commands to return the shutdown and drop thresholds to the default settings.</p>
Step 9	<p>l2protocol-tunnel drop-threshold [point-to-point [pagp lacp udld]] value</p> <p>Example:</p> <pre>Device(config-if)# l2protocol-tunnel drop-threshold point-to-point pagp 500</pre>	<p>(Optional) Configures the threshold for packets-per-second that are accepted for encapsulation. The interface drops packets if the configured threshold is exceeded. If no protocol option is specified, the threshold applies to each of the tunneled Layer 2 protocol types. The range is 1 to 4096. The default is to have no threshold configured.</p> <p>Note If you also set a shutdown threshold on this interface, the drop-threshold value must be less than or equal to the shutdown-threshold value.</p>
Step 10	<p>no cdp enable</p> <p>Example:</p> <pre>Device(config-if)# no cdp enable</pre>	Disables CDP on the interface.
Step 11	<p>spanning-tree bpdu filter enable</p> <p>Example:</p> <pre>Device(config-if)# spanning-tree bpdu filter enable</pre>	Enables BPDU filtering on the interface.
Step 12	<p>exit</p> <p>Example:</p> <pre>Device(config-if)# exit</pre>	Returns to global configuration mode.
Step 13	<p>errdisable recovery cause l2ptguard</p> <p>Example:</p> <pre>Device(config)# errdisable recovery cause l2ptguard</pre>	<p>(Optional) Configures the recovery mechanism from a Layer 2 maximum-rate error so that the interface is reenabled and can try again. Errdisable recovery is disabled by default;</p>

	Command or Action	Purpose
		when enabled, the default time interval is 300 seconds.
Step 14	end Example: Device(config)# end	Returns to privileged EXEC mode.
Step 15	show l2protocol Example: Device# show l2protocol	Displays the Layer 2 tunnel ports on the device, including the protocols configured, the thresholds, and the counters.
Step 16	copy running-config startup-config Example: Device# copy running-config startup-config	(Optional) Saves your entries in the configuration file.

Configuring the Customer Device

Before you begin

For EtherChannels, you need to configure both the SP edge device and the customer device for Layer 2 protocol tunneling.

Procedure

	Command or Action	Purpose
Step 1	enable Example: Device> enable	Enables privileged EXEC mode. Enter your password if prompted.
Step 2	configure terminal Example: Device# configure terminal	Enters global configuration mode.
Step 3	interface interface-id Example: Device(config)# interface gigabitethernet1/0/1	Specifies the interface that is connected to the phone, and enters interface configuration mode.
Step 4	switchport trunk encapsulation dot1q Example: Device(config-if)# switchport trunk encapsulation dot1q	Sets the trunking encapsulation format to IEEE 802.1Q.

	Command or Action	Purpose
Step 5	switchport mode trunk Example: Device(config-if)# switchport mode trunk	Enables trunking on the interface.
Step 6	udld port Example: Device(config-if)# udld port	Enables UDLD in normal mode on the interface.
Step 7	channel-group channel-group-number mode desirable Example: Device(config-if)# channel-group 25 mode desirable	Assigns the interface to a channel group, and specifies desirable for the PAgP mode.
Step 8	exit Example: Device(config-if)# exit	Returns to global configuration mode.
Step 9	interface port-channel port-channel number Example: Device(config)# interface port-channel port-channel 25	Enters port-channel interface mode.
Step 10	shutdown Example: Device(config)# shutdown	Shuts down the interface.
Step 11	no shutdown Example: Device(config)# no shutdown	Enables the interface.
Step 12	end Example: Device(config)# end	Returns to privileged EXEC mode.
Step 13	show l2protocol Example: Device# show l2protocol	Displays the Layer 2 tunnel ports on the device, including the protocols configured, the thresholds, and the counters.
Step 14	copy running-config startup-config Example:	(Optional) Saves your entries in the configuration file.

	Command or Action	Purpose
	Device# <code>copy running-config startup-config</code>	Note Use the no switchport mode trunk , the no uddl enable , and the no channel group channel-group-number mode desirable interface configuration commands to return the interface to the default settings.

Configuration Examples for Layer 2 Protocol Tunneling

The following sections provide various configuration examples for layer 2 protocol tunneling:

Example: Configuring Layer 2 Protocol Tunneling

The following example shows how to configure Layer 2 protocol tunneling for Cisco Discovery Protocol, STP, and VTP and to verify the configuration.

```
Device(config)# interface gigabitethernet1/0/11
Device(config-if)# l2protocol-tunnel cdp
Device(config-if)# l2protocol-tunnel stp
Device(config-if)# l2protocol-tunnel vtp
Device(config-if)# l2protocol-tunnel shutdown-threshold 1500
Device(config-if)# l2protocol-tunnel drop-threshold 1000
Device(config-if)# exit
```

```
Device(config)# end
Device# show l2protocol
```

```
Port Protocol Shutdown Drop Encapsulation Decapsulation Drop
Threshold Threshold Counter Counter Counter
-----
Gi0/11 cdp 1500 1000 2288 2282 0
stp 1500 1000 116 13 0
vtp 1500 1000 3 67 0
pagp ---- ---- 0 0 0
lacp ---- ---- 0 0 0
udld ---- ---- 0 0 0
```

Examples: Configuring the SP Edge and Customer Switches

This example shows how to configure the SP edge switch 1 and edge switch 2. VLANs 17, 18, 19, and 20 are the access VLANs, Fast Ethernet interfaces 1 and 2 are point-to-point tunnel ports with PAGP and UDLD enabled, the drop threshold is 1000, and Fast Ethernet interface 3 is a trunk port.

SP edge switch 1 configuration:

```
Device(config)# interface gigabitethernet1/0/1
Device(config-if)# switchport access vlan 17
Device(config-if)# switchport mode dot1q-tunnel
```

```

Device(config-if)# l2protocol-tunnel point-to-point pagp
Device(config-if)# l2protocol-tunnel point-to-point udld
Device(config-if)# l2protocol-tunnel drop-threshold point-to-point pagp 1000
Device(config-if)# exit
Device(config)# interface gigabitethernet1/0/2
Device(config-if)# switchport access vlan 18
Device(config-if)# switchport mode dot1q-tunnel
Device(config-if)# l2protocol-tunnel point-to-point pagp
Device(config-if)# l2protocol-tunnel point-to-point udld
Device(config-if)# l2protocol-tunnel drop-threshold point-to-point pagp 1000
Device(config-if)# exit
Device(config)# interface gigabitethernet1/0/3
Device(config-if)# switchport trunk encapsulation isl
Device(config-if)# switchport mode trunk

```

SP edge switch 2 configuration:

```

Device(config)# interface gigabitethernet1/0/1
Device(config-if)# switchport access vlan 19
Device(config-if)# switchport mode dot1q-tunnel
Device(config-if)# l2protocol-tunnel point-to-point pagp
Device(config-if)# l2protocol-tunnel point-to-point udld
Device(config-if)# l2protocol-tunnel drop-threshold point-to-point pagp 1000
Device(config-if)# exit
Device(config)# interface gigabitethernet1/0/2
Device(config-if)# switchport access vlan 20
Device(config-if)# switchport mode dot1q-tunnel
Device(config-if)# l2protocol-tunnel point-to-point pagp
Device(config-if)# l2protocol-tunnel point-to-point udld
Device(config-if)# l2protocol-tunnel drop-threshold point-to-point pagp 1000
Device(config-if)# exit
Device(config)# interface gigabitethernet1/0/3
Device(config-if)# switchport trunk encapsulation isl
Device(config-if)# switchport mode trunk

```

This example shows how to configure the customer switch at Site 1. Fast Ethernet interfaces 1, 2, 3, and 4 are set for IEEE 802.1Q trunking, UDLD is enabled, EtherChannel group 1 is enabled, and the port channel is shut down and then enabled to activate the EtherChannel configuration.

```

Device(config)# interface gigabitethernet1/0/1
Device(config-if)# switchport trunk encapsulation dot1q
Device(config-if)# switchport mode trunk
Device(config-if)# udld enable
Device(config-if)# channel-group 1 mode desirable
Device(config-if)# exit
Device(config)# interface gigabitethernet1/0/2
Device(config-if)# switchport trunk encapsulation dot1q
Device(config-if)# switchport mode trunk
Device(config-if)# udld enable
Device(config-if)# channel-group 1 mode desirable
Device(config-if)# exit
Device(config)# interface gigabitethernet1/0/3
Device(config-if)# switchport trunk encapsulation dot1q
Device(config-if)# switchport mode trunk
Device(config-if)# udld enable
Device(config-if)# channel-group 1 mode desirable
Device(config-if)# exit
Device(config)# interface gigabitethernet1/0/4
Device(config-if)# switchport trunk encapsulation dot1q

```

```

Device(config-if)# switchport mode trunk
Device(config-if)# udld enable
Device(config-if)# channel-group 1 mode desirable
Device(config-if)# exit
Device(config)# interface port-channel 1
Device(config-if)# shutdown
Device(config-if)# no shutdown
Device(config-if)# exit

```

Monitoring Tunneling Status

The following table describes the commands used to monitor tunneling status.

Table 20: Commands for Monitoring Tunneling

Command	Purpose
clear l2protocol-tunnel counters	Clears the protocol counters on Layer 2 protocol tunneling ports.
show dot1q-tunnel	Displays IEEE 802.1Q tunnel ports on the device.
show dot1q-tunnel interface <i>interface-id</i>	Verifies if a specific interface is a tunnel port.
show l2protocol-tunnel	Displays information about Layer 2 protocol tunneling ports.
show errdisable recovery	Verifies if the recovery timer from a Layer 2 protocol-tunnel error disable state is enabled.
show l2protocol-tunnel interface <i>interface-id</i>	Displays information about a specific Layer 2 protocol tunneling port.
show l2protocol-tunnel summary	Displays only Layer 2 protocol summary information.
show vlan dot1q tag native	Displays the status of native VLAN tagging on the device.

Feature History for Layer 2 Protocol Tunneling

This table provides release and related information for features explained in this module.

These features are available on all releases subsequent to the one they were introduced in, unless noted otherwise.

Release	Feature	Feature Information
Cisco IOS XE Gibraltar 16.12.1	Layer 2 Protocol Tunneling	Layer 2 protocols allow you to scale topologies to include all remote sites and local sites.

Use Cisco Feature Navigator to find information about platform and software image support. To access Cisco Feature Navigator, go to <http://www.cisco.com/go/cfn>.



CHAPTER 9

Configuring IEEE 802.1Q Tunneling

- [Information About IEEE 802.1Q Tunneling, on page 185](#)
- [Information About Custom EtherType for Cisco Catalyst 9600 Series Supervisor 2 Module, on page 190](#)
- [Restrictions for Custom Ethertypes, on page 191](#)
- [How to Configure IEEE 802.1Q Tunneling, on page 191](#)
- [How to Configure Custom EtherTypes, on page 193](#)
- [Monitoring Tunneling Status, on page 194](#)
- [Example: Configuring an IEEE 802.1Q Tunneling Port, on page 195](#)
- [Example: Configuring a Custom EtherType, on page 195](#)
- [Feature History for IEEE 802.1Q Tunneling, on page 198](#)

Information About IEEE 802.1Q Tunneling

The IEEE 802.1Q Tunneling feature is designed for service providers who carry traffic of multiple customers across their networks and are required to maintain the VLAN and Layer 2 protocol configurations of each customer without impacting the traffic of other customers.

IEEE 802.1Q Tunnel Ports in a Service Provider Network

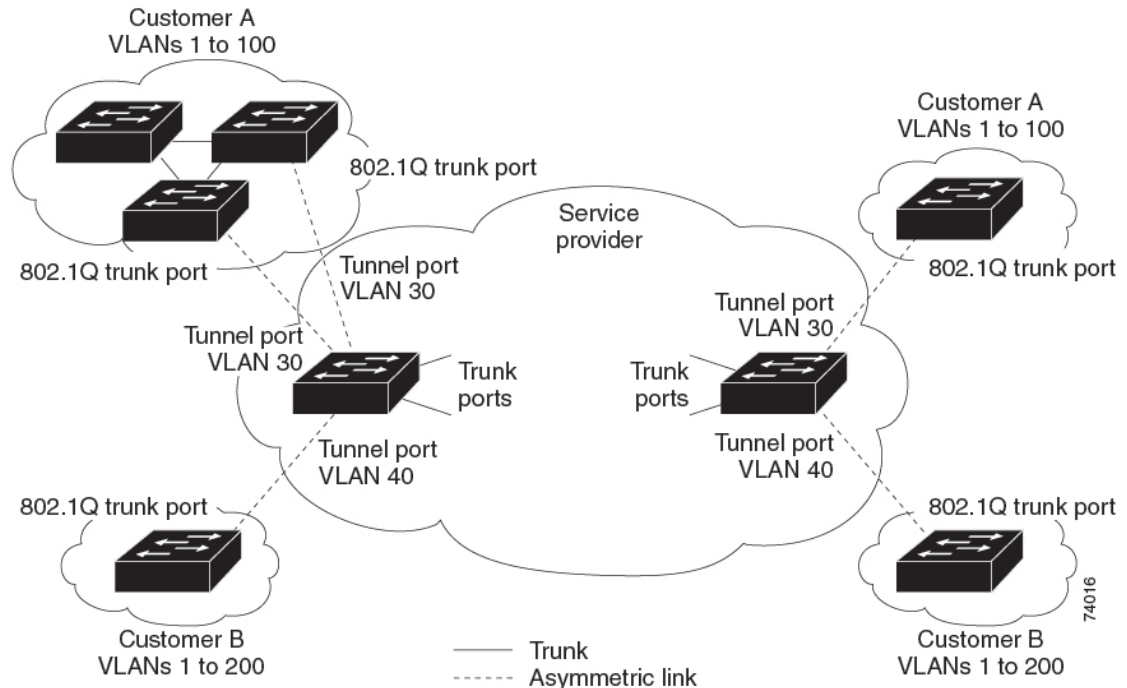
Business customers of service providers often have specific requirements for VLAN IDs and the number of VLANs to be supported. The VLAN ranges required by different customers in the same service-provider network might overlap, and traffic of customers through the infrastructure might be mixed. Assigning a unique range of VLAN IDs to each customer would restrict customer configurations and could easily exceed the VLAN limit (4096) of the IEEE 802.1Q specification.

Using the IEEE 802.1Q tunneling feature, service providers can use a single VLAN to support customers who have multiple VLANs. Customer VLAN IDs are preserved, and traffic from different customers is segregated within the service-provider network, even when they appear to be in the same VLAN. Using IEEE 802.1Q tunneling expands VLAN space by using a VLAN-in-VLAN hierarchy and retagging the tagged packets. A port configured to support IEEE 802.1Q tunneling is called a tunnel port. When you configure tunneling, you assign a tunnel port to a VLAN ID that is dedicated to tunneling. Each customer requires a separate service-provider VLAN ID, but that VLAN ID supports all of the customer's VLANs.

Customer traffic that is tagged in the normal way with appropriate VLAN IDs comes from an IEEE 802.1Q trunk port on the customer device and into a tunnel port on the service-provider edge device. The link between the customer device and the edge device is asymmetric because one end is configured as an IEEE 802.1Q

trunk port, and the other end is configured as a tunnel port. You assign the tunnel port interface to an access VLAN ID that is unique to each customer.

Figure 36: IEEE 802.1Q Tunnel Ports in a Service-Provider Network

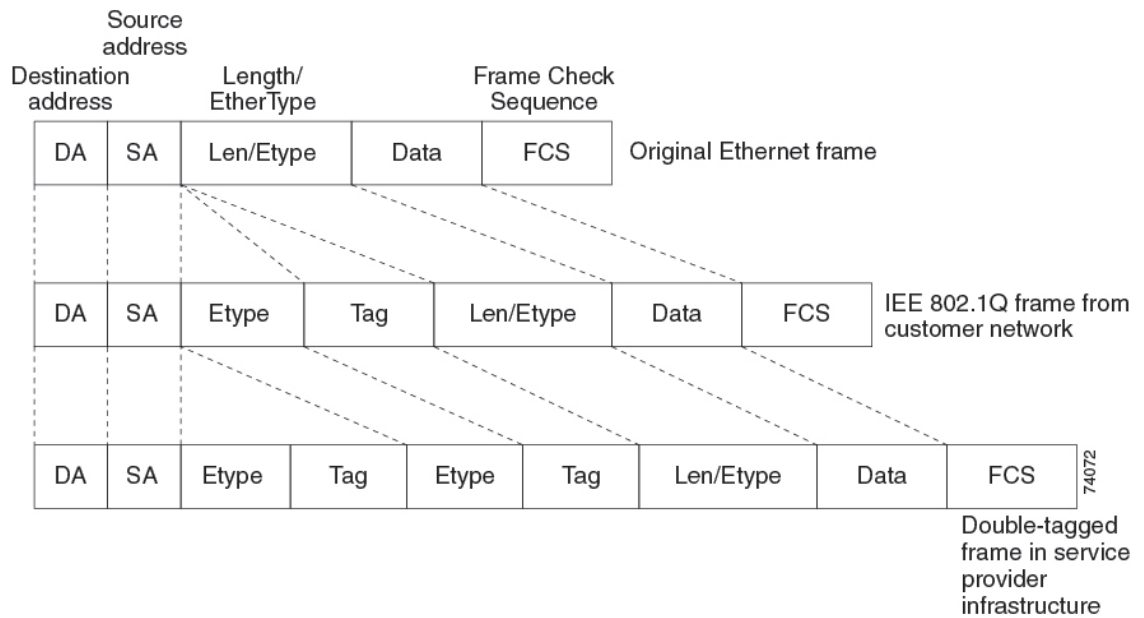


Packets coming from the customer trunk port into the tunnel port on the service-provider edge device are normally IEEE 802.1Q-tagged with the appropriate VLAN ID. The tagged packets remain intact inside the device and when they exit the trunk port into the service-provider network, they are encapsulated with another layer of an IEEE 802.1Q tag (called the metro tag) that contains the VLAN ID that is unique to the customer. The original customer IEEE 802.1Q tag is preserved in the encapsulated packet. Therefore, packets entering the service-provider network are double-tagged, with the outer (metro) tag containing the customer's access VLAN ID, and the inner VLAN ID being that of the incoming traffic.

When the double-tagged packet enters another trunk port in a service-provider core device, the outer tag is stripped as the device processes the packet. When the packet exits another trunk port on the same core device, the same metro tag is again added to the packet.

Figure 37: Original (Normal), IEEE 802.1Q, and Double-Tagged Ethernet Packet Formats

This figure shows the tag structures of the double-tagged packets.



When the packet enters the trunk port of the service-provider egress device, the outer tag is again stripped as the device internally processes the packet. However, the metro tag is not added when the packet is sent out the tunnel port on the edge device into the customer network. The packet is sent as a normal IEEE 802.1Q-tagged frame to preserve the original VLAN numbers in the customer network.

In the above network figure, Customer A was assigned VLAN 30, and Customer B was assigned VLAN 40. Packets entering the edge device tunnel ports with IEEE 802.1Q tags are double-tagged when they enter the service-provider network, with the outer tag containing VLAN ID 30 or 40, appropriately, and the inner tag containing the original VLAN number, for example, VLAN 100. Even if both Customers A and B have VLAN 100 in their networks, the traffic remains segregated within the service-provider network because the outer tag is different. Each customer controls its own VLAN numbering space, which is independent of the VLAN numbering space that is used by other customers and the VLAN numbering space that is used by the service-provider network.

At the outbound tunnel port, the original VLAN numbers on the customer’s network are recovered. It is possible to have multiple levels of tunneling and tagging, but the device supports only one level in this release.

If traffic coming from a customer network is not tagged (native VLAN frames), these packets are bridged or routed as normal packets. All packets entering the service-provider network through a tunnel port on an edge device are treated as untagged packets, whether they are untagged or already tagged with IEEE 802.1Q headers. The packets are encapsulated with the metro tag VLAN ID (set to the access VLAN of the tunnel port) when they are sent through the service-provider network on an IEEE 802.1Q trunk port. The priority field on the metro tag is set to the interface class of service (CoS) priority configured on the tunnel port. (The default is zero if none is configured.)

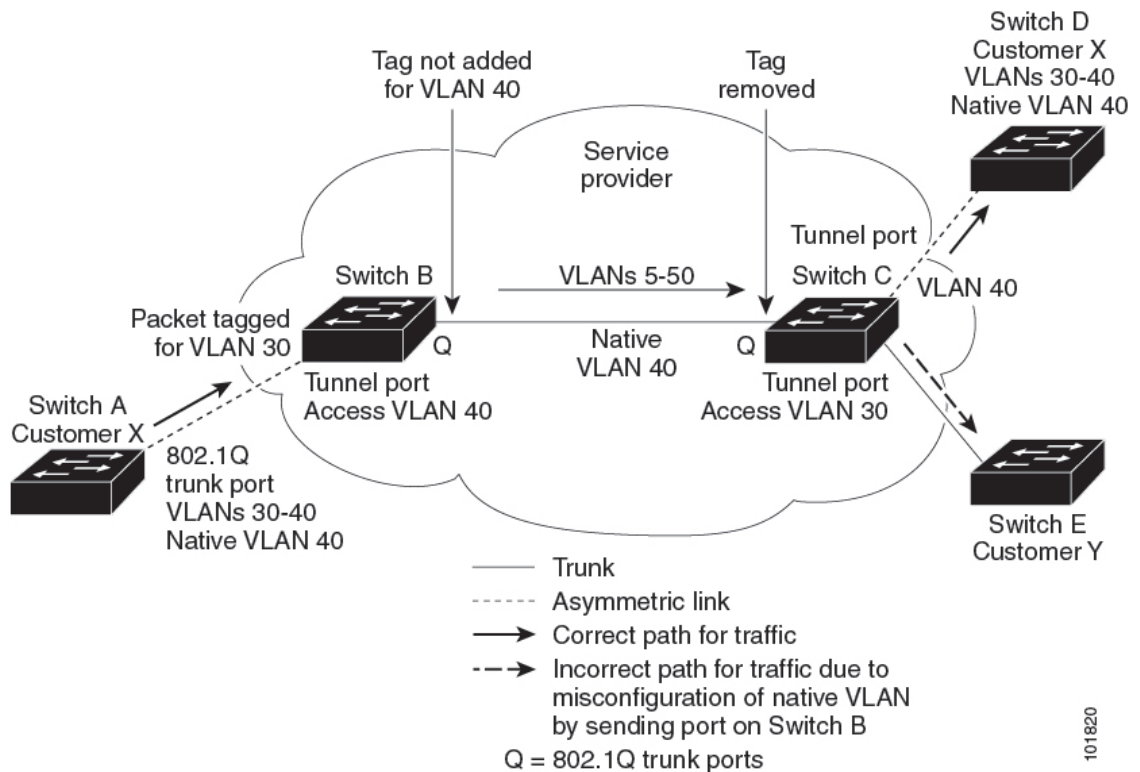
On switches, because 802.1Q tunneling is configured on a per-port basis, it does not matter whether the switch is a standalone device or a member switch. All configuration is done on the active switch.

Native VLANs

When configuring IEEE 802.1Q tunneling on an edge device, you must use IEEE 802.1Q trunk ports for sending packets into the service-provider network. However, packets going through the core of the service-provider network can be carried through IEEE 802.1Q trunks, ISL trunks, or nontrunking links. When IEEE 802.1Q trunks are used in these core devices, the native VLANs of the IEEE 802.1Q trunks must not match any native VLAN of the nontrunking (tunneling) port on the same device because traffic on the native VLAN would not be tagged on the IEEE 802.1Q sending trunk port.

In the following network figure, VLAN 40 is configured as the native VLAN for the IEEE 802.1Q trunk port from Customer X at the ingress edge switch in the service-provider network (Switch B). Switch A of Customer X sends a tagged packet on VLAN 30 to the ingress tunnel port of Switch B in the service-provider network, which belongs to access VLAN 40. Because the access VLAN of the tunnel port (VLAN 40) is the same as the native VLAN of the edge switch trunk port (VLAN 40), the metro tag is not added to tagged packets received from the tunnel port. The packet carries only the VLAN 30 tag through the service-provider network to the trunk port of the egress-edge switch (Switch C) and is misdirected through the egress switch tunnel port to Customer Y.

Figure 38: Potential Problems with IEEE 802.1Q Tunneling and Native VLANs



These are some ways to solve this problem:

- Use the **switchport trunk native vlan tag** per-port command and the **vlan dot1q tag native** global configuration command to configure the edge switches so that all packets going out an IEEE 802.1Q trunk, including the native VLAN, are tagged. If the switch is configured to tag native VLAN packets on all IEEE 802.1Q trunks, the switch drops untagged packets, and sends and receives only tagged packets.



Note `vlan dot1q tag native` global command needs to be enabled to execute the `switchport trunk native vlan tag` command.

- Ensure that the native VLAN ID on the edge switches trunk port is not within the customer VLAN range. For example, if the trunk port carries traffic of VLANs 100 to 200, assign the native VLAN a number outside that range.

System MTU

The default system MTU for traffic on the device is 1500 bytes.

You can configure 10-Gigabit and Gigabit Ethernet ports to support frames larger than 1500 bytes by using the `system mtu bytes` global configuration command.

The system MTU and system jumbo MTU values do not include the IEEE 802.1Q header. Because the IEEE 802.1Q tunneling feature increases the frame size by 4 bytes when the metro tag is added, you must configure all devices in the service-provider network to be able to process maximum frames by adding 4 bytes to the system MTU size.

For example, the device supports a maximum frame size of 1496 bytes with this configuration: The device has a system MTU value of 1500 bytes, and the `switchport mode dot1q tunnel` interface configuration command is configured on a 10-Gigabit or Gigabit Ethernet device port.

IEEE 802.1Q Tunneling and Other Features

Although IEEE 802.1Q tunneling works well for Layer 2 packet switching, there are incompatibilities between some Layer 2 features and Layer 3 switching.

- A tunnel port cannot be a routed port.
- IP routing is not supported on a VLAN that includes IEEE 802.1Q tunnel ports. Packets that are received from a tunnel port are forwarded based only on Layer 2 information. If routing is enabled on a switch virtual interface (SVI) that includes tunnel ports, untagged IP packets received from the tunnel port are recognized and routed by the switch. Customers can access the Internet through its native VLAN. If this access is not needed, you should not configure SVIs on VLANs that include tunnel ports.
- Fallback bridging is not supported on tunnel ports. Because all IEEE 802.1Q-tagged packets that are received from a tunnel port are treated as non-IP packets, if fallback bridging is enabled on VLANs that have tunnel ports that are configured, IP packets would be improperly bridged across VLANs. Therefore, you must not enable fallback bridging on VLANs with tunnel ports.
- Tunnel ports do not support IP access control lists (ACLs).
- Layer 3 quality of service (QoS) ACLs and other QoS features related to Layer 3 information are not supported on tunnel ports. MAC-based QoS is supported on tunnel ports.
- EtherChannel port groups are compatible with tunnel ports as long as the IEEE 802.1Q configuration is consistent within an EtherChannel port group.
- Port Aggregation Protocol (PAgP), Link Aggregation Control Protocol (LACP), and UniDirectional Link Detection (UDLD) are supported on IEEE 802.1Q tunnel ports.

- Dynamic Trunking Protocol (DTP) is not compatible with IEEE 802.1Q tunneling because you must manually configure asymmetric links with tunnel ports and trunk ports.
- VLAN Trunking Protocol (VTP) does not work between devices that are connected by an asymmetrical link or devices that communicate through a tunnel.
- Loopback detection is supported on IEEE 802.1Q tunnel ports.
- When a port is configured as an IEEE 802.1Q tunnel port, spanning-tree bridge protocol data unit (BPDU) filtering is automatically enabled on the interface. Cisco Discovery Protocol (CDP) is automatically disabled on the interface.



Note When you are configuring IEEE 802.1Q tunneling, the BPDU filtering configuration information is not displayed as spanning-tree BPDU filter is automatically enabled. You can verify the BPDU filter information using the **show spanning tree interface** command.

- When an IEEE 802.1Q tunnel port is configured as SPAN source, span filter must be applied for SVLAN to avoid packet loss.
- IGMP/MLD packet forwarding can be enabled on IEEE 802.1Q tunnels. This can be done by disabling IGMP/MLD snooping on the service provider network.

Default IEEE 802.1Q Tunneling Configuration

By default, IEEE 802.1Q tunneling is disabled because the default switchport mode is dynamic auto. Tagging of IEEE 802.1Q native VLAN packets on all IEEE 802.1Q trunk ports is also disabled.

Information About Custom EtherType for Cisco Catalyst 9600 Series Supervisor 2 Module

Cisco Catalyst 9600 Series Supervisor 2 Module uses the default ether type 0x8100 for dot1q and Q-in-Q encapsulations.

Custom EtherType allows configuration of the EtherType per port. The EtherTypes 0x9100 and 0x88a8 are supported using the custom EtherType feature by configuring the **switchport dot1q ether type** command on a Layer 2 interface. 802.1q (0x8100) EtherType is the default EtherType.

With the custom dot1q EtherType, you can select a nonstandard (0x9100 and 0x88a8) 2-byte EtherType in order to identify 802.1Q-tagged frames. The switch is allowed to interoperate with third-party vendors' switches that do not use the standard 0x8100 EtherType to identify 802.1Q-tagged frames. For instance, if 0x9100 EtherType is used as the custom dot1q EtherType on a particular port, incoming frames containing the EtherType 0x9100 are forwarded to the VLAN contained in the tag. Frames that arrive with the default EtherType 0x8100 are treated as untagged.

The interface can be configured with the following EtherTypes:

- 0x9100
- 0x88a8

When configuring a custom EtherType field value, consider the following:

- To use a custom EtherType field value, all the network devices in the traffic path across the network must support the custom EtherType field value.
- You can configure a custom EtherType field value on trunk ports and tunnel ports.
- You can configure a custom EtherType field value on an EtherChannel port.

Restrictions for Custom Ethertypes

- If a custom ethertype 0x9100 or 0x88a8 is configured on a port, all the custom ethertypes are accepted by the RX filter. However, only the configured ethertype will be used during TX. During egress, the configured ethertype 0x9100 or 0x88a8 is inserted and packets received with 8100 ethertypes are treated as untagged (current limitation of untagged packet on trunk port is observed).
- If a custom ethertype is configured under a physical port, all the tagged service instances under the physical port are forced to use that particular ethertype.
- Custom ethertype is not supported on IP configured or routed interfaces.
- Custom ethertype dynamic update from dot1q to tunneling, or from tunneling to dot1q is not supported.
- dot1q tunneling ethertype Canonical Frame Identifier (CFI) preservation is not supported.
- Connectivity Fault Management (CFM) with custom ethertype is not supported.
- 802.1ad is not supported for custom ethertype.
- DHCP snooping is not supported for custom ethertype.
- Each port supports only one EtherType field value. A port that is configured with a custom EtherType field value does not recognize frames that have any other EtherType field value as tagged frames. For example, a trunk port that is configured with a custom EtherType field value does not recognize the standard 0x8100 EtherType field value on 802.1Q-tagged frames and cannot put the frames into the VLAN to which they belong.

How to Configure IEEE 802.1Q Tunneling

Follow these steps to configure a port as an IEEE 802.1Q tunnel port:

Before you begin

- Always use an asymmetrical link between the customer device and the edge device, with the customer device port configured as an IEEE 802.1Q trunk port and the edge device port configured as a tunnel port.
- Assign tunnel ports only to VLANs that are used for tunneling.
- Observe configuration requirements for native VLANs and for and maximum transmission units (MTUs).

Procedure

	Command or Action	Purpose
Step 1	enable Example: Device> enable	Enables privileged EXEC mode. Enter your password if prompted.
Step 2	configure terminal Example: Device# configure terminal	Enters global configuration mode.
Step 3	interface <i>interface-id</i> Example: Device (config)# interface gigabitethernet2/0/1	Enters interface configuration mode for the interface to be configured as a tunnel port. This should be the edge port in the service-provider network that connects to the customer device. Valid interfaces include physical interfaces and port-channel logical interfaces (port channels 1 to 48).
Step 4	switchport access vlan <i>vlan-id</i> Example: Device (config-if)# switchport access vlan 2	Specifies the default VLAN, which is used if the interface stops trunking. This VLAN ID is specific to the particular customer.
Step 5	switchport mode dot1q-tunnel Example: Device (config-if)# switchport mode dot1q-tunnel	Sets the interface as an IEEE 802.1Q tunnel port. Note Use the no switchport mode dot1q-tunnel interface configuration command to return the port to the default state of dynamic desirable.
Step 6	exit Example: Device (config-if)# exit	Returns to global configuration mode.
Step 7	vlan dot1q tag native Example: Device (config)# vlan dot1q tag native	(Optional) Sets the device to enable tagging of native VLAN packets on all IEEE 802.1Q trunk ports. When not set, and a customer VLAN ID is the same as the native VLAN, the trunk port does not apply a metro tag, and packets could be sent to the wrong destination. Note Use the no vlan dot1q tag native global configuration command to disable tagging of native VLAN packets.

	Command or Action	Purpose
Step 8	end Example: Device(config)# end	Returns to privileged EXEC mode.
Step 9	Use one of the following: <ul style="list-style-type: none"> • show dot1q-tunnel • show running-config interface Example: Device# show dot1q-tunnel or Device# show running-config interface	Displays the ports that are configured for IEEE 802.1Q tunneling. Displays the ports that are in tunnel mode.
Step 10	show vlan dot1q tag native Example: Device# show vlan dot1q native	Displays IEEE 802.1Q native VLAN tagging status.
Step 11	copy running-config startup-config Example: Device# copy running-config startup-config	(Optional) Saves your entries in the configuration file.

How to Configure Custom EtherTypes

Custom EtherType allows configuration of the EtherType per port. Use the following procedure to configure custom EtherTypes on Cisco Catalyst 9600 Series Supervisor 2 Module.

Procedure

	Command or Action	Purpose
Step 1	enable Example: Device> enable	Enables privileged EXEC mode, enter your password if prompted.
Step 2	configure terminal Example: Device# configure terminal	Enters global configuration mode.
Step 3	interface interface name Example:	Enters interface configuration mode.

	Command or Action	Purpose
	Device(config)# interface FiftyGigE1/1/0/1	
Step 4	switchport mode trunk Example: Device(config-if)# switchport mode trunk	Configures the interface to operate in trunk mode.
Step 5	switchport trunk allowed vlan all Example: Device(config-if)# switchport trunk allowed vlan all	Specifies all the VLANs defined on the switch to be allowed on the interface.
Step 6	switchport dot1q EtherType <i>EtherType value</i> Example: Device(config-if)# switchport dot1q ethertype 9100	Configures a custom EtherType. Supported custom EtherTypes are 0x9100 and 0x88a8. The EtherType value should be in hex <600-FFFF>.
Step 7	end Example: Device(config-if)# end	Returns to privileged EXEC mode.
Step 8	show platform software fed switch active ifm mappings Example: Device(config-if)# show platform software fed switch active ifm mappings	Displays mapping information about the switch.
Step 9	show platform software fed switch active ifm if-id <i>IF-ID</i> Example: Device(config-if)# show platform software fed switch active ifm if-id 0x4b6	Displays mapping information about the specified interface.

Monitoring Tunneling Status

The following table describes the commands used to monitor tunneling status.

Table 21: Commands for Monitoring Tunneling

Command	Purpose
<code>show dot1q-tunnel</code>	Displays IEEE 802.1Q tunnel ports on the device.
<code>show dot1q-tunnel interface interface-id</code>	Verifies if a specific interface is a tunnel port.
<code>show vlan dot1q tag native</code>	Displays the status of native VLAN tagging on the device.

Example: Configuring an IEEE 802.1Q Tunneling Port

The following example shows how to configure an interface as a tunnel port, enable tagging of native VLAN packets, and verify the configuration. In this configuration, the VLAN ID for the customer connected to Gigabit Ethernet interface 7 on stack member 1 is VLAN 22.

```
Device(config)# interface gigabitethernet1/0/7
Device(config-if)# switchport access vlan 22
% Access VLAN does not exist. Creating vlan 22
Device(config-if)# switchport mode dot1q-tunnel
Device(config-if)# exit
Device(config)# vlan dot1q tag native
Device(config)# end
Device# show dot1q-tunnel interface gigabitethernet1/0/7
Port
-----
Gi1/0/1Port
-----
Device# show vlan dot1q tag native
dot1q native vlan tagging is enabled
```

Example: Configuring a Custom Ethertype

The following example shows how to configure a custom ethertype under an interface.

```
Device> enable
Device# configure terminal
Device(config)# interface Hu2/1/0/1
Device(config-if)# switchport mode trunk
Device(config-if)# switchport trunk allowed vlan all
Device(config-if)# switchport dot1q ethertype 88a8
Device(config-if)# end
Device# show platform software fed switch active ifm mappings
```

```
Interface          IF_ID  Inst Asic Core IFG_ID Port SubPort Mac  First Serdes Last
Serdes  Cntx LPN  GPN  Type Active
HundredGigE2/1/0/1  0x3    0  0  0    0    0    0    129  0    1
0  1  1153 NIF  Y
HundredGigE2/1/0/2  0x4b4  0  0  0    0    0    0    130  2    3
0  2  1154 NIF  Y
HundredGigE2/1/0/3  0x4b5  0  0  0    0    0    0    131  4    5
0  3  1155 NIF  Y
HundredGigE2/1/0/4  0x4b6  0  0  0    0    0    0    132  6    7
0  4  1156 NIF  Y
```

Example: Configuring a Custom Ethertype

```

HundredGigE2/1/0/5      0x4b7  0 0 0 1 0 0 133 8 9
  0 5 1157 NIF Y
HundredGigE2/1/0/6      0x4b8  0 0 0 1 0 0 134 10 11
  0 6 1158 NIF Y
HundredGigE2/1/0/7      0x4b9  0 0 0 1 0 0 135 12 13
  0 7 1159 NIF Y
HundredGigE2/1/0/8      0x4ba  0 0 0 1 0 0 136 14 15
  0 8 1160 NIF Y
HundredGigE2/1/0/9      0x4bb  0 0 1 0 0 0 137 16 17
  0 9 1161 NIF Y
HundredGigE2/1/0/10     0x4bc  0 0 1 0 0 0 138 18 19
  0 10 1162 NIF Y
HundredGigE2/1/0/11     0x4bd  0 0 1 0 0 0 139 20 21
  0 11 1163 NIF Y

```

```
Device# show platform software fed switch active ifm if-id 0x4b6
```

```

Interface IF_ID          : 0x00000000000004b6
Interface Name           : HundredGigE2/1/0/4
Interface Block Pointer  : 0x7ba00bbb5328
Interface Block State    : Ready
Interface State          : Enabled
Interface Admin mode     : Admin Up
Interface Status         : NPD
Interface Ref-Cnt        : 1
Interface Type           : ETHER
Port Type                : SWITCH PORT
Port Location            : LOCAL
Slot                     : 15
Unit                    : 0
Slot Unit                : 4
SNMP IF Index           : 183
GPN                      : 1156
EC Channel               : 0
EC Index                 : 0
QoS Trust Type          : 3 (DSCP)
Ref Count : 1 (feature Ref Counts + 1)
No Feature Reference count Present
IFM Feature Sub block information
Port Physical Subblock
  Affinity ..... [local]
  LPN ..... [4]
  GPN ..... [1156]
  Speed ..... [40GB]
  type ..... [IFM_PORT_TYPE_L2]
  MTU ..... [9222]
  ac profile ..... [IFM_AC_PROFILE_DEFAULT_CUST_ETHER]
Port Subblock [0]
  Mac port oid..... [2426]
  System port oid..... [2430]
  System port gid..... [1359]
  Ethernet port oid..... [2441]
  Voq oid..... [2428]
Platform Subblock
  Asic..... [0]
  Core..... [0]
  Asic Port..... [0]
  Asic Sub Port..... [65535]
  Ifg Id..... [0]
  Mac Num..... [132]
  First Serdes..... [6]
  Last Serdes..... [7]
  FC Mode..... [0]
  FEC Mode..... [0]

```



```

Context Id..... [0]
Port L2 Subblock
L2 Port Mode ..... [port_mode_trunk]
L2 Port Mode set..... [Yes]
Default vlan ..... [0]
Ethertype..... [88a8]
untagged port bd vlan ..... [0]
status..... [0]
ac profile ..... [IFM_AC_PROFILE_DEFAULT_CUST_ETHER]
Events Log
[2023/02/23 14:36:45.121] XCVR enable
[2023/02/23 14:36:45.122] Port mode enable
[2023/02/23 14:36:49.710] link state up
[2023/02/23 14:36:49.712] Mode Dynamic
[2023/02/23 14:36:52.720] Mode Trunk
[2023/02/23 14:36:52.725] Mode Trunk
[2023/02/23 14:44:55.090] STATE DISABLE link down
[2023/02/23 14:44:55.129] link state down
[2023/02/23 14:44:55.132] XCVR disable
[2023/02/23 14:44:55.132] Port mode disable
[2023/02/23 14:44:55.132] link state down
[2023/02/23 14:45:03.992] XCVR enable
--More--

```

```

Device# show interface Hu2/1/0/1 switchport
Name: Hu2/1/0/4
Switchport: Enabled
Administrative Mode: tunnel
Operational Mode: tunnel
Administrative Trunking Encapsulation: dot1q
Operational Trunking Encapsulation: native
Administrative Dot1q Ethertype: 0x9100
Operational Dot1q Ethertype: 0x9100
Negotiation of Trunking: Off
Access Mode VLAN: 1 (default)
Trunking Native Mode VLAN: 1 (default)
Administrative Native VLAN tagging: enabled
Voice VLAN: none
Administrative private-vlan host-association: none
Administrative private-vlan mapping: none
Administrative private-vlan trunk native VLAN: none
Administrative private-vlan trunk Native VLAN tagging: enabled
Administrative private-vlan trunk encapsulation: dot1q
Administrative private-vlan trunk normal VLANs: none
Administrative private-vlan trunk associations: none
Administrative private-vlan trunk mappings: none
Operational private-vlan: none
Trunking VLANs Enabled: ALL
Pruning VLANs Enabled: 2-1001
Capture Mode Disabled
Capture VLANs Allowed: ALL

Protected: false
Unknown unicast blocked: disabled
Unknown multicast blocked: disabled
Vepa Enabled: false
App Interface: false
Appliance trust: none

Device#

```

Feature History for IEEE 802.1Q Tunneling

This table provides release and related information for features explained in this module.

These features are available on all releases subsequent to the one they were introduced in, unless noted otherwise.

Release	Feature	Feature Information
Cisco IOS XE Gibraltar 16.11.1	IEEE 802.1Q Tunneling	The IEEE 802.1Q tunneling feature is designed for service providers who carry traffic of multiple customers across their networks and are required to maintain the VLAN and Layer 2 protocol configurations of each customer without impacting the traffic of other customers.
Cisco IOS XE Dublin 17.11.1	Custom Ethertype	This feature was introduced on Cisco Catalyst 9600X-SUP-2 module .

Use [Cisco Feature Navigator](#) to find information about platform and software image support.



CHAPTER 10

Configuring VLAN Mapping

- [Prerequisites for VLAN Mapping, on page 199](#)
- [Prerequisites for One to One VLAN Mapping, on page 199](#)
- [Restrictions for VLAN Mapping, on page 200](#)
- [Restrictions for One to One VLAN Mapping, on page 200](#)
- [About VLAN Mapping, on page 200](#)
- [Configuration Guidelines for VLAN Mapping, on page 202](#)
- [How to Configure VLAN Mapping, on page 204](#)
- [Feature History for VLAN Mapping, on page 210](#)

Prerequisites for VLAN Mapping

- By default, no VLAN mapping is configured.
- Ensure that you run the **Network Advantage** license. VLAN Mapping is supported only with the **Network Advantage** license level.
- To process control traffic consistently, either enable Layer 2 protocol tunneling (recommended), as follows:

```
!  
Device(config)# interface HundredGigE2/0/36  
Device(config-if)# switchport mode trunk  
Device(config-if)# switchport vlan mapping 20 300  
Device(config-if)# l2protocol-tunnel stp  
Device(config-if)# end
```

or insert a BPDU filter for spanning tree, as follows:

```
!  
Device(config)# interface HundredGigE1/0/1  
Device(config-if)# switchport mode trunk  
Device(config-if)# switchport vlan mapping 10 20  
Device(config-if)# spanning-tree bpdupfilter enable  
Device(config-if)# end
```

Prerequisites for One to One VLAN Mapping

- One-to-One VLAN mapping can be configured only on trunk ports and not on dynamic trunk.

- One-to-One VLAN mapping should be identical on both ports.
- S-VLAN should be created and present in the allowed VLAN list of the trunk port where One-to-One VLAN mapping is configured.

Restrictions for VLAN Mapping

- If VLAN mapping is enabled on an EtherChannel, the configuration does not apply to all member ports of the EtherChannel bundle but applies only to the EtherChannel interface.
- If VLAN mapping is enabled on an EtherChannel and a conflicting mapping translation is enabled on a member port, the configuration is rejected on the member port.
- If a port with VLAN mapping is configured as a part of EtherChannel with a conflicting mapping translation, the port cannot be a member of the port-channel.
- The member port of an EtherChannel is suspended from the EtherChannel bundle if the mode of the port is changed to anything other than 'trunk' mode.
- Default native VLANs, user-configured native VLANs, and reserved VLANs cannot be used for VLAN mapping.
- The S-VLAN used for VLAN mapping cannot be a part of any other Layer 3 configurations, EVPN, or LISP.
- PVLAN support is not available when VLAN mapping is configured.

Restrictions for One to One VLAN Mapping

- When One-to-One VLAN mapping is configured, multiple C-VLANs cannot be mapped to the same S-VLAN
- Merging of C-VLAN and S-VLAN spanning-tree topology is not supported in case of one-to-one vlan mapping.

About VLAN Mapping

In a typical deployment of VLAN mapping, you want service provider to provide a transparent switching infrastructure that includes customers' switches at the remote location as a part of local site. This allows customers to use the same VLAN ID space and run Layer 2 control protocols seamlessly across the provider network. In such scenarios, we recommend that service providers do not impose their VLAN IDs on their customers.

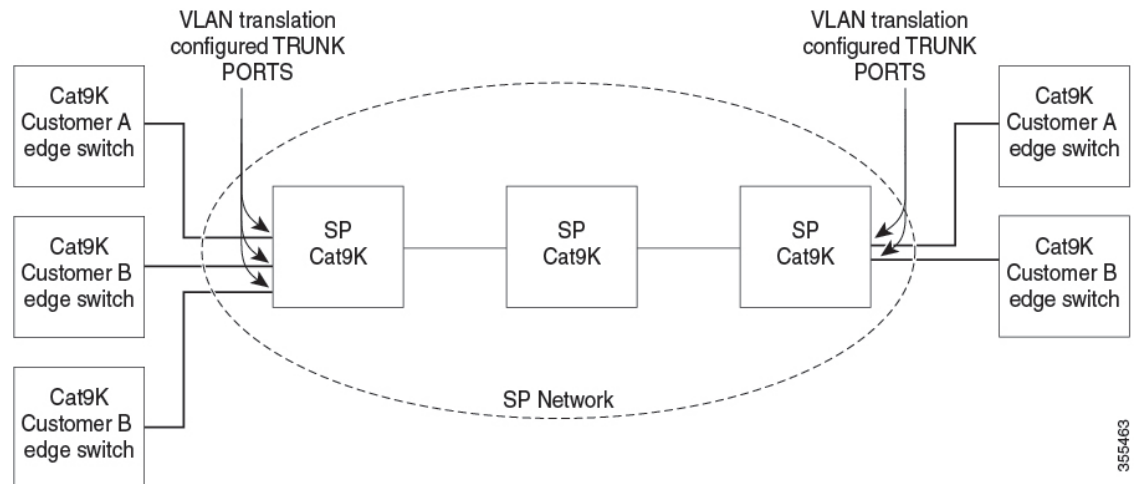
One way to establish translated VLAN IDs (S-VLANs) is to map customer VLANs to VLANs (called VLAN ID translation) on trunk ports that are connected to a customer network. Packets entering the port are mapped to service provider VLAN (S-VLAN) based on the port number and the packet's original customer VLAN-ID (C-VLAN).

Service providers' internal assignments might conflict with a customer's VLAN. To isolate customer traffic, a service provider decides to map a specific VLAN into another one while the traffic is in its cloud.

Deployment Example

In the [figure](#), the service provider provides Layer 2 VPN service to two different customers, A and B. The service provider separates the data and control traffic between the two customers and from the providers' own control traffic. The service provider network must also be transparent to the customer edge devices.

Figure 39: Example of a Service Provider with Layer 2 VPN Service



All forwarding operations on Catalyst 9000 series switch are performed using S-VLAN and not C-VLAN information because the VLAN ID is mapped to the S-VLAN on ingress.



Note When you configure features on a port for VLAN mapping, you always use the S-VLAN rather than C-VLAN.

On an interface configured for VLAN mapping, the specified C-VLAN packets are mapped to the specified S-VLAN when they enter the port. Symmetrical mapping to the customer C-VLAN occurs when packets exit the port.

The switch supports these types of VLAN mapping on trunk ports:

- One-to-one VLAN mapping.
- Selective QinQ.
- QinQ on a trunk port.

Figure 40: Mapping Customer VLANs to Service-Provider VLANs



[Figure](#) shows a topology where a customer uses the same VLANs in multiple sites on different sides of a service-provider network. The C-VLAN IDs is mapped to service-provider VLAN IDs for packet travel across the service-provider backbone. The C-VLAN IDs are retrieved at the other side of the service-provider

backbone for use in the other customer site. Configure the same set of VLAN mappings at a customer-connected port on each side of the service-provider network.

One-to-One VLAN Mapping

One-to-one VLAN mapping occurs at the ingress and egress of the port and maps the customer C-VLAN ID in the 802.1Q tag to the service-provider S-VLAN ID. Packets with VLAN IDs other than the ones with configured VLAN mapping are forwarded as normal traffic.

Selective Q-in-Q

Selective QinQ maps the specified customer VLANs entering the UNI to the specified S-VLAN ID. The S-VLAN ID is added to the incoming unmodified C-VLAN and the packet travels the service provider network double-tagged. At the egress, the S-VLAN ID is removed and the customer VLAN-ID is retained on the packet. By default, packets that do not match the specified customer VLANs are dropped.

Q-in-Q on a Trunk Port

QinQ on a trunk port maps all the customer VLANs entering the UNI to the specified S-VLAN ID. Similar to Selective QinQ, the packet is double-tagged and at the egress, the S-VLAN ID is removed.

Configuration Guidelines for VLAN Mapping



Note • By default, no VLAN mapping is configured.

Guidelines include the following:

- If the VLAN mapping is enabled on an EtherChannel, the configuration does not apply to all member ports of the EtherChannel bundle and applies only to the EtherChannel interface.
- If VLAN mapping is enabled on an EtherChannel and a conflicting mapping translation is enabled on a member port, the configuration is rejected on the member port.
- If a port with VLAN mapping is configured as a part of EtherChannel with a conflicting mapping translation, the port cannot be a member of the port-channel.
- The member port of an EtherChannel is suspended from the EtherChannel bundle if the mode of the port is changed to anything other than 'trunk' mode.
- To process control traffic consistently, either enable Layer 2 protocol tunneling (recommended), as follows:

```

!
Device(config)# interface HundredGigE1/0/1
Device(config-if)# switchport mode trunk
Device(config-if)#switchport vlan mapping 20 300
Device(config-if)# l2protocol-tunnel stp
Device(config-if)# end

```

```
Device(config)# interface HundredGigE2/0/36
Device(config-if)# switchport mode trunk
Device(config-if)# switchport vlan mapping 10 20
Device(config-if)# spanning-tree bpdufilter enable
Device(config-if)# end
```

or insert a BPDU filter for spanning tree, as follows:

```
!
Device(config)# interface HundredGigE1/0/1
Device(config-if)# switchport mode trunk
Device(config-if)# switchport vlan mapping 10 20
Device(config-if)# spanning-tree bpdufilter enable
Device(config-if)# end
```

- Default native VLANs, user-configured native VLANs, and reserved VLANs (range 1002-1005) cannot be used for VLAN mapping.
- The S-VLAN used for VLAN mapping cannot be a part of any other Layer 3 configurations like EVPN or LISP.
- PVLAN support is not available when VLAN mapping is configured.

Configuration Guidelines for One-to-One VLAN Mapping

- One-to-One VLAN mapping can be configured only on trunk ports and not on dynamic trunk.
- One-to-One VLAN mapping should be identical on both ports.
- S-VLAN should be created and present in the allowed VLAN list of the trunk port where One-to-One VLAN mapping is configured.
- When One-to-One VLAN mapping is configured, multiple C-VLANs cannot be mapped to the same S-VLAN.
- Merging of C-VLAN and S-VLAN spanning-tree topology is not supported in case of one-to-one VLAN mapping.

Configuration Guidelines for Selective Q-in-Q

- S-VLAN should be created and present in the allowed VLAN list of the trunk port where Selective Q-in-Q is configured.
- When Selective Q-in-Q is configured, the device supports Layer 2 protocol tunneling for CDP, STP, LLDP, and VTP. For emulated point-to-point network topologies, it also supports PAgP, LACP, and UDLD protocols.
- IP routing is not supported on Selective Q-in-Q enabled ports.
- IPSG is not supported on Selective Q-in-Q enabled ports.
- The tagging of native VLAN packets and selective QinQ ports is mutually exclusive and cannot be supported together on the same port. If the native VLAN tagging global command is enabled on the switch, you should disable the tagging of native VLAN packets on selective QinQ enabled ports using the command **no switchport trunk native vlan tag** command.

Configuration Guidelines for Q-in-Q on a Trunk Port

- S-VLAN should be created and present in the allowed VLAN list of the trunk port where Q-in-Q on a trunk port is configured.
- When Q-in-Q on a trunk port is configured, the device supports Layer 2 protocol tunneling for CDP, STP, LLDP, and VTP. For emulated point-to-point network topologies, it also supports PAgP, LACP, and UDLD protocols.
- Ingress and egress SPAN, and RSPAN are supported on trunk ports with QinQ enabled.
- When QinQ is enabled, the SPAN filtering can be enabled to monitor only the traffic on the mapped VLAN, i.e. S-VLANs.
- IGMP snooping is not supported on the C-VLAN.
- The tagging of native VLAN packets and QinQ on a trunk port are mutually exclusive and cannot be supported together on the same port. If the native VLAN tagging global command is enabled on the switch, you should disable the tagging of native VLAN packets on the QinQ enabled trunk ports using the command **no switchport trunk native vlan tag** command.

How to Configure VLAN Mapping

The following sections provide information about configuring VLAN mapping:

One-to-One VLAN Mapping



Note VLAN Mapping is supported only with the **network-advantage** license level.

To configure one-to-one VLAN mapping to map a customer VLAN ID to a service-provider VLAN ID, perform this task:

Procedure

	Command or Action	Purpose
Step 1	enable Example: Device> enable	Enables privileged EXEC mode. Enter your password if prompted.
Step 2	configure terminal Example: Device# configure terminal	Enters global configuration mode.
Step 3	interface <i>interface-id</i> Example:	Enters interface configuration mode for the interface that is connected to the service-provider network. You can enter a

	Command or Action	Purpose
	Device(config)# interface gigabitethernet1/0/1	physical interface or an EtherChannel port channel.
Step 4	switchport mode trunk Example: Device(config-if)# switchport mode trunk	Configures the interface as a trunk port.
Step 5	switchport vlan mapping <i>vlan-id</i> <i>translated-id</i> Example: Device(config-if)# switchport vlan mapping 2 102	Enters the VLAN IDs to be mapped: <ul style="list-style-type: none"> vlan-id—the customer VLAN ID (C-VLAN) entering the switch from the customer network. The range is from 1 to 4094. translated-id—the assigned service-provider VLAN ID (S-VLAN). The range is from 1 to 4094. <p>By default, the packets with VLAN IDs other than the ones with configured VLAN mapping are forwarded as normal traffic.</p>
Step 6	exit Example: Device(config-if)# exit	Returns to global configuration mode.
Step 7	spanning-tree bpdufilter enable Example: Device(config)# spanning-tree bpdufilter enable	Inserts a BPDU filter for spanning tree. Note To process control traffic consistently, either enable Layer 2 protocol tunneling (recommended) or insert a BPDU filter for spanning tree.
Step 8	end Example: Device(config)# end	Returns to privileged EXEC mode.
Step 9	show vlan mapping Example: Device# show vlan mapping	Verifies the configuration.
Step 10	copy running-config startup-config Example: Device# copy running-config startup-config	(Optional) Saves your entries in the configuration file.

Example

Use **no switchport vlan mapping** command to remove the VLAN mapping information. Entering **no switchport vlan mapping all** command deletes all mapping configurations.

This example shows how to map VLAN IDs 2 to 6 in the customer network to VLANs 101 to 105 in the service-provider network (Figure 3-5). You configure the same VLAN mapping commands for a port in Switch A and Switch B; the traffic on all other VLAN IDs is forwarded as normal traffic.

```
Device> enable
Device# configure terminal
Device(config)# interface gigabiethernet0/1
Device(config-if)# switchport vlan mapping 2 101
Device(config-if)# switchport vlan mapping 3 102
Device(config-if)# switchport vlan mapping 4 103
Device(config-if)# switchport vlan mapping 5 104
Device(config-if)# switchport vlan mapping 6 105
Device(config-if)# exit
```

In the previous example, at the ingress of the service-provider network, VLAN IDs 2 to 6 in the customer network are mapped to VLANs 101 to 105, in the service provider network. At the egress of the service provider network, VLANs 101 to 105 in the service provider network are mapped to VLAN IDs 2 to 6, in the customer network.



Note Packets with VLAN IDs other than the ones with configured VLAN Mapping are forwarded as normal traffic.

Use **show vlan mapping** command to view information about configured vlans.

```
Device> enable
Device# configure terminal
Device(config)# show vlan mapping
Total no of vlan mappings configured: 1
Interface Po5:
VLANs on wire          Translated    VLAN Operation
-----
20                      30           1-to-1
```

Selective Q-in-Q on a Trunk Port

To configure VLAN mapping for selective Q-in-Q on a trunk port, perform this task:



Note You cannot configure one-to-one mapping and selective Q-in-Q on the same interface.

Procedure

	Command or Action	Purpose
Step 1	enable Example:	Enables privileged EXEC mode. Enter your password if prompted.

	Command or Action	Purpose
	Device> enable	
Step 2	configure terminal Example: Device# configure terminal	Enters global configuration mode.
Step 3	interface <i>interface-id</i> Example: Device(config)# interface gigabitethernet1/0/1	Enters interface configuration mode for the interface that is connected to the service-provider network. You can enter a physical interface or an EtherChannel port channel.
Step 4	switchport mode trunk Example: Device(config-if)# switchport mode trunk	Configures the interface as a trunk port.
Step 5	switchport vlan mapping <i>vlan-id</i> dot1q-tunnel <i>outer vlan-id</i> Example: Device(config-if)# switchport vlan mapping 16 dot1q-tunnel 64	Enters the VLAN IDs to be mapped: <ul style="list-style-type: none"> • <i>vlan-id</i> —the customer VLAN ID (C-VLAN) entering the switch from the customer network. The range is from 1 to 4094. You can enter a string of VLAN-IDs. • <i>outer-vlan-id</i> —The outer VLAN ID (S-VLAN) of the service provider network. The range is from 1 to 4094. <p>Use the no form of this command to remove the VLAN mapping configuration. Entering the no switchport vlan mapping all command deletes all mapping configurations.</p>
Step 6	switchport vlan mapping default dot1q-tunnel <i>vlan-id</i> Example: Device(config-if)# switchport vlan mapping default dot1q-tunnel 22	Specifies that all unmapped packets on the port are forwarded with the specified S-VLAN. By default, packets that do not match the mapped VLANs, are dropped. Untagged traffic are forwarded without dropping.
Step 7	exit Example: Device(config-if)# exit	Returns to global configuration mode.
Step 8	spanning-tree bpdupfilter enable Example:	Inserts a BPDU filter for spanning tree.

	Command or Action	Purpose
	Device (config) # spanning-tree bpdupfilter enable	Note To process control traffic consistently, either enable Layer 2 protocol tunneling (recommended) or insert a BPDU filter for spanning tree.
Step 9	end Example: Device (config) # end	Returns to privileged EXEC mode.
Step 10	show interfaces interface-id vlan mapping Example: Device# show interfaces gigabitethernet1/0/1 vlan mapping	Verifies the configuration.
Step 11	copy running-config startup-config Example: Device# copy running-config startup-config	(Optional) Saves your entries in the configuration file.

Example

This example shows how to configure selective QinQ mapping on the port so that traffic with a C-VLAN ID of 2 to 5 enters the switch with an S-VLAN ID of 100. By default, the traffic of any other VLAN ID is dropped.

```
Device (config) # interface GigabitEthernet0/1
Device (config-if) # switchport vlan mapping 2-5 dot1q-tunnel 100
Device (config-if) # exit
```

This example shows how to configure selective QinQ mapping on the port so that traffic with a C-VLAN ID of 2 to 5 enters the switch with an S-VLAN ID of 100. The traffic of any other VLAN ID is forwarded with the S-VLAN ID of 200.

```
Device (config) # interface GigabitEthernet0/1
Device (config-if) # switchport vlan mapping 2-5 dot1q-tunnel 100
Device (config-if) # switchport vlan mapping default dot1q-tunnel 200
Device (config-if) # exit
```

```
Device# show vlan mapping
Total no of vlan mappings configured: 5
Interface Hu1/0/50:
VLANs on wire          Translated VLAN      Operation
-----
2-5                    100                  selective QinQ
*                      200                  default QinQ
```

Q-in-Q on a Trunk Port

To configure VLAN mapping for Q-in-Q on a trunk port, perform this task:

Procedure

	Command or Action	Purpose
Step 1	enable Example: Device> enable	Enables privileged EXEC mode. Enter your password if prompted.
Step 2	configure terminal Example: Device# configure terminal	Enters global configuration mode.
Step 3	interface <i>interface-id</i> Example: Device(config)# interface gigabitethernet1/0/1	Enters interface configuration mode for the interface that is connected to the service-provider network. You can enter a physical interface or an EtherChannel port channel.
Step 4	switchport mode trunk Example: Device(config-if)# switchport mode trunk	Configures the interface as a trunk port.
Step 5	switchport vlan mapping default dot1q-tunnel <i>vlan-id</i> Example: Device(config-if)# switchport vlan mapping default dot1q-tunnel 16	Specifies that all unmapped C-VLAN packets on the port are forwarded with the specified S-VLAN.
Step 6	exit Example: Device(config-if)# exit	Returns to global configuration mode.
Step 7	spanning-tree bpdudfilter enable Example: Device(config)# spanning-tree bpdudfilter enable	Inserts a BPDU filter for spanning tree. Note To process control traffic consistently, either enable Layer 2 protocol tunneling (recommended) or insert a BPDU filter for spanning tree.
Step 8	end Example: Device(config)# end	Returns to privileged EXEC mode.

	Command or Action	Purpose
Step 9	show interfaces <i>interface-id</i> vlan mapping Example: Device# show interfaces gigabitethernet1/0/1 vlan mapping	Verifies the configuration.
Step 10	copy running-config startup-config Example: Device# copy running-config startup-config	(Optional) Saves your entries in the configuration file.

Example

This example shows how to configure QinQ mapping on the port so that traffic of any VLAN ID is forwarded with the S-VLAN ID of 200.

```
Device(config)# interface gigabitethernet0/1
Device(config-if)# switchport vlan mapping default dot1q-tunnel 200
Device(config-if)# exit
```

Feature History for VLAN Mapping

This table provides release and related information for features explained in this module.

These features are available on all releases subsequent to the one they were introduced in, unless noted otherwise.

Release	Feature	Feature Information
Cisco IOS XE Gibraltar 16.11.1	One-to-One VLAN mapping	One-to-One VLAN mapping allows to map customer VLANs to service-provider VLANs on trunk ports that are connected to a customer network.
Cisco IOS XE Gibraltar 16.11.1	Selective Q-in-Q	Support for selective Q-in-Q was introduced
	Q-in-Q on a Trunk Port	Support for Q-in-Q on a trunk port was introduced

Use Cisco Feature Navigator to find information about platform and software image support. To access Cisco Feature Navigator, go to <http://www.cisco.com/go/cfn>.



CHAPTER 11

Troubleshooting Layer 2

- [Overview](#), on page 211
- [Support Articles](#), on page 211
- [Feedback Request](#), on page 212
- [Disclaimer and Caution](#), on page 212

Overview

This chapter provides links to documents authored by Cisco subject matter experts (SMEs). They aim to help you resolve technical issues without requiring a support ticket. If these documents are unable to resolve your issue, we recommend visiting the applicable [Cisco Community](#). There is a wealth of information and advice available from fellow Cisco customers who may have experienced this issue already and provided a solution. If you are not able to find a resolution on the Community, it may be best that you raise a support ticket at [Cisco Support](#). In cases where a support ticket has to be raised, these documents provide guidance about the data that should be collected and added to the support ticket. Specify the support document you referred, and TAC can create an improvement request with the document owner.

Support Articles

The documents in this section were created using specific software and hardware listed in the Components Used section of each article. However, this does not mean that they are limited to what is listed in Components Used, and generally remain relevant for later versions of software and hardware. Note that there could be some changes in the software or hardware that can cause commands to stop working, the syntax to change, or GUIs and CLIs to look different from one release to another.

The following are the support articles associated with this technology:

Document	Description
Configure Verify Troubleshoot QinQ and L2PT on Catalyst 9000 Switches	This document describes how to configure, verify and troubleshoot 802.1Q tunnels (QinQ) and Layer 2 Protocol Tunneling (L2PT) on the Catalyst 9000 family of switches that run Cisco IOS XE software.

Document	Description
Troubleshoot EtherChannels on Catalyst 9000 Switches	This document describes how to understand and troubleshoot EtherChannels on Catalyst 9000 Series Switches.

Feedback Request

Your input helps. A key aspect to improving these support documents is customer feedback. Note that these documents are owned and maintained by multiple teams within Cisco. If you find an issue specific to the document (unclear, confusing, information missing, etc):

- Provide feedback using the **Feedback** button located at the right panel of the corresponding article. The document owner will be notified, and will either update the article, or flag it for removal.
- Include information regarding the section, area, or issue you had with the document and what could be improved. Provide as much detail as possible.

Disclaimer and Caution

The information in this document was created from the devices in a specific lab environment. All of the devices used in this document started with a cleared (default) configuration. If your network is live, ensure that you understand the potential impact of any command.