



Layer 2 Configuration Guide, Cisco IOS XE Bengaluru 17.5.x (Catalyst 9500 Switches)

First Published: 2021-04-01

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 NONINFRINGEMENT 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: www.cisco.com/go/trademarks. 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)

© 2021 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 14

(Optional) Configuring a Secondary Root Device 15

(Optional) Configuring Port Priority 16

(Optional) Configuring Path Cost 17

(Optional) Configuring the Device Priority of a VLAN 18

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

CHAPTER 2

Configuring Loop Detection Guard 25

| | |
|---|----|
| Restrictions for Loop Detection Guard | 25 |
| Information About Loop Detection Guard | 25 |
| Interaction of Loop Detection Guard with Other Features | 27 |
| Spanning Tree Protocol and Loop Detection Guard | 27 |
| VLANs and Loop Detection Guard | 27 |
| How to Configure Loop Detection Guard | 28 |
| Enabling Loop Detection Guard and Error-Disabling the Required Port | 28 |
| Additional References for Configuring Loop Detection Guard | 29 |
| Feature History for Loop Detection Guard | 30 |

CHAPTER 3

Configuring Multiple Spanning-Tree Protocol 31

| | |
|---|----|
| Prerequisites for Multiple Spanning Tree Protocol | 31 |
| Restrictions for MSTP | 31 |
| Information About MSTP | 32 |
| Multiple Spanning Tree Protocol Configuration | 32 |
| Multiple Spanning Tree Protocol Configuration Guidelines | 32 |
| Root Switch Configuration | 33 |
| Multiple Spanning-Tree Regions | 33 |
| Internal Spanning Tree, Common and Internal Spanning Tree, and Common Spanning Tree | 34 |
| Operations Within an Multiple Spanning Tree Region | 34 |
| Operations Between Multiple Spanning Tree Regions | 35 |
| IEEE 802.1s Terminology | 35 |
| Illustration of Multiple Spanning Tree Regions | 36 |
| Hop Count | 36 |
| Boundary Ports | 37 |

| | |
|---|-----------|
| IEEE 802.1s Implementation | 37 |
| Port Role Naming Change | 37 |
| Interoperation Between Legacy and Standard Devices | 38 |
| Detecting Unidirectional Link Failure | 39 |
| Multiple Spanning Tree Protocol and Switch Stacks | 39 |
| Interoperability with IEEE 802.1D Spanning Tree Protocol | 39 |
| Rapid Spanning Tree Protocol Overview | 40 |
| Port Roles and the Active Topology | 40 |
| Rapid Convergence | 41 |
| Synchronization of Port Roles | 42 |
| Bridge Protocol Data Unit Format and Processing | 43 |
| Topology Changes | 44 |
| Protocol Migration Process | 45 |
| Default Multiple Spanning Tree Protocol Configuration | 45 |
| How to Configure MSTP and MSTP Parameters | 46 |
| Specifying the Multiple Spanning Tree Region Configuration and Enabling Multiple Spanning Tree Protocol | 46 |
| (Optional) Configuring the Root Device | 47 |
| (Optional) Configuring a Secondary Root Device | 48 |
| (Optional) Configuring Port Priority | 49 |
| (Optional) Configuring Path Cost | 50 |
| (Optional) Configuring the Device Priority | 51 |
| (Optional) Configuring the Hello Time | 53 |
| Configuring the Forwarding-Delay Time | 53 |
| Configuring the Maximum-Aging Time | 54 |
| (Optional) Configuring the Maximum-Hop Count | 55 |
| (Optional) Specifying the Link Type to Ensure Rapid Transitions | 55 |
| (Optional) Designating the Neighbor Type | 56 |
| Restarting the Protocol Migration Process | 57 |
| Feature History for MSTP | 58 |
| CHAPTER 4 | |
| Configuring Optional Spanning-Tree Features | 61 |
| Information About Optional Spanning-Tree Features | 61 |
| PortFast | 61 |

| | |
|--|----|
| Spanning Tree Protocol PortFast Port Types | 62 |
| Bridge Protocol Data Unit Guard | 63 |
| Bridge Protocol Data Unit Filtering | 63 |
| Bridge Assurance | 63 |
| Guidelines for Configuring Bridge Assurance | 64 |
| UplinkFast | 65 |
| Cross-Stack UplinkFast | 67 |
| How Cross-Stack UplinkFast Works | 67 |
| Events That Cause Fast Convergence | 68 |
| BackboneFast | 69 |
| EtherChannel Guard | 72 |
| Root Guard | 72 |
| Loop Guard | 73 |
| How to Configure Optional Spanning-Tree Features | 74 |
| (Optional) Enabling PortFast | 74 |
| Enabling PortFast Port Types | 75 |
| Configuring the PortFast Default State Globally | 75 |
| Configuring a PortFast Edge Port on a Specified Interface | 76 |
| Configuring a PortFast Network Port on a Specified Interface | 78 |
| (Optional) Enabling Bridge Protocol Data Unit Guard | 79 |
| Enabling BPDU Filtering | 81 |
| Configuring Bridge Assurance | 83 |
| (Optional) Enabling UplinkFast for Use with Redundant Links | 85 |
| (Optional) Disabling UplinkFast | 86 |
| (Optional) Enabling BackboneFast | 86 |
| (Optional) Enabling EtherChannel Guard | 87 |
| (Optional) Enabling Root Guard | 88 |
| (Optional) Enabling Loop Guard | 89 |
| Monitoring the Spanning-Tree Status | 90 |
| Additional References for Optional Spanning Tree Features | 90 |
| Feature History for Optional Spanning Tree Features | 90 |

CHAPTER 5
Configuring EtherChannels 93

| | |
|--------------------------------|----|
| Restrictions for EtherChannels | 93 |
|--------------------------------|----|

| | |
|--|-----|
| Information About EtherChannels | 93 |
| EtherChannel Overview | 93 |
| Channel Groups and Port-Channel Interfaces | 94 |
| Port Aggregation Protocol | 96 |
| Port Aggregation Protocol Modes | 96 |
| Port Aggregation Protocol Learn Method and Priority | 97 |
| Port Aggregation Protocol Interaction with Other Features | 97 |
| Link Aggregation Control Protocol | 98 |
| Link Aggregation Control Protocol Modes | 98 |
| Link Aggregation Control Protocol and Link Redundancy | 99 |
| Link Aggregation Control Protocol Interaction with Other Features | 99 |
| Link Aggregation Control Protocol Interaction with Other Features 1:1 Redundancy | 99 |
| EtherChannel On Mode | 99 |
| Load-Balancing and Forwarding Methods | 100 |
| MAC Address Forwarding | 100 |
| IP Address Forwarding | 100 |
| VLAN ID based Forwarding | 101 |
| Load-Balancing Advantages | 101 |
| EtherChannel and Switch Stacks | 102 |
| Switch Stack and Port Aggregation Protocol | 102 |
| Switch Stacks and Link Aggregation Control Protocol | 102 |
| Default EtherChannel Configuration | 102 |
| EtherChannel Configuration Guidelines | 103 |
| Layer 2 EtherChannel Configuration Guidelines | 104 |
| Layer 3 EtherChannel Configuration Guidelines | 104 |
| Auto-LAG | 104 |
| Auto-LAG Configuration Guidelines | 105 |
| How to Configure EtherChannels | 105 |
| Configuring Layer 2 EtherChannels | 105 |
| Configuring Layer 3 EtherChannels | 107 |
| Configuring EtherChannel Load-Balancing | 109 |
| Configuring EtherChannel Extended Load-Balancing | 111 |
| (Optional) Configuring the Port Aggregation Protocol Learn Method and Priority | 112 |
| Configuring Link Aggregation Control Protocol Hot-Standby Ports | 113 |

| | |
|--|-----|
| (Optional) Configuring the Link Aggregation Control Protocol Max Bundle | 114 |
| Configuring Link Aggregation Control Protocol Port-Channel Standalone Disable | 114 |
| Configuring the Link Aggregation Control Protocol Port Channel Min-Links | 115 |
| (Optional) Configuring the Link Aggregation Control Protocol System Priority | 116 |
| (Optional) Configuring the Link Aggregation Control Protocol Port Priority | 117 |
| Configuring Link Aggregation Control Protocol 1:1 Redundancy | 118 |
| Configuring Configuring Link Aggregation Control Protocol 1:1 Redundancy Fast Rate Timer | 119 |
| Configuring Auto-LAG Globally | 119 |
| Configuring Auto-LAG on a Port Interface | 120 |
| Configuring Persistence with Auto-LAG | 121 |
| Monitoring EtherChannel, Port Aggregation Protocol, and Link Aggregation Control Protocol Status | 122 |
| Configuration Examples for EtherChannels | 122 |
| Example: Configuring Layer 2 EtherChannels | 122 |
| Example: Configuring Layer 3 EtherChannels | 123 |
| Example: Configuring Link Aggregation Control Protocol Hot-Standby Ports | 124 |
| Example: Configuring Link Aggregation Control Protocol 1:1 Redundancy | 124 |
| Example: Configuring Auto LAG | 124 |
| Additional References for EtherChannels | 125 |
| Feature History for EtherChannels | 126 |

CHAPTER 6

| | |
|--|------------|
| Configuring Precision Time Protocol (PTP) | 127 |
| Restrictions and Limitations for PTP | 127 |
| Information About Precision Time Protocol | 128 |
| Ethernet Switches and Delays | 128 |
| Message-Based Synchronisation | 128 |
| Precision Time Protocol Version 2 Message Types | 129 |
| Precision Time Protocol Event Message Sequences | 130 |
| End-to-End Delay Mechanism | 130 |
| Peer-to-Peer Delay Mechanism | 131 |
| Synchronizing the Local Clock | 132 |
| Best Master Clock Algorithm | 133 |
| Precision Time Protocol Clocks | 133 |
| Precision Time Protocol Profiles | 135 |
| Default Profile | 135 |

| | |
|---|-----|
| Precision Time Protocol on an EtherChannel Interface | 136 |
| How to Configure Precision Time Protocol | 136 |
| Configuring Precision Time Protocol Default Profile | 136 |
| Configuring Precision Time Protocol on Layer 2 interface | 138 |
| Configuring Precision Time Protocol on SVI or Layer 3 Interface | 139 |
| Configuring the Source IP on Precision Time Protocol | 139 |
| Configuring PTP Timers | 140 |
| Configuring the Values of Precision Time Protocol Clocks | 141 |
| Configuration Examples for PTP | 142 |
| Example: Layer 2 and Layer 3 PTP Configurations | 142 |
| Example: Configuring Precision Time Protocol on an EtherChannel Interface | 146 |
| Feature History for Precision Time Protocol | 149 |

CHAPTER 7

| | |
|---|------------|
| Configuring Generalized Precision Time Protocol | 151 |
| Restrictions for Generalized Precision Time Protocol over Layer 3 Unicast | 151 |
| Information About Generalized Precision Time Protocol | 151 |
| Generalized Precision Time Protocol on an EtherChannel Interface | 152 |
| Generalized Precision Time Protocol over Layer 3 Unicast | 153 |
| How to Configure Generalized Precision Time Protocol | 154 |
| Enabling Generalized Precision Time Protocol | 155 |
| Enabling Generalized Precision Time Protocol on an Interface | 155 |
| Configuring the Values of Precision Time Protocol Clocks | 156 |
| Configuring Generalized Precision Time Protocol over Layer 3 Unicast | 156 |
| Monitoring Generalized Precision Time Protocol | 158 |
| Verifying Generalized Precision Time Protocol over Layer 3 Unicast Configuration | 158 |
| Configuration Examples for Generalized Precision Time Protocol | 158 |
| Example: Verifying Generalized Precision Time Protocol | 159 |
| Example: Verifying Generalized Precision Time Protocol on an EtherChannel Interface | 161 |
| Example: Configuring Generalized Precision Time Protocol over Layer 3 Unicast | 164 |
| Feature History for Generalized Precision Time Protocol | 165 |

CHAPTER 8

| | |
|--|------------|
| Configuring Resilient Ethernet Protocol | 167 |
| Information About Resilient Ethernet Protocol | 167 |
| Link Integrity | 169 |

| | |
|--|-----|
| Fast Convergence | 170 |
| VLAN Load Balancing | 170 |
| Spanning Tree Interaction | 171 |
| Resilient Ethernet Protocol Ports | 172 |
| How to Configure Resilient Ethernet Protocol | 172 |
| Default Resilient Ethernet Protocol Configuration | 172 |
| Resilient Ethernet Protocol Configuration Guidelines | 173 |
| Configuring Resilient Ethernet Protocol Administrative VLAN | 174 |
| Configuring a REP Interface | 175 |
| Setting Manual Preemption for VLAN Load Balancing | 179 |
| Configuring Simple Network Management Protocol Traps for Resilient Ethernet Protocol | 180 |
| Monitoring Resilient Ethernet Protocol Configurations | 181 |
| Additional References for Resilient Ethernet Protocol | 182 |
| Feature History for Resilient Ethernet Protocol | 182 |

CHAPTER 9
Configuring UniDirectional Link Detection 185

| | |
|---|-----|
| Restrictions for Configuring UniDirectional Link Detection | 185 |
| Information About UniDirectional Link Detection | 185 |
| Fast UDLD | 186 |
| Modes of Operation | 186 |
| Normal Mode | 186 |
| Aggressive Mode | 186 |
| Methods to Detect Unidirectional Links | 187 |
| Neighbor Database Maintenance | 187 |
| Event-Driven Detection and Echoing | 187 |
| UniDirectional Link Detection Reset Options | 188 |
| Default UniDirectional Link Detection Configuration | 188 |
| How to Configure UDLD | 188 |
| Enabling UniDirectional Link Detection Globally | 189 |
| Enabling UniDirectional Link Detection on an Interface | 190 |
| Enabling Fast UniDirectional Link Detection on an Interface | 190 |
| Enabling Fast UniDirectional Link Detection Error Reporting | 191 |
| Disabling UniDirectional Link Detection on Fiber-Optic LAN Interfaces | 192 |
| Monitoring and Maintaining UniDirectional Link Detection | 192 |

| | |
|---|-----|
| Console Error Messages For Fast UniDirectional Link Detection | 193 |
| Additional References for UniDirectional Link Detection | 193 |
| Feature History for UniDirectional Link Detection | 193 |

CHAPTER 10

Configuring Layer 2 Protocol Tunneling 195

| | |
|---|-----|
| Prerequisites for Layer 2 Protocol Tunneling | 195 |
| Restrictions for Layer 2 Protocol Tunneling | 195 |
| Information About Layer 2 Protocol Tunneling | 195 |
| Layer 2 Protocol Tunneling Overview | 196 |
| Layer 2 Protocol Tunneling on Ports | 197 |
| Layer 2 Protocol Tunneling for EtherChannels | 198 |
| Default Layer 2 Protocol Tunneling Configuration | 199 |
| How to Configure Layer 2 Protocol Tunneling | 199 |
| Configuring Layer 2 Protocol Tunneling | 199 |
| How to Configure Layer 2 Protocol Tunneling for EtherChannels | 202 |
| Configuring the SP Edge Switch | 202 |
| Configuring the Customer Device | 205 |
| Configuration Examples for Layer 2 Protocol Tunneling | 207 |
| Example: Configuring Layer 2 Protocol Tunneling | 207 |
| Examples: Configuring the SP Edge and Customer Switches | 207 |
| Monitoring Tunneling Status | 209 |
| Feature History for Layer 2 Protocol Tunneling | 209 |

CHAPTER 11

Configuring IEEE 802.1Q Tunneling 211

| | |
|--|-----|
| Information About IEEE 802.1Q Tunneling | 211 |
| IEEE 802.1Q Tunnel Ports in a Service Provider Network | 211 |
| Native VLANs | 214 |
| System MTU | 215 |
| IEEE 802.1Q Tunneling and Other Features | 215 |
| Default IEEE 802.1Q Tunneling Configuration | 216 |
| How to Configure IEEE 802.1Q Tunneling | 216 |
| Monitoring Tunneling Status | 218 |
| Example: Configuring an IEEE 802.1Q Tunneling Port | 218 |
| Feature History for IEEE 802.1Q Tunneling | 219 |

CHAPTER 12**Configuring VLAN Mapping 221**

- Prerequisites for VLAN Mapping 221
- Prerequisites for One to One VLAN Mapping 222
- Restrictions for VLAN Mapping 222
- Restrictions for One to One VLAN Mapping 222
- About VLAN Mapping 223
 - One-to-One VLAN Mapping 224
 - Selective Q-in-Q 224
 - Q-in-Q on a Trunk Port 225
- Configuration Guidelines for VLAN Mapping 225
 - Configuration Guidelines for One-to-One VLAN Mapping 226
 - Configuration Guidelines for Selective Q-in-Q 226
 - Configuration Guidelines for Q-in-Q on a Trunk Port 226
- How to Configure VLAN Mapping 227
 - One-to-One VLAN Mapping 227
 - Selective Q-in-Q on a Trunk Port 229
 - Q-in-Q on a Trunk Port 232
- Feature History for VLAN Mapping 233

CHAPTER 13**Configuring Audio Video Bridging 235**

- Restrictions for Audio Video Bridging Networks 235
- Introduction to Audio Video Bridging Networks 235
 - Information about Audio Video Bridging 235
 - Audio Video Bridging License Level 236
 - Benefits of Audio Video Bridging 236
 - Components of Audio Video Bridging Network 236
 - Supported SKUs for Audio Video Bridging 238
 - Information About Generalized Precision Time Protocol 238
 - Information about Multiple Stream Reservation Protocol (MSRP) 238
 - Functions of Multiple Stream Reservation Protocol 239
 - Information about Hierarchical QoS 239
 - Information about Multiple VLAN Registration Protocol (MVRP) 240
- Configuring the AVB Network 240

| | |
|---|-----|
| Configuring AVB | 240 |
| Enabling Audio Video Bridging | 240 |
| Configuring Audio Video Bridging | 241 |
| Configuring gPTP | 243 |
| Enabling gPTP | 243 |
| Configuring the Values of Precision Time Protocol Clocks | 244 |
| Configuring HQoS | 245 |
| Enabling HQoS | 245 |
| Hierarchical QoS Policy Formats | 245 |
| Configuring MVRP | 246 |
| Enabling Multiple VLAN Registration Protocol | 246 |
| Configuring Multiple VLAN Registration Protocol on an Interface | 247 |
| Configuring MSRP | 248 |
| Monitoring the AVB Network | 249 |
| Monitoring Audio Video Bridging | 249 |
| Monitoring Generalized Precision Time Protocol | 250 |
| Monitoring Multiple Stream Reservation Protocol | 250 |
| Monitoring Hierarchical QoS | 250 |
| Monitoring Multiple VLAN Registration Protocol | 251 |
| Examples of AVB Configurations and Monitoring | 251 |
| Examples for Audio Video Bridging | 251 |
| Example: Verifying Generalized Precision Time Protocol | 253 |
| Example: Verifying Multiple Stream Reservation Protocol | 257 |
| Example: Verifying Hierarchical QoS | 260 |
| Example: Verifying Multiple VLAN Registration Protocol | 271 |
| Feature History for Audio Video Bridging | 272 |

CHAPTER 14

Configuring Flexlink+ 273

| | |
|-----------------------------------|-----|
| Restrictions for FlexLink+ | 273 |
| Information about FlexLink+ | 273 |
| FlexLink+ Overview | 273 |
| FlexLink+ Configuration | 274 |
| VLAN Load Balancing and FlexLink+ | 274 |
| How to configure Flexlink+ | 276 |

| | |
|--|-----|
| Configuring the Active Port for FlexLink+ | 277 |
| Configuring the Standby Port for FlexLink+ | 277 |
| Configuring VLAN Load Balancing on FlexLink+ | 278 |
| Configuring Propagation of FlexLink+ Topology Change Messages | 280 |
| Configuring Preempt Time Delay | 282 |
| Configuring Manual Preemption for VLAN Load Balancing | 282 |
| Configuration Examples for FlexLink+ | 283 |
| Example: Configuring the Active Port for FlexLink+ | 283 |
| Example: Configuring the Standby Port for FlexLink+ | 283 |
| Example: Configuring VLAN Load Balancing for FlexLink+ | 283 |
| Example: Configuring Propagation of FlexLink+ Topology Change Messages | 284 |
| Feature History for FlexLink+ | 284 |



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 22](#)
- [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 On all the models of the Cisco Catalyst 9500 Series Switches, the long path cost method is the default STP path cost method.



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.

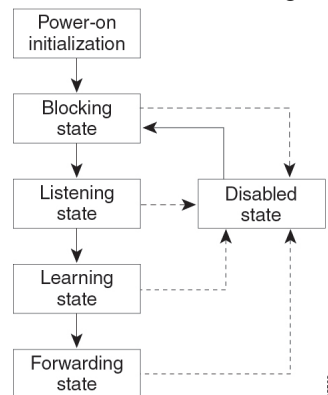
- Disabled—The interface is not participating in spanning tree because of a shutdown port, no link on the port, or no spanning-tree instance running on the port.

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:

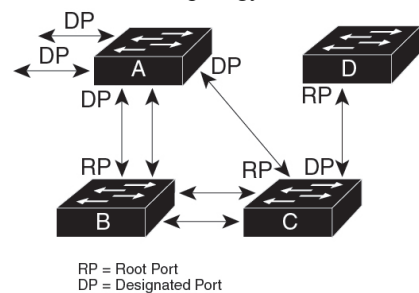
- Discards frames received on the interface
- Discards frames that are switched from another interface for forwarding
- Does not learn addresses
- Does not receive BPDUs

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.

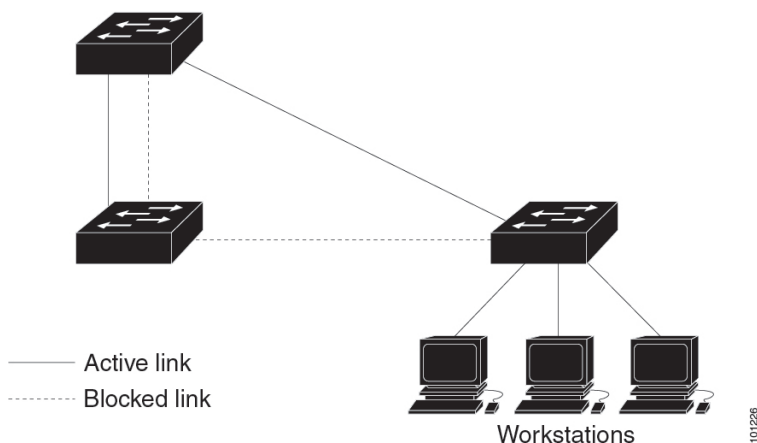
For example, assume that one port on Switch B is a Gigabit Ethernet link and that another port on Switch B (a 10/100 link) is the root port. Network traffic might be more efficient over the Gigabit Ethernet link. By changing the spanning-tree port priority on the Gigabit Ethernet port to a higher priority (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 0x00180C2000000 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

On the Cisco Catalyst 9500 Series Switches (C9500-12Q, C9500-16X, C9500-24Q, C9500-40X models):

- 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 Cisco IOS XE Amsterdam 17.2.1 release, in PVST+ or Rapid PVST+ mode, the device or device stack supports up to 300 spanning-tree instances.

On the Cisco Catalyst 9500 Series High Performance Switches (C9500-32C, C9500-32QC, C9500-48Y4C, C9500-24Y4C models):

- 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.
- 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, in 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. For MST, the number of VLANs that can be mapped to a particular MST instance is 4000 with the Customizable SDM template.

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 25 Gbps: 800 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 25 Gbps: 800 40 Gbps: 500 100 Gbps: 200 1 Tbps: 20 10 Tbps: 2 |

| Feature | Default Setting |
|----------------------|---|
| 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. |

| | Command or Action | Purpose |
|---------------|--|---|
| Step 5 | spanning-tree link-type point-to-point Example: <pre>Device(config-if)# spanning-tree link-type point-to-point</pre> | <p>Specifies that the link type for this port is point-to-point.</p> <p>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.</p> |
| Step 6 | end Example: <pre>Device(config-if)# end</pre> | Returns to privileged EXEC mode. |
| Step 7 | clear spanning-tree detected-protocols Example: <pre>Device# clear spanning-tree detected-protocols</pre> | <p>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.</p> <p>This step is optional if the designated device detects that this device is running rapid PVST+.</p> |

(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: <pre>Device> enable</pre> | <p>Enables privileged EXEC mode.</p> <p>Enter your password if prompted.</p> |
| Step 2 | configure terminal Example: <pre>Device# configure terminal</pre> | Enters global configuration mode. |
| Step 3 | no spanning-tree vlan <i>vlan-id</i> Example: | For <i>vlan-id</i> , the range is 1 to 4094. |

| | Command or Action | Purpose |
|---------------|---|----------------------------------|
| | Device(config)# no spanning-tree vlan 300 | |
| 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. |

| | Command or Action | Purpose |
|---------------|--|----------------------------------|
| 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 [<i>diameter 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. |

| | Command or Action | Purpose |
|---------------|---|---|
| | | Use the same network diameter value that you used when configuring the primary root switch. |
| 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 interface-id 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 port-channel-number). |
| Step 4 | spanning-tree port-priority priority 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 vlan-id port-priority priority 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 |

| | Command or Action | Purpose |
|---------------|---|--|
| | | 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 <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 (port-channel <i>port-channel-number</i>). |
| Step 4 | spanning-tree cost <i>cost</i> 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 <i>vlan-id</i> cost <i>cost</i> 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, |

| | Command or Action | Purpose |
|---------------|--|--|
| | | or a series of VLANs separated by a comma. The range is 1 to 4094. <ul style="list-style-type: none"> 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. |

| | Command or Action | Purpose |
|---------------|---|--|
| | | <ul style="list-style-type: none"> 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. <p>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.</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 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. |
| 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: | Enters global configuration mode. |

| | Command or Action | Purpose |
|---------------|---|--|
| | Device# configure terminal | |
| 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. |
| 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: | 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. |

| | Command or Action | Purpose |
|---------------|---|----------------------------------|
| | Device(config)# spanning-tree transmit hold-count 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 9500 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 Everest 16.5.1a | Spanning Tree Protocol | STP is a Layer 2 link management protocol that provides path redundancy while preventing loops in the network. Support for this feature was introduced only on the C9500-12Q, C9500-16X, C9500-24Q, C9500-40X models of the Cisco Catalyst 9500 Series Switches. |
| Cisco IOS XE Fuji 16.8.1a | Spanning Tree Protocol | Support for this feature was introduced only on the C9500-32C, C9500-32QC, C9500-48Y4C, and C9500-24Y4C models of the Cisco Catalyst 9500 Series Switches. |
| Cisco IOS XE Gibraltar 16.11.1 | Spanning Tree Instances | The number of supported spanning tree instances was increased to 256. Support for this feature was introduced only on the C9500-12Q, C9500-16X, C9500-24Q, C9500-40X models of the Cisco Catalyst 9500 Series Switches. |
| Cisco IOS XE Amsterdam 17.2.1 | Spanning Tree Instances | The number of supported spanning tree instances was increased to 300. Support for this feature was introduced only on the C9500-12Q, C9500-16X, C9500-24Q, C9500-40X models of the Cisco Catalyst 9500 Series Switches. |

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 25](#)
- [Information About Loop Detection Guard, on page 25](#)
- [How to Configure Loop Detection Guard, on page 28](#)
- [Additional References for Configuring Loop Detection Guard, on page 29](#)
- [Feature History for Loop Detection Guard, on page 30](#)

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

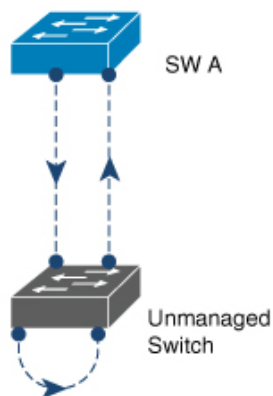
A computer network can experience a network loop where there is more than one Layer 2 path between two endpoints. This is possible when there are multiple connections between two switches in a network or two ports on the same switch are connected to each other. The following figure shows a few examples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

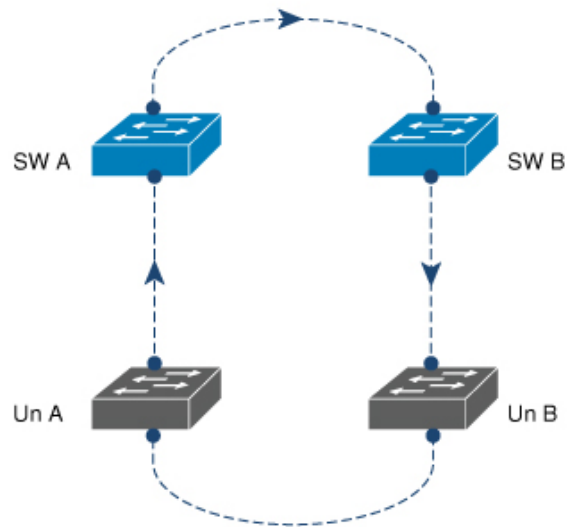
Example 2: This example shows a network loop involving four switches, two within the network (SW A and SW B) and two unmanaged switches (Un A and Un B). Traffic is moving in the following direction SW A to SW B to Un A to Un B and back to SW A, resulting in a network loop.

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

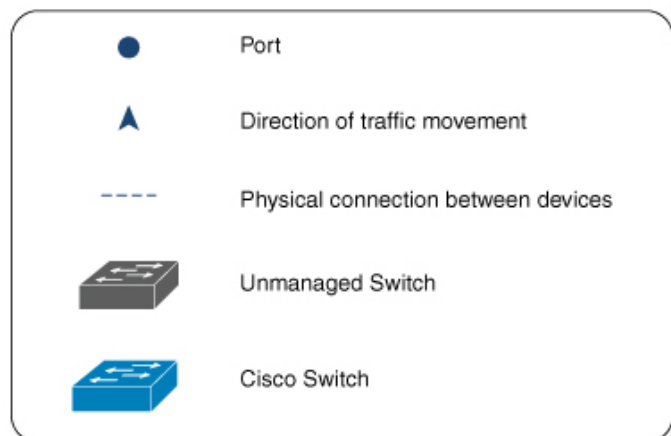
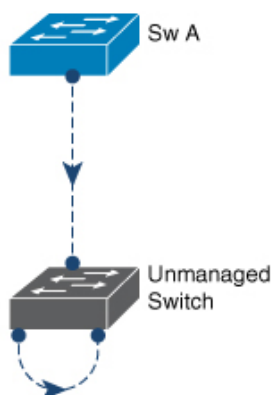
Example 1



Example 2



Example 3



356545

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

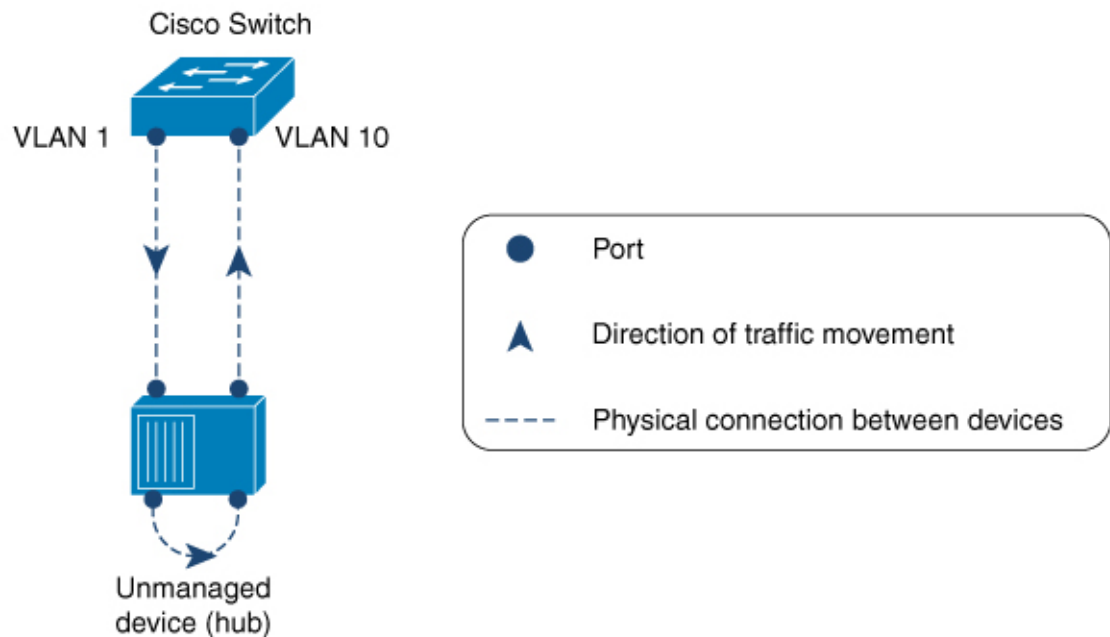
Spanning Tree Protocol and Loop Detection Guard

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 4: A Switch Connected to an Unmanaged Network Hub



356546

How to Configure Loop Detection Guard

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 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 | <p>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.</p> <p>Use the no form of this command to disable this feature.</p> <p>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 detected in one VLAN can result in the error-disabling of all VLAN traffic associated with the trunk port.</p> |
| Step 5 | [no] loopdetect {time action syslog source-port } Example: | 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 |

| | Command or Action | Purpose |
|---------------|---|--|
| | Device(config-if) # loopdetect 7 | <p>an action, the destination port is error-disabled by default.</p> <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 0 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 (Device(config-if) # loopdetect 7), 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 | end Example: Device(config-if) # end | Returns to privileged EXEC mode. |
| Step 7 | show loopdetect Example: Device# show loopdetect | 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 9500 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 configured with STP. Support for this feature was introduced on all the models of the Cisco Catalyst 9500 Series Switches. |

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 31](#)
- [Restrictions for MSTP, on page 31](#)
- [Information About MSTP, on page 32](#)
- [How to Configure MSTP and MSTP Parameters, on page 46](#)
- [Feature History for MSTP, on page 58](#)

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 MSTP

- 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 MSTP

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 spanning-tree instances with the Cisco Catalyst 9500 Series Switches (C9500-12Q, C9500-16X, C9500-24Q, C9500-40X models) and 64 spanning-tree instances with the Cisco Catalyst 9500 Series High Performance Switches (C9500-32C, C9500-32QC, C9500-48Y4C, C9500-24Y4C models). 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 CST 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

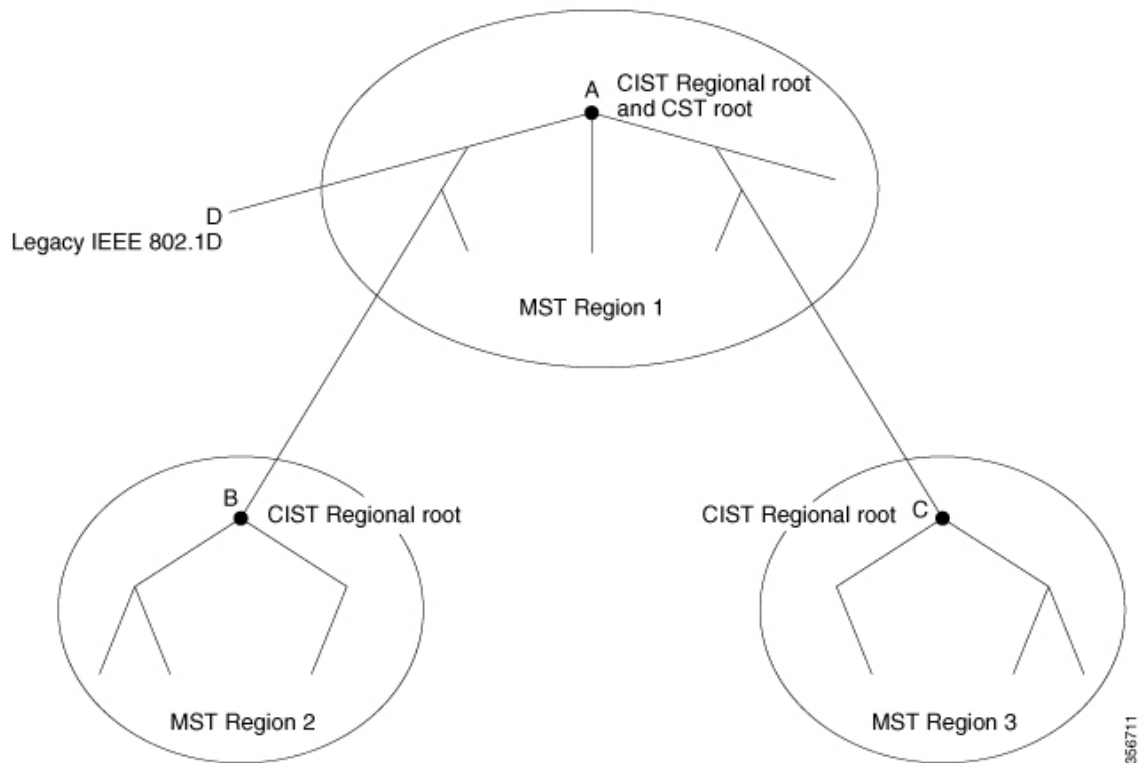
Some MST naming conventions that are used in Cisco's prestandard implementation have been changed to identify some *internal* or *regional* parameters. These parameters are significant only within an MST region, as opposed to external parameters that are relevant to the whole network. Because the CIST 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 5: 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.

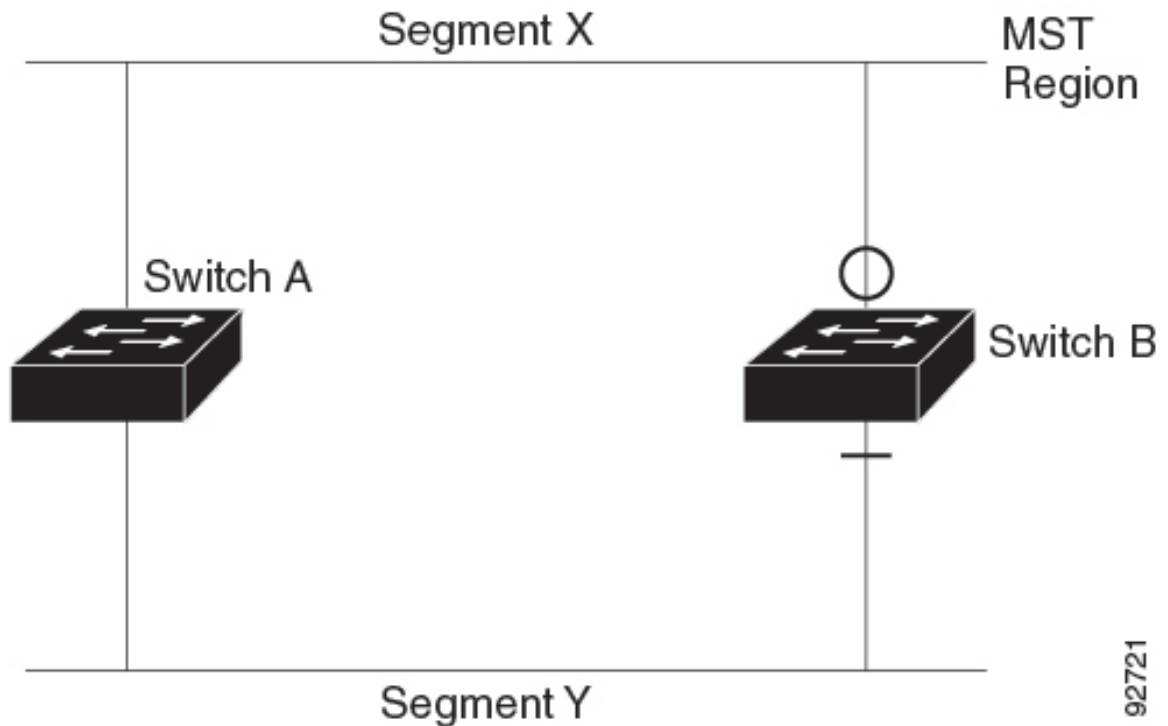
- The boundary port is not the root port of the CIST regional root—The MSTI ports follow the state and role of the CIST port. The standard provides less information, and it might be difficult to understand why an MSTI port can be alternately blocking when it receives no BPDUs (MRecords). In this case, 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 6: 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.

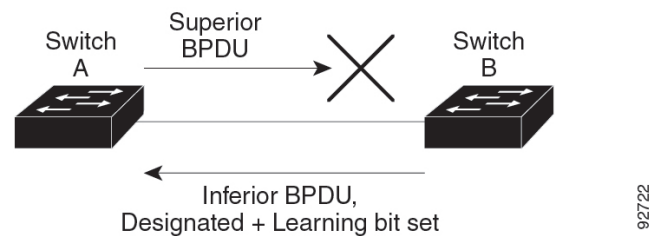
Detecting Unidirectional Link Failure

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 7: 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 8: 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.

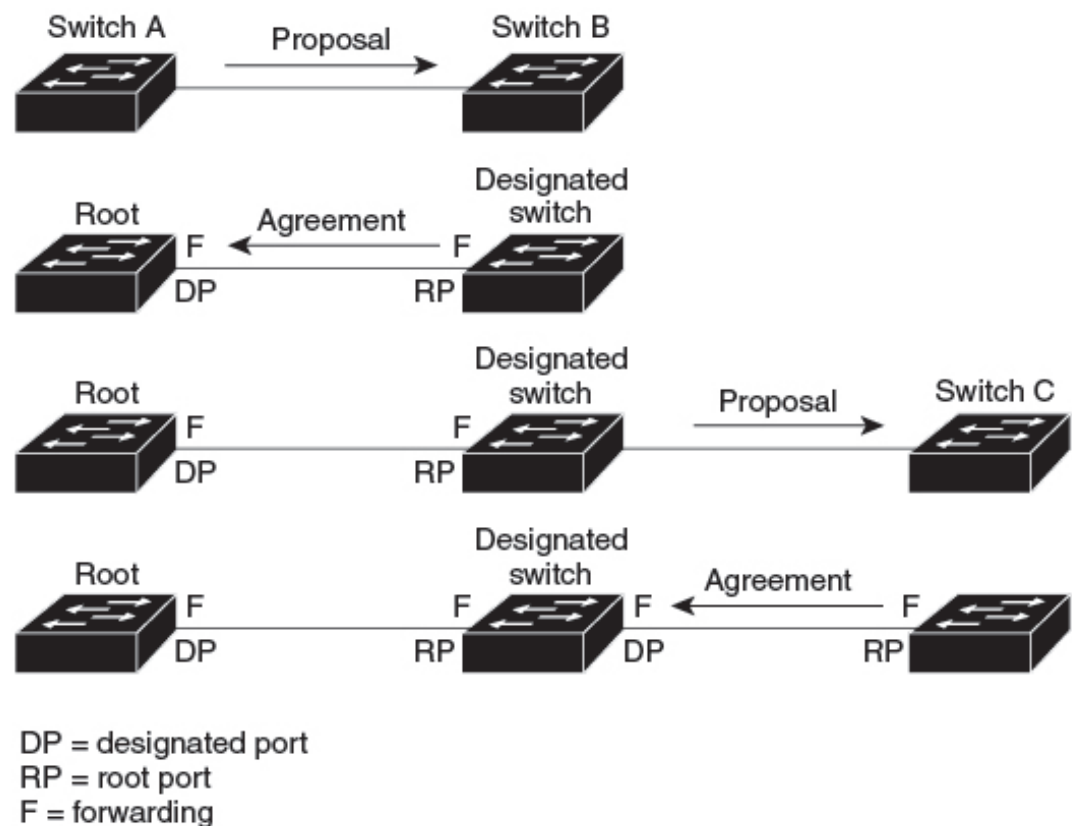
After receiving the proposal message, Switch B selects as its new root port the port from which the proposal message was received, forces all nonedge ports to the blocking state, and sends an agreement message (a BPDU with the agreement flag set) through its new root port.

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.

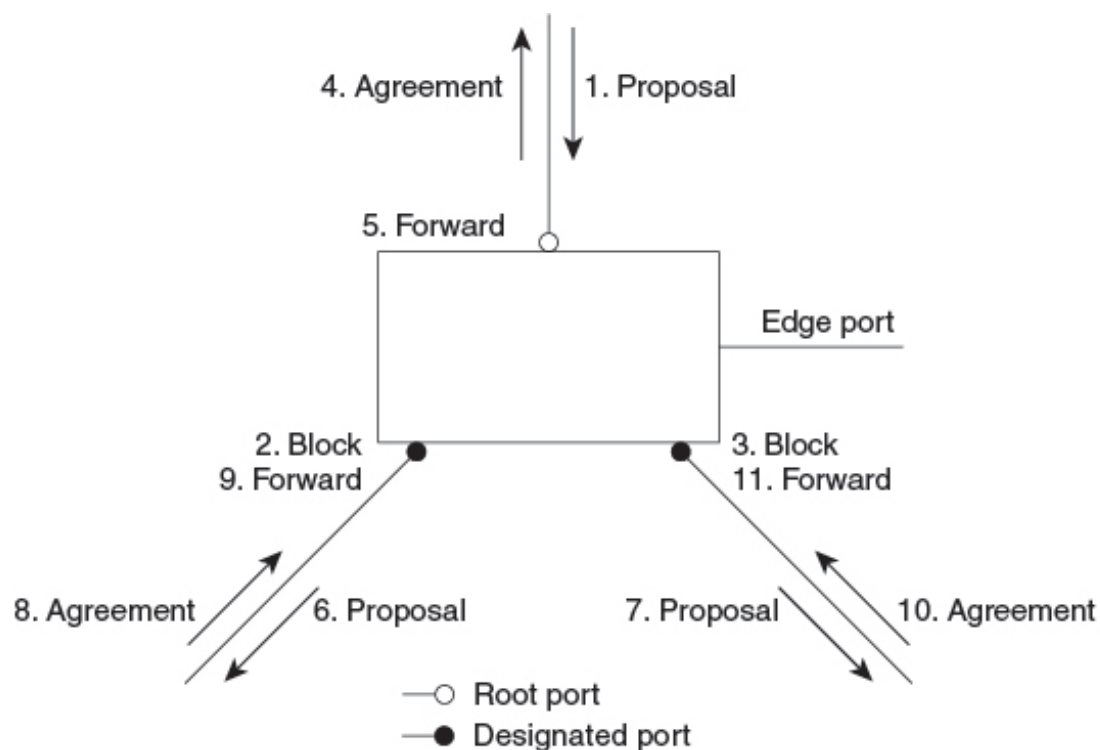
The device is synchronized with superior root information that is received on the root port if all other ports are synchronized. An individual port on the device is synchronized if:

- 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 9: 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.



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 MSTP and MSTP 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 spanning-tree instances with the Cisco Catalyst 9500 Series Switches (C9500-12Q, C9500-16X, C9500-24Q, C9500-40X models) and 64 spanning-tree instances with the Cisco Catalyst 9500 Series High Performance Switches (C9500-32C, C9500-32QC, C9500-48Y4C, C9500-24Y4C models). 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 <i>instance-id</i> vlan <i>vlan-range</i> 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# configure terminal | Enters global configuration mode. |
| Step 3 | spanning-tree mst <i>instance-id</i> root secondary Example: Device(config)# spanning-tree mst 0 root secondary | Configures a device 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)# end | 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> enable | 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 interface-id Example: Device (config) # interface gigabitethernet 1/0/1 | Specifies an interface to configure, and enters interface configuration mode. |
| Step 4 | spanning-tree mst instance-id port-priority priority 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 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. The port-channel range is 1 to 48. |
| Step 4 | spanning-tree mst instance-id cost cost 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. 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. |
| 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 <i>seconds</i> 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 <i>seconds</i> 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 | <p>Enter one of the following commands:</p> <ul style="list-style-type: none"> clear spanning-tree detected-protocols clear spanning-tree detected-protocols interface <i>interface-id</i> <p>Example:</p> <pre>Device# clear spanning-tree detected-protocols</pre> <p>or</p> <pre>Device# clear spanning-tree detected-protocols interface gigabitethernet 1/0/1</pre> | 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 MSTP

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 Everest 16.5.1a | Multiple Spanning-Tree Protocol | <p>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.</p> <p>Support for this feature was introduced only on the C9500-12Q, C9500-16X, C9500-24Q, C9500-40X models of the Cisco Catalyst 9500 Series Switches.</p> |
| Cisco IOS XE Fuji 16.8.1a | Multiple Spanning-Tree Protocol | <p>Support for this feature was introduced only on the C9500-32C, C9500-32QC, C9500-48Y4C, and C9500-24Y4C models of the Cisco Catalyst 9500 Series Switches.</p> |
| Cisco IOS XE Bengaluru 17.5.1 | Number of VLANs | <p>The number of VLANs that can be mapped to a particular MST instance is increased to 4094.</p> <p>Support for this feature was introduced only on the C9500-32C, C9500-32QC, C9500-48Y4C, and C9500-24Y4C models of the Cisco Catalyst 9500 Series Switches.</p> |

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 61](#)
- [How to Configure Optional Spanning-Tree Features, on page 74](#)
- [Monitoring the Spanning-Tree Status, on page 90](#)
- [Additional References for Optional Spanning Tree Features, on page 90](#)
- [Feature History for Optional Spanning Tree Features, on page 90](#)

Information About Optional Spanning-Tree Features

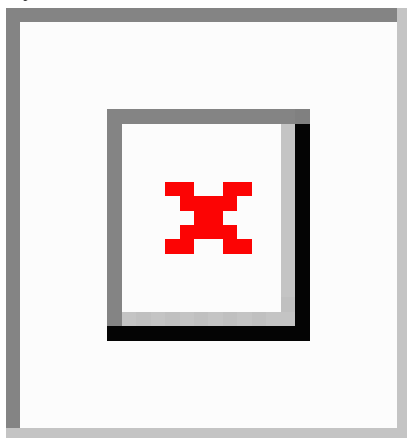
The following sections provide information about Optional Spanning-Tree features:

PortFast

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

Figure 10: PortFast-Enabled Interfaces

You can use PortFast on interfaces 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.



Note PortFast edge feature is supported only on C9500-32C, C9500-32QC, C9500-48Y4C, and C9500-24Y4C models of the Cisco Catalyst 9500 Series Switches

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.



Note Configuring an STP PortFast port type is supported only on the Cisco Catalyst High Performance Series switches (C9500-32C, C9500-32QC, C9500-48Y4C, and C9500-24Y4C models).

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 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 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_100](#).



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 or PortFast enabled ports, spanning tree shuts down ports that are in a PortFast edge-operational or PortFast operational state if any BPDU is received on them. In a valid configuration, PortFast edge-enabled ports or PortFast enabled ports do not receive BPDUs. Receiving a BPDU on a PortFast edge-enabled port or PortFast 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 or PortFast 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 or PortFast enabled interfaces at the global level keeps those interfaces that are in a PortFast edge-operational or PortFast 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 or PortFast enabled interface, the interface loses its PortFast edge-operational status or PortFast operational status, and BPDU filtering is disabled.

Enabling BPDU filtering on an interface without also enabling the PortFast edge feature or PortFast 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.

Bridge Assurance is supported only on the Cisco Catalyst High Performance Series switches (C9500-32C, C9500-32QC, C9500-48Y4C, and C9500-24Y4C models).

The following example shows how Bridge Assurance protects your network from bridging loops. Here, [Figure 11: Network with Normal STP Topology, on page 64](#) shows a normal STP topology, and [Figure 12: Network Loop Due to a Malfunctioning Switch, on page 64](#) demonstrates a potential network problem when the device fails (brain dead) and Bridge Assurance is not enabled on the network.

Figure 11: Network with Normal STP Topology



Figure 12: Network Loop Due to a Malfunctioning Switch



[Figure 13: Network with STP Topology Running Bridge Assurance, on page 64](#) shows the network with Bridge Assurance enabled, and the STP topology progressing normally with bidirectional BDPUs issuing from every STP network port. [Figure 14: Network Problem Averted with Bridge Assurance Enabled, on page 64](#) shows how the potential network problem shown in [Figure 12: Network Loop Due to a Malfunctioning Switch, on page 64](#) does not occur when you have Bridge Assurance enabled on your network.

Figure 13: Network with STP Topology Running Bridge Assurance



Figure 14: 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:

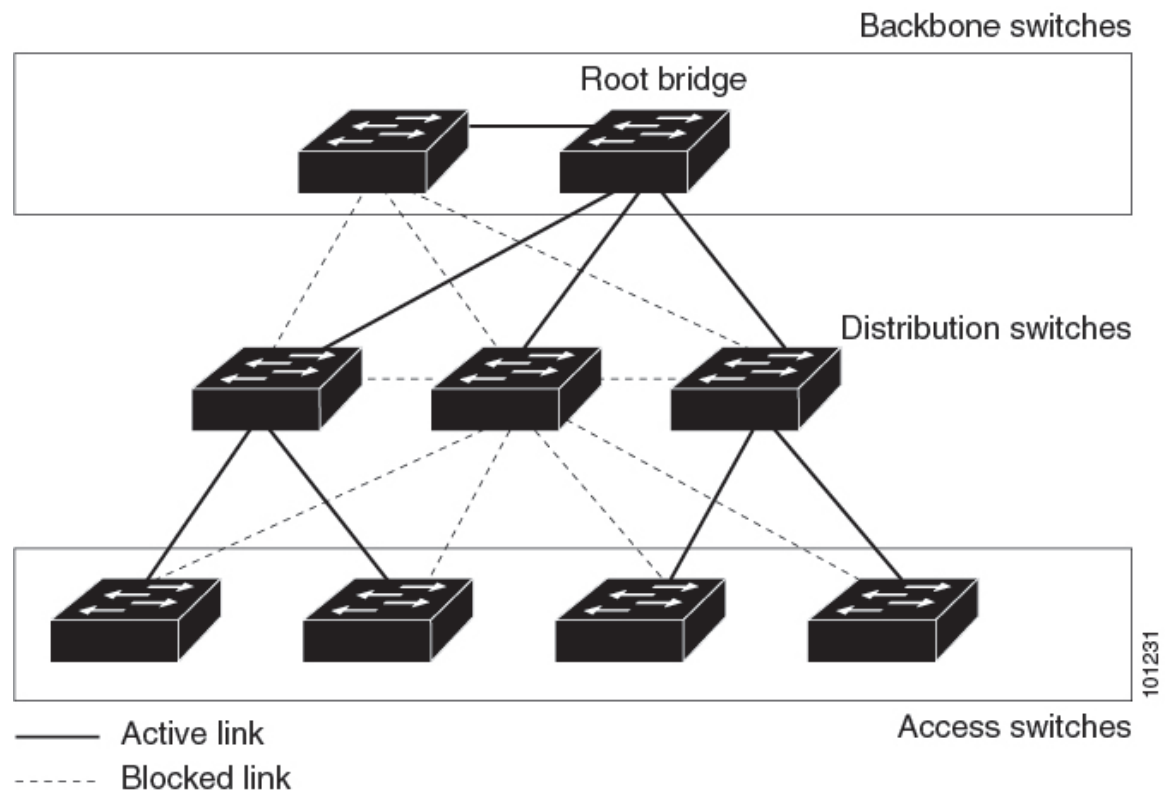
- Bridge Assurance can be enabled or disabled globally.

- 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, BPDU filtering and BPDU 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 15: 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.

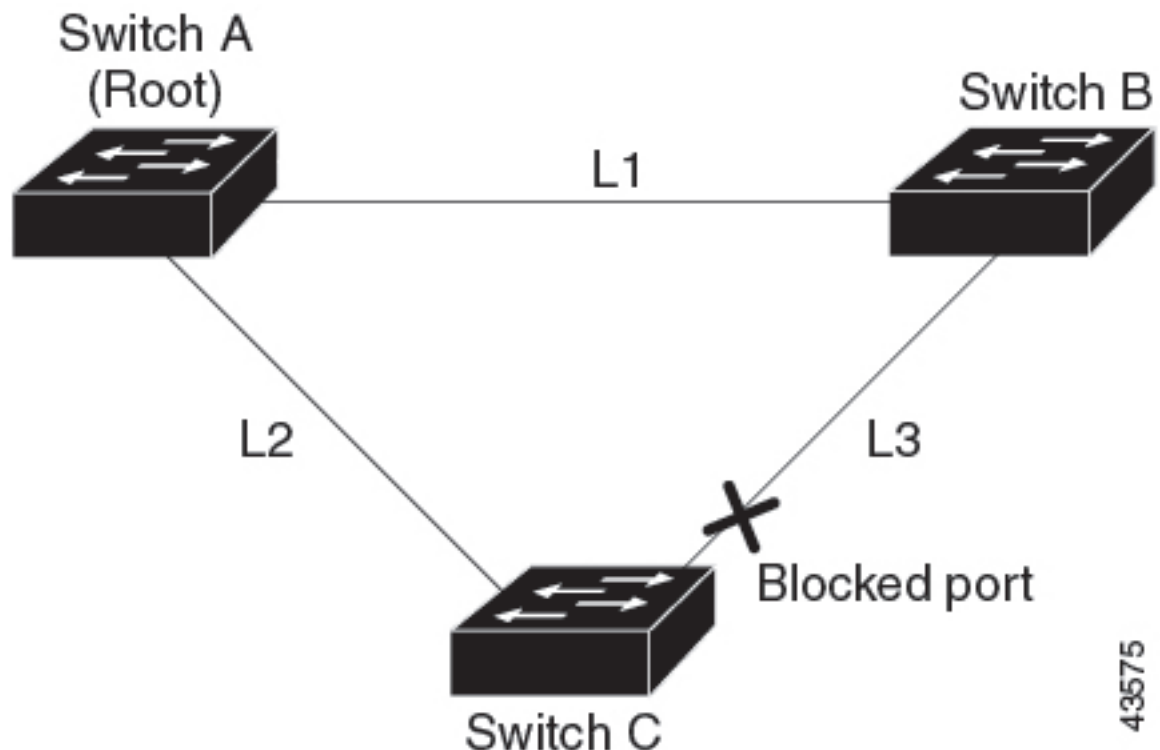


Note UplinkFast is most useful in wiring-closet switches at the access or edge of the network. It is not appropriate for backbone devices. This feature might not be useful for other types of applications.

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 16: UplinkFast Example Before Direct Link Failure

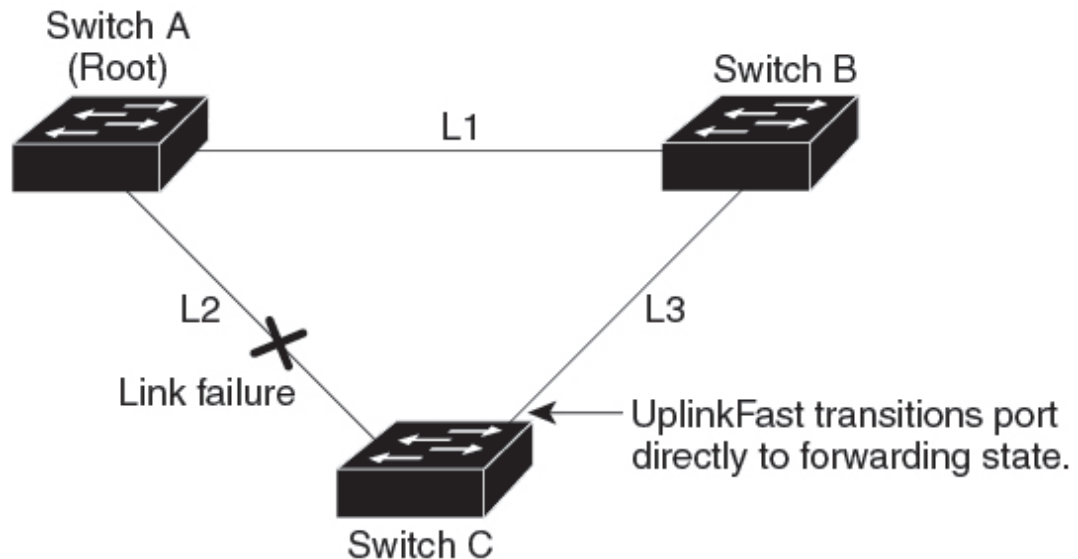
This topology has no link failures. Switch A, the root switch, is connected directly to Switch B over link L1 and to Switch C over link L2. The Layer 2 interface on Switch C that is connected directly to Switch B is in a blocking state.



43575

Figure 17: 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](#).

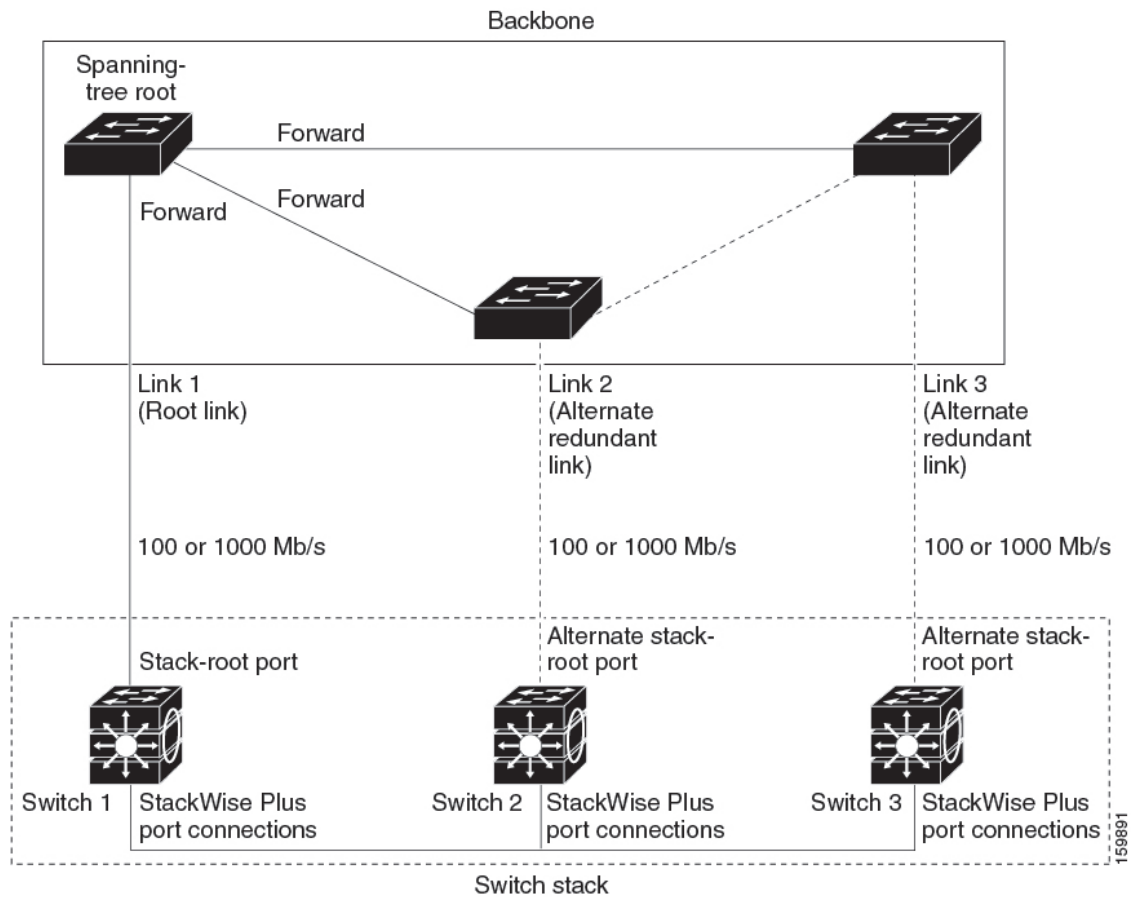
How Cross-Stack UplinkFast Works

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

Figure 18: 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 19: 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.

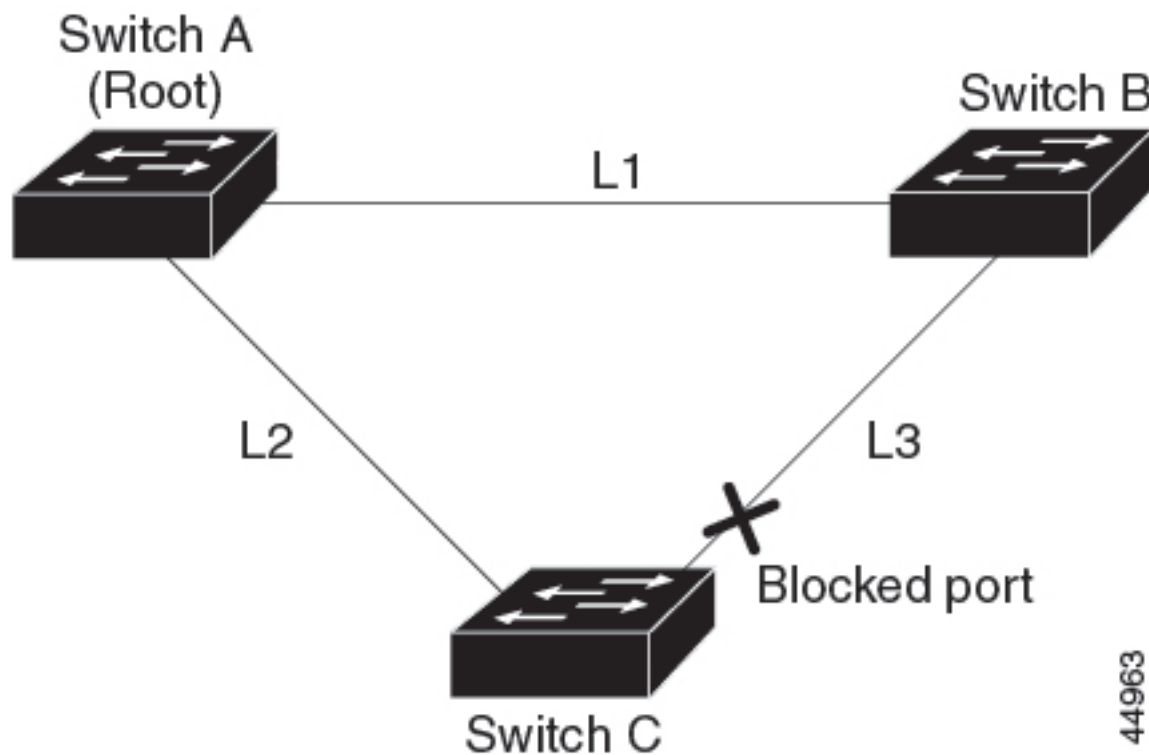
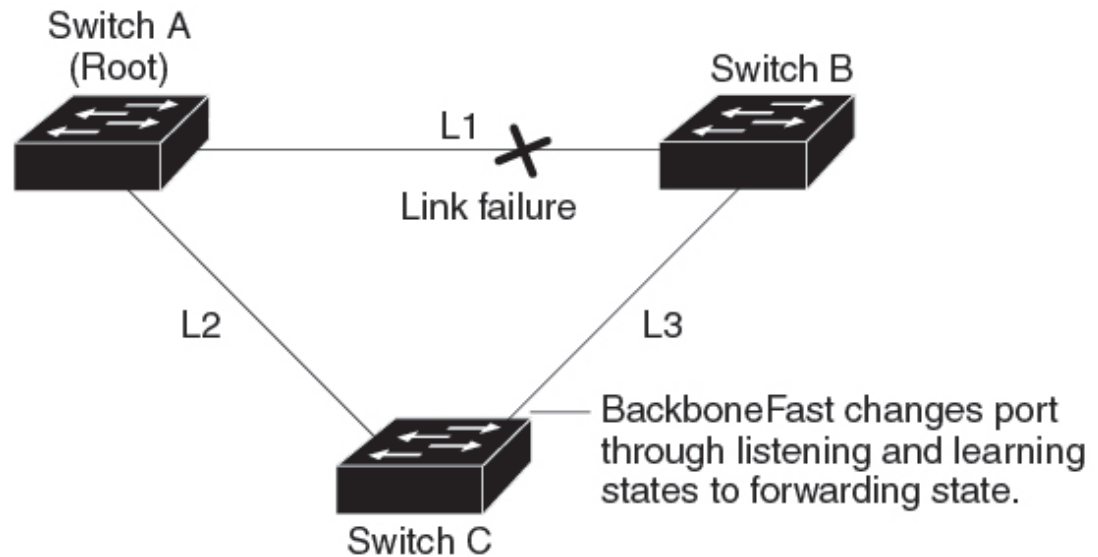


Figure 20: BackboneFast Example After Indirect Link Failure

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

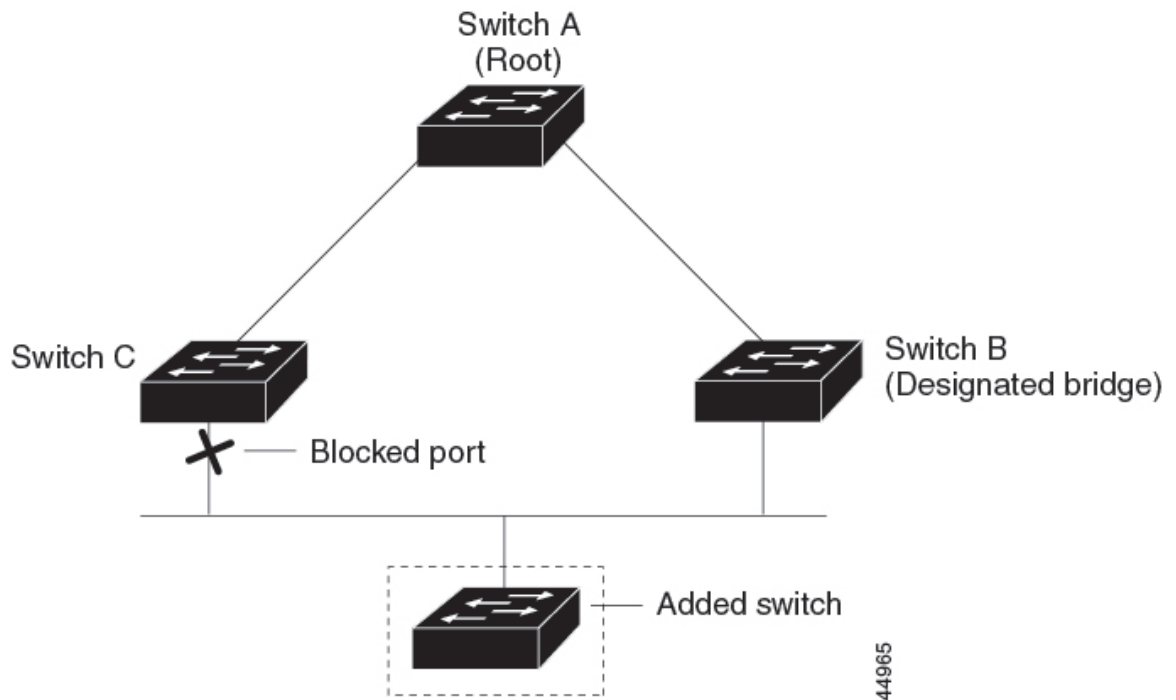
BPDUs from Switch B, Switch C assumes that an indirect failure has occurred. At that point, BackboneFast allows the blocked interface on Switch C to move immediately to the listening state without waiting for the maximum aging time for the interface to expire. BackboneFast then transitions the Layer 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 21: 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

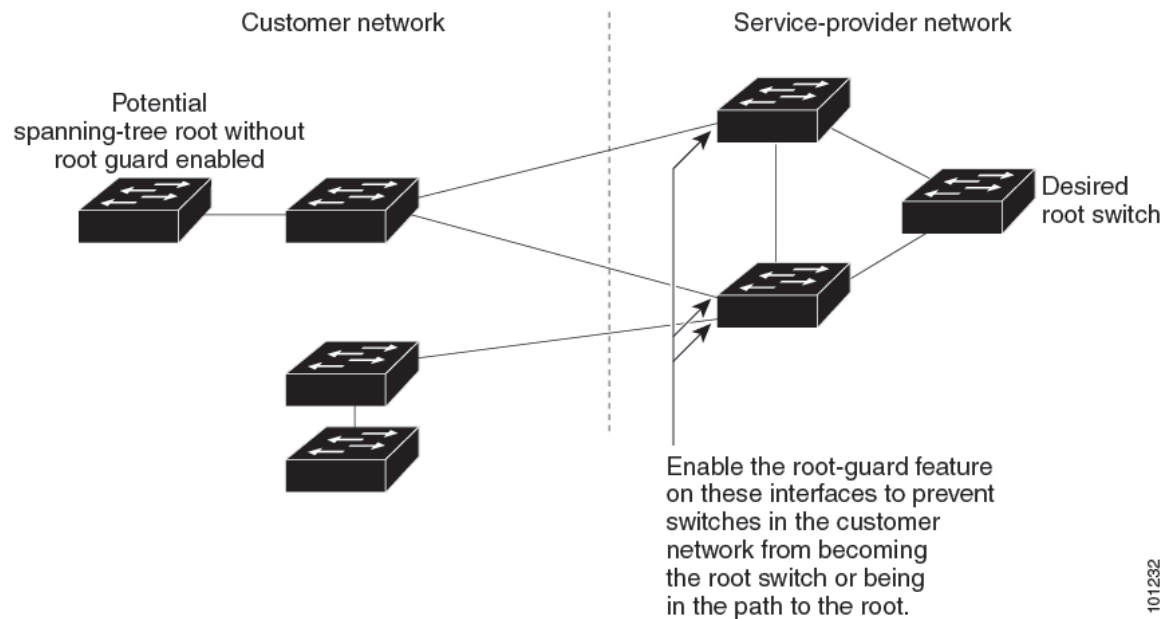
You can use EtherChannel guard to detect an EtherChannel misconfiguration between the switch and a connected device. A misconfiguration can occur if the switch interfaces are configured in an EtherChannel, but the interfaces on the other device are not. A misconfiguration can also occur if the channel 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 22: 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.



101232

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 interface-id 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 | end Example: Device(config-if) # end | 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 | configure terminal Example: Device# configure terminal | Enters global configuration mode. |
| Step 2 | spanning-tree portfast [edge network normal] default Example: Device(config) # spanning-tree portfast edge default | <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 76

[Configuring a PortFast Network Port on a Specified Interface](#), on page 78

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 | Specifies an interface to configure, and enters interface configuration mode. |

| | Command or Action | Purpose |
|---------------|---|--|
| Step 3 | [no] spanning-tree portfast edge Example: Device(config-if)# spanning-tree portfast edge | Enables edge behavior on a Layer 2 access port connected to an end workstation or server. (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 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 4 | end Example: Device(config-if)# end | Exits configuration mode. |
| Step 5 | show spanning-tree interface { {fastethernet gigabitethernet tengigabitethernet } slot /port } { port-channel port_channel_number} 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 75
- [Configuring a PortFast Network Port on a Specified Interface](#), on page 78

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 63.

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)# interface gigabitethernet1/1 | Specifies an interface to configure, and enters interface configuration mode. |
| Step 3 | [no] spanning-tree portfast network Example: Device(config-if)# 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. |

| | Command or Action | Purpose |
|---------------|---|-----------------------------|
| Step 4 | end Example: Device(config-if) # end | Exits configuration mode. |
| Step 5 | 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 75

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

(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 or PortFast 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 | <ul style="list-style-type: none"> For C9500-12Q, C9500-16X, C9500-24Q, C9500-40X models of the Cisco Catalyst 9500 Series Switches use the following command spanning-tree portfast bpduguard default For C9500-32C, C9500-32QC, C9500-48Y4C, C9500-24Y4C, and C9500X-28C8D models of the Cisco Catalyst 9500 Series Switches use the following command spanning-tree portfast edge bpduguard default Example: Device (config) # spanning-tree portfast edge bpduguard default | Enables BPDU guard. |
| Step 4 | interface interface-id 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 | <ul style="list-style-type: none"> For C9500-12Q, C9500-16X, C9500-24Q, C9500-40X models of the Cisco Catalyst 9500 Series Switches use the following command spanning-tree portfast For C9500-32C, C9500-32QC, C9500-48Y4C, C9500-24Y4C, and C9500X-28C8D models of the Cisco Catalyst 9500 Series Switches use the following command spanning-tree portfast edge Example: Device (config-if) # spanning-tree portfast edge | Enables the PortFast edge feature or PortFast feature. |

| | Command or Action | Purpose |
|---------------|---|----------------------------------|
| 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 or PortFast feature. When the port receives a BPDU, it is put it in the error-disabled state.

Enabling BPDU 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 or PortFast 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 or PortFast 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.

This procedure is optional.

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 bpdupfilter default | Globally enables BPDU filtering. By default, BPDU filtering is disabled. |

| | Command or Action | Purpose |
|---------------|---|---|
| | <ul style="list-style-type: none"> For C9500-12Q, C9500-16X, C9500-24Q, C9500-40X models of the Cisco Catalyst 9500 Series Switches use the following command <p>spanning-tree portfast bpdupfilter default</p> <ul style="list-style-type: none"> For C9500-32C, C9500-32QC, C9500-48Y4C, and C9500-24Y4C models of the Cisco Catalyst 9500 Series Switches use the following command <p>spanning-tree portfast edge bpdupfilter default</p> <p>Example:</p> <pre>Device(config)# spanning-tree portfast edge bpdupfilter default</pre> | |
| Step 4 | <p>interface <i>interface-id</i></p> <p>Example:</p> <pre>Device(config)# interface gigabitethernet 1/0/2</pre> | Specifies the interface connected to an end station, and enters interface configuration mode. |
| Step 5 | <ul style="list-style-type: none"> For C9500-12Q, C9500-16X, C9500-24Q, C9500-40X models of the Cisco Catalyst 9500 Series Switches use the following command <p>spanning-tree portfast</p> <ul style="list-style-type: none"> For C9500-32C, C9500-32QC, C9500-48Y4C, and C9500-24Y4C models of the Cisco Catalyst 9500 Series Switches use the following command <p>spanning-tree portfast edge</p> <p>Example:</p> <pre>Device(config-if)# spanning-tree portfast edge</pre> | Enables the PortFast edge feature or PortFast feature. |
| Step 6 | <p>end</p> <p>Example:</p> <pre>Device(config-if)# end</pre> | 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 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 [<i>max-update-rate</i> <i>pkts-per-second</i>] 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. |
| Step 2 | configure terminal Example: Device# configure terminal | Enters global configuration mode. |
| Step 3 | spanning-tree backbonefast Example: Device(config)# spanning-tree backbonefast | Enables BackboneFast. |
| Step 4 | end Example: Device(config)# end | 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> 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 etherchannel guard misconfig Example: Device(config)# spanning-tree etherchannel guard misconfig | Enables EtherChannel guard. |

| | Command or Action | Purpose |
|---------------|---|----------------------------------|
| Step 4 | end Example: Device (config) # end | 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 interface-id 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: | Enables root guard on the interface. |

| | Command or Action | Purpose |
|---------------|---|---|
| | <code>Device(config-if) # spanning-tree guard root</code> | By default, root guard is disabled on all interfaces. |
| Step 5 | end Example: <code>Device(config-if) # end</code> | 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: • <code>show spanning-tree active</code> • <code>show spanning-tree mst</code> Example: <code>Device# show spanning-tree active</code> or <code>Device# show spanning-tree mst</code> | Verifies which interfaces are alternate or root ports. |
| Step 2 | configure terminal Example: <code>Device# configure terminal</code> | Enters global configuration mode. |
| Step 3 | spanning-tree loopguard default Example: <code>Device(config) # spanning-tree loopguard default</code> | Enables loop guard. By default, loop guard is disabled. |
| Step 4 | end Example: <code>Device(config) # end</code> | Returns to privileged EXEC mode. |

Monitoring the Spanning-Tree Status

Table 9: Commands for Monitoring the Spanning-Tree Status

| Command | Purpose |
|---|--|
| show spanning-tree active | Displays spanning-tree information on active interfaces only. |
| 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 lines of spanning-tree state section. |
| For C9500-12Q, C9500-16X, C9500-24Q, C9500-40X models of the Cisco Catalyst 9500 Series Switches use the following command: show spanning-tree mst interface <i>interface-id</i> portfast For C9500-32C, C9500-32QC, C9500-48Y4C, and C9500-24Y4C models of the Cisco Catalyst 9500 Series Switches use the following command 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 9500 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 Everest 16.5.1a | 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. Support for this feature was introduced only on the C9500-12Q, C9500-16X, C9500-24Q, C9500-40X models of the Cisco Catalyst 9500 Series Switches. |
| Cisco IOS XE Fuji 16.8.1a | Optional Spanning Tree Protocol | Support for this feature was introduced only on the C9500-32C, C9500-32QC, C9500-48Y4C, and C9500-24Y4C models of the Cisco Catalyst 9500 Series Switches. |

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 93](#)
- [Information About EtherChannels, on page 93](#)
- [How to Configure EtherChannels, on page 105](#)
- [Monitoring EtherChannel, Port Aggregation Protocol, and Link Aggregation Control Protocol Status, on page 122](#)
- [Configuration Examples for EtherChannels, on page 122](#)
- [Additional References for EtherChannels, on page 125](#)
- [Feature History for EtherChannels, on page 126](#)

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.

Information About EtherChannels

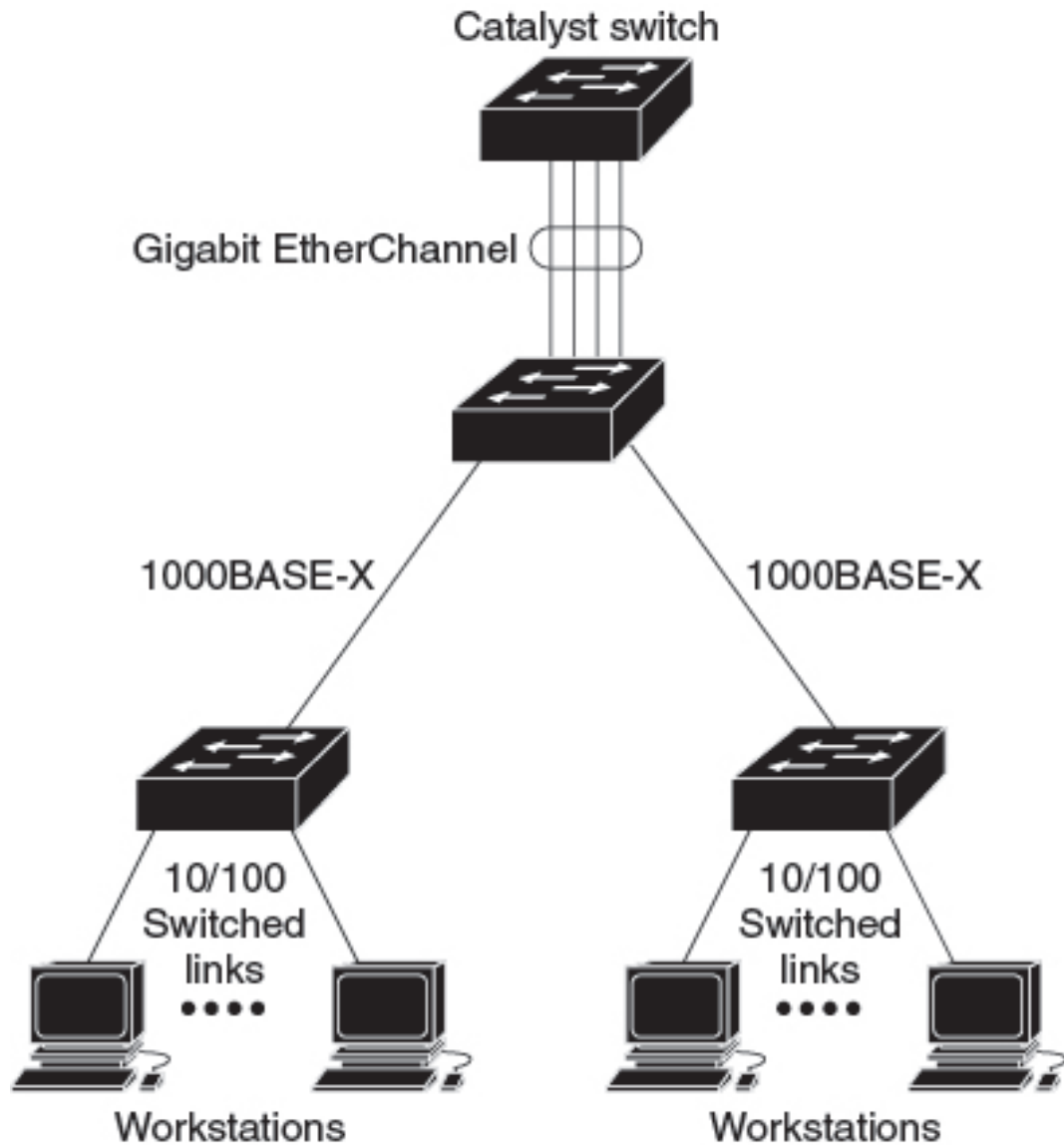
The following sections provide information about EtherChannels and the various modes to configure EtherChannels.

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 23: Typical EtherChannel Configuration

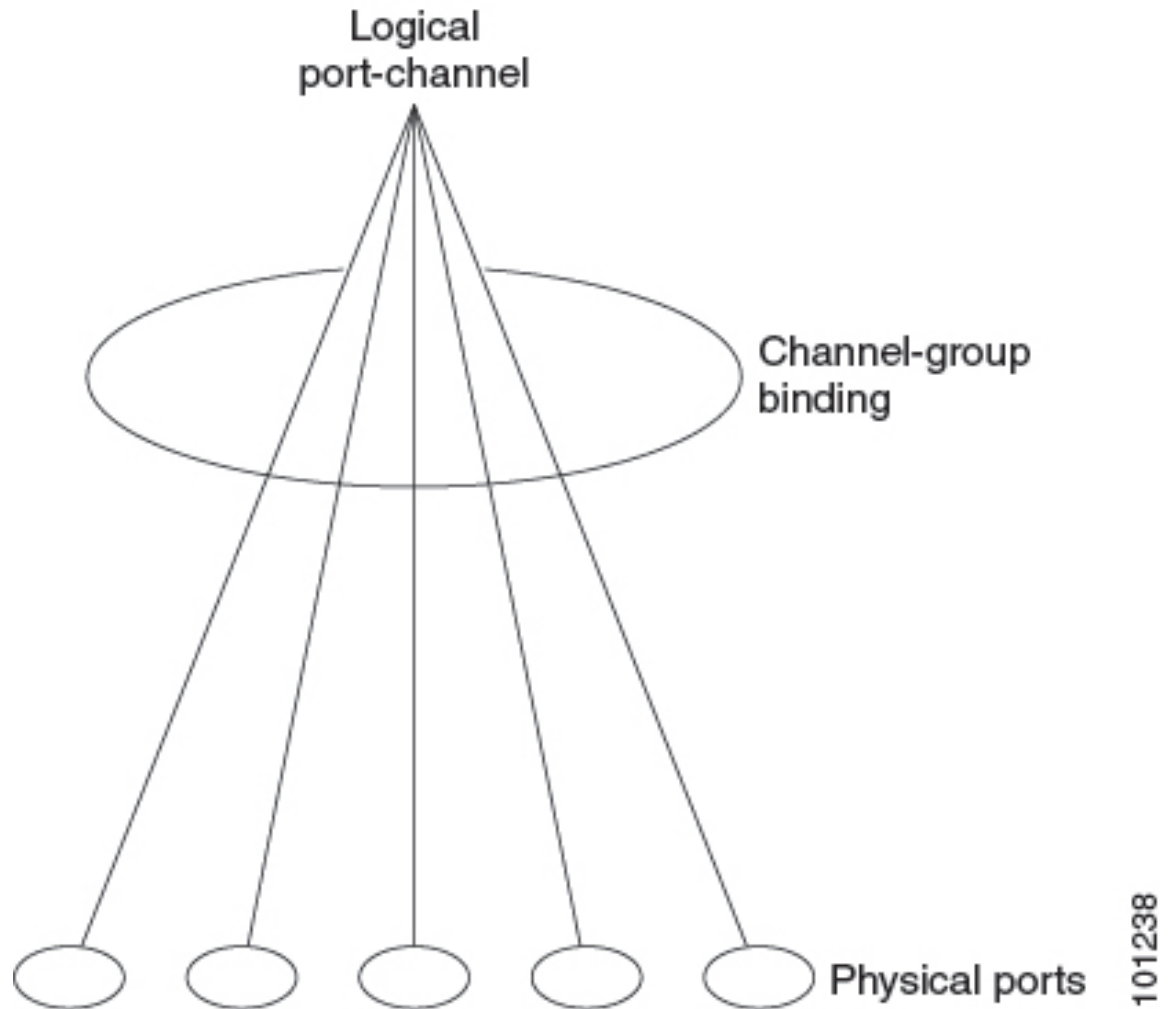


347662

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 24: 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 128. 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.

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 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

EtherChannel balances the traffic load across the links in a channel by reducing part of the binary pattern 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.

- VLAN-based load balancing is supported only on Cisco Catalyst 9500H switches.
- 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 calculated using these parameters. Any changes in one of these parameters will 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

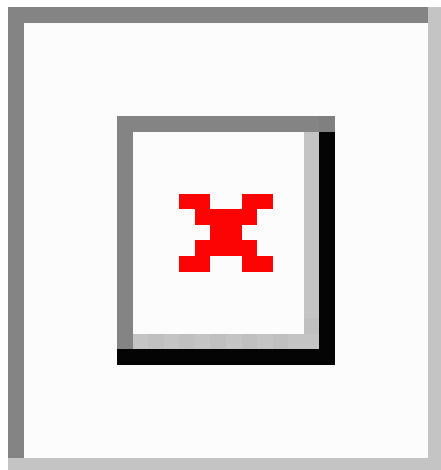
With VLAN ID based forwarding, packets are distributed across the ports in the EtherChannel based on the VLAN ID of the incoming packets and other parameters that are mentioned in the chosen load balance method. Packets with different VLAN IDs will use different ports in the channel and packets with 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 25: 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. |
| LACP port priority | 32768 on all ports. |
| LACP system priority | 32768. |

| Feature | Default Setting |
|----------------|--|
| LACP system ID | LACP system priority and the switch or stack MAC address. |
| Load-balancing | <p>Load distribution on the switch is based on the source-MAC address of the incoming packet.</p> <p>The source-MAC address for Cisco Catalyst 9500 Series switches is src-mac and src-dst-mixed-ip-port for Cisco Catalyst 9500 Series High Performance switches.</p> |

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 128 EtherChannels (non-StackWise Virtual setup) and 126 EtherChannels (StackWise Virtual setup) are supported on a switch or switch stack.
- Configure a PAgP EtherChannel with up to eight Ethernet ports of the same type.
- Configure a LACP EtherChannel with 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.
- 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 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. |

| | Command or Action | Purpose |
|---------------|---|--|
| 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. • 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 |

| | Command or Action | Purpose |
|---------------|---|--|
| | | <p>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.</p> <ul style="list-style-type: none"> • 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: Device> enable | <p>Enables privileged EXEC mode.</p> <p>Enter your password if prompted.</p> |
| 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 | <p>Specifies a physical port, and enters interface configuration mode.</p> <p>Valid interfaces include physical ports.</p> <p>For a PAgP EtherChannel, you can configure up to eight ports of the same type and speed for the same group.</p> <p>For a LACP EtherChannel, you can configure up to 16 Ethernet ports of the same type. Up to</p> |

| | Command or Action | Purpose |
|---------------|--|---|
| | | 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. • 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. |

| | Command or Action | Purpose |
|---------------|---|--|
| | | <ul style="list-style-type: none"> • 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 EtherChannel Load-Balancing

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

This task is optional.

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 { 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 for Cisco Catalyst 9500 Series Switches is src-mac . The default for Cisco Catalyst 9500 Series High Performance Switches is src-dst-mixed-ip-port . The VLAN-based load balancing keywords vlan-dst-ip , vlan-dst-mixed-ip-port , vlan-src-dst-ip , vlan-src-dst-mixed-ip-port , vlan-src-ip , and vlan-src-mixed-ip-port are supported only on Cisco Catalyst 9500 Series High Performance Switches. |

| | Command or Action | Purpose |
|--|-------------------|--|
| | | <p>Select one of these load-distribution methods:</p> <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. • 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. |

| | Command or Action | Purpose |
|---------------|---|--|
| | | <ul style="list-style-type: none"> • 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. |

Configuring EtherChannel Extended Load-Balancing

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

This task is optional.

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 } Example: Device(config)# port-channel load-balance extended dst-ip dst-mac src-ip | Configures an EtherChannel extended load-balancing method. The default for Cisco Catalyst 9500 Series Switches is src-mac . For Cisco Catalyst 9500 Series High Performance Switches, 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-port—Specifies the destination TCP/UDP port. • ipv6-label—Specifies the IPv6 flow label. This keyword is supported only on Cisco Catalyst 9500 Series Switches. |

| | Command or Action | Purpose |
|---------------|---|--|
| | | <ul style="list-style-type: none"> • l3-proto—Specifies the Layer 3 protocol. This keyword is supported only on Cisco Catalyst 9500 Series Switches. • 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: Device(config)# end | 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: 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 1/0/2 | Specifies the port for transmission, and enters interface configuration mode. |
| Step 4 | pagp learn-method physical-port Example: Device(config-if)# pagp learn-method physical port | Selects the PAgP learning method. 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. |

| | Command or Action | Purpose |
|---------------|---|--|
| | | <p>Selects physical-port to connect with another device that is a physical learner.</p> <p>Make sure to configure the port-channel load-balance global configuration command to src-mac.</p> <p>The learning method must be configured the same at both ends of the link.</p> |
| Step 5 | pagp port-priority <i>priority</i> Example: <pre>Device(config-if) # pagp port-priority 200</pre> | <p>Assigns a priority so that the selected port is chosen for packet transmission.</p> <p>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.</p> |
| Step 6 | end Example: <pre>Device(config-if) # end</pre> | 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 128. |
| 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. |

| | Command or Action | Purpose |
|---------------|---|---|
| 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 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. |

(Optional) Configuring the Link Aggregation Control Protocol System Priority

| | Command or Action | Purpose |
|---------------|--|---|
| 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 128. |
| 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. |
| 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 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 interface-id 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:

Example: Configuring Link Aggregation Control Protocol Hot-Standby Ports

```
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 Priority | Admin Key | Oper Key | Port Number | Port State |
|-------|-------|---------|-----------------------|--------------|-------------|----------------|---------------|
| Fa1/1 | FA | hot-sby | 30000* | 0x1 | 0x1 | 0x103 | 0x7 |
| Fa1/2 | SA | bndl | 32768 | 0x1 | 0x1 | 0x102 | 0x3D |

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      Po1 (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 9500 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 Everest 16.5.1a | EtherChannels | EtherChannel provides fault-tolerant high-speed links between switches, routers, and servers. Support for this feature was introduced only on the C9500-12Q, C9500-16X, C9500-24Q, C9500-40X models of the Cisco Catalyst 9500 Series Switches. |
| Cisco IOS XE Fuji 16.8.1a | EtherChannels | Support for this feature was introduced only on the C9500-32C, C9500-32QC, C9500-48Y4C, and C9500-24Y4C models of the Cisco Catalyst 9500 Series Switches. |
| 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. |

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 PTP, on page 127](#)
- [Information About Precision Time Protocol, on page 128](#)
- [How to Configure Precision Time Protocol, on page 136](#)
- [Configuration Examples for PTP, on page 142](#)
- [Feature History for Precision Time Protocol, on page 149](#)

Restrictions and Limitations for PTP

- The output of the **show clock** command on the device and PTP servo clock displayed in the output of the **show platform software fed switch active ptp domain** command are not synchronized. These are two different clocks used on the switch.
- Inter-VLAN is not supported in PTP Transparent Clock Mode.
- PTP is not supported on Cisco StackWise Virtual enabled devices.
- 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)

- You cannot configure PTP on any port of a supervisor module.
- Stateful Switchover (SSO) does not support PTP. The PTP protocol restarts after a switchover.
- MACsec with Precision Time Protocol (PTP) is not supported.

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 [Precision Time Protocol Version 2 Message Types, on page 129](#). 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 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 26: PTP Network



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 assigns 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 130](#)
- [Peer-to-Peer Delay Mechanism, on page 131](#)

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

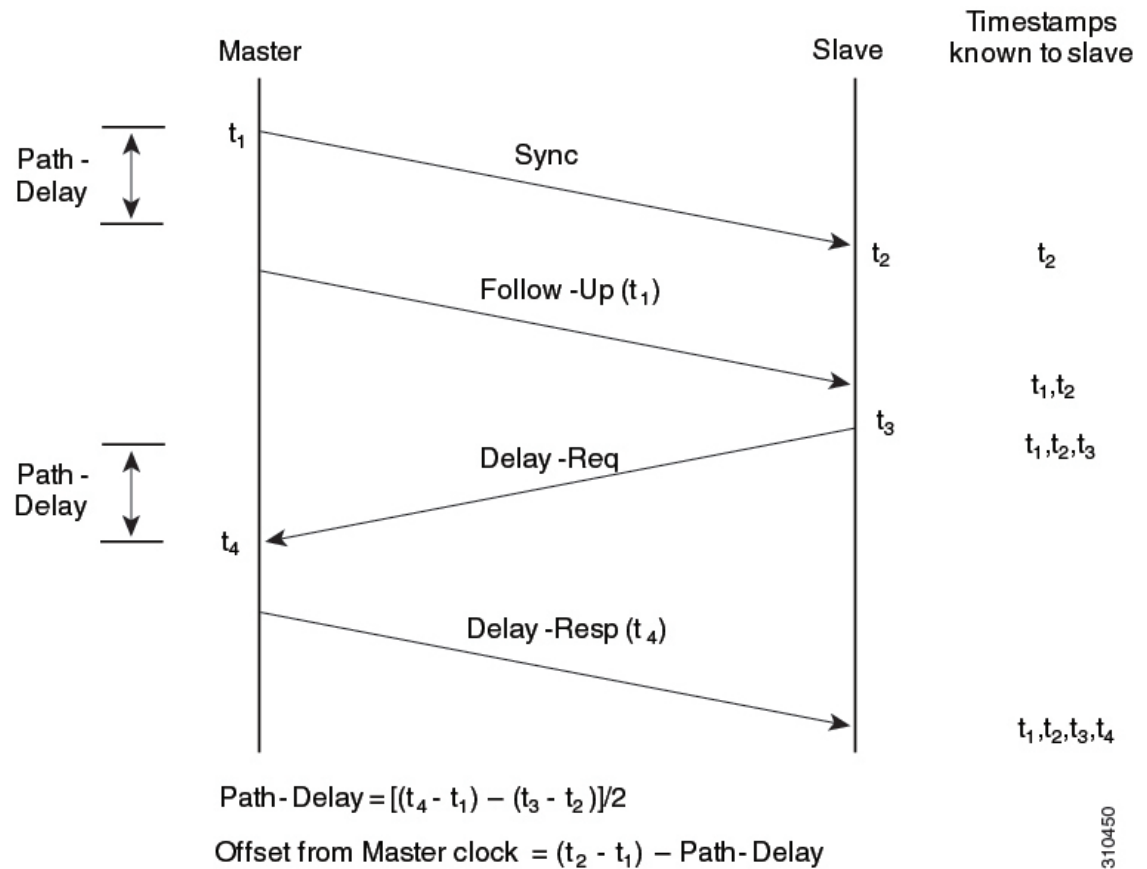
These messages are sent in the following sequence:

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 27: 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 $\text{mean_path_delay}/2$, 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:

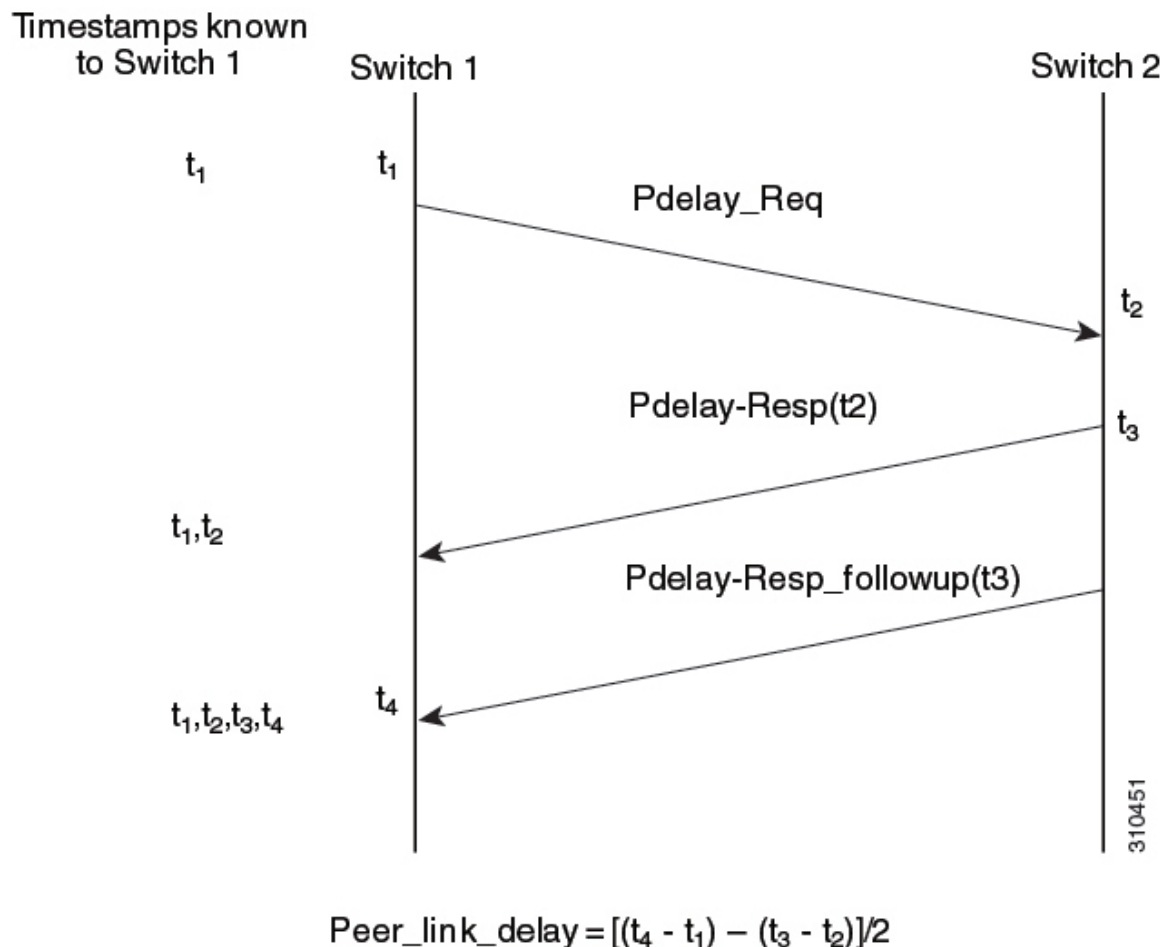
- Pdelay_Req
- Pdelay_Resp
- Pdelay_Resp_Follow_Up

These messages are sent in the following sequence:

1. Port 1 generates timestamp t_1 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 28: 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

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.

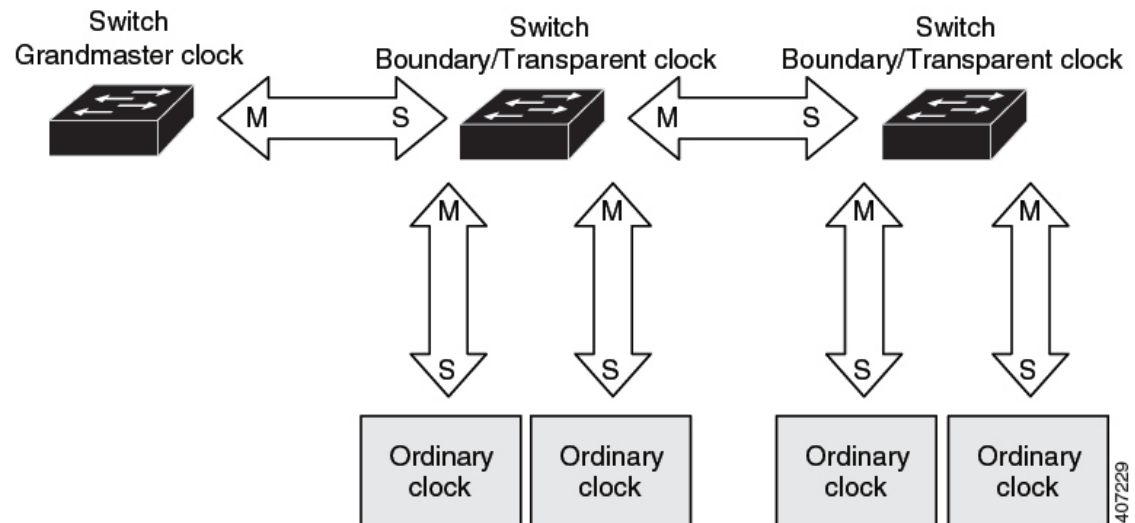
Peer-to-peer (P2P) transparent clocks measure PTP event message transit time the same way E2E transparent clocks do. In addition, P2P transparent clocks measure the upstream link delay. The upstream link delay is

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 29: 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.

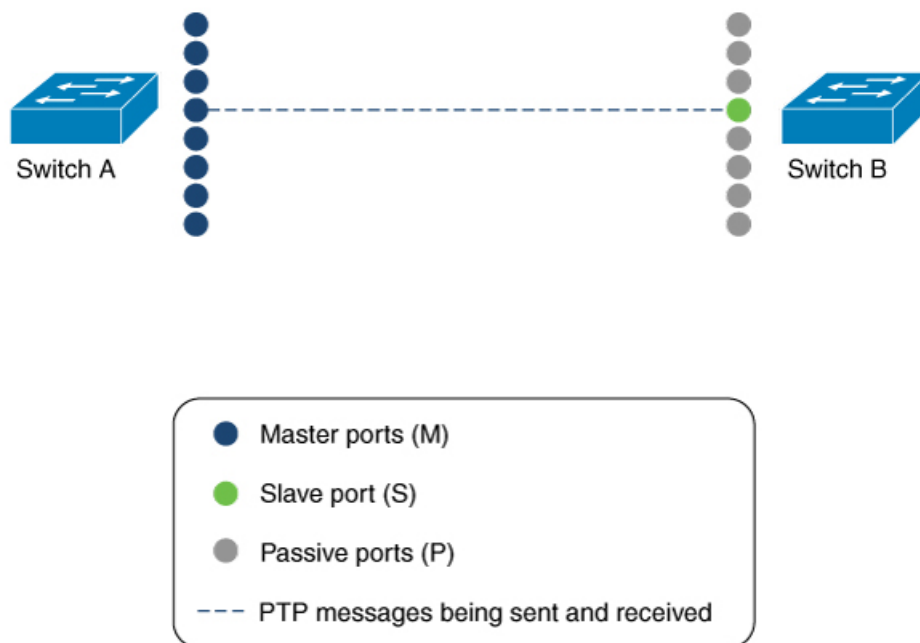
Precision Time Protocol on an EtherChannel Interface

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.

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 sends 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 30: Precision Time Protocol on an EtherChannel Interface



356549

How to Configure Precision Time Protocol

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 p2pttransparent} Example: Device(config)# ptp mode boundary delay-req Device(config)# ptp mode boundary pdelay-req Device(config)# ptp mode e2transparent Device(config)# ptp mode p2pttransparent | Specifies the synchronization clock mode: <ul style="list-style-type: none"> • 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 skew that is caused by clock offsets and network delays. Use this mode when overload or heavy load conditions produce significant delay jitter • 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. • p2pttransparent 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. |

| | Command or Action | Purpose |
|---------------|--|---|
| | | 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. |
| Step 4 | [no]ptp domain <i>value</i> Example: Device(config)# ptp domain 8 | Configures the domain value on PTP. <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 | 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 TenGigabitEthernet1/0/1 | 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)# ptp vlan 5 | 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: | Returns to privileged EXEC mode. |

| | Command or Action | Purpose |
|--|--------------------------------|---------|
| | Device(config-if) # end | |

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. |

| | Command or Action | Purpose |
|---------------|---|---|
| | | <ul style="list-style-type: none"> vlan PTP messages will carry the IP configured on the SVI interface corresponding to the port. <p>Note You can use no ptp source command as default.</p> |
| Step 4 | end Example: Device(config-if) # end | Returns to privileged EXEC mode. |

Configuring PTP 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 converted to seconds with a logarithmic scale:

| Value Entered | Logarithmic Calculation | Value in Seconds |
|---------------|-------------------------|------------------|
| -1 | 2^{-1} | 1/2 |
| 0 | 2^0 | 1 |

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 gigabitethernet2/0/1 | Specifies the physical port to be configured, and enters interface configuration mode. |
| Step 4 | ptp announce {interval value timeout count} Example: | (Optional) Configures the interval between PTP announce messages on an interface or the |

| | Command or Action | Purpose |
|---------------|--|--|
| | Device(config-if) # ptp announce interval 1 | number of PTP intervals before a timeout occurs on an interface. <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 | ptp sync {interval value limit offset-value} Example: Device(config-if) # ptp sync interval 1 | (Optional) Configures the interval between PTP synchronization messages on an interface. <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 | ptp delay-req interval value Example: Device(config-if) # ptp delay-req interval 1 | (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). |
| Step 7 | ptp pdelay-req interval value Example: Device(config-if) # ptp pdelay-req interval 1 | (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). |
| Step 8 | end Example: Device(config-if) # end | Returns to privileged EXEC mode. |

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 | 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 | ptp priority1 value Example: Device(config)# ptp priority1 120 | 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)# ptp priority2 120 | 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)# exit | Returns to global configuration mode. |

Configuration Examples for PTP

The following sections provide various PTP configuration examples:

Example: Layer 2 and Layer 3 PTP Configurations

Example

show ptp port interface *interface-name*

To verify PTP port state, use **show ptp port interface** *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 switch 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 switch active ptp domain 0** command.

```
Device# show platform software fed switch 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.

Example

show ptp port interface *interface-name*

To verify PTP port state, use **show ptp port interface** *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
```

Example: Configuring Precision Time Protocol on an EtherChannel Interface**show platform software fed switch 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 switch active ptp domain 0** command.

```
Device# show platform software fed switch 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 15: 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. |

Example: Configuring Precision Time Protocol on an EtherChannel Interface**Master Clock**

The following command verifies the PTP state on an interface:

```
Device# show ptp brief | exclude FAULTY
Interface          Domain    PTP State
TenGigE1/0/39      0        MASTER
TenGigE1/0/44      0        MASTER
TenGigE1/0/48      0        MASTER
```

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: 3
Number of aggregators:          3

Group  Port-channel  Protocol    Ports
-----+-----+-----+-----
1      Po1 (SU)          LACP        Hu1/0/39 (P)   Hu1/0/44 (P)
                        Hu1/0/48 (P)

```

The following command verifies port state of each 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

```

Example: Configuring Precision Time Protocol on an EtherChannel Interface

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 Fuji 16.8.1a | 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. Support for PTP on Layer 2 ports was introduced. Support for this feature was introduced only on the C9500-12Q, C9500-16X, C9500-24Q, C9500-40X models of the Cisco Catalyst 9500 Series Switches. |
| Cisco IOS XE Gibraltar 16.12.1 | IEEE 1588v2, Precision Time Protocol (PTP) support | PTP Version 2 (PTPv2) was introduced. Support for this feature was introduced only on the C9500-32C, C9500-32QC, C9500-48Y4C, and C9500-24Y4C models of the Cisco Catalyst 9500 Series Switches. |
| Cisco IOS XE Gibraltar 16.12.1 | PTP on native Layer 3 ports | Support for PTP on native Layer 3 ports was introduced. Support for this feature was introduced on all the models of the Cisco Catalyst 9500 Series Switches. |
| Cisco IOS XE Amsterdam 17.2.1 | IEEE 1588v2 PTP on EtherChannel Interfaces | Support for PTP on an EtherChannel was introduced. Support for this feature was introduced on all the models of the Cisco Catalyst 9500 Series Switches. |
| Cisco IOS XE Bengaluru 17.5.1 | Port primary | Support was introduced to set a port permanently as primary (master) using the ptp role primary command. |

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 Generalized Precision Time Protocol

- [Restrictions for Generalized Precision Time Protocol over Layer 3 Unicast, on page 151](#)
- [Information About Generalized Precision Time Protocol, on page 151](#)
- [How to Configure Generalized Precision Time Protocol, on page 154](#)
- [Monitoring Generalized Precision Time Protocol, on page 158](#)
- [Verifying Generalized Precision Time Protocol over Layer 3 Unicast Configuration, on page 158](#)
- [Configuration Examples for Generalized Precision Time Protocol, on page 158](#)
- [Feature History for Generalized Precision Time Protocol, on page 165](#)

Restrictions for Generalized Precision Time Protocol over Layer 3 Unicast

Generalized Precision Time Protocol over Layer 3 Unicast feature is not supported on stacked devices.

Information About Generalized Precision Time Protocol

Generalized precision time protocol (PTP) is an IEEE 802.1AS standard that provides a mechanism to synchronize the clocks of the bridges and end-point devices in a network. Generalized PTP defines the mechanism to elect the grandmaster clock (using Best Master Clock Algorithm [BMCA]) among the time-aware bridges and the talker and listener. The grandmaster is the root of the timing hierarchy that gets established in a time-aware network and distributes time to the nodes below to enable synchronization.

Time synchronization also requires determining the link delay and switch delays in the network nodes. A generalized PTP switch is an IEEE 1588 boundary clock, which also determines the link delay using the peer-to-peer delay mechanism. The delays that are computed are included in the correction field of the PTP messages and relayed to the endpoints. The talker and listener use this generalized PTP time as a shared clock reference, which is used to relay and recover the media clock. Generalized PTP currently defines only domain 0, which is what the generalized PTP switch supports.

The peer-to-peer delay mechanism runs on Spanning Tree Protocol-blocked (STP-blocked) ports as well. No other PTP messages are sent over blocked ports.

In a PTP domain, BMCA organizes clocks and ports in an hierarchical fashion, which includes clocks and port states:

Clocks

- Grandmaster (GM or GMC)
- Boundary Clock (BC)

Port States

- Master (M)
- Slave (S)
- Passive (P)

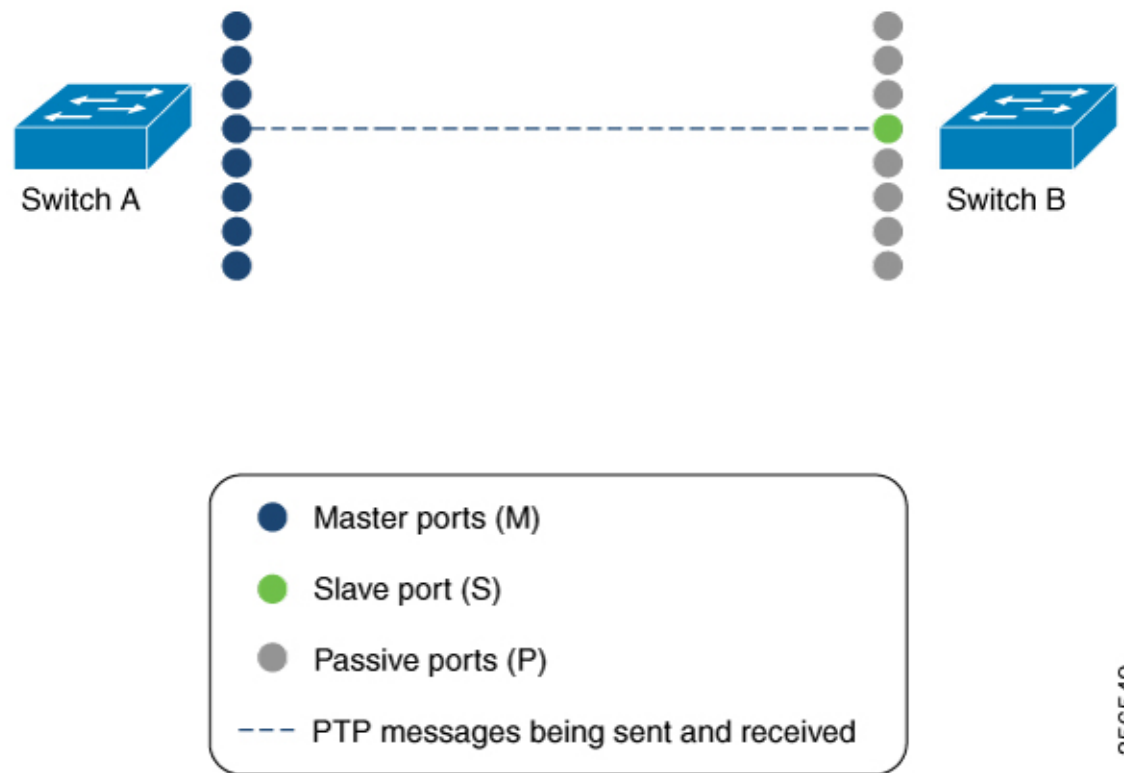
Generalized Precision Time Protocol on an EtherChannel Interface

An EtherChannel interface allows multiple physical Ethernet links to be combined into one logical channel. Configuring an 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 generalized PTP is configured.

For example, in [Figure 31: Generalized Precision Time Protocol on an EtherChannel Interface](#) shows that two switches (Switch A and Switch B) are connected through an eight-member EtherChannel. If you consider Switch A as the master clock, all the ports that are a 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 the 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 receive generalized PTP messages. In Switch B, only the slave port exchanges generalized PTP messages. There is no exchange of generalized PTP messages in the passive ports.

Figure 31: Generalized Precision Time Protocol on an EtherChannel Interface



356549

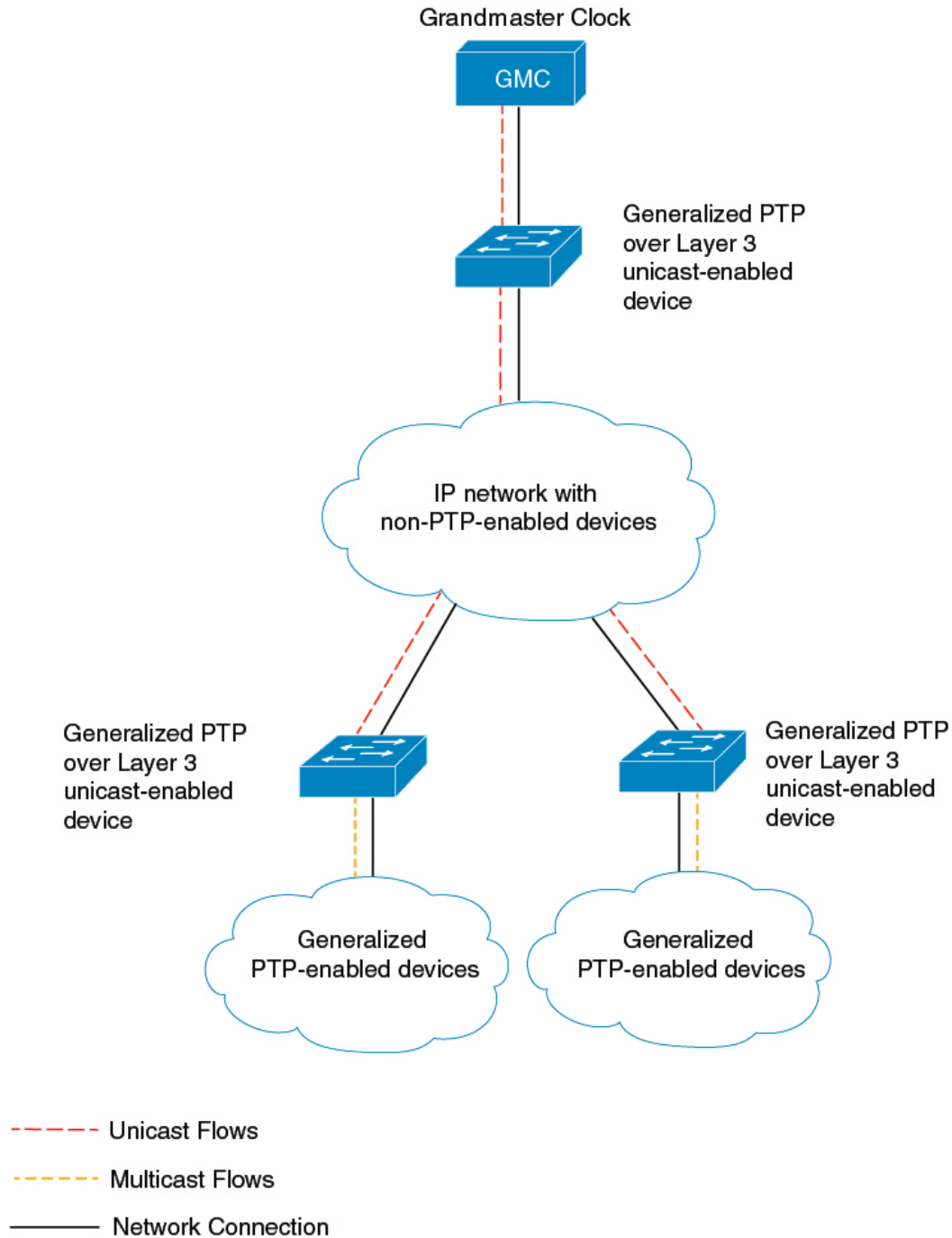
Generalized Precision Time Protocol over Layer 3 Unicast

A generalized PTP network consists of Layer 2 devices that are connected to a grandmaster clock that is usually a high-precision clock such as GPS. But for generalized PTP networks that span across multiple floors or even across multiple buildings, configuring a high-precision grandmaster clock on each floor or building increases the cost of deployment. Also, such networks are connected over Layer 3 devices; all Layer 3 devices do not support generalized PTP and certain Layer 3 devices do not support multicast routing.

The Generalized Precision Time Protocol over Layer 3 Unicast feature is a solution introduced to support generalized PTP networks connected over Layer 3 devices. Layer 3 devices, such as the Cisco Catalyst 9500 Series Switches, are configured with this feature. A high-precision grandmaster clock is connected to the primary device that is enabled with this feature. Layer 3 devices that are enabled with this feature synchronize their clocks using PTP boundary clock's end-to-end delay mechanism messages. They also synchronize all the clocks in the generalized PTP networks that are connected to them.

The following figure displays a network, with generalized PTP over Layer 3 unicast configured:

Figure 32: Generalized PTP over Layer 3 Unicast



457814

How to Configure Generalized Precision Time Protocol

This section describes the various configurations available for generalized PTP.

Enabling Generalized Precision Time Protocol

To enable generalized PTP on a 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 | [no]ptp profile dot1as Example: Device(config)# ptp profile dot1as | Generalized PTP is enabled globally. Use the no form of this command to disable generalized PTP globally. |
| Step 4 | end Example: Device(config)# end | Returns to privileged EXEC mode. |

Enabling Generalized Precision Time Protocol on an Interface

To enable generalized PTP on an 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 interface-id Example: Device(config)# interface te1/1/1 | Defines the interface to be configured as a trunk, and enters interface configuration mode. The interface that you specify can be a part of an EtherChannel. |
| Step 4 | ptp enable Example: | Enables generalized PTP on all the interfaces. To disable generalized PTP on a port, use the no form of this command: |

| | Command or Action | Purpose |
|---------------|--|---|
| | Device (config-if) # ptp enable | Device (config-if) # no ptp enable |
| Step 5 | end Example: Device (config-if) # end | Returns to privileged EXEC mode. |

Configuring the Values of Precision Time Protocol Clocks

Follow these steps to configure the values of PTP clocks, priority1 and priority2:

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 priority1 value Example: Device (config) # ptp priority1 120 | 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) # ptp priority2 120 | 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) # exit | Returns to global configuration mode. |

Configuring Generalized Precision Time Protocol over Layer 3 Unicast

To configure generalized PTP over Layer 3 unicast, perform this procedure.



Note You can configure more than one IPv4 unicast connection that connects to a different boundary clock under the same property name.

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(config)# configure terminal | Enters global configuration mode. |
| Step 3 | ptp property word Example: Device(config)# ptp property cisco1 | Sets the PTP property name and enters property configuration mode. |
| Step 4 | transport unicast ipv4 local loopback value Example: Device(config-property)# transport unicast ipv4 local loopback 0 | Configures a unicast IPv4 connection from a loopback interface and enters property transport sub-config mode. <i>value</i> : Loopback interface number. The maximum number of sessions that are supported is 127. |
| Step 5 | peer {ip ip_address vrf word ip ip_address} Example: Device(config-property-transport)# peer ip 192.0.2.1 | Connects to a peer PTP-aware device. <ul style="list-style-type: none"> • vrf word: Default virtual routing and forwarding (VRF) or user-defined VRF. • ip ip_address: IP address of a peer PTP device. |
| Step 6 | source ip interface interface_id Example: Device(config-property-transport)# source ip interface GigabitEthernet 1/0/1 | (Optional) Configures the source IP address instead of the loopback interface ID. <i>interface_id</i> : Source IP address. |
| Step 7 | exit Example: Device(config-property-transport)# exit | Exits property transport sub-config mode and returns to property mode. |
| Step 8 | exit Example: Device(config-property)# exit | Exits property mode and returns to global configuration mode. |
| Step 9 | ptp dot1as extend property word Example: Device(config)# ptp dot1as extend property cisco1 | Enables IEEE 802.1AS profile extending on the configured PTP property name. |

Monitoring Generalized Precision Time Protocol

Use the following commands in privileged EXEC mode to monitor generalized PTP.

Table 16: Commands to Monitor Generalized Precision Time Protocol

| Command | Purpose |
|--|---|
| show ptp brief | Displays the brief status of PTP on all interfaces. |
| show ptp clock | Displays PTP clock information. |
| show ptp parent | Displays the parent clock information. |
| show ptp port | Displays the PTP port information. |
| show platform software fed switch active ptp if-id {interface-id} | Displays details about the PTP status on a port. |

Verifying Generalized Precision Time Protocol over Layer 3 Unicast Configuration

Use the following commands in privileged EXEC mode to verify generalized PTP over Layer 3 unicast configurations.

Table 17: Commands to Verify Generalized PTP over Layer 3 Unicast Configuration

| Command | Purpose |
|---|--|
| show ptp transport properties | Displays the PTP profile and properties, including the transport method, loopback interface number, and PTP state. |
| show ptp port loopback value | Displays the PTP configurations of the specified loopback interface. |
| show platform software fed active ptp interface loopback value | Displays the PTP connection details and events of the specified loopback interface. |

Configuration Examples for Generalized Precision Time Protocol

The following sections provide configuration examples for generalized PTP.

Example: Verifying Generalized Precision Time Protocol

The following is a sample output of the **show ptp brief** command:

```
Device# show ptp brief
Interface                               Domain    PTP State
FortyGigabitEthernet1/1/1              0         FAULTY
FortyGigabitEthernet1/1/2              0         SLAVE
GigabitEthernet1/1/1                   0         FAULTY
GigabitEthernet1/1/2                   0         FAULTY
GigabitEthernet1/1/3                   0         FAULTY
GigabitEthernet1/1/4                   0         FAULTY
TenGigabitEthernet1/0/1                 0         FAULTY
TenGigabitEthernet1/0/2                 0         FAULTY
TenGigabitEthernet1/0/3                 0         MASTER
TenGigabitEthernet1/0/4                 0         FAULTY
TenGigabitEthernet1/0/5                 0         FAULTY
TenGigabitEthernet1/0/6                 0         FAULTY
TenGigabitEthernet1/0/7                 0         MASTER
TenGigabitEthernet1/0/8                 0         FAULTY
TenGigabitEthernet1/0/9                 0         FAULTY
TenGigabitEthernet1/0/10                0         FAULTY
TenGigabitEthernet1/0/11                0         MASTER
TenGigabitEthernet1/0/12                0         FAULTY
TenGigabitEthernet1/0/13                0         FAULTY
TenGigabitEthernet1/0/14                0         FAULTY
TenGigabitEthernet1/0/15                0         FAULTY
TenGigabitEthernet1/0/16                0         FAULTY
TenGigabitEthernet1/0/17                0         FAULTY
TenGigabitEthernet1/0/18                0         FAULTY
TenGigabitEthernet1/0/19                0         MASTER
TenGigabitEthernet1/0/20                0         FAULTY
TenGigabitEthernet1/0/21                0         FAULTY
TenGigabitEthernet1/0/22                0         FAULTY
TenGigabitEthernet1/0/23                0         FAULTY
TenGigabitEthernet1/0/24                0         FAULTY
TenGigabitEthernet1/1/1                 0         FAULTY
TenGigabitEthernet1/1/2                 0         FAULTY
TenGigabitEthernet1/1/3                 0         FAULTY
TenGigabitEthernet1/1/4                 0         FAULTY
TenGigabitEthernet1/1/5                 0         FAULTY
TenGigabitEthernet1/1/6                 0         FAULTY
TenGigabitEthernet1/1/7                 0         FAULTY
TenGigabitEthernet1/1/8                 0         FAULTY
```

The following is a sample output of the **show ptp clock** command:

```
Device# show ptp clock
PTP CLOCK INFO
PTP Device Type: Boundary clock
PTP Device Profile: IEEE 802/1AS Profile
Clock Identity: 0x4:6C:9D:FF:FE:4F:95:0
Clock Domain: 0
Number of PTP ports: 38
PTP Packet priority: 4
Priority1: 128
Priority2: 128
Clock Quality:
  Class: 248
  Accuracy: Unknown
  Offset (log variance): 16640
Offset From Master(ns): 0
```

Example: Verifying Generalized Precision Time Protocol

```

Mean Path Delay(ns): 0
Steps Removed: 3
Local clock time: 00:12:13 UTC Jan 1 1970

```

The following is a sample output of the **show ptp parent** command:

```

Device# show ptp parent
PTP PARENT PROPERTIES
  Parent Clock:
    Parent Clock Identity: 0xB0:7D:47:FF:FE:9E:B6:80
    Parent Port Number: 3
    Observed Parent Offset (log variance): 16640
    Observed Parent Clock Phase Change Rate: N/A

  Grandmaster Clock:
    Grandmaster Clock Identity: 0x4:6C:9D:FF:FE:67:3A:80
    Grandmaster Clock Quality:
      Class: 248
      Accuracy: Unknown
      Offset (log variance): 16640
      Priority1: 0
      Priority2: 128

```

The following is a sample output of the **show ptp port** command:

```

Device# show ptp port
PTP PORT DATASET: FortyGigabitEthernet1/1/1
  Port identity: clock identity: 0x4:6C:9D:FF:FE:4E:3A:80
  Port identity: port number: 1
  PTP version: 2
  Port state: FAULTY
  Delay request interval(log mean): 5
  Announce receipt time out: 3
  Peer mean path delay(ns): 0
  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

PTP PORT DATASET: FortyGigabitEthernet1/1/2
  Port identity: clock identity: 0x4:6C:9D:FF:FE:4E:3A:80
  Port identity: port number: 2
  PTP version: 2
  Port state: FAULTY
  Delay request interval(log mean): 5
  Announce receipt time out: 3
  Peer mean path delay(ns): 0
  Announce interval(log mean): 1
--More--

```

The following is a sample output of the **show ptp port** command for an interface:

```

Device# show ptp port gi1/0/26
PTP PORT DATASET: GigabitEthernet1/0/26
  Port identity: clock identity: 0x4:6C:9D:FF:FE:4E:3A:80
  Port identity: port number: 28
  PTP version: 2
  Port state: MASTER
  Delay request interval(log mean): 5
  Announce receipt time out: 3
  Peer mean path delay(ns): 0
  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
```

The following is a sample output of the **show platform software fed switch active ptp if-id** command for an interface:

```
Device# show platform software fed switch active ptp if-id 0x20
Displaying port data for if_id 20
=====

Port Mac Address 04:6C:9D:4E:3A:9A
Port Clock Identity 04:6C:9D:FF:FE:4E:3A:80
Port number 28
PTP Version 2
domain_value 0
dot1as capable: FALSE
sync_recpt_timeout_time_interval 375000000 nanoseconds
sync_interval 125000000 nanoseconds
neighbor_rate_ratio 0.000000
neighbor_prop_delay 0 nanoseconds
compute_neighbor_rate_ratio: TRUE
compute_neighbor_prop_delay: TRUE
port_enabled: TRUE
ptt_port_enabled: TRUE
current_log_pdelay_req_interval 0
pdelay_req_interval 0 nanoseconds
allowed_lost_responses 3
neighbor_prop_delay_threshold 2000 nanoseconds
is_measuring_delay : FALSE
Port state: : MASTER
sync_seq_num 22023
delay_req_seq_num 23857
num sync messages transmitted 0
num sync messages received 0
num followup messages transmitted 0
num followup messages received 0
num pdelay requests transmitted 285695
num pdelay requests received 0
num pdelay responses transmitted 0
num pdelay responses received 0
num pdelay followup responses transmitted 0
num pdelay followup responses received 0
```

Example: Verifying Generalized Precision Time Protocol on an EtherChannel Interface

The following examples show how to verify generalized PTP on an EtherChannel interface (see [Figure 31: Generalized Precision Time Protocol on an EtherChannel Interface](#)).

Master Clock

The following is a sample output of the **show ptp brief** command used to verify the PTP state on an interface:

```
Device# show ptp brief | exclude FAULTY
Interface          Domain    PTP State
TenGigE1/0/39      0         MASTER
```

Example: Verifying Generalized Precision Time Protocol on an EtherChannel Interface

```
TenGigE1/0/44          0          MASTER
TenGigE1/0/48          0          MASTER
```

The following is a sample output of the **show etherchannel summary** command used to verify if the 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: 3
Number of aggregators:          3

Group  Port-channel  Protocol    Ports
-----+-----+-----+-----
1      Po1(SU)        LACP        Hu1/0/39(P)  Hu1/0/44(P)
                        Hu1/0/48(P)
```

The following is a sample output of the **show ptp port** command used to verify the port state of each 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 is a sample output of the **show ptp brief** command 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 is a sample output of the **show etherchannel summary** command used to verify if the 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 is a sample output of the **show ptp port** command used to verify the 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

```

Example: Configuring Generalized Precision Time Protocol over Layer 3 Unicast

```

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

```

Example: Configuring Generalized Precision Time Protocol over Layer 3 Unicast

The following examples shows how to configure generalized PTP over Layer 3 unicast on Device 1 and Device 2:

Figure 33: Example for Generalized PTP over Layer 3 Unicast



The following example shows how to configure generalized PTP over Layer 3 unicast on Device 1:

```

Device1> enable
Device1# configure terminal
Device1(config)# interface Loopback0
Device1(config-if)# ip address 192.0.2.1 255.255.255.255
Device1(config-if)# exit
Device1(config)# ptp property gptpproperty
Device1(config-property)# transport unicast ipv4 local Loopback0
Device1(config-property-transport)# peer ip 198.51.100.1
Device1(config-property-transport)# exit
Device1(config-property)# exit
Device1(config)# ptp dotlas extend property gptpproperty
Device1(config)# end

```

The following example shows how to configure generalized PTP over Layer 3 unicast on Device 2:

```
Device2> enable
Device2# configure terminal
Device2(config)# interface Loopback0
Device2(config-if)# ip address 198.51.100.1 255.255.255.255
Device2(config-if)# exit
Device2(config)# ptp property gptpproperty
Device2(config-property)# transport unicast ipv4 local Loopback0
Device2(config-property-transport)# peer ip 192.0.2.1
Device2(config-property-transport)# exit
Device2(config-property)# exit
Device2(config)# ptp dot1as extend property gptpproperty
Device2(config)# end
```

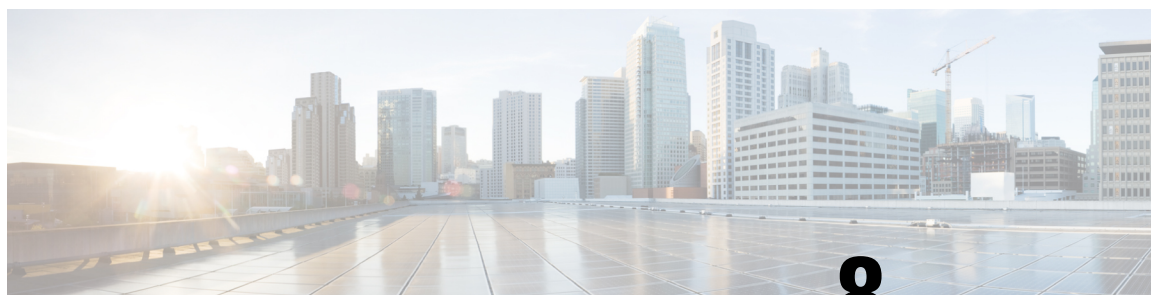
Feature History for Generalized Precision Time Protocol

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

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

| Release | Feature | Feature Information |
|-------------------------------|--|---|
| Cisco IOS XE Fuji 16.8.1a | Generalized Precision Time Protocol | Generalized Precision Time Protocol (PTP) is an IEEE 802.1AS standard that provides a mechanism to synchronize the clocks of the bridges and end-point devices in a network. Support for this feature was introduced on all the models of the Cisco Catalyst 9500 Series Switches. |
| Cisco IOS XE Amsterdam 17.2.1 | IEEE802.1AS (gPTP) support on EtherChannel Interfaces | From this release the interface on which you configure generalized PTP can be part of an EtherChannel. Support for this feature was introduced on all the models of the Cisco Catalyst 9500 Series Switches. |
| Cisco IOS XE Bengaluru 17.5.1 | Generalized Precision Time Protocol over Layer 3 Unicast | Generalized PTP over Layer 3 Unicast feature allows message-based synchronization across non-PTP-enabled devices and with unicast PTP configured on Layer 3 devices. |

Use the 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 Resilient Ethernet Protocol

- [Information About Resilient Ethernet Protocol, on page 167](#)
- [How to Configure Resilient Ethernet Protocol, on page 172](#)
- [Monitoring Resilient Ethernet Protocol Configurations, on page 181](#)
- [Additional References for Resilient Ethernet Protocol, on page 182](#)
- [Feature History for Resilient Ethernet Protocol , on page 182](#)

Information About Resilient Ethernet Protocol



Note The Resilient Ethernet Protocol is not supported in the Cisco IOS XE Bengaluru 17.4.x release.

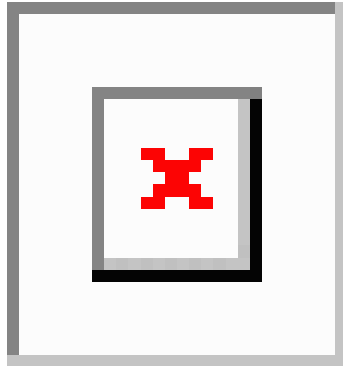
Resilient Ethernet Protocol (REP) is a Cisco proprietary protocol that provides an alternative to Spanning Tree Protocol (STP) to control network loops, handle link failures, and improve convergence time. REP controls a group of ports connected in a segment, ensures that the segment does not create any bridging loops, and responds to link failures within the segment. REP provides a basis for constructing more complex networks and supports VLAN load balancing.



Note The feature is supported on Cisco Catalyst Series Switches with the Network Essentials license.

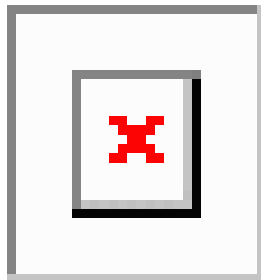
REP segment is a chain of ports connected to each other and configured with a segment ID. Each segment consists of standard (non-edge) segment ports and two user-configured edge ports. A switch can have no more than two ports that belong to the same segment, and each segment port can have only one external neighbor. A segment can go through a shared medium, but on any link, only two ports can belong to the same segment. REP is supported only on Trunk ports.

The figure below shows an example of a segment consisting of six ports spread across four switches. Ports E1 and E2 are configured as edge ports. When all ports are operational (as in the segment on the left), a single port is blocked, shown by the diagonal line. This blocked port is also known as the Alternate port (ALT port). When there is a failure in the network, the blocked port returns to the forwarding state to minimize network disruption.

Figure 34: REP Open Segment

The segment shown in the figure above is an open segment; there is no connectivity between the two edge ports. The REP segment cannot cause a bridging loop, and you can safely connect the segment edges to any network. All hosts connected to switches inside the segment have two possible connections to the rest of the network through the edge ports, but only one connection is accessible at any time. If a failure occurs on any segment or on any port on a REP segment, REP unblocks the ALT port to ensure that connectivity is available through the other gateway.

The segment below is a closed segment, also known as Ring Segment, with both edge ports located on the same router. With this configuration, you can create a redundant connection between any two routers in the segment.

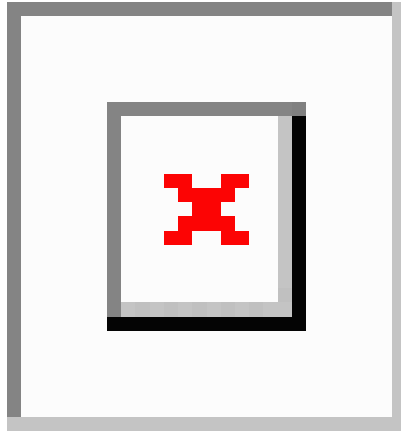
Figure 35: REP Ring Segment

REP segments have the following characteristics:

- If all ports in a segment are operational, one port (referred to as the ALT port) is in the blocked state for each VLAN. If VLAN load balancing is configured, two ALT ports in the segment control the blocked state of VLANs.
- If a port is not operational, and cause a link failure, all ports forward traffic on all VLANs to ensure connectivity.
- In case of a link failure, alternate ports are unblocked as quickly as possible. When the failed link is restored, a logically blocked port per VLAN is selected with minimal disruption to the network.

You can construct almost any type of network based on REP segments.

In access ring topologies, the neighboring switch might not support REP as shown in the figure below. In this case, you can configure the non-REP facing ports (E1 and E2) as edge no-neighbor ports. The edge no-neighbor port can be configured to send an STP topology change notice (TCN) towards the aggregation switch.

Figure 36: Edge No-Neighbor Ports

REP has these limitations:

- You must configure each segment port; an incorrect configuration can cause forwarding loops in the networks.
- REP can manage only a single failed port within the segment; multiple port failures within the REP segment cause loss of network connectivity.
- You should configure REP only in networks with redundancy. Configuring REP in a network without redundancy causes loss of connectivity.
- REP is not supported in a Cisco StackWise Virtual (SVL) solution.

Link Integrity

REP does not use an end-to-end polling function between edge ports to verify link integrity. It implements local link failure detection. The REP Link Status Layer (LSL) detects its REP-aware neighbor and establishes connectivity within the segment. All the VLANs are blocked on an interface until the neighbor is detected. After the neighbor is identified, REP determines which neighbor port should become the alternate port and which ports should forward traffic.

Each port in a segment has a unique port ID. The port ID format is similar to that used by the spanning tree algorithm: a port number (unique on the bridge) associated to a MAC address (unique in the network). When a segment port is coming up, its LSL starts sending packets that include the segment ID and the port ID. The port is declared as operational after it performs a three-way handshake with a neighbor in the same segment.

A segment port does not become operational if:

- No neighbor has the same segment ID.
- More than one neighbor has the same segment ID.
- A neighbor does not acknowledge a local port as a peer.

Each port creates an adjacency with its immediate neighbor. After the neighbor adjacencies are created, the ports negotiate with each other to determine the blocked port for the segment, which will function as the alternate port. All the other ports become unblocked. By default, REP packets are sent to a bridge protocol data unit-class MAC address. The packets can also be sent to a Cisco multicast address, which is used only

to send blocked port advertisement (BPA) messages when there is a failure in the segment. The packets are dropped by the devices not running REP.

Fast Convergence

REP runs on a physical link basis and not on a per-VLAN basis. Only one hello message is required for all the VLANs, and this reduces the load on the protocol. We recommend that you create VLANs consistently on all the switches in a given segment and configure the same allowed VLANs on the REP trunk ports. To avoid the delay introduced by relaying messages in software, REP also allows some packets to be flooded to a regular multicast address. These messages operate at the hardware flood layer (HFL) and are flooded to the entire network, not just the REP segment. Switches that do not belong to the segment treat them as data traffic. You can control flooding of these messages by configuring an administrative VLAN for the entire domain or for a particular segment.

VLAN Load Balancing

One edge port in the REP segment acts as the primary edge port; and another as the secondary edge port. It is the primary edge port that always participates in VLAN load balancing in the segment. REP VLAN balancing is achieved by blocking some VLANs at a configured alternate port and all the other VLANs at the primary edge port. When you configure VLAN load balancing, you can specify the alternate port in one of three ways:

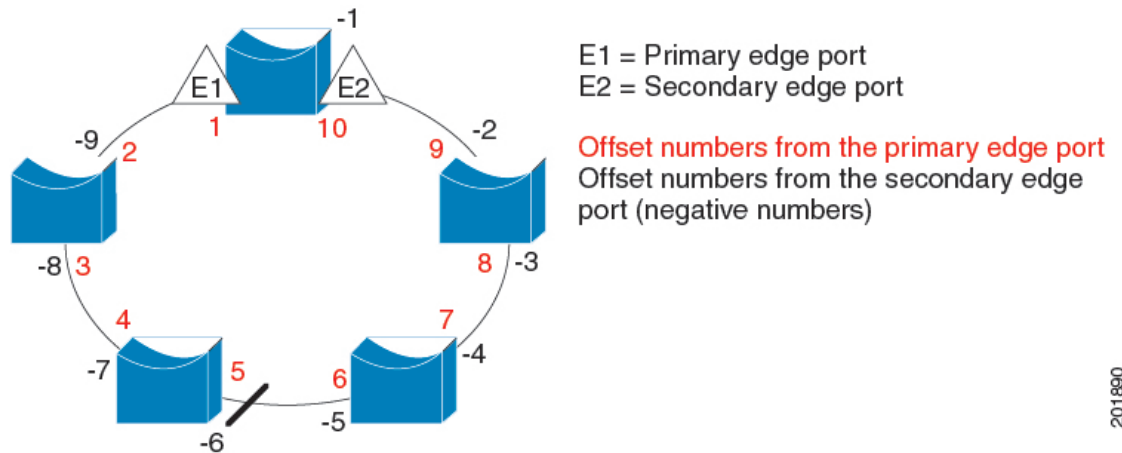
- By entering the port ID of the interface. To identify the port ID of a port in the segment, enter the **show interface rep detail** interface configuration command for the port.
- By entering the **preferred** keyword to select the port that you previously configured as the preferred alternate port with the **rep segment segment-id preferred** interface configuration command.
- By entering the neighbor offset number of a port in the segment, which identifies the downstream neighbor port of an edge port. The neighbor offset number range is -256 to +256; a value of 0 is invalid. The primary edge port has an offset number of 1; positive numbers above 1 identify downstream neighbors of the primary edge port. Negative numbers indicate the secondary edge port (offset number -1) and its downstream neighbors.



Note Configure offset numbers on the primary edge port by identifying a port's downstream position from the primary (or secondary) edge port. Never enter an offset value of 1 because that is the offset number of the primary edge port.

The following figure shows neighbor offset numbers for a segment, where E1 is the primary edge port and E2 is the secondary edge port. The red numbers inside the ring are numbers offset from the primary edge port; the black numbers outside of the ring show the offset numbers from the secondary edge port. Note that you can identify all the ports (except the primary edge port) by either a positive offset number (downstream position from the primary edge port) or a negative offset number (downstream position from the secondary edge port). If E2 became the primary edge port, its offset number would then be 1 and E1 would be -1.

Figure 37: Neighbor Offset Numbers in a Segment



201890

When the REP segment is complete, all the VLANs are blocked. When you configure VLAN load balancing, you must also configure triggers in one of two ways:

- Manually trigger VLAN load balancing at any time by entering the **rep preempt segment** *segment-id* privileged EXEC command on the switch that has the primary edge port.
- Configure a preempt delay time by entering the **rep preempt delay** *seconds* interface configuration command. After a link failure and recovery, VLAN load balancing begins after the configured preempt time period elapses. Note that the delay timer restarts if another port fails before the time has elapsed.



Note When VLAN load balancing is configured, it does not start working until triggered by either manual intervention or a link failure and recovery.

When VLAN load balancing is triggered, the primary edge port sends out a message to alert all the interfaces in the segment about the preemption. When the secondary port receives the message, the message is sent to the network to notify the alternate port to block the set of VLANs specified in the message and to notify the primary edge port to block the remaining VLANs.

You can also configure a particular port in the segment to block all the VLANs. Only the primary edge port initiates VLAN load balancing, which is not possible if the segment is not terminated by an edge port on each end. The primary edge port determines the local VLAN load-balancing configuration.

Reconfigure the primary edge port to reconfigure load balancing. When you change the load-balancing configuration, the primary edge port waits for the **rep preempt segment** command or for the configured preempt delay period after a port failure and recovery, before executing the new configuration. If you change an edge port to a regular segment port, the existing VLAN load-balancing status does not change. Configuring a new edge port might cause a new topology configuration.

Spanning Tree Interaction

REP does not interact with STP, but it can coexist. A port that belongs to a segment is removed from spanning tree control and STP BPDUs are not accepted or sent from segment ports. Therefore, STP cannot run on a segment.

To migrate from an STP ring configuration to REP segment configuration, begin by configuring a single port in the ring as part of the segment and continue by configuring contiguous ports to minimize the number of segments. Each segment always contains a blocked port, so multiple segments means multiple blocked ports and a potential loss of connectivity. When the segment has been configured in both directions up to the location of the edge ports, you then configure the edge ports.

Resilient Ethernet Protocol Ports

REP segments consist of Failed, Open, or Alternate ports:

- A port configured as a regular segment port starts as a failed port.
- After the neighbor adjacencies are determined, the port transitions to alternate port state, blocking all the VLANs on the interface. Blocked-port negotiations occur, and when the segment settles, one blocked port remains in the alternate role and all the other ports become open ports.
- When a failure occurs in a link, all the ports move to the Failed state. When the Alternate port receives the failure notification, it changes to the Open state, forwarding all the VLANs.

A regular segment port that is converted to an edge port, or an edge port that is converted to a regular segment port, does not always result in a topology change. If you convert an edge port into a regular segment port, VLAN load balancing is not implemented unless it has been configured. For VLAN load balancing, you must configure two edge ports in the segment.

A segment port that is reconfigured as a spanning tree port restarts according to the spanning tree configuration. By default, this is a designated blocking port. If PortFast is configured or if STP is disabled, the port goes into the forwarding state.

How to Configure Resilient Ethernet Protocol

A segment is a collection of ports that are connected to one another in a chain and configured with a segment ID. To configure REP segments, configure the REP administrative VLAN (or use the default VLAN 1) and then add the ports to the segment, using interface configuration mode. You should configure two edge ports in a segment, with one of them being the primary edge port and the other the secondary edge port by default. A segment should have only one primary edge port. If you configure two ports in a segment as primary edge ports, for example, ports on different switches, the REP selects one of them to serve as the segment's primary edge port. If required, you can configure the location to which segment topology change notices (STCNs) and VLAN load balancing are to be sent.

Default Resilient Ethernet Protocol Configuration

REP is disabled on all the interfaces. When enabled, the interface is a regular segment port unless it is configured as an edge port.

When REP is enabled, the task of sending segment topology change notices (STCNs) is disabled, all the VLANs are blocked, and the administrative VLAN is VLAN 1.

When VLAN load balancing is enabled, the default is manual preemption with the delay timer disabled. If VLAN load balancing is not configured, the default after manual preemption is to block all the VLANs in the primary edge port.

Resilient Ethernet Protocol Configuration Guidelines

Follow these guidelines when configuring REP:

- REP is supported on 1-Gigabit Ethernet and 10-Gigabit Ethernet interfaces.
- We recommend that you begin by configuring one port and then configure contiguous ports to minimize the number of segments and the number of blocked ports.
- If more than two ports in a segment fail when no external neighbors are configured, one port goes into a forwarding state for the data path to help maintain connectivity during configuration. In the **show rep interface** command output, the Port Role for this port shows as “Fail Logical Open”; the Port Role for the other failed port shows as “Fail No Ext Neighbor”. When the external neighbors for the failed ports are configured, the ports go through the alternate port state transitions and eventually go to an open state or remain as the alternate port, based on the alternate port selection mechanism.
- REP ports must be Layer 2 IEEE 802.1Q or Trunk ports.
- We recommend that you configure all trunk ports in the segment with the same set of allowed VLANs.
- Be careful when configuring REP through a Telnet connection. Because REP blocks all VLANs until another REP interface sends a message to unblock it. You might lose connectivity to the router if you enable REP in a Telnet session that accesses the router through the same interface.
- You cannot run REP and STP on the same segment or interface.
- If you connect an STP network to a REP segment, be sure that the connection is at the segment edge. An STP connection that is not at the edge could cause a bridging loop because STP does not run on REP segments. All STP BPDUs are dropped at REP interfaces.
- You must configure all trunk ports in the segment with the same set of allowed VLANs, or a misconfiguration occurs.
- If REP is enabled on two ports on a switch, both ports must be either regular segment ports or edge ports. REP ports follow these rules:
 - There is no limit to the number of REP ports on a switch; however, only two ports on a switch can belong to the same REP segment.
 - If only one port on a switch is configured in a segment, the port should be an edge port.
 - If two ports on a switch belong to the same segment, they must be both edge ports, both regular segment ports, or one regular port and one edge no-neighbor port. An edge port and regular segment port on a switch cannot belong to the same segment.
 - If two ports on a switch belong to the same segment and one is configured as an edge port and one as a regular segment port (a misconfiguration), the edge port is treated as a regular segment port.
- REP interfaces come up in a blocked state and remain in a blocked state until they are safe to be unblocked. You need to be aware of this status to avoid sudden connection losses.
- REP sends all LSL PDUs in untagged frames on the native VLAN. The BPA message sent to the Cisco multicast address is sent on the administration VLAN, which is VLAN 1 by default.
- You can configure how long a REP interface remains up without receiving a hello from a neighbor. You can use the **rep lsl-age-timer** value interface configuration command to set the time from 120 ms to

10000 ms. The LSL hello timer is then set to the age-timer value divided by 3. In normal operation, three LSL hellos are sent before the age timer on the peer switch expires and checks for hello messages.

- EtherChannel port channel interfaces do not support LSL age-timer values less than 1000 ms. If you try to configure a value less than 1000 ms on a port channel, you receive an error message and the command is rejected.
- REP ports cannot be configured as one of the following port types:
 - Switched Port Analyzer (SPAN) destination port
 - Tunnel port
 - Access port
- REP is supported on EtherChannels, but not on an individual port that belongs to an EtherChannel.
- There can be a maximum of 26 REP segments per switch.
- If private VLANs are configured in a REP topology for an isolated trunk port, the secondary VLAN type must be mapped to the private VLAN isolated type only. A secondary VLAN cannot be mapped to a private VLAN community type for an isolated trunk port.

Configuring Resilient Ethernet Protocol Administrative VLAN

To avoid the delay created by link-failure messages, and VLAN-blocking notifications during load balancing, REP floods packets to a regular multicast address at the hardware flood layer (HFL). These messages are flooded to the whole network, and not just the REP segment. You can control the flooding of these messages by configuring an administrative VLAN.

Follow these guidelines when configuring the REP administrative VLAN:

- If you do not configure an administrative VLAN, the default is VLAN 1.
- The administrative VLAN cannot be the RSPAN VLAN.

If the REP domain has multiple REP segments that are mutually exclusive, it is difficult to maintain a loop-less single admin VLAN across the entire REP domain. Starting with the Cisco IOS XE 17.2.1 release, you can configure multiple REP VLANs and manage multiple REP segments which are mutually exclusive to each other. You can configure additional admin VLANs by specifying a segment id in the rep admin vlan command.

Existing configurations that use a single, global REP admin VLAN will continue working as earlier. REP segments that have not been assigned a specific admin VLAN will use the global admin VLAN. HFL packets will be flooded into the admin VLAN configured for the segment.

To configure the REP administrative VLAN for an REP segment, follow these steps:

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# configure terminal | Enters global configuration mode. |
| Step 3 | rep admin vlan <i>vlan-id</i> segment <i>segment-id</i> Example: Device(config)# rep admin vlan 2 segment 4 | Specifies the administrative VLAN for the segment. The VLAN range is from 2 to 4094. The segment id number ranges from 1-1024. To set the admin VLAN to 1, which is the default, enter the no rep admin vlan global configuration command. |
| Step 4 | end Example: Device(config)# end | Exits global configuration mode and returns to privileged EXEC mode. |
| Step 5 | show interface [<i>interface-id</i>] rep detail Example: Device# show interface gigabitethernet1/1 rep detail | (Optional) Verifies the configuration on a REP interface. |
| Step 6 | copy running-config startup config Example: Device# copy running-config startup config | (Optional) Saves your entries in the switch startup configuration file. |

Configuring a REP Interface

To configure REP, enable REP on each segment interface and identify the segment ID. This task is mandatory, and must be done before other REP configurations. You must also configure a primary and secondary edge port on each segment. All the other steps are optional.

Follow these steps to enable and configure REP on an 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. |

| | Command or Action | Purpose |
|---------------|---|---|
| Step 3 | interface <i>interface-id</i> Example: Device(config)# interface gigabitethernet1/1 | Specifies the interface, and enters interface configuration mode. The interface can be a physical Layer 2 interface or a port channel (logical interface). |
| Step 4 | switchport mode trunk Example: Device(config-if)# switchport mode trunk | Configures the interface as a Layer 2 trunk port. |
| Step 5 | rep segment <i>segment-id</i> [edge [no-neighbor] [primary]] [preferred] Example: Device(config-if)# rep segment 1 edge no-neighbor primary | <p>Enables REP on the interface and identifies a segment number. The segment ID range is from 1 to 1024.</p> <p>Note You must configure two edge ports, including one primary edge port, for each segment.</p> <p>These optional keywords are available:</p> <ul style="list-style-type: none"> • (Optional) edge—Configures the port as an edge port. Each segment has only two edge ports. Entering the keyword edge without the keyword primary configures the port as the secondary edge port. • (Optional) primary—Configures the port as the primary edge port, the port on which you can configure VLAN load balancing. • (Optional) no-neighbor—Configures a port with no external REP neighbors as an edge port. The port inherits all the properties of an edge port, and you can configure the properties the same way you would for an edge port. <p>Note Although each segment can have only one primary edge port, if you configure edge ports on two different switches and enter the keyword primary on both the switches, the configuration is valid. However, REP selects only one of these ports as the segment primary edge port. You can identify the primary edge port for a segment by entering the show rep topology command in privileged EXEC mode.</p> |

| | Command or Action | Purpose |
|---------------|--|---|
| | | <ul style="list-style-type: none"> • (Optional) preferred—Indicates that the port is the preferred alternate port or the preferred port for VLAN load balancing. <p>Note Configuring a port as preferred does not guarantee that it becomes the alternate port; it merely gives the port a slight edge over equal contenders. The alternate port is usually a previously failed port.</p> |
| Step 6 | rep stcn { <i>interface interface id</i> segment id-list stp } Example: Device(config-if) # rep stcn segment 25-50 | (Optional) Configures the edge port to send segment topology change notices (STCNs). <ul style="list-style-type: none"> • interface interface-id—Designates a physical interface or port channel to receive STCNs. • segment id-list—Identifies one or more segments to receive STCNs. The range is from 1 to 1024. • stp—Sends STCNs to STP networks. <p>Note Spanning Tree (MST) mode is required on edge no-neighbor nodes when rep stcn stp command is configured for sending STCNs to STP networks.</p> |
| Step 7 | rep block port { <i>id port-id</i> <i>neighbor-offset</i> preferred } vlan { <i>vlan-list</i> all } Example: Device(config-if) # rep block port id 0009001818D68700 vlan 1-100 | (Optional) Configures VLAN load balancing on the primary edge port, identifies the REP alternate port in one of three ways (<i>id port-id</i> , <i>neighbor_offset</i> , preferred), and configures the VLANs to be blocked on the alternate port. <ul style="list-style-type: none"> • id port-id—Identifies the alternate port by port ID. The port ID is automatically generated for each port in the segment. You can view interface port IDs by entering the show interface type number rep [detail] privileged EXEC command. • <i>neighbor_offset</i>—Number to identify the alternate port as a downstream neighbor from an edge port. The range is from -256 to 256, with negative numbers indicating the downstream neighbor from the secondary edge port. A value of 0 is |

| | Command or Action | Purpose |
|---------------|--|---|
| | | <p>invalid. Enter -1 to identify the secondary edge port as the alternate port.</p> <p>Note Because you enter the rep block port command at the primary edge port (offset number 1), you cannot enter an offset value of 1 to identify an alternate port.</p> <ul style="list-style-type: none"> • preferred—Selects the regular segment port that is previously identified as the preferred alternate port for VLAN load balancing. • vlan vlan-list—Blocks one VLAN or a range of VLANs. • vlan all—Blocks all the VLANs. <p>Note Enter this command only on the REP primary edge port.</p> |
| Step 8 | rep preempt delay <i>seconds</i> Example: Device(config-if) # rep preempt delay 100 | <p>(Optional) Configures a preempt time delay.</p> <ul style="list-style-type: none"> • Use this command if you want VLAN load balancing to be automatically triggered after a link failure and recovery. • The time delay range is between 15 to 300 seconds. The default is manual preemption with no time delay. <p>Note Enter this command only on the REP primary edge port.</p> |
| Step 9 | rep lsl-age-timer <i>value</i> Example: Device(config-if) # rep lsl-age-timer 2000 | <p>(Optional) Configures a time (in milliseconds) for which the REP interface remains up without receiving a hello from a neighbor.</p> <p>The range is from 120 to 10000 ms in 40-ms increments. The default is 5000 ms (5 seconds).</p> <p>Note</p> <ul style="list-style-type: none"> • EtherChannel port channel interfaces do not support LSL age-timer values that are less than 1000 ms. • Both the ports on the link should have the same LSL age that is configured in order to avoid link flaps. |

| | Command or Action | Purpose |
|----------------|--|---|
| Step 10 | end Example: Device(config-if)# end | Exits global configuration mode and returns to privileged EXEC mode. |
| Step 11 | show interface [<i>interface-id</i>] rep [detail] Example: Device# show interface gigabitethernet1/1 rep detail | (Optional) Displays the REP interface configuration. |
| Step 12 | copy running-config startup-config Example: Device# copy running-config startup-config | (Optional) Saves your entries in the router startup configuration file. |

Setting Manual Preemption for VLAN Load Balancing

If you do not enter the **rep preempt delay seconds** interface configuration command on the primary edge port to configure a preemption time delay, the default is to manually trigger VLAN load balancing on the segment. Be sure that all the other segment configurations have been completed before manually preempting VLAN load balancing. When you enter the **rep preempt delay segment segment-id** command, a confirmation message is displayed before the command is executed because preemption might cause network disruption.

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 | rep preempt segment <i>segment-id</i> Example: Device(config)# rep preempt segment 100 The command will cause a momentary traffic disruption. Do you still want to continue? [confirm] | Manually triggers VLAN load balancing on the segment. You need to confirm the command before it is executed. |
| Step 4 | end Example: Device# end | Exits global configuration mode and returns to privileged EXEC mode. |

| | Command or Action | Purpose |
|---------------|---|---|
| Step 5 | show rep topology segment <i>segment-id</i> Example: Device# show rep topology segment 100 | (Optional) Displays REP topology information. |
| Step 6 | end Example: Device# end | Exits privileged EXEC mode. |

Configuring Simple Network Management Protocol Traps for Resilient Ethernet Protocol

You can configure a router to send REP-specific traps to notify the Simple Network Management Protocol (SNMP) server of link-operational status changes and port role changes.

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 | snmp mib rep trap-rate <i>value</i> Example: Device(config)# snmp mib rep trap-rate 500 | Enables the switch to send REP traps, and sets the number of traps that are sent per second. <ul style="list-style-type: none"> Enter the number of traps sent per second. The range is from 0 to 1000. The default is 0 (no limit is imposed; a trap is sent at every occurrence). |
| Step 4 | end Example: Device(config)# end | Returns to privileged EXEC mode. |
| Step 5 | show running-config Example: Device# show running-config | (Optional) Displays the running configuration, which can be used to verify the REP trap configuration. |
| Step 6 | copy running-config startup-config Example: | (Optional) Saves your entries in the switch startup configuration file. |

| | Command or Action | Purpose |
|--|---|---------|
| | Device# copy running-config startup-config | |

Monitoring Resilient Ethernet Protocol Configurations



Note When the port on the peer side is down the **show rep topology** command displays both the primary and the secondary ports as secondary ports.

This is an example of the output for the **show interface** *[interface-id]* **rep** *[detail]* command. For this display, the REP configuration and status on an uplink port is shown.

```
Device# show interfaces TenGigabitEthernet4/1 rep detail
```

```
TenGigabitEthernet4/1 REP enabled
Segment-id: 3 (Primary Edge)
PortID: 03010015FA66FF80
Preferred flag: No
Operational Link Status: TWO_WAY
Current Key: 02040015FA66FF804050
Port Role: Open
Blocked VLAN: <empty>
Admin-vlan: 1
Preempt Delay Timer: disabled
Configured Load-balancing Block Port: none
Configured Load-balancing Block VLAN: none
STCN Propagate to: none
LSL PDU rx: 999, tx: 652
HFL PDU rx: 0, tx: 0
BPA TLV rx: 500, tx: 4
BPA (STCN, LSL) TLV rx: 0, tx: 0
BPA (STCN, HFL) TLV rx: 0, tx: 0
EPA-ELECTION TLV rx: 6, tx: 5
EPA-COMMAND TLV rx: 0, tx: 0
EPA-INFO TLV rx: 135, tx: 136
```

This is an example of the output for the **show interface** *[interface-id]* **rep** *[detail]* command. For this display, the REP configuration and status on a downlink port is shown.

```
Device#show interface TenGigabitEthernet5/0/27 rep detail
```

```
TenGigabitEthernet5/0/27 REP enabled
Segment-id: 1 (Segment)
PortID: 019B380E4D9ACAC0
Preferred flag: No
Operational Link Status: NO_NEIGHBOR
Current Key: 019B380E4D9ACAC0696B
Port Role: Fail No Ext Neighbor
Blocked VLAN: 1-4094
Admin-vlan: 1
Preempt Delay Timer: 100 sec
LSL Ageout Timer: 2000 ms
LSL Ageout Retries: 5
```

```

Configured Load-balancing Block Port: 09E9380E4D9ACAC0
Configured Load-balancing Block VLAN: 1-100
STCN Propagate to: segment 25
LSL PDU rx: 292, tx: 340
HFL PDU rx: 0, tx: 0
BPA TLV rx: 0, tx: 0
BPA (STCN, LSL) TLV rx: 0, tx: 0
BPA (STCN, HFL) TLV rx: 0, tx: 0
EPA-ELECTION TLV rx: 0, tx: 0
EPA-COMMAND TLV rx: 0, tx: 0
EPA-INFO TLV rx: 0, tx: 0

```

This is an example for the **show rep topology** [**segment** *segment-id*] [**archive**] [**detail**] command. For this display, the REP topology information for all the segments is shown.

Device# **show rep topology**

```

REP Segment 1
BridgeName      PortName      Edge Role
-----
10.64.106.63    Te5/4         Pri  Open
10.64.106.228   Te3/4         Open
10.64.106.228   Te3/3         Open
10.64.106.67    Te4/3         Open
10.64.106.67    Te4/4         Alt
10.64.106.63    Te4/4         Sec  Open

REP Segment 3
BridgeName      PortName      Edge Role
-----
10.64.106.63    Gi50/1        Pri  Open
SVT_3400_2      Gi0/3         Open
SVT_3400_2      Gi0/4         Open
10.64.106.68    Gi40/2        Open
10.64.106.68    Gi40/1        Open
10.64.106.63    Gi50/2        Sec  Alt

```

Additional References for Resilient Ethernet Protocol

Related Documents

| Related Topic | Document Title |
|---------------|--|
| REP commands | See the Layer 2/3 Commands section of the <i>Command Reference (Catalyst 9500 Series Switches)</i> |

Feature History for Resilient Ethernet 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 Everest 16.5.1a | Resilient Ethernet Protocol | REP is a Cisco proprietary protocol that provides an alternative to STP to control network loops, handle link failures, and improve convergence time. On Cisco Catalyst 9500 Series Switches, support was introduced only on the C9500-40X model. |
| Cisco IOS XE Fuji 16.8.1a | Resilient Ethernet Protocol | Support for the feature was introduced on all switch models with 1-Gigabit Ethernet and 10-Gigabit Ethernet interfaces. |
| Cisco IOS XE Fuji 16.9.1 | Resilient Ethernet Protocol Support on Downlink Ports | Support for configuring REP on downlink ports was introduced. |
| Cisco IOS XE Amsterdam 17.2.1 | Multiple Administrative VLANs for Resilient Ethernet Protocol | Support for configuring multiple administrative VLANs in REP 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 9

Configuring UniDirectional Link Detection

- [Restrictions for Configuring UniDirectional Link Detection, on page 185](#)
- [Information About UniDirectional Link Detection, on page 185](#)
- [How to Configure UDLD, on page 188](#)
- [Monitoring and Maintaining UniDirectional Link Detection, on page 192](#)
- [Console Error Messages For Fast UniDirectional Link Detection, on page 193](#)
- [Additional References for UniDirectional Link Detection, on page 193](#)
- [Feature History for UniDirectional Link Detection, on page 193](#)

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 UDLD

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.

Fast UDLD functionality has been made available on Catalyst Catalyst 9500 Series Switches High Performance devices starting with the Cisco IOS XE Fuji 16.9.1

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 UDLD

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. Note Use the no udld port interface configuration command to disable UDLD on a specified fiber-optic port. |
| 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: | Enters global configuration mode. |

| | Command or Action | Purpose |
|---------------|---|---|
| | Device# <code>configure terminal</code> | |
| Step 2 | interface <i>interface-id</i> Example: Device(config)# <code>interface gigabitethernet 1/0/1</code> | Specifies the port to be enabled for Fast UDLD, and enters interface configuration mode. |
| Step 3 | udld fast-hello <i>message time interval</i> Example: Device(config-if)# <code>udld fast-hello 200</code> | 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. <p>Note Fast UDLD can be enabled only if UDLD is already enabled on the specified port.</p> |
| Step 4 | end Example: Device(config-if)# <code>end</code> | 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# <code>configure terminal</code> | Enters global configuration mode. |
| Step 2 | udld fast-hello error-reporting Example: Device(config)# <code>udld fast-hello error-reporting</code> | Enables the display of console messages to report the error upon detection of a link failure. <p>Note The detected unidirectional link will not be disabled if udld fast-hello error-reporting has been enabled.</p> |
| Step 3 | end Example: Device(config)# <code>end</code> | 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 9500 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 Everest 16.5.1a | 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. Support for this feature was introduced only on the C9500-12Q, C9500-16X, C9500-24Q, C9500-40X models of the Cisco Catalyst 9500 Series Switches. |
| Cisco IOS XE Fuji 16.8.1a | UniDirectional Link Detection (UDLD) | Support for this feature was introduced only on the C9500-32C, C9500-32QC, C9500-48Y4C, and C9500-24Y4C models of the Cisco Catalyst 9500 Series Switches. |

| Release | Feature | Feature Information |
|--------------------------|-----------|---|
| Cisco IOS XE Fuji 16.9.1 | Fast UDLD | <p>Enables subsecond UDLD. The UDLD protocol helps monitor a physical connection (such as monitoring wrong cabling) to detect unidirectional links to avoid spanning-tree topology loops or silent drop traffic.</p> <p>Support for this feature was introduced only on the C9500-12Q, C9500-16X, C9500-24Q, C9500-40X models of the Cisco Catalyst 9500 Series Switches.</p> |

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 10

Configuring Layer 2 Protocol Tunneling

- [Prerequisites for Layer 2 Protocol Tunneling, on page 195](#)
- [Restrictions for Layer 2 Protocol Tunneling, on page 195](#)
- [Information About Layer 2 Protocol Tunneling, on page 195](#)
- [How to Configure Layer 2 Protocol Tunneling, on page 199](#)
- [How to Configure Layer 2 Protocol Tunneling for EtherChannels, on page 202](#)
- [Configuration Examples for Layer 2 Protocol Tunneling, on page 207](#)
- [Monitoring Tunneling Status, on page 209](#)
- [Feature History for Layer 2 Protocol Tunneling, on page 209](#)

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.



Note

This is applicable on the C9500-32C, C9500-32QC, C9500-48Y4C, and C9500-24Y4C models of the Cisco Catalyst 9500 Series Switches.

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 38: Layer 2 Protocol Tunneling

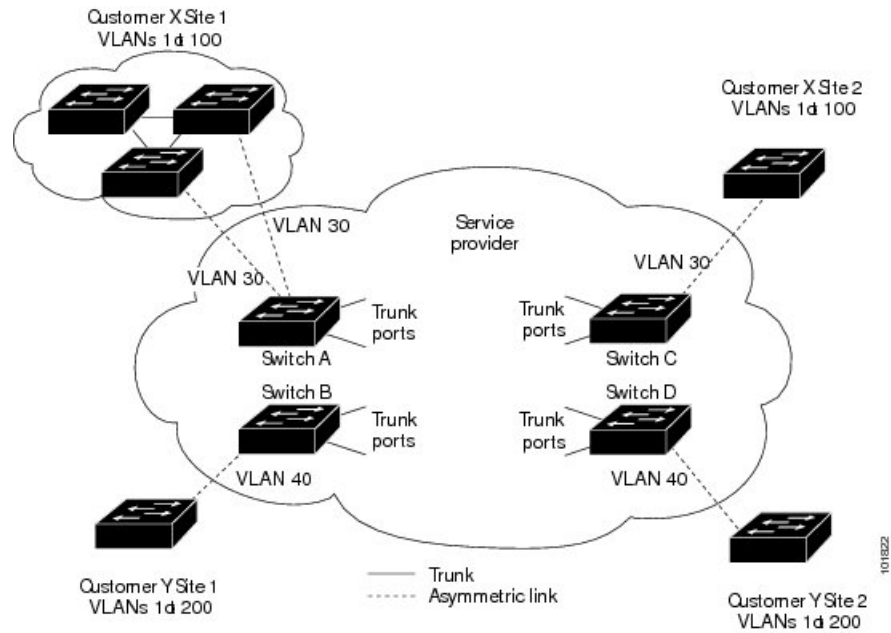
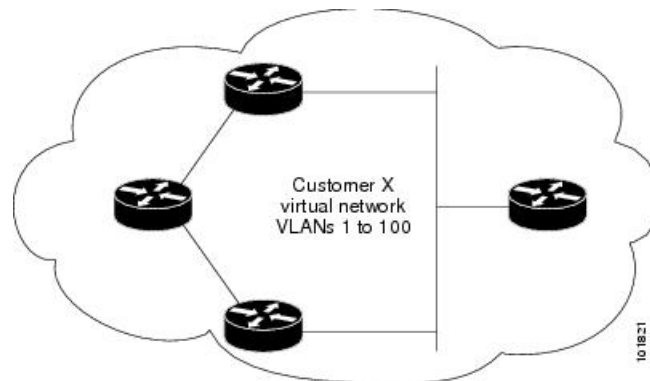


Figure 39: Layer 2 Network Topology Without Proper Convergence



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.

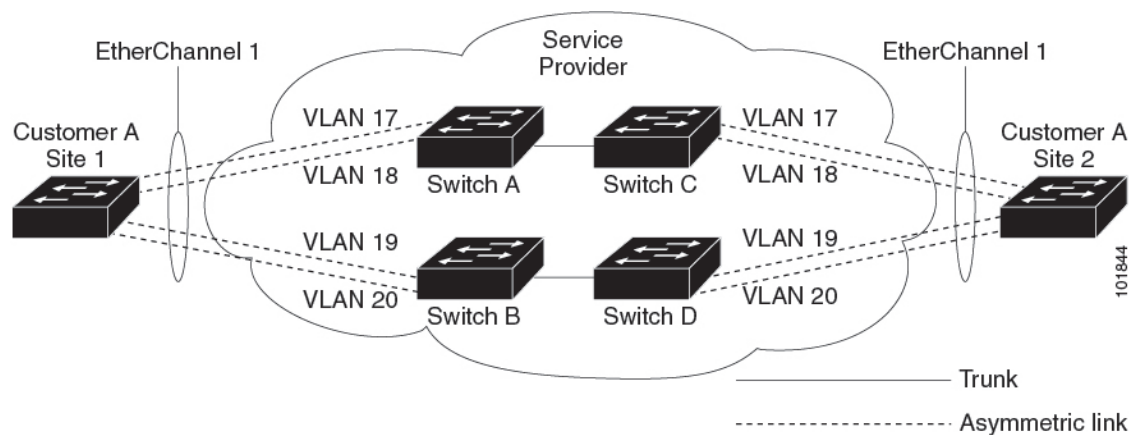
Layer 2 Protocol Tunneling for EtherChannels

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 40: 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: <pre>Device(config)# interface gigabitethernet1/0/1</pre> | 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 • switchport mode trunk Example: <pre>Device(config-if)# switchport mode dot1q-tunnel</pre> or <pre>Device(config-if)# switchport mode trunk</pre> | 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: <pre>Device(config-if)# l2protocol-tunnel cdp</pre> | 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: <pre>Device(config-if)# l2protocol-tunnel shutdown-threshold 100 cdp</pre> | (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. |

| | Command or Action | Purpose |
|----------------|---|---|
| | | <p>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.</p> |
| 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 bpdupfilter enable</p> <p>Example:</p> | Inserts a BPDU filter for spanning tree. |

| | Command or Action | Purpose |
|----------------|---|--|
| | Device(config)# spanning-tree bpduguard enable | Note While configuring Layer 2 Protocol Tunneling on a trunk port, you must enable a BPDU filter for spanning tree. |
| Step 11 | end Example: Device(config)# end | Returns to privileged EXEC mode. |
| Step 12 | 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 13 | copy running-config startup-config Example: Device# copy running-config startup-config | (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> 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 native vlan vlan-id | Configures the native VLAN. |

| | Command or Action | Purpose |
|---------------|---|---|
| | Example: <pre>Device(config-if)# switchport trunk native vlan 2</pre> | 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> Example: <pre>Device(config-if)# switchport trunk allowed vlan 1,2,4-3003,3005-4094</pre> | Note Specifies the list of allowed VLANs. 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. |
| Step 6 | Use one of the following: <ul style="list-style-type: none"> • switchport mode dot1q-tunnel • switchport mode trunk Example: <pre>Device(config-if)# switchport mode dot1q-tunnel</pre> or <pre>Device(config-if)# switchport mode trunk</pre> | Configures the interface as an IEEE 802.1Q tunnel port or as a trunk port. |
| Step 7 | l2protocol-tunnel point-to-point [pagp lacp udld] Example: <pre>Device(config-if)# l2protocol-tunnel point-to-point pagp</pre> | Note (Optional) Enables point-to-point protocol tunneling for the desired protocol. If no keyword is entered, tunneling is enabled for all three protocols. 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. 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. |
| Step 8 | l2protocol-tunnel shutdown-threshold [point-to-point [pagp lacp udld]] <i>value</i> | (Optional) Configures the threshold for packets-per-second that are accepted for |

| | Command or Action | Purpose |
|----------------|--|--|
| | Example: <pre>Device(config-if)# l2protocol-tunnel shutdown-threshold point-to-point pagp 100</pre> | <p>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> <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 | l2protocol-tunnel drop-threshold [point-to-point [pagp lacp udld]] value Example: <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 | no cdp enable Example: <pre>Device(config-if)# no cdp enable</pre> | Disables CDP on the interface. |
| Step 11 | spanning-tree bpdu filter enable Example: <pre>Device(config-if)# spanning-tree bpdu filter enable</pre> | Enables BPDU filtering on the interface. |
| Step 12 | exit Example: | Returns to global configuration mode. |

| | Command or Action | Purpose |
|----------------|---|--|
| | <code>Device(config-if)# exit</code> | |
| Step 13 | errdisable recovery cause l2ptguard Example: <code>Device(config)# errdisable recovery cause l2ptguard</code> | (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. |
| Step 14 | end Example: <code>Device(config)# end</code> | Returns to privileged EXEC mode. |
| Step 15 | show l2protocol Example: <code>Device# show l2protocol</code> | 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: <code>Device# copy running-config startup-config</code> | (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: <code>Device> enable</code> | Enables privileged EXEC mode. Enter your password if prompted. |
| Step 2 | configure terminal Example: <code>Device# configure terminal</code> | Enters global configuration mode. |
| Step 3 | interface interface-id Example: <code>Device(config)# interface gigabitethernet1/0/1</code> | Specifies the interface that is connected to the phone, and enters interface configuration mode. |

| | Command or Action | Purpose |
|----------------|---|--|
| Step 4 | switchport trunk encapsulation dot1q Example: Device(config-if)# switchport trunk encapsulation dot1q | Sets the trunking encapsulation format to IEEE 802.1Q. |
| 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 <i>channel-group-number</i> 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 <i>port-channel number</i> 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. |

| | Command or Action | Purpose |
|----------------|---|---|
| Step 14 | copy running-config startup-config Example: <pre>Device# copy running-config startup-config</pre> | (Optional) Saves your entries in the configuration file. Note Use the no switchport mode trunk , the no udld 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. Support for this feature was introduced on all the models of the Cisco Catalyst 9500 Series Switches. |

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

Configuring IEEE 802.1Q Tunneling

- [Information About IEEE 802.1Q Tunneling, on page 211](#)
- [How to Configure IEEE 802.1Q Tunneling, on page 216](#)
- [Monitoring Tunneling Status, on page 218](#)
- [Example: Configuring an IEEE 802.1Q Tunneling Port, on page 218](#)
- [Feature History for IEEE 802.1Q Tunneling, on page 219](#)

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.

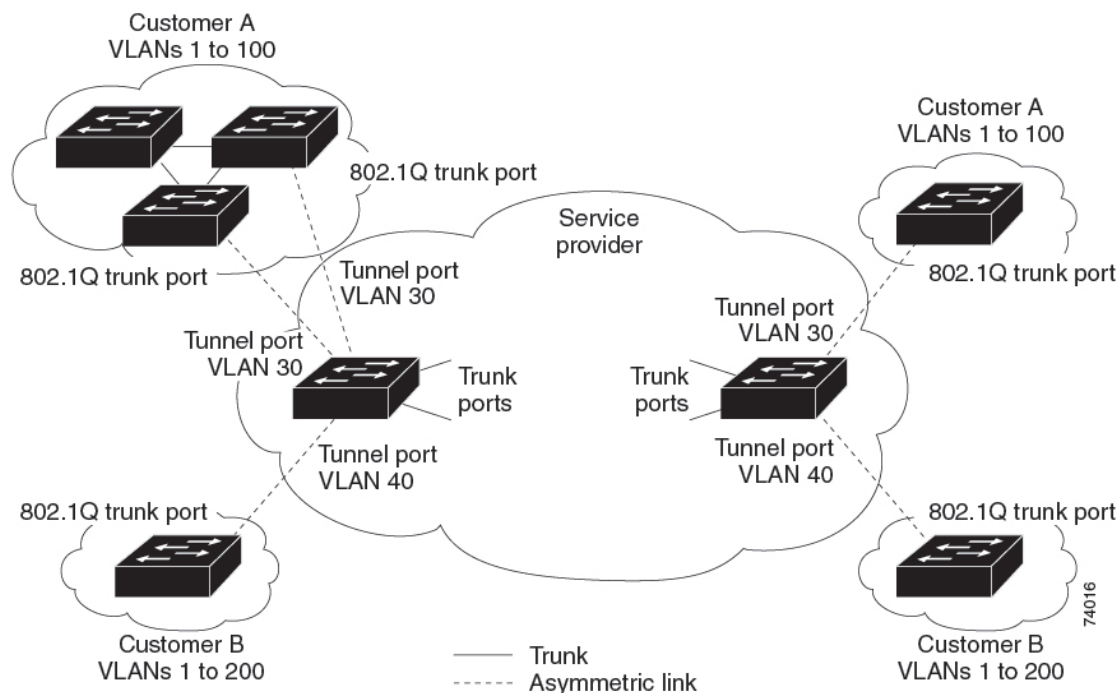
IEEE 802.1Q Tunnel Ports in a Service Provider Network

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 41: IEEE 802.1Q Tunnel Ports in a Service-Provider Network

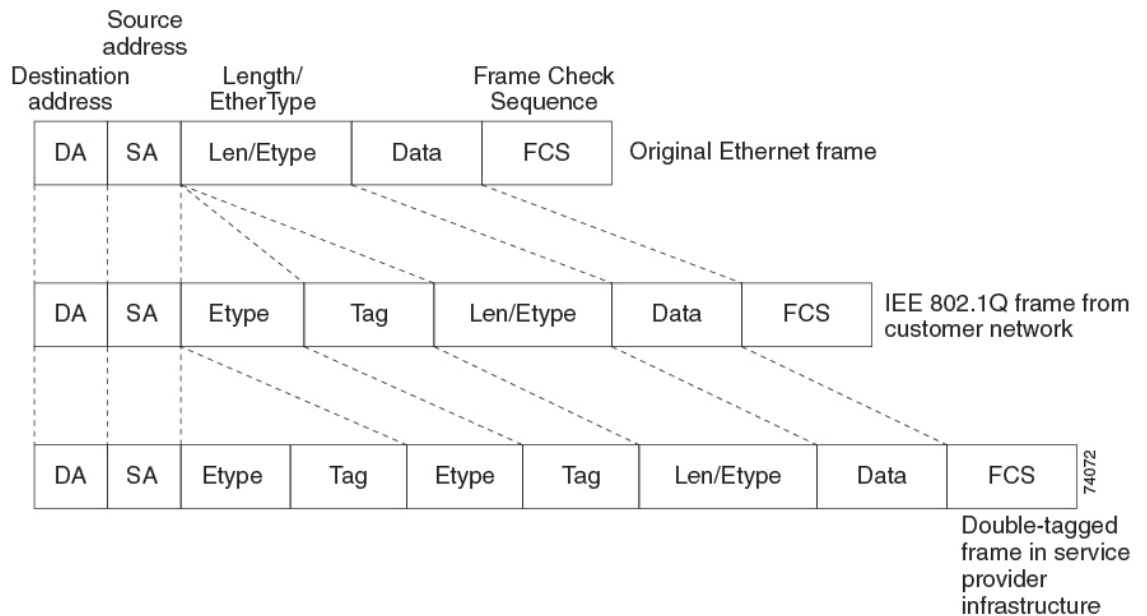


Packets coming from the customer trunk port into the tunnel port on the service-provider edge device are normally IEEE 802.1Q-tagged with the appropriate VLAN ID. The tagged packets remain intact inside the device and when they exit the trunk port into the service-provider network, they are encapsulated 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 42: 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.)

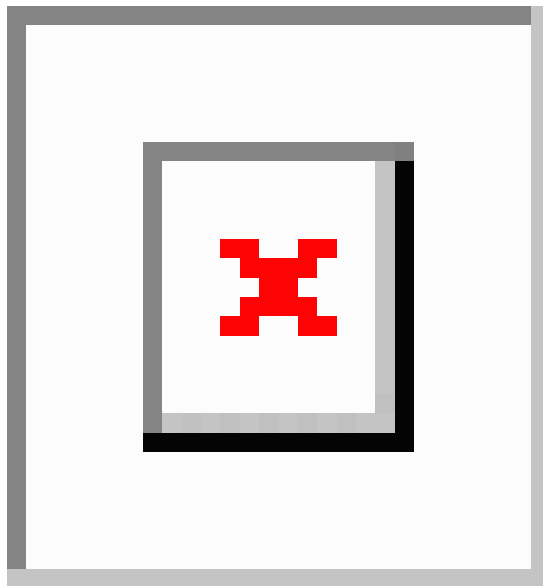
On switches, because 802.1Q tunneling is configured on a per-port basis, it does not matter whether the switch is a standalone device or a member switch. All configuration is done on the active switch.

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 43: 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.

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. |

| | Command or Action | Purpose |
|---------------|--|--|
| 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. |
| 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 | Displays the ports configured for IEEE 802.1Q tunneling. Displays the ports that are in tunnel mode. |

| | Command or Action | Purpose |
|----------------|---|--|
| | or Device# <code>show running-config interface</code> | |
| Step 10 | <code>show vlan dot1q tag native</code> Example: Device# <code>show vlan dot1q native</code> | Displays IEEE 802.1Q native VLAN tagging status. |
| Step 11 | <code>copy running-config startup-config</code> Example: Device# <code>copy running-config startup-config</code> | (Optional) Saves your entries in the configuration file. |

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

```

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 Everest 16.5.1a | IEEE 802.1Q Tunneling | <p>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.</p> <p>Support for this feature was introduced only on the C9500-12Q, C9500-16X, C9500-24Q, C9500-40X models of the Cisco Catalyst 9500 Series Switches.</p> |
| Cisco IOS XE Fuji 16.8.1a | IEEE 802.1Q Tunneling | <p>Support for this feature was introduced only on the C9500-32C, C9500-32QC, C9500-48Y4C, and C9500-24Y4C models of the Cisco Catalyst 9500 Series Switches.</p> |

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 12

Configuring VLAN Mapping

- [Prerequisites for VLAN Mapping, on page 221](#)
- [Prerequisites for One to One VLAN Mapping, on page 222](#)
- [Restrictions for VLAN Mapping, on page 222](#)
- [Restrictions for One to One VLAN Mapping, on page 222](#)
- [About VLAN Mapping, on page 223](#)
- [Configuration Guidelines for VLAN Mapping, on page 225](#)
- [How to Configure VLAN Mapping, on page 227](#)
- [Feature History for VLAN Mapping, on page 233](#)

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:

Applicable for C9500-12Q, C9500-16X, C9500-24Q, C9500-40X models of the Cisco Catalyst 9500 Series Switches.

```
!  
Device(config)# interface HundredGigE1/0/1  
Device(config-if)# switchport mode access  
Device(config-if)# l2protocol-tunnel stp  
Device(config-if)# end
```

Applicable for C9500-32C, C9500-32QC, C9500-48Y4C, and C9500-24Y4C models of the Cisco Catalyst 9500 Series Switches.

```
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
```

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 bpdudfilter 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. This is applicable to C9500-12Q, C9500-16X, C9500-24Q, C9500-40X models of the Cisco Catalyst 9500 Series Switches and C9500-32C, C9500-32QC, C9500-48Y4C, and C9500-24Y4C models of the Cisco Catalyst 9500 Series Switches.
- If a port with VLAN mapping is configured as a part of EtherChannel with a conflicting mapping translation, the port restricted from becoming a member of the port-channel. This is applicable to C9500-12Q, C9500-16X, C9500-24Q, C9500-40X models of the Cisco Catalyst 9500 Series Switches and C9500-32C, C9500-32QC, C9500-48Y4C, and C9500-24Y4C models of the Cisco Catalyst 9500 Series Switches.
- 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. This is applicable to C9500-12Q, C9500-16X, C9500-24Q, C9500-40X models of the Cisco Catalyst 9500 Series Switches and C9500-32C, C9500-32QC, C9500-48Y4C, and C9500-24Y4C models of the Cisco Catalyst 9500 Series Switches.
- 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



Note Q-in-Q on a Trunk Port is not supported on the C9500-12Q, C9500-16X, C9500-24Q, C9500-40X models of the Cisco Catalyst 9500 Series Switches.

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.

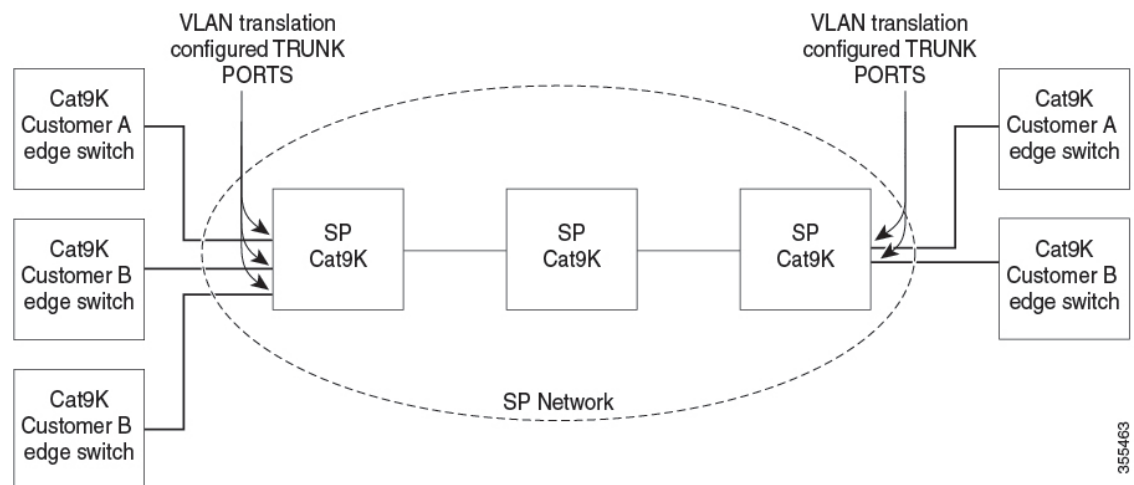
One way to establish translated VLAN IDs (S-VLANs) is to map customer VLANs to VLANs (called VLAN ID translation) on trunk ports that are connected to a customer network. Packets entering the port are mapped to service provider VLAN (S-VLAN) based on the port number and the packet's original customer 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 44: 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 45: Mapping Customer VLANs to Service-Provider VLANs

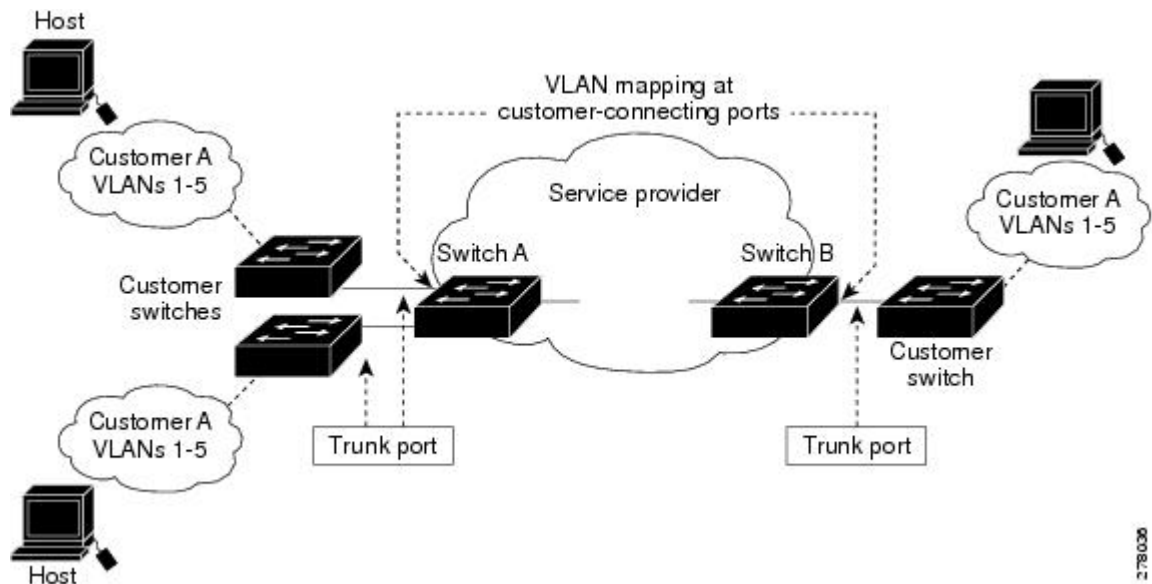


Figure shows a topology where a customer uses the same VLANs in multiple sites on different sides of a service-provider network. The C-VLAN IDs are mapped to service-provider VLAN IDs for packet travel across the service-provider backbone. The C-VLAN IDs are retrieved at the other side of the service-provider 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. This is applicable for C9500-12Q, C9500-16X, C9500-24Q, C9500-40X models of the Cisco Catalyst 9500 Series Switches and C9500-32C, C9500-32QC, C9500-48Y4C, and C9500-24Y4C models of the Cisco Catalyst 9500 Series Switches.

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.
- Maximum number of VLAN mapping configurations that are supported on Cisco Catalyst 9500 High Performance series switches is 1000 system wide.
- Maximum number of VLAN mapping configurations that are supported on Cisco Catalyst 9500 series switches is 512 system wide.

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 belonging to an EtherChannel is configured with a VLAN mapping and the EtherChannel is configured with a conflicting VLAN mapping, then the port is removed from the EtherChannel (Applicable for C9500-12Q, C9500-16X, C9500-24Q, C9500-40X models of the Cisco Catalyst 9500 Series Switches)
- If a port with VLAN mapping is configured as a part of EtherChannel with a conflicting mapping translation, the port is restricted from becoming a member of the port-channel (Applicable for C9500-32C, C9500-32QC, C9500-48Y4C, and C9500-24Y4C models of the Cisco Catalyst 9500 Series Switches).
- 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
```

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 bpdudfilter 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.
- Applicable for C9500-32C, C9500-32QC, C9500-48Y4C, and C9500-24Y4C models of the Cisco Catalyst 9500 Series Switches.

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.
- Applicable for C9500-32C, C9500-32QC, C9500-48Y4C, and C9500-24Y4C models of the Cisco Catalyst 9500 Series Switches.

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 interface-id 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. |

| | Command or Action | Purpose |
|----------------|--|---|
| 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"> • <i>vlan-id</i> —the customer VLAN ID (C-VLAN) entering the switch from the customer network. The range is from 1 to 4094. • <i>translated-id</i> —the assigned service-provider VLAN ID (S-VLAN). The range is from 1 to 4094. By default, the packets with VLAN IDs other than the ones with configured VLAN mapping are forwarded as normal traffic. |
| 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 | Enables privileged EXEC mode. |
| | Example: | 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 interface-id 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 vlan-id dot1q-tunnel outer vlan-id Example: Device(config-if)# switchport vlan mapping 16 dot1q-tunnel 64 | <p>Enters the VLAN IDs to be mapped:</p> <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. You can enter a string of VLAN-IDs. • outer-vlan-id —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 vlan-id Example: Device(config-if)# switchport vlan mapping default dot1q-tunnel 22 | <p>Specifies that all unmapped packets on the port are forwarded with the specified S-VLAN.</p> <p>By default, packets that do not match the mapped VLANs, are dropped.</p> <p>Untagged traffic are forwarded without dropping.</p> |
| 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 bpduguard 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 interface-id 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 vlan-id 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 bpdupfilter enable Example: Device(config)# spanning-tree bpdupfilter 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 Fuji 16.8.1a | One-to-One VLAN mapping | <p>One-to-One VLAN mapping allows to map customer VLANs to service-provider VLANs on trunk ports connected to a customer network.</p> <p>Support for this feature was introduced only on the C9500-32C, C9500-32QC, C9500-48Y4C, and C9500-24Y4C models of the Cisco Catalyst 9500 Series Switches.</p> |
| Cisco IOS XE Gibraltar 16.11.1 | One-to-One VLAN mapping | <p>One-to-One VLAN mapping allows to map customer VLANs to service-provider VLANs on trunk ports connected to a customer network.</p> <p>Support for this feature was introduced only on the C9500-12Q, C9500-16X, C9500-24Q, C9500-40X models of the Cisco Catalyst 9500 Series Switches.</p> |

| Release | Feature | Feature Information |
|--------------------------------|------------------------|---|
| Cisco IOS XE Gibraltar 16.11.1 | Selective Q-in-Q | Support for selective Q-in-Q was introduced Support for this feature was introduced only on the C9500-32C, C9500-32QC, C9500-48Y4C, and C9500-24Y4C models of the Cisco Catalyst 9500 Series Switches. |
| | Q-in-Q on a Trunk Port | Support for Q-in-Q on a trunk port was introduced Support for this feature was introduced only on the C9500-32C, C9500-32QC, C9500-48Y4C, and C9500-24Y4C models of the Cisco Catalyst 9500 Series Switches. |
| Cisco IOS XE Bengaluru 17.5.1 | Selective Q-in-Q | Support for selective Q-in-Q was introduced Support for this feature was introduced on the C9500-12Q, C9500-16X, C9500-24Q, C9500-40X models of the Cisco Catalyst 9500 Series Switches. |

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 13

Configuring Audio Video Bridging

The following sections provide information about various aspects relating to audio and video bridging networks.

- [Restrictions for Audio Video Bridging Networks, on page 235](#)
- [Introduction to Audio Video Bridging Networks, on page 235](#)
- [Configuring the AVB Network, on page 240](#)
- [Monitoring the AVB Network, on page 249](#)
- [Examples of AVB Configurations and Monitoring, on page 251](#)
- [Feature History for Audio Video Bridging, on page 272](#)

Restrictions for Audio Video Bridging Networks

- AVB is not supported on stacked systems.
- AVB is not supported on EtherChannel interfaces.
- AVB is supported only on STP-enabled network.
- The maximum number of MSRP streams supported is 200 in one device.
- The C9300-48HX, C9300-48TX, C9300-24HX, C9300X-48HXN, C9300-24UX, C9300-48UXM, C9300-24UXB, and C9300-48UN switch models do not support gPTP at 100Mbps speed.

Introduction to Audio Video Bridging Networks

Information about Audio Video Bridging

Audio and video equipment deployments have traditionally been analog single-purpose point-to-point one-way links. Migration to digital transmission also continued to retain the point-to-point one-way links architecture. The dedicated connection model resulted in a mass of cabling in professional and consumer applications, which was hard to manage and operate.

In order to accelerate the adoption to Ethernet based audio/video deployments in an interoperable way IEEE came up with the IEEE Audio Video Bridging standards - IEEE 802.1BA. This defines a mechanism where endpoints and the network will function as a whole to enable high quality A/V streaming across consumer applications to professional audio-video over an Ethernet infrastructure.

**Note**

- AVB is not supported on stacked systems.
- AVB is not supported on EtherChannel interfaces.
- AVB is supported only on STP-enabled network.

Audio Video Bridging License Level

Audio video bridging is supported on the Network Advantage license.

Benefits of Audio Video Bridging

AVB is a mechanism to enable Ethernet based audio-video transmission which has the following benefits:

- Guaranteed max latency
- Time synchronized
- Bandwidth guaranteed
- Professional grade

Components of Audio Video Bridging Network

AVB protocols operate only in domains where every device is AVB capable. The AVB network comprises of AVB talkers, AVB listeners, AVB switches and the grandmaster clock source.

- AVB Talker - An AVB end station that is the source or producer of a stream, i.e. microphones, video camera, and so on.
- AVB Listener - An AVB end station that is the destination or consumer of a stream, i.e. speaker, video screen, and so on.
- AVB Switch - An Ethernet switch that complies with IEEE802.1 AVB standards.
- AVB stream: A data stream associated with a stream reservation compliant with the Stream Reservation Protocol (SRP).

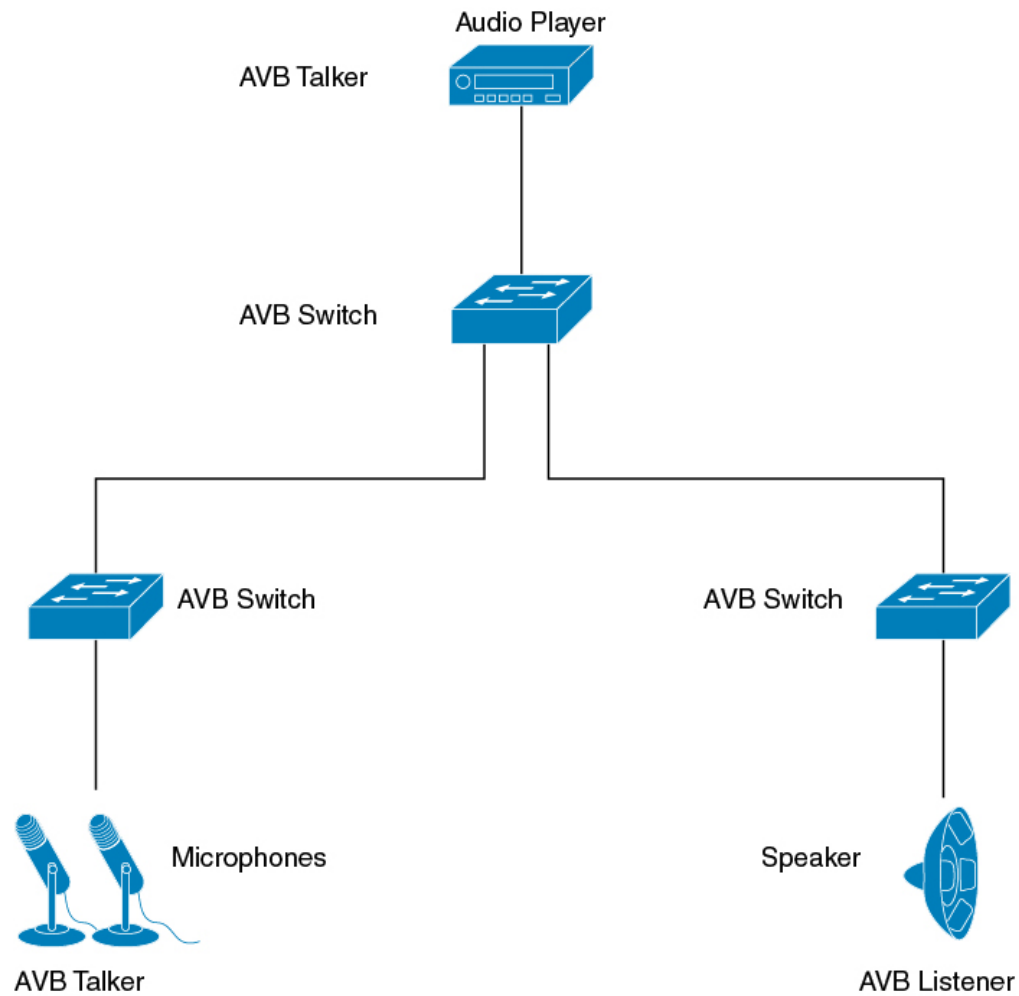
**Note**

In some instances, the word “bridge” is used. In this context, it references to a switch.

The IEEE 802.1BA specification requires that an AVB talker must be grandmaster capable. In a typical deployment a network node can also be the grandmaster, provided it can either source or derive timing from a grandmaster capable device and provide the timing to the AVB network using IEEE 802.1AS.

[Figure 46: AVB Network](#) shows a simple illustration of AVB network with different components.

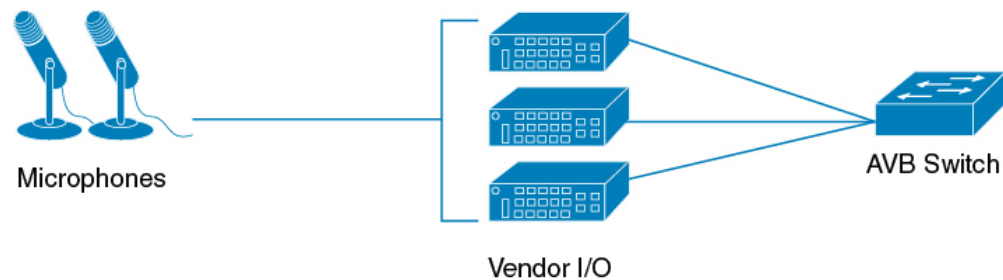
Figure 46: AVB Network



354698

In many instances, the Audio/Video end points (Microphone, Speaker, and so on) are analog devices. AVB end-point vendors introduce Digital Signal Processors (DSP) and I/O devices that provide extensive audio/video processing and aggregate the end-points into an AVB Ethernet interface, as shown in [Figure 47: Vendor Audio I/O System, on page 237](#).

Figure 47: Vendor Audio I/O System



354699

Supported SKUs for Audio Video Bridging

AVB is supported on the following Catalyst 9500 SKUs.

- C9500-24Q
- C9500-12Q
- C9500-40X
- C9500-16X

Information About Generalized Precision Time Protocol

Generalized Precision Time Protocol (gPTP) is an IEEE 802.1AS standard, which provides a mechanism to synchronize clocks of the bridges and end point devices in an AVB network. It defines the mechanism to elect the grandmaster clock (BMCA) among the time-aware bridges and talker and listener. The grandmaster is the root of the timing hierarchy that gets established in the time-aware network and distributes time to nodes below to enable synchronization.

Time synchronization also requires determining the link delay and switch delays in the network nodes. A gPTP switch is an IEEE 1588 boundary clock, which also determines the link delay using the peer-to-peer delay mechanism. The delays computed are included in the correction field of the PTP messages and relayed to the end-points. The talker and listener use this gPTP time as a shared clock reference, which is used to relay and recover the media clock. gPTP currently defines only domain 0, which is what the switch supports.

The peer to peer delay mechanism runs on STP blocked ports as well. No other PTP messages are sent over blocked ports.

In a PTP domain, Best Master Clock (BMC) algorithm organizes Clocks and Ports into a hierarchical fashion, which includes clocks and port states:

Clocks

- Grandmaster (GM/GMC)
- Boundary Clock (BC)

Port States

- Master (M)
- Slave (S)
- Passive (P)

Information about Multiple Stream Reservation Protocol (MSRP)

Multiple Stream Reservation Protocol (MSRP) provides a mechanism for end stations to reserve network resources that will guarantee the transmission and reception of data streams across a network with the requested QoS. It is one of the core protocols required on an AVB device (talker, listener and switches). It allows talkers to advertise streams across a network of AVB switches and listeners to register for receiving the streams.

MSRP is the key software protocol module for supporting AVB. It enables stream establishment and teardown. It interfaces with gPTP to update the latency information for the streams. It interfaces with the QoS module

to setup the hardware resources that would guarantee requested bandwidth for the streams. It also provides the QoS shaping parameters required for the credit based shaper.

Functions of Multiple Stream Reservation Protocol

MSRP performs the following functions:

- Allows Talkers to advertise Streams and Listeners to discover and register for Streams.
- Establishes a path through an Ethernet between a Talker and one or more Listeners.
- Provides guaranteed bandwidth for AVB Streams.
- Guarantees an upper bound on latency.
- Discovers and reports the worst case end-to-end latency between the Talker and each of its Listeners.
- Reports failure reason and location when a path between the Talker and a Listener cannot satisfy bandwidth requirements.
- Supports multiple classes of traffic with different latency targets.
- Protects best effort traffic from starvation by limiting AVB traffic.
- MSRP Talker declarations are not forwarded along the STP blocked ports.
- MSRP listens to the STP TCN notification to generate MSRP declarations tear /modify / establish streams.

Information about Hierarchical QoS

AVB networks guarantee bandwidth and minimum bounded latency for the time-sensitive audio and video streams. AVB defines Class A and Class B as the time-sensitive streams, based on the worst-case latency targets of the traffic from talker to listener.

The latency targets for the two streams are listed as below:

- SR-Class A: 2ms
- SR-Class B: 50ms

The sum of the worst-case latency contributions per hop should result in an overall end-to-end latency of 2 ms or less for SR-Class A and 50ms or less for SR-Class B. A typical AVB deployment of 7 hops from talker to listener meets these latency requirements.

The priority code points map the traffic to the specific stream. Frame forwarding behavior is based on this priority. A credit-based shaper is used to shape the transmission of these streams in accordance with the bandwidth that has been reserved on a given outbound queue so that the latency targets are met.

AVB supports hierarchical QoS. AVB Hierarchical QoS policy is two level Parent-Child Policy. AVB Parent policy segregates audio, video traffic streams(SR-Class A , SR-Class B) and Network Control packets from standard best-effort Ethernet traffic (Non-SR) and manage streams accordingly. Hierarchical QoS allows you to specify QoS behavior at multiple policy levels, which provides a high degree of granularity in traffic management. You can use hierarchical policies to:

- Allow a parent class to shape multiple queues in a child policy.
- Apply specific policy map actions on the aggregate traffic.

- Apply class-specific policy map actions.

You can modify only ingress and egress HQoS child policy's class-map and its actions using **policy-map** *AVB-Output-Child-Policy* and **policy-map** *AVB-Input-Child-Policy* command.



Note You should not modify the PCP in child policy to map with PCP configured in Parent Policy, for example SR Class A Cos 3 and SR Class B Cos 2.

Hierarchical Policing

Hierarchical policing is supported on ingress and egress interfaces. Hierarchical QoS separates the SR and Non-SR class related rules into parent and child policies respectively. AVB SR classes are completely controlled by MSRP client and hence, parent policies containing SR class attributes are governed by MSRP. The end user has complete control over child policies which contain Non-SR class attributes and can modify only the child policies.

AVB HQoS child policies are user modifiable and NVGENed to preserve the configuration if user saves the configuration to the startup-config. So, AVB HQoS child policy configurations are retained even after reload.

Information about Multiple VLAN Registration Protocol (MVRP)

Multiple VLAN Registration Protocol (MVRP) is an application based on MRP. MVRP provides a mechanism for dynamic maintenance of the contents of Dynamic VLAN Registration Entries for each vlan ids, and for propagating the information they contain to other Bridges. This information allows MVRP-aware devices to dynamically establish and update their knowledge of the set of vlan ids associated with VLANs that currently have active members, and through which Ports those members can be reached.

MVRP, from an AVB perspective, is mandatory on the talkers and the listeners. Independent of AVB, MVRP is an IEEE 802.1Q requirement on the VLAN-aware switches. However, manual configuration of VLANS on the switches is sufficient for AVB.



Note VTP should be in the disabled mode or transparent mode for MVRP to work.

Configuring the AVB Network

Configuring AVB

This section describes the various configurations available for AVB.

Enabling Audio Video Bridging

You can enable AVB using the below command on the switch.

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 | avb Example: <pre>Device(config)# avb</pre> | Enables AVB on the switch. |
| Step 4 | avb strict Example: <pre>Device(config)# avb strict</pre> | Enables AVB on the switch. This command is used in combination with the avb command to enable AVB. Note This command will be deprecated in the future releases. |
| Step 5 | end Example: <pre>Device(config)# end</pre> | Returns to privileged EXEC mode. |

What to do next

To disable AVB on the switch, use the **no** form of the command.

Configuring Audio Video Bridging

You can configure the interfaces along the connectivity path for AVB devices as dot1q trunk ports by using the below commands.

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 tel1/1/1 | Defines the interface to be configured as a trunk, and enters interface configuration mode. |
| Step 4 | switchport mode trunk Example: Device(config-if)# switchport mode trunk | Configures the port as a trunk port. |
| Step 5 | exit Example: Device(config-if)# exit | Returns to global configuration mode. |
| Step 6 | vlan 2 Example: Device(config)# vlan 2 | Configures VLAN 2 on the switch. Note VLAN 2 is the default AVB VLAN. If you need to configure another VLAN as the default AVB VLAN, use the command in Step 7. |
| Step 7 | avb vlan <i>vlan-id</i> Example: Device(config)# avb vlan 10 | (Optional) Sets the specified VLAN as the default AVB VLAN on the switch. Use this command when you need to set the default AVB VLAN other than VLAN 2. The range for <i>vlan-id</i> varies from 2 to 4094. |
| Step 8 | avb Example: Device(config-vlan)# avb | Configures AVB on the specified interface. |

| | Command or Action | Purpose |
|---------------|--|----------------------------------|
| Step 9 | end Example: Device(config) # end | Returns to privileged EXEC mode. |

What to do next

To disable AVB on the switch, use the **"no"** form of the command.

Configuring gPTP

This section describes the various configurations available for gPTP.

Enabling gPTP

When AVB is enabled on the switch, gPTP for AVB also gets enabled.

You can also enable gPTP globally using the command given below:

Procedure

| | Command or Action | Purpose |
|---------------|--|---|
| Step 1 | enable Example: Device> enable | Enables privileged EXEC mode. <ul style="list-style-type: none"> • Enter your password if prompted. |
| Step 2 | configure terminal Example: Device# configure terminal | Enters global configuration mode. |
| Step 3 | [no]ptp profile dot1as Example: Device(config) # ptp profile dot1as | gPTP is enabled globally when you enable AVB. Use the no form of this command to disable gPTP globally. |
| Step 4 | end Example: Device(config) # end | Returns to privileged EXEC mode. |

Enabling gPTP on an interface

You can also enable gPTP on an interface using the command given below:

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 the global configuration mode. |
| Step 3 | interface <i>interface-id</i> Example: Device(config)# interface tel1/1/1 | Defines the interface to be configured as a trunk, and enters interface configuration mode. The interface that you specify can be part of an EtherChannel. |
| Step 4 | ptp enable Example: Device(config-if)# ptp enable | Enables gPTP on all the interfaces. To disable gPTP on a port, use the no form of this command as shown below: Device(config-if)# no ptp enable |
| Step 5 | end Example: Device(config-if)# end | Returns to privileged EXEC mode. |

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 | 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 priority1 <i>value</i> Example: Device(config)# ptp priority1 120 | 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. |

| | Command or Action | Purpose |
|---------------|---|--|
| Step 4 | ptp priority2 <i>value</i> Example: Device(config) # ptp priority2 120 | 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) # exit | Returns to global configuration mode. |

Configuring HQoS

This section describes the various configurations available for HQoS.

Enabling HQoS

When AVB is enabled on the switch, HQoS for AVB also gets enabled.

Hierarchical QoS Policy Formats

This following example shows hierarchical remarking policy at the ingress interface:

```

policy-map AVB-Input-Child-Policy
  class VOIP-DATA-CLASS
    set dscp EF
  class MULTIMEDIA-CONF-CLASS
    set dscp AF41
  class BULK-DATA-CLASS
    set dscp AF11
  class TRANSACTIONAL-DATA-CLASS
    set dscp AF21
  class SCAVENGER-DATA-CLASS
    set dscp CS1
  class SIGNALING-CLASS
    set dscp CS3
  class class-default
    set dscp default

policy-map AVB-Input-Policy-Remark-AB
  class AVB-SR-A-CLASS
    set cos 0 (set 0 for boundary & SR class A PCP value for core port)
  class AVB-SR-B-CLASS
    set cos 0 (set 0 for boundary & SR class B PCP value for core port)
  class class-default
    service-policy AVB-Input-Child-Policy

policy-map AVB-Input-Policy-Remark-A
  class AVB-SR-A-CLASS
    set cos 0 (set 0 for boundary & SR class A PCP value for core port)
  class class-default
    service-policy AVB-Input-Child-Policy

policy-map AVB-Input-Policy-Remark-B
  class AVB-SR-B-CLASS
    set cos 0 (set 0 for boundary & SR class B PCP value for core port)
  class class-default
    service-policy AVB-Input-Child-Policy

```

```

policy-map AVB-Input-Policy-Remark-None
  class class-default
    service-policy AVB-Input-Child-Policy

```

This following example shows hierarchical queuing policy at the egress interface:

```

policy-map AVB-Output-Child-Policy
  class VOIP-PRIORITY-QUEUE
    bandwidth remaining percent 30
    queue-buffers ratio 10
  class MULTIMEDIA-CONFERENCING-STREAMING-QUEUE
    bandwidth remaining percent 15
    queue-limit dscp AF41 percent 80
    queue-limit dscp AF31 percent 80
    queue-limit dscp AF42 percent 90
    queue-limit dscp AF32 percent 90
    queue-buffers ratio 10
  class TRANSACTIONAL-DATA-QUEUE
    bandwidth remaining percent 15
    queue-limit dscp AF21 percent 80
    queue-limit dscp AF22 percent 90
    queue-buffers ratio 10
  class BULK-SCAVENGER-DATA-QUEUE
    bandwidth remaining percent 15
    queue-limit dscp AF11 percent 80
    queue-limit dscp AF12 percent 90
    queue-limit dscp CS1 percent 80
    queue-buffers ratio 15
  class class-default
    bandwidth remaining percent 25
    queue-buffers ratio 25

policy-map AVB-Output-Policy
  class AVB-SR-A-CLASS
    priority level 1 (Shaper value based on stream registration)
  class AVB-SR-B-CLASS
    priority level 2 (Shaper value based on stream registration)
  class CONTROL-MGMT-QUEUE
    priority level 3 percent 15
  class class-default
    bandwidth remaining percent 100
    queue-buffers ratio 80
    service-policy AVB-Output-Child-Policy

```

Configuring MVRP

This section describes the various configurations available for MVRP.

Enabling Multiple VLAN Registration Protocol

You can enable MVRP on the switches in the topology to enable VLAN propagation using the below command.



Note You must change VTP mode to **transparent** or **off** , before enabling dynamic vlan creation via MVRP.

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 | mvrp global Example: <pre>Device(config)# mvrp global</pre> | Enters the MVRP Global configuration mode. |
| Step 4 | vtp mode {transparent off} Example: <pre>Device(config)# vtp mode transparent</pre> Example: <pre>Device(config)# vtp mode off</pre> | Sets the VTP to transparent or off mode. |
| Step 5 | mvrp vlan create Example: <pre>Device(config)# mvrp vlan create</pre> | Enables MVRP on the switches. |

Configuring Multiple VLAN Registration Protocol on an Interface

You can configure MVRP on the switch interfaces using the below commands.

Procedure

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

| | Command or Action | Purpose |
|---------------|---|---|
| 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 tel1/1/1</pre> | Defines the interface to be configured as a trunk, and enters interface configuration mode. |
| Step 4 | mvrp registration {fixed forbidden normal} Example: <pre>Device(config-if)# mvrp registration fixed</pre> | Registers MVRP with the MAD instance. <ul style="list-style-type: none"> • fixed - Fixed registration • forbidden - Forbidden registration • normal - Normal registration |
| Step 5 | mvrp timer {join leave leave-all periodic} Example: <pre>Device(config-if)# mvrp timer join</pre> | Configures the MVRP timer. <ul style="list-style-type: none"> • join - Timer controls the interval between transmit opportunities that are applied to the ASM • leave - The timer controls the RSM waits in the LV state before transiting to the MT state • leave-all - The timer control the frequency with which the LeaveAll SM generates LeaveAll PDUs • periodic - Periodic timer |
| Step 6 | exit Example: <pre>Device(config-if)# exit</pre> | Returns to global configuration mode. |

Configuring MSRP

You can configure MSRP timer values using the below commands:

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 | avb [msrp-join-timer <i>milliseconds</i> msrp-leave-timer <i>milliseconds</i> msrp-leaveall-timer <i>milliseconds</i> msrp-tx-slow] Example: Device(config)# avb msrp-leave-timer 6000 | Configures the MSRP timer. <ul style="list-style-type: none"> • msrp-join-timer <i>milliseconds</i>: Configures the MSRP join timer value in milliseconds. • msrp-leave-timer <i>milliseconds</i>: Configures the MSRP leave timer value in milliseconds. • msrp-leaveall-timer <i>milliseconds</i>: Configures the MSRP leaveall timer value in milliseconds. • msrp-tx-slow: Slows down the default packet sending rate with a gap of 100 milliseconds. |
| Step 4 | end Example: Device(config)# end | Returns to privileged EXEC mode. |

Monitoring the AVB Network

Monitoring Audio Video Bridging

To display the AVB details, use the commands in the following table:

| Command | Purpose |
|------------------------|--------------------------------------|
| show avb domain | Displays the AVB domain. |
| show avb stream | Displays the AVB stream information. |

Monitoring Generalized Precision Time Protocol

To display the gPTP protocol details, use the commands in the following table:

| Command | Purpose |
|--|---|
| show ptp brief | Displays a brief status of ptp on the interfaces. |
| show ptp clock | Displays ptp clock information. |
| show ptp parent | Displays the parent clock information. |
| show ptp port | Displays the ptp port information. |
| show platform software fed switch active ptp if-id {interface-id} | Displays details info about ptp status on the port. |

Monitoring Multiple Stream Reservation Protocol

To display the MSRP details, use the commands in the following table:

| Command | Purpose |
|-----------------------------------|--|
| show msrp streams | Displays MSRP stream information. |
| show msrp streams detailed | Displays detailed MSRP stream information. |
| show msrp streams brief | Displays MSRP stream information in brief. |
| show msrp port bandwidth | Displays MSRP port bandwidth information. |

Monitoring Hierarchical QoS

To display the HQoS details, use the commands in the following table:

| Command | Purpose |
|--|--|
| show run | Displays all the child policy map details. |
| show policy-map | Displays the details of the policy map configuration. |
| show platform hardware fed switch active qos queue stats interface <i>interface-id</i> | Displays the QoS statistics for different queue mappings in AVB. |
| show platform hardware fed switch active qos queue config interface <i>interface-id</i> | Displays the QoS queue configurations. |
| show policy-map interface <i>interface-id</i> [input output] | Displays the AVB QoS statistics. Packet counters for ingress and bytes counters for egress are accounted for QoS Statistics. |

Monitoring Multiple VLAN Registration Protocol

To display the MVRP details, use the commands in the following table:

| Command | Purpose |
|----------------------------|--------------------------------------|
| show mvrp summary | Displays MVRP summary information. |
| show mvrp interface | Displays interface MVRP information. |

Examples of AVB Configurations and Monitoring

Examples for Audio Video Bridging

This example shows how you can view the AVB domain.

```
Device#show avb domain
```

```
AVB Class-A
  Priority Code Point      : 3
  VLAN                    : 2
  Core ports              : 1
  Boundary ports          : 67
```

```
AVB Class-B
  Priority Code Point      : 2
  VLAN                    : 2
  Core ports              : 1
  Boundary ports          : 67
```

| Interface | State | Delay | PCP | VID | Information |
|-----------|-------|-------|-----|-----|-----------------------|
| Te1/0/1 | down | N/A | | | Oper state not up |
| Te1/0/2 | down | N/A | | | Oper state not up |
| Te1/0/3 | down | N/A | | | Oper state not up |
| Te1/0/4 | down | N/A | | | Oper state not up |
| Te1/0/5 | up | N/A | | | Port is not asCapable |
| Te1/0/6 | down | N/A | | | Oper state not up |
| Te1/0/7 | down | N/A | | | Oper state not up |
| Te1/0/8 | down | N/A | | | Oper state not up |
| Te1/0/9 | down | N/A | | | Oper state not up |
| Te1/0/10 | down | N/A | | | Oper state not up |
| Te1/0/11 | down | N/A | | | Oper state not up |
| Te1/0/12 | down | N/A | | | Oper state not up |
| Te1/0/13 | down | N/A | | | Oper state not up |
| Te1/0/14 | down | N/A | | | Oper state not up |
| Te1/0/15 | down | N/A | | | Oper state not up |
| Te1/0/16 | down | N/A | | | Oper state not up |

| | | | | | |
|----------|------|-------|---|---|-----------------------|
| Tel/0/17 | down | N/A | | | Oper state not up |
| Tel/0/18 | down | N/A | | | Oper state not up |
| Tel/0/19 | up | N/A | | | Port is not asCapable |
| Tel/0/20 | down | N/A | | | Oper state not up |
| Tel/0/21 | down | N/A | | | Oper state not up |
| Tel/0/22 | down | N/A | | | Oper state not up |
| Tel/0/23 | up | N/A | | | Port is not asCapable |
| Tel/0/24 | down | N/A | | | Oper state not up |
| Tel/0/25 | down | N/A | | | Oper state not up |
| Tel/0/26 | down | N/A | | | Oper state not up |
| Tel/0/27 | down | N/A | | | Oper state not up |
| Tel/0/28 | down | N/A | | | Oper state not up |
| Tel/0/29 | up | N/A | | | Port is not asCapable |
| Tel/0/30 | down | N/A | | | Oper state not up |
| Tel/0/31 | down | N/A | | | Oper state not up |
| Tel/0/32 | down | N/A | | | Oper state not up |
| Tel/0/33 | down | N/A | | | Oper state not up |
| Tel/0/34 | down | N/A | | | Oper state not up |
| Tel/0/35 | up | N/A | | | Port is not asCapable |
| Tel/0/36 | down | N/A | | | Oper state not up |
| Tel/0/37 | down | N/A | | | Oper state not up |
| Tel/0/38 | down | N/A | | | Oper state not up |
| Tel/0/39 | up | 507ns | | | |
| Class- | A | core | 3 | 2 | |
| Class- | B | core | 2 | 2 | |
| Tel/0/40 | down | N/A | | | Oper state not up |
| Tel/0/41 | down | N/A | | | Oper state not up |
| Tel/0/42 | down | N/A | | | Oper state not up |
| Tel/0/43 | down | N/A | | | Oper state not up |
| Tel/0/44 | down | N/A | | | Oper state not up |
| Tel/0/45 | down | N/A | | | Oper state not up |
| Tel/0/46 | down | N/A | | | Oper state not up |
| Tel/0/47 | down | N/A | | | Oper state not up |
| Tel/0/48 | down | N/A | | | Oper state not up |
| Tel/1/1 | down | N/A | | | Oper state not up |
| Tel/1/2 | down | N/A | | | Oper state not up |
| Tel/1/3 | down | N/A | | | Oper state not up |
| Tel/1/4 | down | N/A | | | Oper state not up |
| Tel/1/5 | down | N/A | | | Oper state not up |
| Tel/1/6 | down | N/A | | | Oper state not up |
| Tel/1/7 | down | N/A | | | Oper state not up |
| Tel/1/8 | down | N/A | | | Oper state not up |
| Tel/1/9 | down | N/A | | | Oper state not up |
| Tel/1/10 | down | N/A | | | Oper state not up |
| Tel/1/11 | down | N/A | | | Oper state not up |
| Tel/1/12 | down | N/A | | | Oper state not up |
| Tel/1/13 | down | N/A | | | Oper state not up |
| Tel/1/14 | down | N/A | | | Oper state not up |
| Tel/1/15 | down | N/A | | | Oper state not up |

```

Tel1/1/16      down      N/A      Oper state not up
Fo1/1/1        down      N/A      Oper state not up
Fo1/1/2        down      N/A      Oper state not up
Fo1/1/3        down      N/A      Oper state not up
Fo1/1/4        down      N/A      Oper state not up

```

This example shows how you can view the AVB stream information.

Device# **show avb stream**

```

Stream ID:      0011.0100.0001:1      Incoming Interface:  Tel1/1/1
  Destination   : 91E0.F000.FE00
  Class         : A
  Rank          : 1
  Bandwidth     : 6400 Kbit/s

```

Outgoing Interfaces:

```

-----
Interface      State      Time of Last Update      Information
-----
Tel1/1/1       Ready      Tue Apr 26 01:25:40.634

```

```

Stream ID:      0011.0100.0002:2      Incoming Interface:  Tel1/1/1
  Destination   : 91E0.F000.FE01
  Class         : A
  Rank          : 1
  Bandwidth     : 6400 Kbit/s

```

Outgoing Interfaces:

```

-----
Interface      State      Time of Last Update      Information
-----
Tel1/1/1       Ready      Tue Apr 26 01:25:40.634

```

Example: Verifying Generalized Precision Time Protocol

This command can be used to see a brief status of ptp on the interfaces.

Device# **show ptp brief**

Example: Verifying Generalized Precision Time Protocol

| Interface | Domain | PTP State |
|---------------------------|--------|-----------|
| FortyGigabitEthernet1/1/1 | 0 | FAULTY |
| FortyGigabitEthernet1/1/2 | 0 | SLAVE |
| GigabitEthernet1/1/1 | 0 | FAULTY |
| GigabitEthernet1/1/2 | 0 | FAULTY |
| GigabitEthernet1/1/3 | 0 | FAULTY |
| GigabitEthernet1/1/4 | 0 | FAULTY |
| TenGigabitEthernet1/0/1 | 0 | FAULTY |
| TenGigabitEthernet1/0/2 | 0 | FAULTY |
| TenGigabitEthernet1/0/3 | 0 | MASTER |
| TenGigabitEthernet1/0/4 | 0 | FAULTY |
| TenGigabitEthernet1/0/5 | 0 | FAULTY |
| TenGigabitEthernet1/0/6 | 0 | FAULTY |
| TenGigabitEthernet1/0/7 | 0 | MASTER |
| TenGigabitEthernet1/0/8 | 0 | FAULTY |
| TenGigabitEthernet1/0/9 | 0 | FAULTY |
| TenGigabitEthernet1/0/10 | 0 | FAULTY |
| TenGigabitEthernet1/0/11 | 0 | MASTER |
| TenGigabitEthernet1/0/12 | 0 | FAULTY |
| TenGigabitEthernet1/0/13 | 0 | FAULTY |
| TenGigabitEthernet1/0/14 | 0 | FAULTY |
| TenGigabitEthernet1/0/15 | 0 | FAULTY |
| TenGigabitEthernet1/0/16 | 0 | FAULTY |
| TenGigabitEthernet1/0/17 | 0 | FAULTY |
| TenGigabitEthernet1/0/18 | 0 | FAULTY |
| TenGigabitEthernet1/0/19 | 0 | MASTER |
| TenGigabitEthernet1/0/20 | 0 | FAULTY |
| TenGigabitEthernet1/0/21 | 0 | FAULTY |
| TenGigabitEthernet1/0/22 | 0 | FAULTY |
| TenGigabitEthernet1/0/23 | 0 | FAULTY |
| TenGigabitEthernet1/0/24 | 0 | FAULTY |
| TenGigabitEthernet1/1/1 | 0 | FAULTY |
| TenGigabitEthernet1/1/2 | 0 | FAULTY |
| TenGigabitEthernet1/1/3 | 0 | FAULTY |
| TenGigabitEthernet1/1/4 | 0 | FAULTY |
| TenGigabitEthernet1/1/5 | 0 | FAULTY |
| TenGigabitEthernet1/1/6 | 0 | FAULTY |
| TenGigabitEthernet1/1/7 | 0 | FAULTY |
| TenGigabitEthernet1/1/8 | 0 | FAULTY |

This command can be used to view ptp clock information.

Device# **show ptp clock**

PTP CLOCK INFO

PTP Device Type: Boundary clock
 PTP Device Profile: IEEE 802/1AS Profile
 Clock Identity: 0x4:6C:9D:FF:FE:4F:95:0
 Clock Domain: 0
 Number of PTP ports: 38


```
PTP Packet priority: 4
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: 3
Local clock time: 00:12:13 UTC Jan 1 1970
```

This command can be used to view the parent clock information.

```
Device# show ptp parent
```

PTP PARENT PROPERTIES

```
Parent Clock:
Parent Clock Identity: 0xB0:7D:47:FF:FE:9E:B6:80
Parent Port Number: 3
Observed Parent Offset (log variance): 16640
Observed Parent Clock Phase Change Rate: N/A

Grandmaster Clock:
Grandmaster Clock Identity: 0x4:6C:9D:FF:FE:67:3A:80
Grandmaster Clock Quality:
    Class: 248
    Accuracy: Unknown
    Offset (log variance): 16640
    Priority1: 0
    Priority2: 128
```

This command can be used to view the ptp port information.

```
Device# show ptp port
```

```
PTP PORT DATASET: FortyGigabitEthernet1/1/1
Port identity: clock identity: 0x4:6C:9D:FF:FE:4E:3A:80
Port identity: port number: 1
PTP version: 2
Port state: FAULTY
Delay request interval(log mean): 5
Announce receipt time out: 3
Peer mean path delay(ns): 0
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
```

```

PTP PORT DATASET: FortyGigabitEthernet1/1/2
  Port identity: clock identity: 0x4:6C:9D:FF:FE:4E:3A:80
  Port identity: port number: 2
  PTP version: 2
  Port state: FAULTY
  Delay request interval(log mean): 5
  Announce receipt time out: 3
  Peer mean path delay(ns): 0
  Announce interval(log mean): 1
--More--

```

This command can be used to view the port information for a particular interface.

Device# **show ptp port gi1/0/26**

```

PTP PORT DATASET: GigabitEthernet1/0/26
  Port identity: clock identity: 0x4:6C:9D:FF:FE:4E:3A:80
  Port identity: port number: 28
  PTP version: 2
  Port state: MASTER
  Delay request interval(log mean): 5
  Announce receipt time out: 3
  Peer mean path delay(ns): 0
  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

```

This command can be used to view the

Device# **show platform software fed switch active ptp if-id 0x20**

Displaying port data for if_id 20

```

=====
Port Mac Address 04:6C:9D:4E:3A:9A
Port Clock Identity 04:6C:9D:FF:FE:4E:3A:80
Port number 28
PTP Version 2
domain_value 0
dot1as_capable: FALSE
sync_recpt_timeout_time_interval 375000000 nanoseconds
sync_interval 125000000 nanoseconds
neighbor_rate_ratio 0.000000
neighbor_prop_delay 0 nanoseconds
compute_neighbor_rate_ratio: TRUE
compute_neighbor_prop_delay: TRUE

```

```

port_enabled: TRUE
ptt_port_enabled: TRUE
current_log_pdelay_req_interval 0
pdelay_req_interval 0 nanoseconds
allowed_lost_responses 3
neighbor_prop_delay_threshold 2000 nanoseconds
is_measuring_delay : FALSE
Port state: : MASTER
sync_seq_num 22023
delay_req_seq_num 23857
num sync messages transmitted 0
num sync messages received 0
num followup messages transmitted 0
num followup messages received 0
num pdelay requests transmitted 285695
num pdelay requests received 0
num pdelay responses transmitted 0
num pdelay responses received 0
num pdelay followup responses transmitted 0
num pdelay followup responses received 0

```

Example: Verifying Multiple Stream Reservation Protocol

This example shows how you can view the MSRP stream information.

Device# **show msrp streams**

```

-----
Stream ID Talker Listener
Advertise Fail Ready ReadyFail AskFail
R | D R | D R | D R | D R | D
-----
YY:YY:YY:YY:YY:YY:0001 1 | 2 0 | 0 1 | 0 0 | 1 1 | 0
zz:zz:zz:zz:zz:zz:0002 1 | 0 0 | 1 1 | 0 0 | 0 0 | 1
-----

```

This example shows how you can view the detailed MSRP stream information.

Device# **show msrp streams detail**

```

Stream ID:          0011.0100.0001:1
  Stream Age: 01:57:46 (since Mon Apr 25 23:41:11.413)
  Create Time: Mon Apr 25 23:41:11.413
  Destination Address: 91E0.F000.FE00
  VLAN Identifier: 1
  Data Frame Priority: 3 (Class A)
  MaxFrameSize: 100
  MaxIntervalFrames: 1 frames/125us
  Stream Bandwidth: 6400 Kbit/s

```

Example: Verifying Multiple Stream Reservation Protocol

```
Rank: 1
Received Accumulated Latency: 20
Stream Attributes Table:
```

```
-----
Interface          Attr State      Direction      Type
-----
Gil/0/1            Register        Talker         Advertise
Attribute Age: 01:57:46 (since Mon Apr 25 23:41:11.413)
MRP Applicant: Very Anxious Observer, send None
MRP Registrar: In
Accumulated Latency: 20
-----
Tel/1/1            Declare         Talker         Advertise
Attribute Age: 00:19:52 (since Tue Apr 26 01:19:05.525)
MRP Applicant: Quiet Active, send None
MRP Registrar: In
Accumulated Latency: 20
-----
Tel/1/1            Register        Listener        Ready
Attribute Age: 00:13:17 (since Tue Apr 26 01:25:40.635)
MRP Applicant: Very Anxious Observer, send None
MRP Registrar: In
-----
Gil/0/1            Declare         Listener        Ready
Attribute Age: 00:13:17 (since Tue Apr 26 01:25:40.649)
MRP Applicant: Quiet Active, send None
MRP Registrar: In
-----
```

This example shows how you can view the MSRP stream information in brief.

```
Device# show msrp streams brief
```

Legend: R = Registered, D = Declared.

```
-----
Stream ID          Destination          Bandwidth    Talkers    Listeners
  Fail                                     (Kbit/s)    R | D      R | D
-----
0011.0100.0001:1   91E0.F000.FE00       6400         1 | 1      1 | 1
  No
0011.0100.0002:2   91E0.F000.FE01       6400         1 | 1      1 | 1
  No
0011.0100.0003:3   91E0.F000.FE02       6400         1 | 1      1 | 1
  No
0011.0100.0004:4   91E0.F000.FE03       6400         1 | 1      1 | 1
```

```

No
0011.0100.0005:5      91E0.F000.FE04      6400      1 | 1      1 | 1
No
0011.0100.0006:6      91E0.F000.FE05      6400      1 | 1      1 | 1
No
0011.0100.0007:7      91E0.F000.FE06      6400      1 | 1      1 | 1
No
0011.0100.0008:8      91E0.F000.FE07      6400      1 | 1      1 | 1
No
0011.0100.0009:9      91E0.F000.FE08      6400      1 | 1      1 | 1
No
0011.0100.000A:10     91E0.F000.FE09      6400      1 | 1      1 | 1
No

```

This example shows how you can view the MSRP port bandwidth information.

Device# **show msrp port bandwidth**

```

-----
Ethernet      Capacity      Assigned      Available      Reserved
Interface      (Kbit/s)      A | B          A | B          A | B
-----
Tel1/0/1       10000000      75 | 0         75 | 75        0 | 0
Tel1/0/2       10000000      75 | 0         75 | 75        0 | 0
Tel1/0/3       1000000      75 | 0         75 | 75        0 | 0
Tel1/0/4       10000000      75 | 0         75 | 75        0 | 0
Tel1/0/5       10000000      75 | 0         75 | 75        0 | 0
Tel1/0/6       10000000      75 | 0         75 | 75        0 | 0
Tel1/0/8       10000000      75 | 0         75 | 75        0 | 0
Tel1/0/9       10000000      75 | 0         75 | 75        0 | 0
Tel1/0/10      10000000      75 | 0         75 | 75        0 | 0
Tel1/0/11      10000000      75 | 0         75 | 75        0 | 0
Tel1/0/12      10000000      75 | 0         75 | 75        0 | 0
Tel1/0/13      1000000      75 | 0         75 | 75        0 | 0
Tel1/0/14      10000000      75 | 0         75 | 75        0 | 0
Tel1/0/15      10000000      75 | 0         75 | 75        0 | 0
Tel1/0/16      10000000      75 | 0         75 | 75        0 | 0
Tel1/0/17      10000000      75 | 0         75 | 75        0 | 0
Tel1/0/18      10000000      75 | 0         75 | 75        0 | 0
Tel1/0/19      1000000      75 | 0         75 | 75        0 | 0
Tel1/0/20      10000000      75 | 0         75 | 75        0 | 0
Tel1/0/21      10000000      75 | 0         75 | 75        0 | 0
Tel1/0/22      10000000      75 | 0         75 | 75        0 | 0
Tel1/0/23      10000000      75 | 0         75 | 75        0 | 0
Tel1/0/24      10000000      75 | 0         75 | 75        0 | 0
Gi1/1/1        1000000      75 | 0         75 | 75        0 | 0
Gi1/1/2        1000000      75 | 0         75 | 75        0 | 0
Gi1/1/3        1000000      75 | 0         75 | 75        0 | 0
Gi1/1/4        1000000      75 | 0         75 | 75        0 | 0

```

| | | | | |
|---------|-----------|--------|---------|-------|
| Tel/1/1 | 100000000 | 75 0 | 75 75 | 0 0 |
| Tel/1/2 | 100000000 | 75 0 | 75 75 | 0 0 |
| Tel/1/3 | 100000000 | 75 0 | 75 75 | 0 0 |
| Tel/1/4 | 100000000 | 75 0 | 75 75 | 0 0 |
| Tel/1/5 | 100000000 | 75 0 | 75 75 | 0 0 |
| Tel/1/6 | 100000000 | 75 0 | 75 75 | 0 0 |
| Tel/1/7 | 100000000 | 75 0 | 75 75 | 0 0 |
| Tel/1/8 | 100000000 | 75 0 | 75 75 | 0 0 |
| Fo1/1/1 | 400000000 | 75 0 | 75 75 | 0 0 |
| Fo1/1/2 | 400000000 | 75 0 | 75 75 | 0 0 |

Example: Verifying Hierarchical QoS

This example shows how you can view all the policy-map configuration details when AVB is enabled.

Device# **show policy-map**

```

Policy Map AVB-Input-Policy-Remark-B
  Class AVB-SR-CLASS-A
    set cos 3
  Class AVB-SR-CLASS-B
    set cos 0
  Class class-default
    service-policy AVB-Input-Child-Policy

Policy Map AVB-Input-Policy-Remark-A
  Class AVB-SR-CLASS-A
    set cos 0
  Class AVB-SR-CLASS-B
    set cos 2
  Class class-default
    service-policy AVB-Input-Child-Policy

Policy Map AVB-Output-Policy-Default
  Class AVB-SR-CLASS-A
    priority level 1 1 (%)
  Class AVB-SR-CLASS-B
    priority level 2 1 (%)
  Class AVB-CONTROL-MGMT-QUEUE
    priority level 3 15 (%)
  Class class-default
    bandwidth remaining 100 (%)
    queue-buffers ratio 70
    service-policy AVB-Output-Child-Policy

Policy Map AVB-Input-Policy-Remark-AB
  Class AVB-SR-CLASS-A
    set cos 0
  Class AVB-SR-CLASS-B
    set cos 0
  Class class-default

```

```
service-policy AVB-Input-Child-Policy

Policy Map AVB-Input-Policy-Remark-None
  Class AVB-SR-CLASS-A
    set cos 3
  Class AVB-SR-CLASS-B
    set cos 2
  Class class-default
    service-policy AVB-Input-Child-Policy

Policy Map AVB-Input-Child-Policy
  Class AVB-VOIP-DATA-CLASS
    set dscp ef
  Class AVB-MULTIMEDIA-CONF-CLASS
    set dscp af41
  Class AVB-BULK-DATA-CLASS
    set dscp af11
  Class AVB-TRANSACTIONAL-DATA-CLASS
    set dscp af21
  Class AVB-SCAVENGER-DATA-CLASS
    set dscp cs1
  Class AVB-SIGNALING-CLASS
    set dscp cs3
  Class class-default
    set dscp default

Policy Map AVB-Output-Child-Policy
  Class AVB-VOIP-PRIORITY-QUEUE
    bandwidth remaining 30 (%)
    queue-buffers ratio 30
  Class AVB-MULTIMEDIA-CONF-STREAMING-QUEUE
    bandwidth remaining 15 (%)
    queue-limit dscp af41 percent 80
    queue-limit dscp af31 percent 80
    queue-limit dscp af42 percent 90
    queue-limit dscp af32 percent 90
    queue-buffers ratio 15
  Class AVB-TRANSACTIONAL-DATA-QUEUE
    bandwidth remaining 15 (%)
    queue-limit dscp af21 percent 80
    queue-limit dscp af22 percent 90
    queue-buffers ratio 15
  Class AVB-BULK-SCAVENGER-DATA-QUEUE
    bandwidth remaining 15 (%)
    queue-limit dscp af11 percent 80
    queue-limit dscp af12 percent 90
    queue-limit dscp cs1 percent 80
    queue-buffers ratio 15
  Class class-default
    bandwidth remaining 25 (%)
    queue-buffers ratio 25
```

This example shows how you can view all the policy-map configuration details when AVB is disabled.

```
Device# show policy-map

Building configuration...

Current configuration : 2079 bytes
!
policy-map AVB-Input-Child-Policy
class AVB-VOIP-DATA-CLASS
    set dscp ef
class AVB-MULTIMEDIA-CONF-CLASS
    set dscp af41
class AVB-BULK-DATA-CLASS
    set dscp af11
class AVB-TRANSACTIONAL-DATA-CLASS
    set dscp af21
class AVB-SCAVENGER-DATA-CLASS
    set dscp cs1
class AVB-SIGNALING-CLASS
    set dscp cs3
class class-default
    set dscp default
policy-map AVB-Output-Child-Policy
class AVB-VOIP-PRIORITY-QUEUE
    bandwidth remaining percent 30
    queue-buffers ratio 30
class AVB-MULTIMEDIA-CONF-STREAMING-QUEUE
    bandwidth remaining percent 15
    queue-limit dscp af41 percent 80
    queue-limit dscp af31 percent 80
    queue-limit dscp af42 percent 90
    queue-limit dscp af32 percent 90
    queue-buffers ratio 15
class AVB-TRANSACTIONAL-DATA-QUEUE
    bandwidth remaining percent 15
    queue-limit dscp af21 percent 80
    queue-limit dscp af22 percent 90
    queue-buffers ratio 15
class AVB-BULK-SCAVENGER-DATA-QUEUE
    bandwidth remaining percent 15
    queue-limit dscp af11 percent 80
    queue-limit dscp af12 percent 90
    queue-limit dscp cs1 percent 80
    queue-buffers ratio 15
class class-default
    bandwidth remaining percent 25
```



```
    queue-buffers ratio 25
!
end
```

This example shows how you can view all the class-map configuration details when AVB is enabled.

Device# **show class-map**

```
Class Map match-any AVB-VOIP-DATA-CLASS (id 31)
  Match  dscp ef (46)
  Match  cos  5

Class Map match-any AVB-BULK-DATA-CLASS (id 33)
  Match  access-group name AVB-BULK-DATA-CLASS-ACL

Class Map match-any AVB-VOIP-PRIORITY-QUEUE (id 37)
  Match  dscp cs4 (32) cs5 (40) ef (46)
  Match  precedence 4  5
  Match  cos  5

Class Map match-any AVB-MULTIMEDIA-CONF-CLASS (id 32)
  Match  access-group name AVB-MULTIMEDIA-CONF-CLASS-ACL

Class Map match-any AVB-SIGNALING-CLASS (id 36)
  Match  access-group name AVB-SIGNALING-CLASS-ACL

Class Map match-any AVB-MULTIMEDIA-CONF-STREAMING-QUEUE (id 38)
  Match  dscp af41 (34) af42 (36) af43 (38)
  Match  dscp af31 (26) af32 (28) af33 (30)
  Match  cos  4

Class Map match-any AVB-BULK-SCAVENGER-DATA-QUEUE (id 40)
  Match  dscp cs1 (8) af11 (10) af12 (12) af13 (14)
  Match  precedence 1
  Match  cos  1

Class Map match-any AVB-TRANSACTIONAL-DATA-CLASS (id 34)
  Match  access-group name AVB-TRANSACTIONAL-DATA-CLASS-ACL

Class Map match-any AVB-TRANSACTIONAL-DATA-QUEUE (id 39)
  Match  dscp af21 (18) af22 (20) af23 (22)

Class Map match-any AVB-SR-CLASS-B (id 42)
  Match  cos  2

Class Map match-any AVB-SR-CLASS-A (id 41)
  Match  cos  3
```

```

Class Map match-any AVB-SCAVENGER-DATA-CLASS (id 35)
  Match access-group name AVB-SCAVENGER-DATA-CLASS-ACL

Class Map match-any AVB-CONTROL-MGMT-QUEUE (id 43)
  Match ip dscp cs2 (16)
  Match ip dscp cs3 (24)
  Match ip dscp cs6 (48)
  Match ip dscp cs7 (56)
  Match ip precedence 6
  Match ip precedence 7
  Match ip precedence 3
  Match ip precedence 2
  Match cos 6
  Match cos 7

```

This example shows how you can view all the class-map configuration details when AVB is disabled.

Device# **show class-map**

Building configuration...

```

Current configuration : 2650 bytes
!
class-map match-any AVB-VOIP-DATA-CLASS
match dscp ef
  match cos 5
class-map match-any AVB-BULK-DATA-CLASS
match access-group name AVB-BULK-DATA-CLASS-ACL
class-map match-any AVB-VOIP-PRIORITY-QUEUE
match dscp cs4 cs5 ef
  match precedence 4 5
  match cos 5
class-map match-any AVB-MULTIMEDIA-CONF-CLASS
match access-group name AVB-MULTIMEDIA-CONF-CLASS-ACL
class-map match-any AVB-SIGNALING-CLASS
match access-group name AVB-SIGNALING-CLASS-ACL
class-map match-any AVB-MULTIMEDIA-CONF-STREAMING-QUEUE
match dscp af41 af42 af43
  match dscp af31 af32 af33
  match cos 4
class-map match-any AVB-BULK-SCAVENGER-DATA-QUEUE
match dscp cs1 af11 af12 af13
  match precedence 1
  match cos 1
class-map match-any AVB-TRANSACTIONAL-DATA-CLASS
match access-group name AVB-TRANSACTIONAL-DATA-CLASS-ACL
class-map match-any AVB-TRANSACTIONAL-DATA-QUEUE
match dscp af21 af22 af23
class-map match-any AVB-SCAVENGER-DATA-CLASS

```

```
match access-group name AVB-SCAVENGER-DATA-CLASS-ACL
end
```

This example shows how you can view all the AVB QoS statistics.

```
Device# show policy-map interface gigabitEthernet 1/0/15
```

```
GigabitEthernet1/0/15
```

```
Service-policy input: AVB-Input-Policy-Remark-AB
```

```
Class-map: AVB-SR-CLASS-A (match-any)
  0 packets
  Match: cos 3
        0 packets, 0 bytes
        5 minute rate 0 bps
  QoS Set
    cos 0
```

```
Class-map: AVB-SR-CLASS-B (match-any)
  0 packets
  Match: cos 2
        0 packets, 0 bytes
        5 minute rate 0 bps
  QoS Set
    cos 0
```

```
Class-map: class-default (match-any)
  0 packets
  Match: any
```

```
Service-policy : AVB-Input-Child-Policy
```

```
Class-map: AVB-VOIP-DATA-CLASS (match-any)
  0 packets
  Match: dscp ef (46)
        0 packets, 0 bytes
        5 minute rate 0 bps
  Match: cos 5
        0 packets, 0 bytes
        5 minute rate 0 bps
  QoS Set
    cos 3
```

```
Class-map: AVB-MULTIMEDIA-CONF-CLASS (match-any)
  0 packets
  Match: access-group name AVB-MULTIMEDIA-CONF-CLASS-ACL
        0 packets, 0 bytes
        5 minute rate 0 bps
  QoS Set
```

```

    dscp af41

Class-map: AVB-BULK-DATA-CLASS (match-any)
  0 packets
  Match: access-group name AVB-BULK-DATA-CLASS-ACL
    0 packets, 0 bytes
    5 minute rate 0 bps
  QoS Set
    dscp af11

Class-map: AVB-TRANSACTIONAL-DATA-CLASS (match-any)
  0 packets
  Match: access-group name AVB-TRANSACTIONAL-DATA-CLASS-ACL
    0 packets, 0 bytes
    5 minute rate 0 bps
  QoS Set
    dscp af21

Class-map: AVB-SCAVENGER-DATA-CLASS (match-any)
  0 packets
  Match: access-group name AVB-SCAVENGER-DATA-CLASS-ACL
    0 packets, 0 bytes
    5 minute rate 0 bps
  QoS Set
    dscp cs1

Class-map: AVB-SIGNALING-CLASS (match-any)
  0 packets
  Match: access-group name AVB-SIGNALING-CLASS-ACL
    0 packets, 0 bytes
    5 minute rate 0 bps
  QoS Set
    dscp cs3

Class-map: class-default (match-any)
  0 packets
  Match: any
  QoS Set
    dscp default

Service-policy output: AVB-Output-Policy-Default

queue stats for all priority classes:
  Queueing
  priority level 3

  (total drops) 0
  (bytes output) 7595

queue stats for all priority classes:
  Queueing

```

```
priority level 2

(total drops) 0
(bytes output) 0

queue stats for all priority classes:
Queueing
priority level 1

(total drops) 0
(bytes output) 0

Class-map: AVB-SR-CLASS-A (match-any)
  0 packets
  Match: cos 3
    0 packets, 0 bytes
    5 minute rate 0 bps
  Priority: 1% (10000 kbps), burst bytes 250000,

  Priority Level: 1

Class-map: AVB-SR-CLASS-B (match-any)
  0 packets
  Match: cos 2
    0 packets, 0 bytes
    5 minute rate 0 bps
  Priority: 1% (10000 kbps), burst bytes 250000,

  Priority Level: 2

Class-map: AVB-CONTROL-MGMT-QUEUE (match-any)
  0 packets
  Match: ip dscp cs2 (16)
    0 packets, 0 bytes
    5 minute rate 0 bps
  Match: ip dscp cs3 (24)
    0 packets, 0 bytes
    5 minute rate 0 bps
  Match: ip dscp cs6 (48)
    0 packets, 0 bytes
    5 minute rate 0 bps
  Match: ip dscp cs7 (56)
    0 packets, 0 bytes
    5 minute rate 0 bps
  Match: ip precedence 6
    0 packets, 0 bytes
    5 minute rate 0 bps
  Match: ip precedence 7
    0 packets, 0 bytes
    5 minute rate 0 bps
  Match: ip precedence 3
```

```

    0 packets, 0 bytes
    5 minute rate 0 bps
Match: ip precedence 2
    0 packets, 0 bytes
    5 minute rate 0 bps
Match: cos 6
    0 packets, 0 bytes
    5 minute rate 0 bps
Match: cos 7
    0 packets, 0 bytes
    5 minute rate 0 bps
Priority: 15% (150000 kbps), burst bytes 3750000,

Priority Level: 3

Class-map: class-default (match-any)
  0 packets
  Match: any
  Queueing

  (total drops) 0
  (bytes output) 0
  bandwidth remaining 80%
  queue-buffers ratio 70

Service-policy : AVB-Output-Child-Policy

Class-map: AVB-VOIP-PRIORITY-QUEUE (match-any)
  0 packets
  Match: dscp cs4 (32) cs5 (40) ef (46)
    0 packets, 0 bytes
    5 minute rate 0 bps
  Match: precedence 4 5
    0 packets, 0 bytes
    5 minute rate 0 bps
  Match: cos 5
    0 packets, 0 bytes
    5 minute rate 0 bps
  Queueing

  (total drops) 0
  (bytes output) 0
  bandwidth remaining 30%
  queue-buffers ratio 30

Class-map: AVB-MULTIMEDIA-CONF-STREAMING-QUEUE (match-any)
  0 packets
  Match: dscp af41 (34) af42 (36) af43 (38)
    0 packets, 0 bytes
    5 minute rate 0 bps
  Match: dscp af31 (26) af32 (28) af33 (30)

```

```
    0 packets, 0 bytes
    5 minute rate 0 bps
Match: cos 4
    0 packets, 0 bytes
    5 minute rate 0 bps
Queueing

queue-limit dscp 26 percent 80
queue-limit dscp 28 percent 90
queue-limit dscp 34 percent 80
queue-limit dscp 36 percent 90
(total drops) 0
(bytes output) 0
bandwidth remaining 15%

queue-buffers ratio 15

Class-map: AVB-TRANSACTIONAL-DATA-QUEUE (match-any)
  0 packets
  Match: dscp af21 (18) af22 (20) af23 (22)
    0 packets, 0 bytes
    5 minute rate 0 bps
  Match: cos 0
    0 packets, 0 bytes
    5 minute rate 0 bps
  Queueing

  queue-limit dscp 18 percent 80
  queue-limit dscp 20 percent 90
  (total drops) 0
  (bytes output) 0
  bandwidth remaining 15%

  queue-buffers ratio 15

Class-map: AVB-BULK-SCAVENGER-DATA-QUEUE (match-any)
  0 packets
  Match: dscp cs1 (8) af11 (10) af12 (12) af13 (14)
    0 packets, 0 bytes
    5 minute rate 0 bps
  Match: precedence 1
    0 packets, 0 bytes
    5 minute rate 0 bps
  Match: cos 1
    0 packets, 0 bytes
    5 minute rate 0 bps
  Queueing

  queue-limit dscp 8 percent 80
  queue-limit dscp 10 percent 80
  queue-limit dscp 12 percent 90
```

```

(total drops) 0
(bytes output) 0
bandwidth remaining 15%

queue-buffers ratio 15

Class-map: class-default (match-any)
 0 packets
Match: any
Queueing

(total drops) 0
(bytes output) 0
bandwidth remaining 25%
queue-buffers ratio 25

```

The following is a sample output from the **show platform hardware fed switch active qos queue config interface interface-id** command.

```

Device# show platform hardware fed switch active qos queue config interface t1/0/11
DATA Port:2 GPN:11 AFD:Disabled QoSMap:2 HW Queues: 16 - 23
  DrainFast:Disabled PortSoftStart:1 - 3600

```

| | DTS | Hardmax | Softmax | PortSMin | GlblSMin | PortStEnd | |
|----------|-------|---------|---------------|----------|----------|-----------|--------------|
| | ----- | ----- | ----- | ----- | ----- | ----- | |
| 0 | 0 | 9 | 33 | 3 | 33 | 0 | 0 |
| 1 | 0 | 9 | 33 | 4 | 2400 | 99 | 99 |
| 2 | 1 | 6 | 30 | 4 | 2400 | 90 | 90 |
| 3 | 1 | 5 | 0 | 4 | 2400 | 189 | 189 |
| 4 | 1 | 5 | 0 | 4 | 2400 | 90 | 90 |
| 5 | 1 | 5 | 0 | 4 | 2400 | 90 | 90 |
| 6 | 1 | 5 | 0 | 4 | 2400 | 90 | 90 |
| 7 | 1 | 5 | 0 | 4 | 2400 | 153 | 153 |
| Priority | | | Shaped/shared | | weight | | shaping_step |
| | | | ----- | | ----- | | ----- |
| 0 | | 1 | Shaped | | 16383 | | 163 |
| 1 | | 2 | Shaped | | 16383 | | 163 |
| 2 | | 3 | Shaped | | 125 | | 153 |
| 3 | | 7 | Shared | | 50 | | 0 |
| 4 | | 7 | Shared | | 100 | | 0 |
| 5 | | 7 | Shared | | 100 | | 0 |
| 6 | | 7 | Shared | | 100 | | 0 |
| 7 | | 7 | Shared | | 60 | | 0 |

The following is a sample output from the **show platform hardware fed switch active qos queue stats interface interface-id** command.

```

Device# show platform hardware fed switch active qos queue stats interface t1/0/15
DATA Port:8 Enqueue Counters

```


| Queue | Buffers | Enqueue-TH0 | Enqueue-TH1 | Enqueue-TH2 |
|-------|---------|-------------|-------------|-------------|
| 0 | 1 | 0 | 0 | 23788459506 |
| 1 | 0 | 0 | 0 | 30973507838 |
| 2 | 0 | 0 | 12616270 | 13164040 |
| 3 | 0 | 0 | 0 | 0 |
| 4 | 0 | 0 | 0 | 0 |
| 5 | 0 | 0 | 0 | 0 |
| 6 | 0 | 0 | 0 | 0 |
| 7 | 0 | 0 | 0 | 119616 |

DATA Port:8 Drop Counters

| Queue | Drop-TH0 | Drop-TH1 | Drop-TH2 | SBufDrop | QebDrop |
|-------|----------|----------|----------|----------|---------|
| 0 | 0 | 0 | 0 | 0 | 0 |
| 1 | 0 | 0 | 0 | 0 | 0 |
| 2 | 0 | 0 | 0 | 0 | 0 |
| 3 | 0 | 0 | 0 | 0 | 0 |
| 4 | 0 | 0 | 0 | 0 | 0 |

Example: Verifying Multiple VLAN Registration Protocol

This example shows how you can view the MVRP summary information.

Device# **show mvrp summary**

```

MVRP global state           : enabled
MVRP VLAN creation          : enabled
VLANs created via MVRP      : 2,567
MAC learning auto provision  : disabled
Learning disabled on VLANs  : none

```

This example shows how you can view the interface MVRP information.

Device# **show mvrp interface**

```

Port      Status      Registrar State
Tel1/0/47  on              normal
Tel1/1/3   off             normal

Port      Join Timeout    Leave Timeout    Leaveall Timeout    Periodic
                                Timeout
Tel1/0/47  20                60               1000                100
Tel1/1/3   20                60               1000                100

Port      Vlans Declared

```

```

Tel/0/47      1-2,567,900
Tel/1/3       none

Port          Vlans Registered
Tel/0/47      2,567
Tel/1/3       none

Port          Vlans Registered and in Spanning Tree Forwarding State
Tel/0/47      2,567
Tel/1/3       none

```

Feature History for Audio Video Bridging

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 Fuji 16.8.1a | Audio Video Bridging (AVB): IEEE 802.1BA | AVB is a standard based mechanism where endpoints and the network will function as a whole to enable high quality A/V streaming across consumer applications to professional audio-video over an Ethernet infrastructure. Support for this feature was introduced on all the models of the Cisco Catalyst 9500 Series Switches. |
| Cisco IOS XE Gibraltar 16.12.5 | AVB MSRP | MSRP commands have been introduced to configure MSRP timer values. |
| Cisco IOS XE Amsterdam 17.2.1 | IEEE802.1AS (gPTP) support on EtherChannel Interfaces | Starting with this release the interface on which you configure gPTP can be part of an EtherChannel. Support for this feature was introduced on all the models of the Cisco Catalyst 9500 Series Switches. |

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 14

Configuring Flexlink+

- [Restrictions for FlexLink+, on page 273](#)
- [Information about FlexLink+, on page 273](#)
- [How to configure Flexlink+, on page 276](#)
- [Configuration Examples for FlexLink+, on page 283](#)
- [Feature History for FlexLink+, on page 284](#)

Restrictions for FlexLink+

- FlexLink+ is not supported in a Cisco StackWise Virtual solution.
- FlexLink+ is supported only on Layer 2 trunk ports and port channels and is not supported on interfaces configured on Layer 3 ports and on VLANs.



Note FlexLink+ is not supported on port channels configured with access mode.

Information about FlexLink+

The following sections provide information about FlexLink+

FlexLink+ Overview

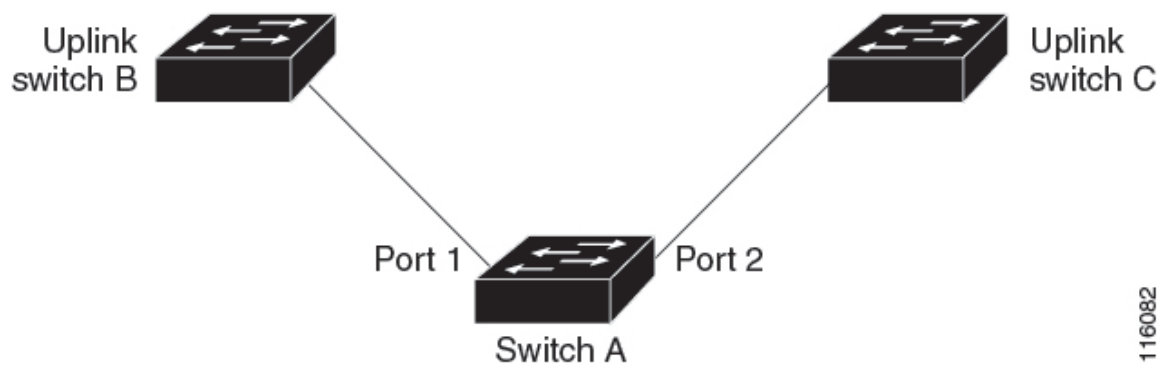
The FlexLink+ feature enables the user to configure a pair of a Layer 2 interfaces (trunk ports or port channels) where one interface is configured to act as a backup to the other. FlexLink+ provides an alternative solution to the Spanning Tree Protocol (STP) when you require simple link redundancy between two network nodes. STP is a complete solution that provides link redundancy and prevents loops in the network. If you need fast link redundancy between two nodes in the network, it is simpler and quicker to use FlexLink+. Flexlinks are typically configured in service provider or enterprise networks where customers do not want to run STP on the device. If the device is running STP, Flexlinks are not necessary because STP already provides link-level redundancy or backup.

In FlexLink+, when one of the links is up and forwarding traffic, the other link is in standby mode, ready to begin forwarding traffic if the active link shuts down. If the primary link shuts down, the standby link starts forwarding traffic. When the active link comes back up, it goes into standby mode and does not forward traffic.

FlexLink+ Configuration

In the following figure, ports 1 and 2 on switch A are connected to uplink switches B and C. Because they are configured with FlexLink+, only one of the interfaces is forwarding traffic; the other is in standby mode. If port 1 is the active link, it begins forwarding traffic between port 1 and switch B; the link between port 2 (the backup link) and switch C is not forwarding traffic. If port 1 goes down, port 2 comes up and starts forwarding traffic to switch C. When port 1 comes back up, it goes into standby mode and does not forward traffic; port 2 continues forwarding traffic.

Figure 48: FlexLink+ Topology



If STP is configured on the uplink switch interfaces that connect to the FlexLink+ ports (Switch B and Switch C in this case), we recommend running the **spanning-tree portfast trunk** command on such uplink switch interfaces, for faster convergence.

Flexlink+ includes an optimization for improved multicast traffic convergence. The optimization uses Layer 2 multicast snooping mechanisms and requires that the uplink switches connected to the Flexlink+ configured ports have the same Layer 2 multicast snooping feature that is enabled.

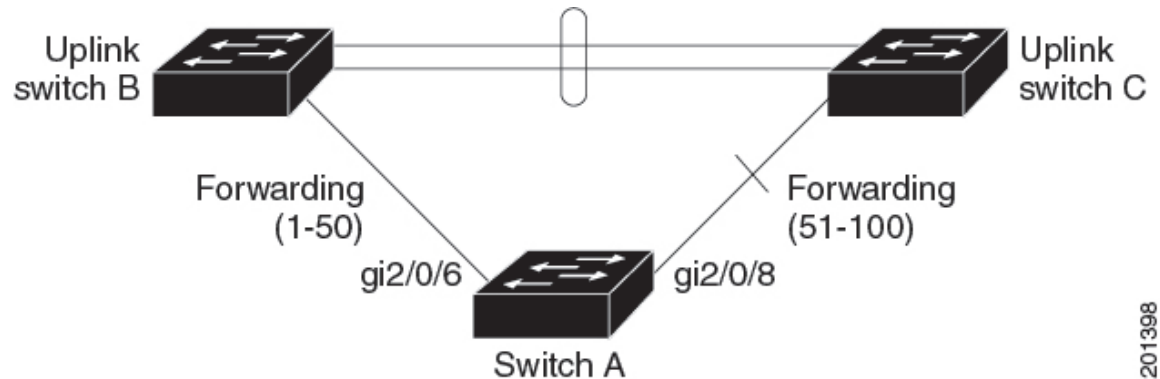


Note For IPv4 multicast IGMP snooping is on by default. If IGMP snooping needs to be disabled on the uplink switches it must also be disabled on the Flexlink+ host switch. Otherwise IGMP reports may be looped around the active and standby Flexlink+ ports leading to excessively high CPU utilization.

VLAN Load Balancing and FlexLink+

VLAN load-balancing allows you to configure a FlexLink+ pair so that both ports simultaneously forward the traffic for some mutually exclusive VLANs. For example, if FlexLink+ ports are configured for 1-100 VLANs, the traffic of the first 50 VLANs is forwarded on one port and the rest on the other port. If one of the ports fails, the other active port forwards all the traffic. When the failed port comes back up, it resumes forwarding traffic in the preferred VLANs. This way, apart from providing redundancy, this FlexLink+ pair is used for load balancing. FlexLink+ VLAN load-balancing doesn't impose any restrictions on uplink switches.

Figure 49: VLAN Load Balancing in FlexLink+ Topology



201398

When you configure VLAN load balancing, you must also configure triggers in one of two ways:

- Manually trigger VLAN load balancing at any time by entering the **rep preempt segment** privileged EXEC command on the switch that has the primary edge port.
- Configure a preempt delay time by entering the **rep preempt delay** interface configuration command. After a link failure and recovery, VLAN load balancing begins after the configured preempt time period elapses. The delay timer restarts if another port fails before the time has elapsed.



Note When VLAN load balancing is configured, it does not start working until triggered by either manual intervention or a link failure and recovery.

When VLAN load balancing is triggered, the primary edge port sends out a message to alert all interfaces in the segment about the preemption. When the secondary port receives the message, it is reflected into the network to notify the alternate port to block the set of VLANs specified in the message and to notify the primary edge port to block the remaining VLANs.

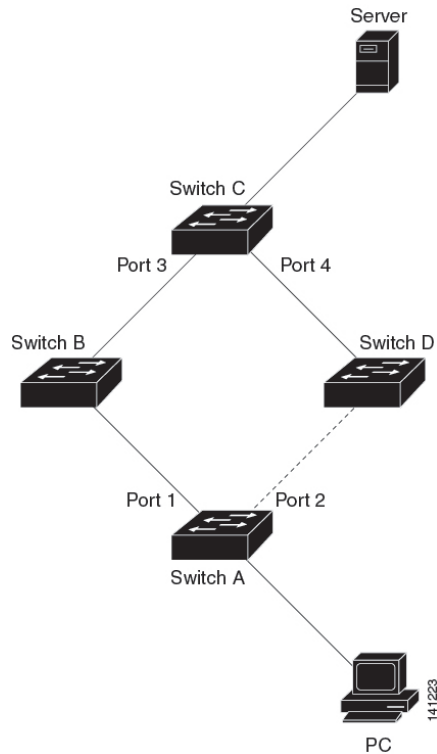
You can also configure a particular port in the segment to block all VLANs. Only the primary edge port initiates VLAN load balancing, which is not possible if the segment is not terminated by an edge port on each end. The primary edge port determines the local VLAN load balancing configuration.

Reconfigure the primary edge port to reconfigure load balancing. When you change the load balancing configuration, the primary edge port again waits for the **rep preempt segment** command or for the configured preempt delay period after a port failure and recovery before executing the new configuration. If you change an edge port to a regular segment port, the existing VLAN load balancing status does not change. Configuring a new edge port might cause a new topology configuration.

When VLAN load balancing is enabled, the default is manual preemption with the delay timer disabled. If VLAN load balancing is not configured, the default after manual preemption is to block all VLANs at the primary edge port.

When a primary link fails, FlexLink+ transmits dummy multicast packets over the new active interface. The dummy multicast packet format is:

- Destination: 01:00:0c:cd:cd:cd
- Source: MAC address of the hosts or ports on the newly active Flex Link port.

Figure 50: Transmission of Dummy Multicast Packets in FlexLink+ Topology

In the above figure, Ports 1 and 2 on switch A are connected to switches B and D through a Flex Link pair. Port 1 is forwarding traffic, and port 2 is in the blocking state. Traffic from the PC to the server is forwarded from port 1 to port 3. The MAC address of the PC has been learned on port 3 of switch C. Traffic from the server to the PC is forwarded from port 3 to port 1.

If port 1 shuts down, port 2 starts forwarding traffic. If there is no traffic from the PC to the server after failover to port 2, switch C does not learn the MAC address of the PC on port 4, and because of that, switch C keeps forwarding traffic from the server to the PC out of port 3. There is traffic loss from the server to the PC because port 1 is down. To alleviate this problem, the feature sends out a dummy multicast packet with the source MAC address of the PC over port 2. Switch C learns the PC MAC address on port 4 and starts forwarding traffic from the server to the PC out of port 4. One dummy multicast packet is sent out for every MAC address.

**Note**

- Local administrative shut down or a link that starts forwarding again due to preemption is not considered a link failure. In those cases, the feature flushes the dynamic hosts and does not move them.
- Static MAC addresses that are configured on a Flex Link port are restored when it starts forwarding again.

How to configure Flexlink+

The following sections provide information on how to configure Flexlink+.

Configuring the Active Port for FlexLink+

To configure the active port for FlexLink+, follow 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# interface Port-channel2 | Specifies the interface, and enters interface configuration mode. |
| Step 4 | switchport trunk allowed vlan <i>vlan-list</i> Example: Device(config-if)# switchport trunk allowed vlan 20-23,40,41 | Configures the allowed VLANs for an interface. |
| Step 5 | switchport mode trunk Example: Device(config-if)# switchport mode trunk | Configures the interface as a Layer 2 trunk. |
| Step 6 | rep segment <i>segment-id</i> edge no-neighbor primary Example: Device(config-if)# rep segment 1023 edge no-neighbor primary | Specifies that the port is the primary edge port where you can configure the active port of FlexLink+. A segment has only one primary edge port. |

Configuring the Standby Port for FlexLink+

To configure the standby port for FlexLink+, follow this procedure:

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# interface Port-channel7 | Specifies the interface, and enters interface configuration mode. |
| Step 4 | switchport trunk allowed vlan <i>vlan-list</i> Example: Device(config-if)# switchport trunk allowed vlan 20-23,40,41 | Configures the allowed VLANs for an interface. |
| Step 5 | switchport mode trunk Example: Device(config-if)# switchport mode trunk | Configures the interface as a Layer 2 trunk. |
| Step 6 | rep segment <i>segment-id</i> edge no-neighbor preferred Example: Device(config-if)# rep segment 1023 edge no-neighbor preferred | Specifies the segment edge as one with no external REP neighbor. Specifies that the port is the standby port for FlexLink+. <div> Note <ul style="list-style-type: none"> • Use the preferred keyword to ensure that the standby port becomes the blocking port. This optional keyword indicates that the port is the preferred alternate port or the preferred port for VLAN load balancing. • Configuring a port as preferred does not guarantee that it becomes the alternate port; it merely gives the port a slight edge over equal contenders. The alternate port is usually a previously failed port. </div> |

Configuring VLAN Load Balancing on FlexLink+

To configure VLAN 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 | interface interface-id Example: Device(config)# interface gigabitethernet2/0/8 | Specifies the interface, and enters interface configuration mode. The interface can be a physical Layer 2 interface or a port channel (logical interface). |
| Step 4 | switchport mode trunk Example: Device(config-if)# switchport mode trunk | Configures the interface as a Layer 2 trunk. |
| Step 5 | rep segment segment-id edge no-neighbor primary Example: Device(config-if)# rep segment 300 edge no-neighbor primary | Specifies that the port is the primary edge port. |
| Step 6 | rep block port port-number vlan vlan-range Example: Device(config-if)# rep block port 2 vlan 1-50 | Forwarding traffic for VLANs 1-50 is blocked on the standby port. Forwarding of traffic for VLANs 51-100 is blocked on the active port. |
| Step 7 | exit Example: Device(config-if) exit | Exits the Interface configuration mode. |
| Step 8 | interface interface-id Example: Device(config)# interface gigabitethernet2/0/6 | Specifies the interface, and enters interface configuration mode. The interface can be a physical Layer 2 interface or a port channel (logical interface). |

| | Command or Action | Purpose |
|----------------|--|---|
| Step 9 | switchport mode trunk Example: Device(config-if)# switchport mode trunk | Configures the interface as a Layer 2 trunk. |
| Step 10 | rep segment <i>segment-id</i> edge no-neighbor Example: Device(config-if)# rep segment 300 edge no-neighbor | Specifies the segment edge as one with no external REP neighbor. Specifies that the port is the standby port for Flexlink+. |
| Step 11 | end Example: Device(config-if)# end | Exits to privileged EXEC mode. |

Configuring Propagation of FlexLink+ Topology Change Messages

When the FlexLink+ protocol is deployed as part of a larger domain, you can configure the propagation of FlexLink+ topology change messages to the next tier devices. To configure the propagation of FlexLink+ topology change messages, 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 gigabitethernet2/0/8 | Specifies the interface, and enters interface configuration mode. The interface can be a physical Layer 2 interface or a port channel (logical interface). |
| Step 4 | switchport mode trunk Example: Device(config-if)# switchport mode trunk | Configures the interface as a Layer 2 trunk. |
| Step 5 | rep segment <i>segment-id</i> edge no-neighbor primary | Specifies the port as the primary edge port. |

| | Command or Action | Purpose |
|----------------|--|--|
| | Example: Device(config-if) # rep segment 300 edge no-neighbor primary | |
| Step 6 | rep stcn stp Example: Device(config-if) # rep stcn stp | Propagates FlexLink+ topology change messages to the next tier devices. |
| Step 7 | rep block port port-number vlan vlan-range Example: Device(config-if) # rep block port 2 vlan 1-50 | Forwarding traffic for VLANs 1-50 is blocked on the standby port. Forwarding of traffic for VLANs 51-100 is blocked on the active port. |
| Step 8 | exit Example: Device(config-if) exit | Exits the Interface configuration mode. |
| Step 9 | switchport mode trunk Example: Device(config-if) # switchport mode trunk | Configures the interface as a Layer 2 trunk. |
| Step 10 | interface interface-id Example: Device(config) # interface gigabitethernet2/0/6 | Specifies the interface, and enters interface configuration mode. The interface can be a physical Layer 2 interface or a port channel (logical interface). |
| Step 11 | rep segment segment-id edge no-neighbor Example: Device(config-if) # rep segment 300 edge no-neighbor | Specifies the segment edge as one with no external REP neighbor. Specifies that the port is the standby port for FlexLink+. |
| Step 12 | rep stcn stp Example: Device(config-if) # rep stcn stp | Propagates FlexLink+ topology change messages to the next tier devices. |
| Step 13 | end Example: Device(config-if) # end | Exits to privileged EXEC mode. |

Configuring Preempt Time Delay

To configure a preempt time delay for VLAN 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 | interface interface-id Example: Device(config)# interface gigabitethernet2/0/8 | Specifies the interface, and enters interface configuration mode. The interface can be a physical Layer 2 interface or a port channel (logical interface). |
| Step 4 | switchport mode trunk Example: Device(config-if)# switchport mode trunk | Configures the interface as a Layer 2 trunk. |
| Step 5 | rep preempt delay seconds Example: Device(config-if)# rep preempt delay 30 | Configures a preempt time delay. Automatically triggers VLAN load balancing after a link failure and recovery. The time delay range is between 15 to 300 seconds. The default is manual preemption with no time delay. Note Enter this command only on the REP primary edge port. |

Configuring Manual Preemption for VLAN Load Balancing

If you do not enter a preemption time delay, the default is to manually trigger VLAN load balancing on the segment. Ensure that all other segment configurations are complete before manually preempting VLAN load balancing. When you enter the **rep preempt delay segment** command, a confirmation message appears before the command is executed because preemption can cause network disruption.

Procedure

| | Command or Action | Purpose |
|---------------|---|---|
| Step 1 | enable Example: Device> enable | Enables privileged EXEC mode. Enter your password if prompted. |
| Step 2 | rep preempt segment <i>segment-id</i> Example: Device# rep preempt segment 300 | Manually triggers VLAN load balancing on the segment. The range for segment id is from 1 to 1024. |
| Step 3 | show rep topology segment <i>segment-id</i> Example: Device# show rep topology segment 300 | Displays REP topology information for the segment. |

Configuration Examples for FlexLink+

The following sections provide examples for configuring FlexLink+.

Example: Configuring the Active Port for FlexLink+

This example shows how to configure the active port for FlexLink+.

```
Device# interface Port-channel2
Device(config-if)# switchport trunk allowed vlan 20-23,40,41
Device(config-if)# switchport mode trunk
Device(config-f)# rep segment 1023 edge no-neighbor primary
```

Example: Configuring the Standby Port for FlexLink+

This example shows how to configure the standby port for FlexLink+.

```
Device# interface Port-channel7
Device(config-if)# switchport trunk allowed vlan 20-23,40,41
Device(config-if)# switchport mode trunk
Device(config-f)# rep segment 1023 edge no-neighbor preferred
```

Example: Configuring VLAN Load Balancing for FlexLink+

The following example shows VLAN load balancing configured on a FlexLink+ interface. VLANs 1-50 are blocked on the active port and VLANs 51-100 are blocked on the standby port.

```
Device(config)# interface gigabitethernet2/0/8
Device(config-if)# switchport mode trunk
Device(config-if)# rep segment 300 edge no-neighbor primary
Device(config-if)# rep block port 2 vlan 1-50
Device(config-if)# exit
Device(config)# interface gigabitethernet2/0/6
Device(config-if)# switchport mode trunk
```

Example: Configuring Propagation of FlexLink+ Topology Change Messages

```
Device(config-if)# rep segment 300 edge no-neighbor
Device(config-if)# end
```

Example: Configuring Propagation of FlexLink+ Topology Change Messages

The following example shows how to configure the propagation of FlexLink+ topology change messages to the next tier devices.

```
Device(config)# interface gigabitethernet2/0/8
Device(config-if)# switchport mode trunk
Device(config-if)# rep segment 300 edge no-neighbor primary
Device(config-if)# rep stcn stp
Device(config-if)# rep block port 2 vlan 1-50
Device(config-if)# exit
Device(config)# interface gigabitethernet2/0/6
Device(config-if)# switchport mode trunk
Device(config-if)# rep segment 300 edge no-neighbor
Device(config-if)# rep stcn stp
Device(config-if)# end
```

Feature History for FlexLink+

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 | FlexLink+ | <p>The FlexLink+ feature enables the user to configure a pair of a Layer 2 interfaces (trunk ports or port channels) where one interface is configured to act as a backup to the other interface.</p> <p>Support for this feature was introduced only on the C9500-12Q, C9500-16X, C9500-24Q, C9500-40X models of the Cisco Catalyst 9500 Series Switches.</p> |

| Release | Feature | Feature Information |
|-------------------------------|--|---|
| Cisco IOS XE Amsterdam 17.2.1 | VLAN Load Balancing for FlexLink+ Preemption for VLAN Load Balancing FlexLink+ Dummy Multicast Packets | <p>VLAN load balancing feature was introduced on FlexLink+. VLAN load-balancing allows you to configure a FlexLink+ pair so that both ports can simultaneously forward the traffic for some mutually exclusive VLANs.</p> <p>You can trigger VLAN load balancing either by manually triggering it or by configuring a preempt delay.</p> <p>When a primary link fails, FlexLink+ transmits dummy multicast packets over the new active interface. These packets help learn the source MAC address.</p> <p>Support for this feature was introduced only on the C9500-12Q, C9500-16X, C9500-24Q, C9500-40X models of the Cisco Catalyst 9500 Series Switches.</p> |

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>.

