# Cisco Nexus 3550-T NX-OS Layer 2 Switching Configuration Guide, Release 10.6(x)

**First Published:** 2025-08-13

**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

# CONTENTS

# Preface

This preface includes the following sections:

# Audience

This publication is for network administrators who install, configure, and maintain Cisco Nexus switches.

# Document Conventions

Command descriptions use the following conventions:

| Convention | Description |
|---|---|
| **bold** | Bold text indicates the commands and keywords that you enter literally as shown. |
| *Italic* | Italic text indicates arguments for which the user supplies the values. |
| [x] | Square brackets enclose an optional element (keyword or argument). |
| [x \| y] | Square brackets enclosing keywords or arguments separated by a vertical bar indicate an optional choice. |
| {x \| y} | Braces enclosing keywords or arguments separated by a vertical bar indicate a required choice. |
| [x {y \| z}] | Nested set of square brackets or braces indicate optional or required choices within optional or required elements. Braces and a vertical bar within square brackets indicate a required choice within an optional element. |
| `variable` | Indicates a variable for which you supply values, in context where italics cannot be used. |
| string | A nonquoted set of characters. Do not use quotation marks around the string or the string will include the quotation marks. |

Examples use the following conventions:

| Convention | Description |
|---|---|
| `screen font` | Terminal sessions and information the switch displays are in screen font. |
| **`boldface screen font`** | Information you must enter is in boldface screen font. |
| *italic screen font* | Arguments for which you supply values are in italic screen font. |
| < > | Nonprinting characters, such as passwords, are in angle brackets. |
| [ ] | Default responses to system prompts are in square brackets. |
| !, # | An exclamation point (!) or a pound sign (#) at the beginning of a line of code indicates a comment line. |

# Related Documentation for Cisco Nexus 3500 Series Switches

The entire Cisco Nexus 3500 Series switch documentation set is available at the following URL:

https://www.cisco.com/c/en/us/support/switches/nexus-3000-series-switches/tsd-products-support-series-home.html

# Documentation Feedback

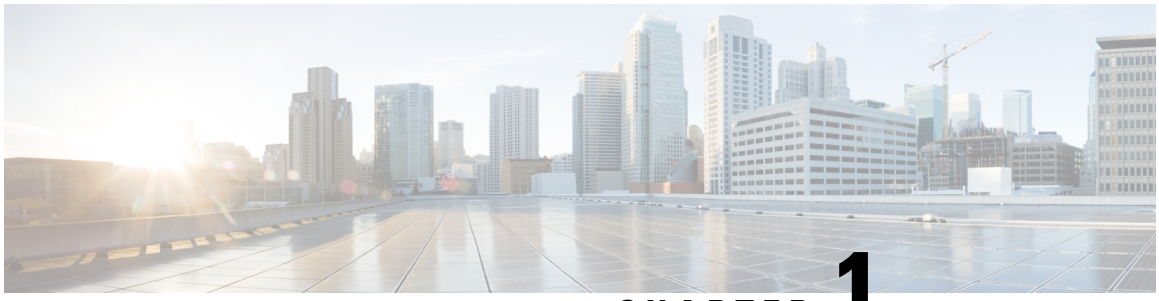
To provide technical feedback on this document, or to report an error or omission, please send your comments to nexus3k-docfeedback@cisco.com. We appreciate your feedback.

# Communications, Services, and Additional Information

- To receive timely, relevant information from Cisco, sign up at Cisco Profile Manager.
- To get the business results you're looking for with the technologies that matter, visit Cisco Services.
- To submit a service request, visit Cisco Support.
- To discover and browse secure, validated enterprise-class apps, products, solutions and services, visit Cisco DevNet.
- To obtain general networking, training, and certification titles, visit Cisco Press.
- To find warranty information for a specific product or product family, access Cisco Warranty Finder.

### Cisco Bug Search Tool

Cisco Bug Search Tool (BST) is a web-based tool that acts as a gateway to the Cisco bug tracking system that maintains a comprehensive list of defects and vulnerabilities in Cisco products and software. BST provides you with detailed defect information about your products and software.



CHAPTER 1

# New and Changed Information

This section contains the new and changed information for a release.

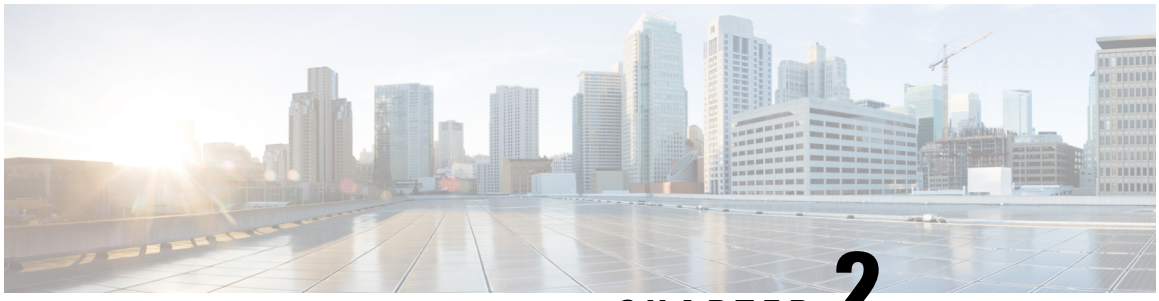


# New and Changed Information

*Table 1: New and Changed Information for Cisco Nexus 3550-T NX-OS Release 10.6(x)*

| Feature | Description | Changed in Release | Where Documented |
|---|---|---|---|
| NA | No new features added for this release. | 10.6(1)F | NA |

New and Changed Information



CHAPTER 2

# Overview

This preface includes the following sections:



# Licensing Requirements

For a complete explanation of Cisco NX-OS licensing recommendations and how to obtain and apply licenses, see the *Cisco NX-OS Licensing Guide*.

# Layer 2 Ethernet Switching Overview

### Layer 2 Ethernet Switching Overview

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

The device solves congestion problems caused by high-bandwidth devices and a large number of users by assigning each device (for example, a server) to its own collision domain. Because each LAN port connects to a separate Ethernet collision domain, servers in a switched environment achieve full access to the bandwidth.

# VLANs

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

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

All ports are assigned to the default VLAN (VLAN1) when the device first comes up. A VLAN interface, or switched virtual interface (SVI), is a Layer 3 interface that is created to provide communication between VLANs.

The devices support a maximum of 255 VLANs (the range is from 1 to 4095) in accordance with the IEEE 802.1Q standard.

**Note** Inter-Switch Link (ISL) trunking is not supported on the Cisco NX-OS.

# Spanning Tree

This section discusses the implementation of the Spanning Tree Protocol (STP) on the software. Spanning tree is used to refer to IEEE 802.1w and IEEE 802.1s. When the IEEE 802.1D Spanning Tree Protocol is referred to in the publication, 802.1D is stated specifically.

## STP Overview

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

802.1D is the original standard for STP, and many improvements have enhanced the basic loop-free STP. Additionally, the entire standard was reworked to make the loop-free convergence process faster to keep up with the faster equipment.

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

Although the software can interoperate with legacy 802.1D systems, the system runs MST. MST is the default STP protocol for Cisco Nexus devices.

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

## Rapid PVST+

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

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

## MST

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

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

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

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

You can force specified interfaces to send prestandard, rather than standard, MST messages using the command-line interface.

## STP Extensions

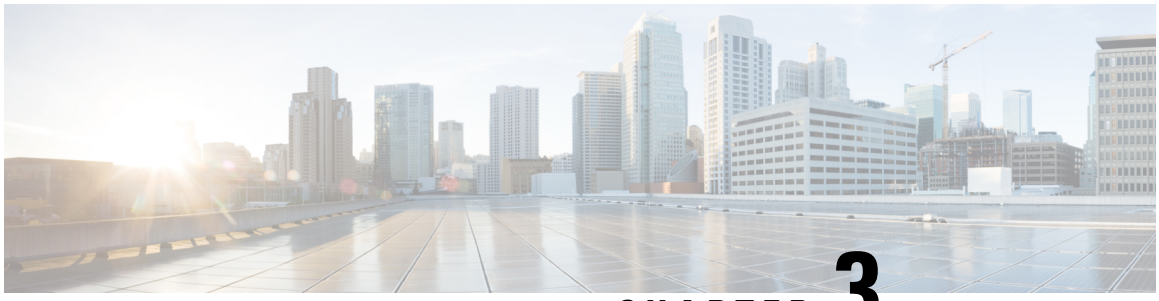
The software supports the following Cisco proprietary features:

- Spanning tree port types—The default spanning tree port type is normal. You can configure interfaces connected to Layer 2 hosts as edge ports and interfaces connected to Layer 2 switches or bridges as network ports.
- Bridge Assurance—Once you configure a port as a network port, Bridge Assurance sends BPDUs on all ports and moves a port into the blocking state if it no longer receives BPDUs. This enhancement is available only when you are running MST.
- BPDU Guard—BPDU Guard shuts down the port if that port receives a BPDU.
- BPDU Filter—BPDU Filter suppresses sending and receiving BPDUs on the port.
- Loop Guard—Loop Guard helps prevent bridging loops that could occur because of a unidirectional link failure on a point-to-point link.
- Root Guard—STP root guard prevents a port from becoming root port or blocked port. If a port configured for root guard receives a superior BPDU, the port immediately goes to the root-inconsistent (blocked) state.

# Related Topics

The following documents are related to the Layer 2 switching features:

- *Cisco Nexus® 3550-T Interfaces Configuration Guide*
- *Cisco Nexus® 3550-T Security Configuration Guide*
- *Cisco Nexus® 3550-T System Management Configuration Guide*



CHAPTER 3

# Configuring Layer 2 Switching



# Information About Layer 2 Switching

You can configure Layer 2 switching ports as access or trunk ports. Trunks carry the traffic of multiple VLANs over a single link and allow you to extend VLANs across an entire network. All Layer 2 switching ports maintain MAC address tables.

**Note** See *Cisco Nexus 3550-T Interfaces Configuration Guide*, for information on creating interfaces.

## Switching Frames Between Segments

Each LAN port on a device can connect to a single workstation, server, or to another device through which workstations or servers connect to the network.

To reduce signal degradation, the device considers each LAN port to be an individual segment. When stations connected to different LAN ports need to communicate, the device forwards frames from one LAN port to the other at wire speed to ensure that each session receives full bandwidth.

To switch frames between LAN ports efficiently, the device maintains an address table. When a frame enters the device, it associates the media access control (MAC) address of the sending network device with the LAN port on which it was received.

## Building the Address Table and Address Table Changes

The device dynamically builds the address table by using the MAC source address of the frames received. When the device receives a frame for a MAC destination address not listed in its address table, it floods the

frame to all LAN ports of the same VLAN except the port that received the frame. When the destination station replies, the device adds its relevant MAC source address and port ID to the address table. The device then forwards subsequent frames to a single LAN port without flooding all LAN ports.

You can configure MAC addresses, which are called static MAC addresses, to statically point to specified interfaces on the device. These static MAC addresses override any dynamically learned MAC addresses on those interfaces. You cannot configure broadcast addresses as static MAC addresses. The static MAC entries are retained across a reboot of the device.

The address table can store a number of MAC address entries depending on the hardware I/O module. The device uses an aging mechanism, defined by a configurable aging timer, so if an address remains inactive for a specified number of seconds, it is removed from the address table.

## Layer 3 Static MAC Addresses

You can configure a static MAC address for the following Layer 3 interfaces:

- Layer 3 interfaces
- Layer 3 port channels
- VLAN network interface

**Note** When you configure static MAC on SVI interface, ensure that the first 42 bits match the Virtual Device Context (VDC) MAC.

See the Cisco Nexus Series NX-OS Interfaces Configuration Guide, for information on configuring Layer 3 interfaces.

# Prerequisites for Configuring MAC Addresses

MAC addresses have the following prerequisites:

- You must be logged onto the device.
- If necessary, install the Advanced Services license.

# Default Settings for Layer 2 Switching

This table lists the default setting for Layer 2 switching parameters.

***Table 2: Default Layer 2 Switching Parameters***

| Parameters | Default |
|---|---|
| Aging time | 1800 seconds |

# Configuring Layer 2 Switching by Steps

**Note** If you are familiar with the Cisco IOS CLI, be aware that the Cisco NX-OS commands for this feature might differ from the Cisco IOS commands that you would use.

## Configuring a Static MAC Address

You can configure MAC addresses, which are called static MAC addresses, to statically point to specified interfaces on the device. These static MAC addresses override any dynamically learned MAC addresses on those interfaces. You cannot configure broadcast or multicast addresses as static MAC addresses.

### Procedure

| | Command or Action | Purpose |
|---|---|---|
| **Step 1** | **config t**<br>**Example:**<br>`switch# config t`<br>`switch(config)#` | Enters configuration mode. |
| **Step 2** | **mac address-table static** *mac-address* **vlan** *vlan-id* {[ **interface** {*type slot/port*} \| **port-channel** *number*]}<br>**Example:**<br>`switch(config)# mac address-table static 1.1.1 vlan 2 interface ethernet 1/2` | Specifies a static MAC address to add to the Layer 2 MAC address table. |
| **Step 3** | **exit**<br>**Example:**<br>`switch(config)# exit`<br>`switch#` | Exits the configuration mode. |
| **Step 4** | (Optional) **show mac address-table static**<br>**Example:**<br>`switch# show mac address-table static` | Displays the static MAC addresses. |
| **Step 5** | (Optional) **copy running-config startup-config**<br>**Example:**<br>`switch# copy running-config startup-config` | Copies the running configuration to the startup configuration. |

### Example

This example shows how to put a static entry in the Layer 2 MAC address table:

```
switch# config t
switch(config)# mac address-table static 1.1.1 vlan 2 interface ethernet 1/2
switch(config)#
```

# Configuring a Static MAC Address on a Layer 3 Interface

You can configure static MAC addresses on Layer 3 interfaces. You cannot configure broadcast or multicast addresses as static MAC addresses.

See Cisco Nexus 3550-T Series NX-OS Interfaces Configuration Guide, for information on configuring Layer 3 interfaces.

### Procedure

| | Command or Action | Purpose |
|---|---|---|
| **Step 1** | **config t**<br>**Example:**<br>`switch# config t`<br>`switch(config)#` | Enters configuration mode. |
| **Step 2** | **interface** [**ethernet** *slot/port* \| **ethernet** *slot/port.number* \| **port-channel** *number* \| **vlan** *vlan-id*]<br>**Example:**<br>`switch(config)# interface ethernet 1/3` | Specifies the Layer 3 interface and enters the interface configuration mode.<br>**Note**<br>You must create the Layer 3 interface before you can assign the static MAC address. |
| **Step 3** | **mac-address** *mac-address*<br>**Example:**<br>`switch(config-if)# mac-address 22ab.47dd.ff89`<br>`switch(config-if)#` | Specified a static MAC address to add to the Layer 3 interface. |
| **Step 4** | **exit**<br>**Example:**<br>`switch(config-if)# exit`<br>`switch(config)#` | Exits the interface mode. |
| **Step 5** | (Optional) **show interface** [**ethernet** *slot/port* \| **ethernet** *slot/port.number* \| **port-channel** *number* \| **vlan** *vlan-id*]<br>**Example:**<br>`switch# show interface ethernet 1/3` | Displays information about the Layer 3 interface. |

| | Command or Action | Purpose |
|---|---|---|
| **Step 6** | (Optional) **copy running-config startup-config**<br>**Example:**<br>`switch# copy running-config startup-config` | Copies the running configuration to the startup configuration. |

### Example

This example shows how to configure the Layer 3 interface on slot 1, port 3 with a static MAC address:

```
switch# config t
switch(config)# interface ethernet 1/3
switch(config-if)# mac-address 22ab.47dd.ff89
switch(config-if)#
```

This example shows how to configure the SVI MAC address:

```
switch(config-if)# show vdc
vdc_id  vdc_name    state        mac                type         lc
------  --------     -----       ----------         ---------    ------
1       triton5     active       64:3f:5f:84:37:9a   Ethernet    None

switch(config-if)# interface vlan 10
switch(config-if)# mac-address 64:3f:5f:84:37:93
```

# Configuring the Aging Time for the MAC Table

You can configure the amount of time that a MAC address entry (the packet source MAC address and port on which that packet was learned) remains in the MAC table, which contains the Layer 2 information.

**Note** MAC addresses are aged out up to two times the configured MAC address table aging timeout.

**Note** You can also configure the MAC aging time in interface configuration mode or VLAN configuration mode.

### Procedure

| | Command or Action | Purpose |
|---|---|---|
| **Step 1** | **config t**<br>**Example:**<br>`switch# config t`<br>`switch(config)#` | Enters configuration mode. |

| | Command or Action | Purpose |
|---|---|---|
| **Step 2** | **mac address-table aging-time** *seconds*<br>**Example:**<br>`switch(config)# mac address-table aging-time 600` | Specifies the time before an entry ages out and is discarded from the Layer 2 MAC address table. The range is from 120 to 918000; the default is 1800 seconds. Entering the value 0 disables the MAC aging. |
| **Step 3** | **exit**<br>**Example:**<br>`switch(config)# exit`<br>`switch#` | Exits the configuration mode. |
| **Step 4** | (Optional) **show mac address-table aging-time**<br>**Example:**<br>`switch# show mac address-table aging-time` | Displays the aging time configuration for MAC address retention. |
| **Step 5** | (Optional) **copy running-config startup-config**<br>**Example:**<br>`switch# copy running-config startup-config` | Copies the running configuration to the startup configuration. |

### Example

This example shows how to set the ageout time for entries in the Layer 2 MAC address table to 600 seconds (10 minutes):

```
switch# config t
switch(config)# mac address-table aging-time 600
switch(config)#
```

# Clearing Dynamic Addresses from the MAC Table

You can clear all dynamic Layer 2 entries in the MAC address table. (You can also clear entries by designated interface or VLAN.)

### Procedure

| | Command or Action | Purpose |
|---|---|---|
| **Step 1** | **clear mac address-table dynamic** {**address** *mac_addr*} {**interface** [**ethernet** *slot/port* \| **port-channel** *channel-number*]} {**vlan** *vlan_id*}<br>**Example:**<br>`switch# clear mac address-table dynamic` | Clears the dynamic address entries from the MAC address table in Layer 2. |

| | Command or Action | Purpose |
|---|---|---|
| **Step 2** | (Optional) **show mac address-table**<br>**Example:**<br>`switch# show mac address-table` | Displays the MAC address table. |

### Example

This example shows how to clear the dynamic entries in the Layer 2 MAC address table:

```
switch# clear mac address-table dynamic
switch#
```

# Verifying the Layer 2 Switching Configuration

To display Layer 2 switching configuration information, perform one of the following tasks:

| Command | Purpose |
|---|---|
| **show mac address-table** | Displays information about the MAC address table. |
| **show mac address-table aging-time** | Displays information about the aging time set for the MAC address entries. |
| **show mac address-table static** | Displays information about the static entries on the MAC address table. |
| **show interface** [*interface*] **mac-address** | Displays the MAC addresses and the burn-in MAC address for the interfaces. |

# Configuration Example for Layer 2 Switching

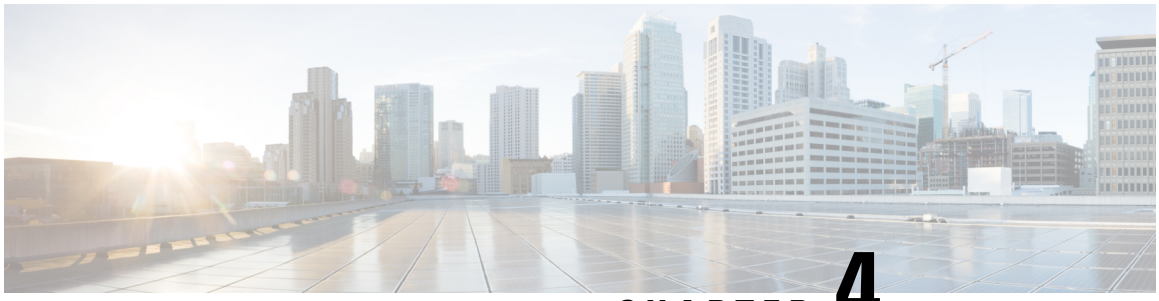
The following example shows how to add a static MAC address and how to modify the default global aging time for MAC addresses:

```
switch# configure terminal
switch(config)# mac address-table static 0000.0000.1234 vlan 10 interface ethernet 1/15
switch(config)# mac address-table aging-time 120
```

# Additional References for Layer 2 Switching -- CLI Version

### Related Documents

| Related Topic | Document Title |
|---|---|
| Static MAC addresses | *Cisco Nexus® 3550-T Security Configuration Guide* |
| Interfaces | *Cisco Nexus® 3550-T Interfaces Configuration Guide* |
| System management | *Cisco Nexus® 3550-T System Management Configuration Guide* |
| Licensing | *Cisco Nexus 3550-T NX-OS Smart Licensing Using Policy User Guide* |
| Release Notes | *Cisco Standalone Series NX-OS Release Notes* |



CHAPTER 4

# Configuring Rapid PVST+ Using Cisco NX-OS



## Information About Rapid PVST+

**Note** See the *Cisco Nexus 3550-T Series NX-OS Interfaces Configuration Guide*, for information on creating Layer 2 interfaces.

The Spanning Tree Protocol (STP) was implemented to provide a loop-free network at Layer 2 of the network. Rapid PVST+ is an updated implementation of STP that allows you to create one spanning tree topology for each VLAN. Rapid PVST+ is the default STP mode on the device.

**Note** Spanning tree is used to refer to IEEE 802.1w and IEEE 802.1s. If the IEEE 802.1D Spanning Tree Protocol is discussed in this publication, then 802.1D is stated specifically.

**Note** Rapid PVST+ is the default STP mode.

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

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

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

# STP

STP is a Layer 2 link-management protocol that provides path redundancy while preventing loops in the network.

## Overview of STP

In order for a Layer 2 Ethernet network to function properly, only one active path can exist between any two stations. STP 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.

When you create fault-tolerant internetworks, you must have a loop-free path between all nodes in a network. The STP algorithm calculates the best loop-free path throughout a switched Layer 2 network. Layer 2 LAN ports send and receive STP frames, which are called Bridge Protocol Data Units (BPDUs), at regular intervals. Network devices do not forward these frames but use the frames to construct a loop-free path.

Multiple active paths between end stations cause loops in the network. If a loop exists in the network, end stations might receive duplicate messages and network devices might learn end station MAC addresses on multiple Layer 2 LAN ports.

STP defines a tree with a root bridge and a loop-free path from the root to all network devices in the Layer 2 network. STP forces redundant data paths into a blocked state. If a network segment in the spanning tree fails and a redundant path exists, the STP algorithm recalculates the spanning tree topology and activates the blocked path.

When two Layer 2 LAN ports on a network device are part of a loop, the STP port priority and port path-cost setting determine which port on the device is put in the forwarding state and which port is put in the blocking state. The STP port priority value is the efficiency with which that location allows the port to pass traffic. The STP port path-cost value is derived from the media speed.

## How a Topology is Created

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

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

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

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

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

**Note** The **mac-address bpdu source version 2** command enables STP to use the new Cisco MAC address (00:26:0b:xx:xx:xx) as the source address of BPDUs generated on vPC ports.

To apply this command, you must have identical configurations for both vPC peer switches or peers.

Cisco strongly recommends that you disable ether channel guard on the edge devices before issuing this command to minimize traffic disruption from STP inconsistencies. Re-enable the ether channel guard after updating on both peers.

## Bridge ID

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

### Bridge Priority Value

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

You can only specify a device bridge ID (used by the spanning tree algorithm to determine the identity of the root bridge; the lowest number is preferred) as a multiple of 4096.

**Note** In this device, the extended system ID is always enabled; you cannot disable the extended system ID.

### Extended System ID

The device always uses the 12-bit extended system ID.

***Figure 1: Bridge ID with Extended System ID***

This figure shows the 12-bit extended system ID field that is part of the bridge ID.


This table shows how the system ID extension combined with the bridge ID functions as the unique identifier for a VLAN.

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

| Bridge 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 | Bit 1 |
| 32768 | 16384 | 8192 | 4096 | 2048 | 1024 | 512 | 256 | 128 | 64 | 32 | 16 | 8 | 4 | 2 | 1 |

### STP MAC Address Allocation

**Note** MAC address reduction is always enabled on the device.

Because MAC address reduction is always enabled on the device, you should also enable MAC address reduction on all other Layer 2 connected network devices to avoid undesirable root bridge election and spanning tree topology issues.

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

- 0
- 4096
- 8192
- 12288
- 16384
- 20480
- 24576
- 28672
- 32768
- 36864
- 40960
- 45056
- 49152
- 53248
- 57344
- 61440

STP uses the extended system ID plus a MAC address to make the bridge ID unique for each VLAN.

**Note** If another bridge in the same spanning tree domain does not run the MAC address reduction feature, it could win the root bridge ownership because of the finer granularity in the selection of its bridge ID.

## BPDUs

Network devices transmit BPDUs throughout the STP instance. Each network device sends configuration BPDUs to communicate and compute the spanning tree topology. Each configuration BPDU contains the following minimal information:

- The unique bridge ID of the network device that the transmitting network device believes to be the root bridge
- The STP path cost to the root
- The bridge ID of the transmitting bridge
- The message age
- The identifier of the transmitting port
- Values for the hello, forward delay, and max-age protocol timer
- Additional information for STP extension protocols

When a network device transmits a Rapid PVST+ BPDU frame, all network devices connected to the VLAN on which the frame is transmitted receive the BPDU. When a network device receives a BPDU, it does not forward the frame but instead uses the information in the frame to calculate a BPDU. If the topology changes, the device initiates a BPDU exchange.

A BPDU exchange results in the following:

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

## Election of the Root Bridge

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

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

The STP root bridge is the logical center of each spanning tree topology in a Layer 2 network. All paths that are not needed to reach the root bridge from anywhere in the Layer 2 network are placed in STP blocking mode.

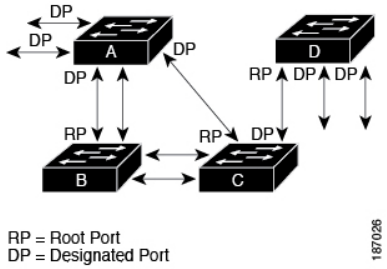
BPDUs contain information about the transmitting bridge and its ports, including bridge and MAC addresses, bridge priority, port priority, and path cost. STP uses this information to elect the root bridge for the STP instance, to elect the root port that leads to the root bridge, and to determine the designated port for each Layer 2 segment.

## Creating the Spanning Tree Topology

By lowering the numerical value of the ideal network device so that it becomes the root bridge, you force an STP recalculation to form a new spanning tree topology with the ideal network device as the root.

*Figure 2: Spanning Tree Topology*

In this figure, switch A is elected as the root bridge because the bridge priority of all the network devices is set to the default (32768) and switch A has the lowest MAC address. However, due to traffic patterns, the number of forwarding ports, or link types, switch A might not be the ideal root bridge.




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

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

# Rapid PVST+

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

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

## Overview of Rapid PVST+

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

**Note** Rapid PVST+ is the default STP mode for the device.

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

**Note** Rapid PVST+ supports one STP instance for each VLAN.

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

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

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

**Note** We recommend that you configure all ports connected to a Layer 2 host as edge ports.

- Root port—If Rapid PVST+ selects a new root port, it blocks the old root port and immediately transitions the new root port to the forwarding state.
- Point-to-point links—If you connect a port to another port through a point-to-point link and the local port becomes a designated port, it negotiates a rapid transition with the other port by using the proposal-agreement handshake to ensure a loop-free topology.

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

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

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

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

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

**Note** The TCA flag is used only when the device is interacting with devices that are running legacy 802.1D STP.

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

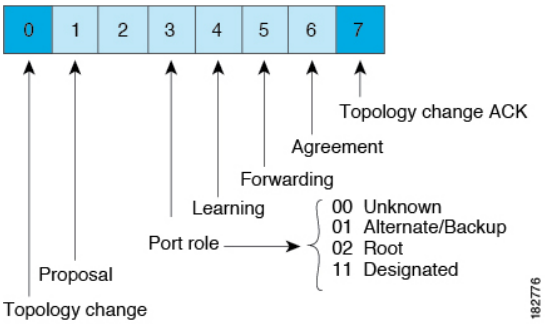
## Rapid PVST+ BPDUs

Rapid PVST+ and 802.1w use all six bits of the flag byte to add the following:

- The role and state of the port that originates the BPDU
- The proposal and agreement handshake

***Figure 3: Rapid PVST+ Flag Byte in BPDU***

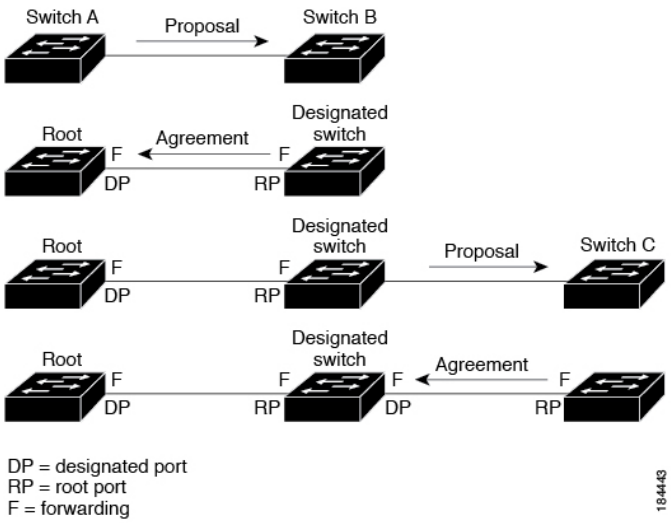
This figure shows the use of the BPDU flags in Rapid PVST+.




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

## Proposal and Agreement Handshake

***Figure 4: Proposal and Agreement Handshaking for Rapid Convergence***




In this figure, switch A is connected to switch B through a point-to-point link, and all of the ports are in the blocking state. Assume that the priority of switch A is a smaller numerical value than the priority of switch B. Switch A sends a proposal message (a configuration BPDU with the proposal flag set) to switch B, proposing itself as the designated switch.

After receiving the proposal message, switch B selects as its new root port the port from which the proposal message was received, forces all 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 the agreement message from switch B, switch A also immediately transitions its designated port to the forwarding state. No loops in the network can form because switch B blocked all of its nonedge ports and because there is a point-to-point link between switches A and B.

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

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

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

## Protocol Timers

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

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

| Variable | Description |
|---|---|
| Hello timer | Determines how often each device broadcasts BPDUs to other network devices. The default is 2 seconds, and the range is from 1 to 10. |
| Forward delay timer | Determines how long each of the listening and learning states last before the port begins forwarding. This timer is generally not used by the protocol, but it is used when interoperating with the 802.1D spanning tree. The default is 15 seconds, and the range is from 4 to 30 seconds. |
| Maximum age timer | Determines the amount of time that protocol information received on a port is stored by the network device. This timer is generally not used by the protocol, but it is used when interoperating with the 802.1D spanning tree. The default is 20 seconds; the range is from 6 to 40 seconds. |

## Port Roles

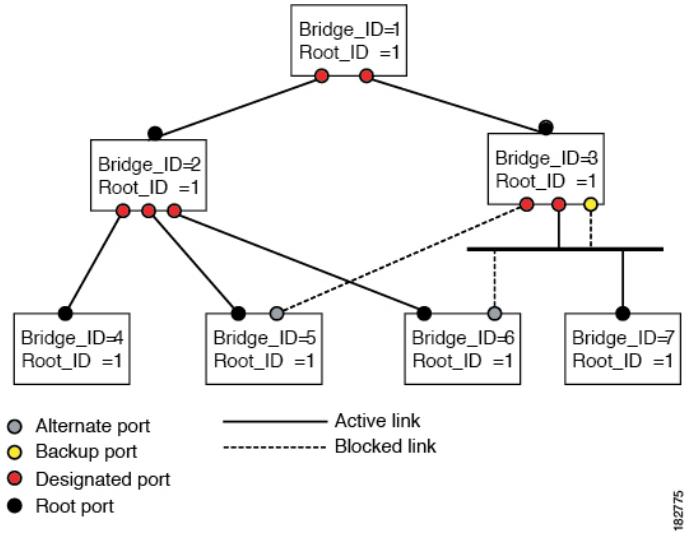
Rapid PVST+ provides rapid convergence of the spanning tree by assigning port roles and learning the active topology. Rapid PVST+ builds upon the 802.1D STP to select the device with the highest switch priority (lowest numerical priority value) as the root bridge. Rapid PVST+ assigns one of these port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

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

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

***Figure 5: Sample Topology Demonstrating Port Roles***




This figure shows port roles. 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.

## Rapid PVST+ Port State Overview

Propagation delays can occur when protocol information passes through a switched LAN. As a result, topology changes can take place at different times and at different places in a switched network. When a Layer 2 LAN port transitions directly from nonparticipation in the spanning tree topology to the forwarding state, it can create temporary data loops. Ports must wait for new topology information to propagate through the switched LAN before starting to forward frames.

Each Layer 2 LAN port on the device that uses Rapid PVST+ or MST exists in one of the following four states:

- Blocking—The Layer 2 LAN port does not participate in frame forwarding.
- Learning—The Layer 2 LAN port prepares to participate in frame forwarding.
- Forwarding—The Layer 2 LAN port forwards frames.
- Disabled—The Layer 2 LAN port does not participate in STP and is not forwarding frames.

When you enable Rapid PVST+, every port in the device, VLAN, and network goes through the blocking state and the transitory states of learning at power up. If properly configured, each Layer 2 LAN port stabilizes to the forwarding or blocking state.

When the STP algorithm places a Layer 2 LAN port in the forwarding state, the following process occurs:

1. The Layer 2 LAN port is put into the blocking state while it waits for protocol information that suggests it should go to the learning state.
2. The Layer 2 LAN port waits for the forward delay timer to expire, moves the Layer 2 LAN port to the learning state, and restarts the forward delay timer.
3. In the learning state, the Layer 2 LAN port continues to block frame forwarding as it learns the end station location information for the forwarding database.
4. The Layer 2 LAN port waits for the forward delay timer to expire and then moves the Layer 2 LAN port to the forwarding state, where both learning and frame forwarding are enabled.

### Blocking State

A Layer 2 LAN port in the blocking state does not participate in frame forwarding.

A Layer 2 LAN port in the blocking state performs as follows:

- Discards frames received from the attached segment.
- Discards frames switched from another port for forwarding.
- Does not incorporate the end station location into its address database. (There is no learning on a blocking Layer 2 LAN port, so there is no address database update.)
- Receives BPDUs and directs them to the system module.
- Receives, processes, and transmits BPDUs received from the system module.
- Receives and responds to control plane messages.

### Learning State

A Layer 2 LAN port in the learning state prepares to participate in frame forwarding by learning the MAC addresses for the frames. The Layer 2 LAN port enters the learning state from the blocking state.

A Layer 2 LAN port in the learning state performs as follows:

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

### Forwarding State

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

A Layer 2 LAN port in the forwarding state performs as follows:

- Forwards frames received from the attached segment.
- Forwards frames switched from another port for forwarding.
- Incorporates the end station location information into its address database.
- Receives BPDUs and directs them to the system module.
- Processes BPDUs received from the system module.
- Receives and responds to control plane messages.

### Disabled State

A Layer 2 LAN port in the disabled state does not participate in frame forwarding or STP. A Layer 2 LAN port in the disabled state is virtually nonoperational.

A disabled Layer 2 LAN port performs as follows:

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

### Summary of Port States

This table lists the possible operational and Rapid PVST+ states for ports and whether the port is included in the active topology.

***Table 5: Port State Active Topology***

| Operational Status | Port State | Is Port Included in the Active Topology? |
|---|---|---|
| Enabled | Blocking | No |
| Enabled | Learning | Yes |
| Enabled | Forwarding | Yes |
| Disabled | Disabled | No |

## 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, Rapid PVST+ forces all other ports to synchronize with the new root information.

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

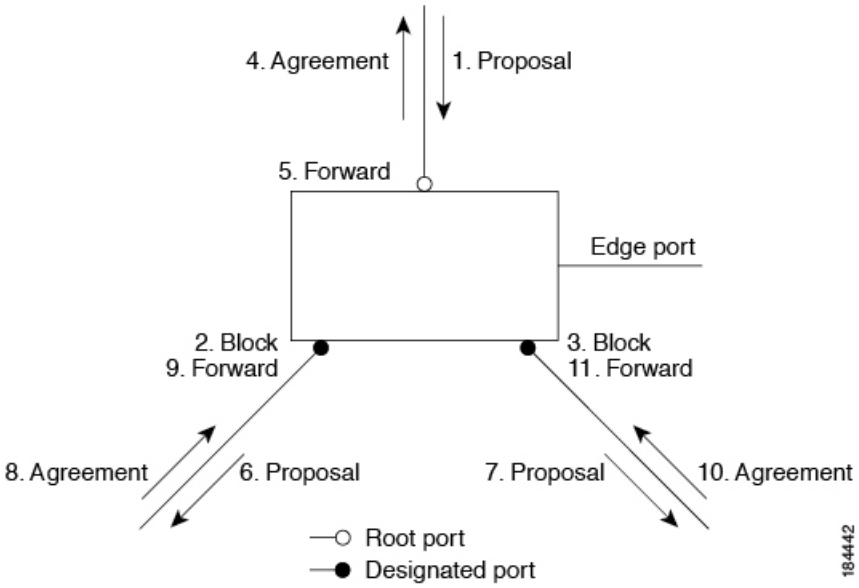
- That port is in the blocking state.
- It is an edge port (a port configured to be at the edge of the network).

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

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

***Figure 6: Sequence of Events During Rapid Convergence***

This figure shows the sequence of events during synchronization.




### Processing Superior BPDU Information

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

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

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

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

### Processing Inferior BPDU Information

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

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

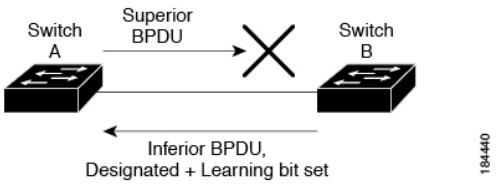
## Detecting Unidirectional Link Failure:Rapid PVST+

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 using the Unidirectional Link Detection (UDLD) feature. This feature is based on the dispute mechanism.

See the *Cisco Nexus 3550-T Series NX-OS Interfaces Configuration Guide*, for information on UDLD.

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

***Figure 7: Detecting Unidirectional Link Failure***




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

## Port Cost

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

This table shows how the STP port path-cost default value is determined from the media speed and path-cost calculation method of a LAN interface.

***Table 6: Default Port Cost***

| Bandwidth | Short Path-Cost Method of Port Cost | Long Path-Cost Method of Port Cost |
|---|---|---|
| 10 Mbps | 100 | 2,000,000 |

| Bandwidth | Short Path-Cost Method of Port Cost | Long Path-Cost Method of Port Cost |
|---|---|---|
| 100 Mbps | 19 | 200,000 |
| 1 Gbps | 4 | 20,000 |
| 10 Gbps | 2 | 2,000 |
| 40 Gbps | 1 | 500 |
| 100 Gbps | 1 | 200 |
| 400 Gbps | 1 | 50 |

If a loop occurs, STP considers the port cost when selecting a LAN interface to put into the forwarding state.

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

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

## Port Priority

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

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

# Rapid PVST+ and IEEE 802.1Q Trunks

The 802.1Q trunks impose some limitations on the STP strategy for a network. In a network of Cisco network devices connected through 802.1Q trunks, the network devices maintain one instance of STP for each VLAN allowed on the trunks. However, non-Cisco 802.1Q network devices maintain only one instance of STP for all VLANs allowed on the trunks, which is the Common Spanning Tree (CST).

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

For more information on 802.1Q trunks, see the *Cisco Nexus 3550-T Series NX-OS Interfaces Configuration Guide*.

# Rapid PVST+ Interoperation with Legacy 802.1D STP

Rapid PVST+ can interoperate with devices that are running the legacy 802.1D protocol. The device knows that it is interoperating with equipment running 802.1D when it receives a BPDU version 0. The BPDUs for Rapid PVST+ are version 2. If the BPDU received is an 802.1w BPDU version 2 with the proposal flag set, the device sends an agreement message after all of the other ports are synchronized. If the BPDU is an 802.1D BPDU version 0, 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.

The device interoperates with legacy 802.1D devices as follows:

- Notification—Unlike 802.1D BPDUs, 802.1w does not use TCN BPDUs. However, for interoperability with 802.1D devices, the device processes and generates TCN BPDUs.
- Acknowledgment—When an 802.1w device receives a TCN message on a designated port from an 802.1D device, it replies with an 802.1D configuration BPDU with the TCA bit set. However, if the TC-while timer (the same as the TC timer in 802.1D) is active on a root port connected to an 802.1D device and a configuration BPDU with the TCA set is received, the TC-while timer is reset.

  This method of operation is required only for 802.1D devices. The 802.1w BPDUs do not have the TCA bit set.
- Protocol migration—For backward compatibility with 802.1D devices, 802.1w selectively sends 802.1D configuration BPDUs and TCN BPDUs on a per-port basis.

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

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

**Note** If you want all devices on the same LAN segment to reinitialize the protocol on each interface, you must reinitialize Rapid PVST+.

# Rapid PVST+ Interoperation with 802.1s MST

Rapid PVST+ interoperates seamlessly with the IEEE 802.1s Multiple Spanning Tree (MST) standard. No user configuration is needed. To disable this seamless interoperation, you can use PVST Simulation.

# High Availability for Rapid PVST+

The software supports high availability for Rapid PVST+. However, the statistics and timers are not restored when Rapid PVST+ restarts. The timers start again and the statistics begin from 0.

# Prerequisites for Configuring Rapid PVST+

Rapid PVST+ has the following prerequisites:

- You must be logged onto the device.

# Guidelines and Limitations for Configuring Rapid PVST+

Rapid PVST+ has the following configuration guidelines and limitations:

- **show** commands with the **internal** keyword are not supported.
- For VLAN configuration limits please see the *Cisco Nexus 3550-T Series NX-OS Verified Scalability Guide*.
- Port channeling—The port-channel bundle is considered as a single port. The port cost is the aggregation of all the configured port costs assigned to that channel.
- We recommend that you configure all ports connected to Layer 2 hosts as STP edge ports.
- Always leave STP enabled.
- Do not change timers because changing timers can adversely affect stability.
- Keep user traffic off the management VLAN; keep the management VLAN separate from the user data.
- Choose the distribution and core layers as the location of the primary and secondary root switches.
- When you connect two Cisco devices through 802.1Q trunks, the switches exchange spanning tree BPDUs on each VLAN allowed on the trunks. The BPDUs on the native VLAN of the trunk are sent untagged to the reserved 802.1D spanning tree multicast MAC address (01-80-C2-00-00-00). The BPDUs on all VLANs on the trunk are sent tagged to the reserved Cisco Shared Spanning Tree Protocol (SSTP) multicast MAC address (01-00-0c-cc-cc-cd).

# Default Settings for Rapid PVST+

This table lists the default settings for Rapid PVST+ parameters.

***Table 7: Default Rapid PVST+ Parameters***

| Parameters | Default |
|---|---|
| Spanning Tree | Enabled on all VLANs. |
| Spanning Tree mode | Rapid PVST+<br>**Caution**<br>Changing the spanning tree mode disrupts the traffic because all spanning tree instances are stopped for the previous mode and started for the new mode. |
| VLAN | All ports assigned to VLAN1. |

| Parameters | Default |
|---|---|
| Extended system ID | Always enabled. |
| MAC address reduction | Always enabled. |
| Bridge ID priority | 32769 (default bridge priority plus system ID extension of default VLAN1). |
| Port state | Blocking (changes immediately after convergence). |
| Port role | Designated (changes after convergence). |
| Port/VLAN priority | 128. |
| Path-cost calculation method | Short. |
| Port/VLAN cost | Auto<br>The default port cost is determined by the media speed and path-cost method calculation, as follows:<br>• 1 Gigabit Ethernet:<br>• short: 4<br>• long: 20,000<br>• 10 Gigabit Ethernet:<br>• short: 2<br>• long: 2,000<br>• 40 Gigabit Ethernet:<br>• short: 1<br>• long: 500 |
| Hello time | 2 seconds. |
| Forward delay time | 15 seconds. |
| Maximum aging time | 20 seconds. |
| Link type | Auto<br>The default link type is determined by the duplex, as follows:<br>• Full duplex: point-to-point link<br>• Half duplex: shared link |

# Configuring Rapid PVST+

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

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

## Enabling Rapid PVST+ - CLI Version

If you disable Rapid PVST+ on any VLANs, you must reenable Rapid PVRST+ on the specified VLANs. If you have enabled MST on the device and now want to use Rapid PVST+, you must enable Rapid PVST+ on the device.

Rapid PVST+ is the default STP mode. You cannot simultaneously run MST and Rapid PVST+ in the same chassis.

**Note** When you change the spanning tree mode, traffic is disrupted because all spanning tree instances are stopped for the previous mode and started for the new mode.

### Procedure

| | Command or Action | Purpose |
|---|---|---|
| **Step 1** | **config t**<br>**Example:**<br>`switch# config t`<br>`switch(config)#` | Enters configuration mode. |
| **Step 2** | **spanning-tree mode rapid-pvst**<br>**Example:**<br>`switch(config)# spanning-tree mode rapid-pvst` | Enables Rapid PVST+ on the device. Rapid PVST+ is the default spanning tree mode.<br>**Note**<br>Changing the spanning tree mode disrupts traffic because all spanning tree instances are stopped for the previous mode and started for the new mode. |
| **Step 3** | **exit**<br>**Example:**<br>`switch(config)# exit`<br>`switch#` | Exits configuration mode. |
| **Step 4** | (Optional) **show running-config spanning-tree all**<br>**Example:** | Displays information about the currently running STP configuration. |

| | Command or Action | Purpose |
|---|---|---|
| | `switch# show running-config spanning-tree all` | |
| **Step 5** | (Optional) **copy running-config startup-config**<br>**Example:**<br>`switch# copy running-config startup-config` | Copies the running configuration to the startup configuration. |

### Example

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

```
switch# config t
switch(config)# spanning-tree mode rapid-pvst
switch(config)# exit
switch#
```

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

# Disabling or Enabling Rapid PVST+ Per VLAN - CLI Version

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

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

### Procedure

| | Command or Action | Purpose |
|---|---|---|
| **Step 1** | **config t**<br>**Example:**<br>`switch# config t`<br>`switch(config)#` | Enters configuration mode. |
| **Step 2** | **spanning-tree vlan** *vlan-range* or **no spanning-tree vlan** *vlan-range*<br>**Example:**<br>`switch(config)# spanning-tree vlan 5` | • **spanning-tree vlan** *vlan-range*<br>Enables Rapid PVST+ (default STP) on a per VLAN basis. The *vlan-range* value can be 2 through 3967 except for reserved VLAN values.<br>• **no spanning-tree vlan** *vlan-range* |

| | Command or Action | Purpose |
|---|---|---|
| | | Disables Rapid PVST+ on the specified VLAN. See the Caution for information regarding this command. |
| **Step 3** | **exit**<br>**Example:**<br>`switch(config)# exit`<br>`switch#` | Exits configuration mode. |
| **Step 4** | (Optional) **show spanning-tree**<br>**Example:**<br>`switch# show spanning-tree` | Displays the STP configuration. |
| **Step 5** | (Optional) **copy running-config startup-config**<br>**Example:**<br>`switch# copy running-config startup-config` | Copies the running configuration to the startup configuration. |

### Example

This example shows how to enable STP on VLAN 5:

```
switch# config t
switch(config)# spanning-tree vlan 5
switch(config)# exit
switch#
```

**Note** Do not disable spanning tree on a VLAN unless all switches and bridges in the VLAN have spanning tree disabled. You cannot disable spanning tree on some switches and bridges in a VLAN and leave it enabled on other switches and bridges in the VLAN. This action can have unexpected results because switches and bridges with spanning tree enabled will have incomplete information regarding the physical topology of the network.

**Caution** We do not recommend disabling spanning tree even in a topology that is free of physical loops. Spanning tree serves as a safeguard against misconfigurations and cabling errors. Do not disable spanning tree in a VLAN without ensuring that no physical loops are present in the VLAN.

**Note** Because STP is enabled by default, entering the **show running** command to view the resulting configuration does not display the command that you entered to enable STP.

# Configuring the Root Bridge ID

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

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

When you enter the **spanning-tree vlan** *vlan-range* **root primary** command, the device sets the bridge priority to 24576 if this value will cause the device to become the root for the specified VLANs. If any root bridge for the specified VLAN has a bridge priority lower than 24576, the device sets the bridge priority for the specified VLANs to 4096 less than the lowest bridge priority.

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

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

### Procedure

| | Command or Action | Purpose |
|---|---|---|
| **Step 1** | **config t**<br>**Example:**<br>`switch# config t`<br>`switch(config)#` | Enters configuration mode. |
| **Step 2** | **spanning-tree vlan** *vlan-range* **root primary**<br>**Example:**<br>`switch(config)# spanning-tree vlan 2 root primary` | Sets the bridge priority for the spanning tree. |
| **Step 3** | **exit**<br>**Example:**<br>`switch(config)# exit`<br>`switch#` | Exits configuration mode. |
| **Step 4** | (Optional) **show spanning-tree**<br>**Example:**<br>`switch# show spanning-tree` | Displays the STP configuration. |
| **Step 5** | (Optional) **copy running-config startup-config**<br>**Example:** | Copies the running configuration to the startup configuration. |

| | Command or Action | Purpose |
|---|---|---|
| | `switch# copy running-config startup-config` | |

### Example

This example shows how to configure the device as the root bridge:

```
switch# config t
switch(config)# spanning-tree vlan 2 root primary
switch(config)# exit
switch#
```

# Configuring a Secondary Root Bridge-CLI Version

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

Enter the **diameter** keyword to specify the Layer 2 network diameter (that is, the maximum number of bridge hops between any two end stations in the Layer 2 network). When you specify the network diameter, the software automatically selects an optimal hello time, forward delay time, and maximum age time for a network of that diameter, which can significantly reduce the STP convergence time. You can enter the **hello-time** keyword to override the automatically calculated hello time.

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

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

### Procedure

| | Command or Action | Purpose |
|---|---|---|
| **Step 1** | **config t**<br>**Example:**<br>`switch# config t`<br>`switch(config)#` | Enters configuration mode. |
| **Step 2** | **spanning-tree vlan** *vlan-range* **root secondary** [**diameter** *dia* [**hello-time** *hello-time*]]<br>**Example:**<br>`switch(config)# spanning-tree vlan 5 root secondary diameter 4` | Configures a device as the secondary root bridge. The *vlan-range* value can be 2 through 3967 (except for reserved VLAN values). The *dia* default is 7. The *hello-time* can be from 1 |

| | Command or Action | Purpose |
|---|---|---|
| | | to 10 seconds, and the default value is 2 seconds. |
| **Step 3** | **exit**<br>**Example:**<br>`switch(config)# exit`<br>`switch#` | Exits configuration mode. |
| **Step 4** | (Optional) **show spanning-tree vlan** *vlan_id*<br>**Example:**<br>`switch# show spanning-tree vlan 5` | Displays the STP configuration for the specified VLANs. |
| **Step 5** | (Optional) **copy running-config startup-config**<br>**Example:**<br>`switch# copy running-config startup-config` | Copies the running configuration to the startup configuration. |

### Example

This example shows how to configure the device as the secondary root bridge for VLAN 5 with a network diameter of 4:

```
switch# config t
switch(config)# spanning-tree vlan 5 root secondary diameter 4
switch(config)# exit
switch#
```

# Configuring the Rapid PVST+ Bridge Priority of a VLAN

You can configure the Rapid PVST+ bridge priority of a VLAN. This is another method of configuring root bridges.

**Note** Be careful when using this configuration. We recommend that you configure the primary root and secondary root to modify the bridge priority.

### Procedure

| | Command or Action | Purpose |
|---|---|---|
| **Step 1** | **config t**<br>**Example:**<br>`switch# config t`<br>`switch(config)#` | Enters configuration mode. |

| | Command or Action | Purpose |
|---|---|---|
| **Step 2** | **spanning-tree vlan** *vlan-range* **priority** *value*<br>**Example:**<br>`switch(config)# spanning-tree vlan 5 priority 8192` | Configures the bridge priority of a VLAN. Valid values are 0, 4096, 8192, 12288, 16384, 20480, 24576, 28672, 32768, 36864, 40960, 45056, 49152, 53248, 57344, and 61440. All other values are rejected. The default value is 32768. |
| **Step 3** | **exit**<br>**Example:**<br>`switch(config)# exit`<br>`switch#` | Exits configuration mode. |
| **Step 4** | (Optional) **show spanning-tree vlan** *vlan_id*<br>**Example:**<br>`switch# show spanning-tree vlan 5` | Displays the STP configuration for the specified VLANs. |
| **Step 5** | (Optional) **copy running-config startup-config**<br>**Example:**<br>`switch# copy running-config startup-config` | Copies the running configuration to the startup configuration. |

### Example

This example shows how to configure the priority of VLAN 5 on Gigabit Ethernet port 1/4 to 8192:

```
switch# config t
switch(config)# spanning-tree vlan 5 priority 8192
switch(config)# exit
switch#
```

# Configuring the Rapid PVST+ Port Priority - CLI Version

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

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

### Procedure

| | Command or Action | Purpose |
|---|---|---|
| **Step 1** | **config t**<br>**Example:** | Enters configuration mode. |

| | Command or Action | Purpose |
|---|---|---|
| | `switch# config t`<br>`switch(config)#` | |
| **Step 2** | **interface** *type slot/port*<br>**Example:**<br>`switch(config)# interface ethernet 1/4`<br>`switch(config-if)#` | Specifies the interface to configure and enters the interface configuration mode. |
| **Step 3** | **spanning-tree** [**vlan** *vlan-list*] **port-priority** *priority*<br>**Example:**<br>`switch(config-if)# spanning-tree port-priority 160` | Configures the port priority for the LAN interface. The *priority* value can be from 0 to 224. A lower value indicates a higher priority. The priority values are 0, 32, 64, 96, 128, 160, 192, and 224. All other values are rejected. The default value is 128. |
| **Step 4** | **exit**<br>**Example:**<br>`switch(config-if)# exit`<br>`switch(config)#` | Exits interface mode. |
| **Step 5** | (Optional) **show spanning-tree interface** {**ethernet** *slot/port* \| *port channel channel-number*}<br>**Example:**<br>`switch# show spanning-tree interface ethernet 1/10` | Displays the STP configuration for the specified interface. |
| **Step 6** | (Optional) **copy running-config startup-config**<br>**Example:**<br>`switch(config)# copy running-config startup-config` | Copies the running configuration to the startup configuration. |

### Example

This example shows how to configure the port priority of Ethernet access port 1/4 to 160:

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

# Configuring the Rapid PVST+ Path-Cost Method and Port Cost - CLI Version

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

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

### Procedure

| | Command or Action | Purpose |
|---|---|---|
| **Step 1** | **config t**<br>**Example:**<br>`switch# config t`<br>`switch(config)#` | Enters configuration mode. |
| **Step 2** | **spanning-tree pathcost method** {**long** \| **short**}<br>**Example:**<br>`switch(config)# spanning-tree pathcost method long` | Selects the method used for Rapid PVST+ path-cost calculations. The default method is the short method. |
| **Step 3** | **interface** *type slot/port*<br>**Example:**<br>`switch(config)# interface ethernet 1/4`<br>`switch(config-if)` | Specifies the interface to configure and enters the interface configuration mode. |
| **Step 4** | **spanning-tree** [**vlan** *vlan-id*] **cost** [*value* \| *auto*]<br>**Example:**<br>`switch(config-if)# spanning-tree cost 1000` | Configures the port cost for the LAN interface. The cost value, depending on the path-cost calculation method, can be as follows:<br>• short—1 to 65535<br>• long—1 to 200000000<br>**Note**<br>You configure this parameter per port on access ports and per VLAN on trunk ports.<br>The default is **auto**, which sets the port cost on both the path-cost calculation method and the media speed. |
| **Step 5** | **exit**<br>**Example:**<br>`switch(config-if)# exit`<br>`switch(config)#` | Exits interface mode. |
| **Step 6** | (Optional) **show spanning-tree pathcost method**<br>**Example:**<br>`switch# show spanning-tree pathcost method` | Displays the STP path-cost method. |

| | Command or Action | Purpose |
|---|---|---|
| **Step 7** | (Optional) **copy running-config startup-config**<br>**Example:**<br>`switch(config)# copy running-config startup-config` | Copies the running configuration to the startup configuration. |

### Example

This example shows how to configure the port cost of Ethernet access port 1/4 to 1000:

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

# Configuring the Rapid PVST+ Hello Time for a VLAN - CLI Version

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

**Note** Be careful when using this configuration because you may disrupt the Spanning Tree. For most situations, we recommend that you configure the primary root and secondary root to modify the hello time.

### Procedure

| | Command or Action | Purpose |
|---|---|---|
| **Step 1** | **config t**<br>**Example:**<br>`switch# config t`<br>`switch(config)#` | Enters configuration mode. |
| **Step 2** | **spanning-tree vlan** *vlan-range* **hello-time** *value*<br>**Example:**<br>`switch(config)# spanning-tree vlan 5 hello-time 7` | Configures the hello time of a VLAN. The hello time value can be from 1 to 10 seconds, and the default is 2 seconds. |
| **Step 3** | **exit**<br>**Example:**<br>`switch(config)# exit`<br>`switch#` | Exits configuration mode. |

| | Command or Action | Purpose |
|---|---|---|
| **Step 4** | (Optional) **show spanning-tree vlan** *vlan_id*<br>**Example:**<br>`switch# show spanning-tree vlan 5` | Displays the STP configuration per VLAN. |
| **Step 5** | (Optional) **copy running-config startup-config**<br>**Example:**<br>`switch# copy running-config startup-config` | Copies the running configuration to the startup configuration. |

### Example

This example shows how to configure the hello time for VLAN 5 to 7 seconds:

```
switch# config t
switch(config)# spanning-tree vlan 5 hello-time 7
switch(config)# exit
switch#
```

# Configuring the Rapid PVST+ Forward Delay Time for a VLAN - CLI Version

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

### Procedure

| | Command or Action | Purpose |
|---|---|---|
| **Step 1** | **config t**<br>**Example:**<br>`switch# config t`<br>`switch(config)#` | Enters configuration mode. |
| **Step 2** | **spanning-tree vlan** *vlan-range* **forward-time** *value*<br>**Example:**<br>`switch(config)# spanning-tree vlan 5 forward-time 21` | Configures the forward delay time of a VLAN. The forward delay time value can be from 4 to 30 seconds, and the default is 15 seconds. |
| **Step 3** | **exit**<br>**Example:**<br>`switch(config)# exit`<br>`switch#` | Exits configuration mode. |
| **Step 4** | (Optional) **show spanning-tree vlan** *vlan_id*<br>**Example:**<br>`switch# show spanning-tree vlan 5` | Displays the STP configuration per VLAN. |

| | Command or Action | Purpose |
|---|---|---|
| **Step 5** | (Optional) **copy running-config startup-config**<br>**Example:**<br>`switch# copy running-config startup-config` | Copies the running configuration to the startup configuration. |

### Example

This example shows how to configure the forward delay time for VLAN 5 to 21 seconds:

```
switch# config t
switch(config)# spanning-tree vlan 5 forward-time 21
switch(config)# exit
switch#
```

# Configuring the Rapid PVST+ Maximum Age Time for a VLAN - CLI Version

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

### Procedure

| | Command or Action | Purpose |
|---|---|---|
| **Step 1** | **config t**<br>**Example:**<br>`switch# config t`<br>`switch(config)#` | Enters configuration mode. |
| **Step 2** | **spanning-tree vlan** *vlan-range* **max-age** *value*<br>**Example:**<br>`switch(config)# spanning-tree vlan 5 max-age 36` | Configures the maximum aging time of a VLAN. The maximum aging time value can be from 6 to 40 seconds, and the default is 20 seconds. |
| **Step 3** | **exit**<br>**Example:**<br>`switch(config)# exit`<br>`switch#` | Exits configuration mode. |
| **Step 4** | (Optional) **show spanning-tree vlan** *vlan_id*<br>**Example:**<br>`switch# show spanning-tree vlan 5` | Displays the STP configuration per VLAN. |
| **Step 5** | (Optional) **copy running-config startup-config**<br>**Example:** | Copies the running configuration to the startup configuration. |

| | Command or Action | Purpose |
|---|---|---|
| | `switch# copy running-config startup-config` | |

### Example

This example shows how to configure the maximum aging time for VLAN 5 to 36 seconds:

```
switch# config t
switch(config)# spanning-tree vlan 5 max-age 36
switch(config)# exit
switch#
```

# Specifying the Link Type for Rapid PVST+ - CLI Version

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

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

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

### Procedure

| | Command or Action | Purpose |
|---|---|---|
| **Step 1** | **config t**<br>**Example:**<br>`switch# config t`<br>`switch(config)#` | Enters configuration mode. |
| **Step 2** | **interface** *type slot/port*<br>**Example:**<br>`switch(config)# interface ethernet 1/4`<br>`switch(config-if)#` | Specifies the interface to configure and enters the interface configuration mode. |
| **Step 3** | **spanning-tree link-type** {*auto* \| *point-to-point* \| *shared*}<br>**Example:**<br>`switch(config-if)# spanning-tree link-type point-to-point` | Configures the link type to be either a point-to-point link or shared link. The system reads the default value from the device connection, as follows: half duplex links are shared and full-duplex links are point to point. If the link type is shared, the STP falls back to 802.1D. The default is auto, which sets the link type based on the duplex setting of the interface. |
| **Step 4** | **exit**<br>**Example:** | Exits interface mode. |

| | Command or Action | Purpose |
|---|---|---|
| | `switch(config-if)# exit`<br>`switch(config)#` | |
| **Step 5** | (Optional) **show spanning-tree**<br>**Example:**<br>`switch# show spanning-tree` | Displays the STP configuration. |
| **Step 6** | (Optional) **copy running-config startup-config**<br>**Example:**<br>`switch(config)# copy running-config startup-config` | Copies the running configuration to the startup configuration. |

### Example

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

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

# Reinitializing the Protocol for Rapid PVST+

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

### Procedure

| | Command or Action | Purpose |
|---|---|---|
| **Step 1** | **clear spanning-tree detected-protocol** [**interface** {**ethernet** *slot/port* \| **port channel** *channel-number*}]<br>**Example:**<br>`switch# clear spanning-tree detected-protocol` | Reinitializes Rapid PVST+ on all interfaces on the device or specified interfaces. |

### Example

This example shows how to reinitialize Rapid PVST+ on the Ethernet interface on slot 1, port 8:

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

# Verifying the Rapid PVST+ Configurations

To display Rapid PVST+ configuration information, perform one of the following tasks:

| Command | Purpose |
|---|---|
| **show running-config spanning-tree** [ **all**] | Displays STP information. |
| **show spanning-tree summary** | Displays summary STP information. |
| **show spanning-tree detail** | Displays detailed STP information. |
| **show spanning-treeshow spanning-tree** {**vlan***vlan-id* \| **interface** {[**ethernet***slot/port*] \| [**port-channel***channel-number*]}} [**detail**] | Displays STP information per VLAN and interface. |
| **show spanning-tree vlanshow spanning-tree vlan** *vlan-id* **bridge** | Displays information on the STP bridge. |

# Displaying and Clearing Rapid PVST+ Statistics -- CLI Version

To display Rapid PVRST+ configuration information, perform one of the following tasks:

| Command | Purpose |
|---|---|
| **clear spanning-tree counters** [**interface** *type slot/port* \| **vlan***vlan-id*] | Clears the counters for STP. |
| **show spanning-tree** {**vlan** *vlan-id* \| **interface** {[**ethernet** *slot/port*] \| [**port-channel** *channel-number*]}} **detail** | Displays information about STP by interface or VLAN including BPDUs sent and received. |

# Rapid PVST+ Example Configurations

The following example shows how to configure Rapid PVST+:

```
switch# configure terminal
switch(config)# spanning-tree port type edge bpduguard default
switch(config)# spanning-tree port type edge bpdufilter default
switch(config)# spanning-tree port type network default
switch(config)# spanning-tree vlan 1-10 priority 24576
switch(config)# spanning-tree vlan 1-10 hello-time 1
switch(config)# spanning-tree vlan 1-10 forward-time 9
switch(config)# spanning-tree vlan 1-10 max-age 13

switch(config)# interface Ethernet 1/1 switchport
switch(config-if)# spanning-tree port type edge
switch(config-if)# exit

switch(config)# spanning-tree port type edge
switch(config-if)# switchport
switch(config-if)# switchport mode trunk
```

```
switch(config-if)# spanning-tree guard root
switch(config-if)# exit
switch(config)#
```

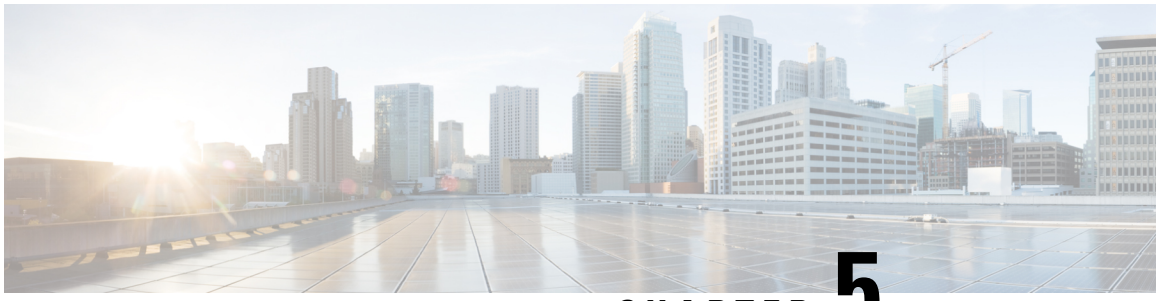
# Additional References for Rapid PVST+ -- CLI Version

### Related Documents

| Related Topic | Document Title |
|---|---|
| Layer 2 interfaces | *Cisco Nexus 3550-T Series NX-OS Interfaces Configuration Guide* |
| System management | *Cisco Nexus 3550-T Series NX-OS System Management Configuration Guide* |

### Standards

| Standards | Title |
|---|---|
| IEEE 802.1Q-2006 (formerly known as IEEE 802.1s), IEEE 802.1D-2004 (formerly known as IEEE 802.1w), IEEE 802.1D, IEEE 802.1t | — |



CHAPTER 5

# Configuring MST Using Cisco NX-OS



## Information About MST

**Note** See the *Cisco Nexus® 3550-T Interfaces Configuration* section, for information on creating Layer 2 interfaces.

MST, which is the IEEE 802.1s standard, allows you to assign two or more VLANs to a spanning tree instance. MST is not the default spanning tree mode; Rapid per VLAN Spanning Tree (Rapid PVST+) is the default mode. MST instances with the same name, revision number, and VLAN-to-instance mapping combine to form an MST region. The MST region appears as a single bridge to spanning tree configurations outside the region. MST forms a boundary to that interface when it receives an IEEE 802.1D Spanning Tree Protocol (STP) message from a neighboring device.

**Note** Spanning tree is used to refer to IEEE 802.1w and IEEE 802.1s. If the IEEE 802.1D Spanning Tree Protocol is discussed in this publication, 802.1D is stated specifically.

### MST Overview

**Note** RSTP (Rapid Spanning Tree Protocol) is the default spanning tree mode.

MST maps multiple VLANs into a spanning tree instance, with each instance having a spanning tree topology independent of other spanning tree instances. This architecture provides multiple forwarding paths for data traffic, enables load balancing, and reduces the number of STP instances required to support a large number of VLANs. MST improves the fault tolerance of the network because a failure in one instance (forwarding path) does not affect other instances (forwarding paths).

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

MAC address reduction is always enabled on the device. You cannot disable this feature.

MST improves spanning tree operation and maintains backward compatibility with original 802.1D spanning tree STP versions:

**Note**

- IEEE 802.1 was defined in the Rapid Spanning Tree Protocol (RSTP) and was incorporated into IEEE 802.1D.
- IEEE 802.1 was defined in MST and was incorporated into IEEE 802.1Q

  .

# MST Regions

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

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

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

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

Each device can support only single MST instance (Instance 0), in a single MST region. You can assign a VLAN to only one MST instance at a time.

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

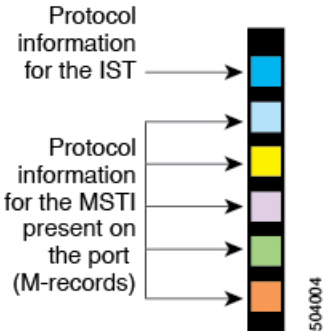
**Note** We do not recommend that you partition the network into a large number of regions.

# MST BPDUs

Each device has only one MST BPDU per interface, and that BPDU carries an M-record for each MSTI on the device. Only the IST sends BPDUs for the MST region; all M-records are encapsulated in that one BPDU

that the IST sends. Because the MST BPDU carries information for all instances, the number of BPDUs that need to be processed to support MST is significantly reduced.

***Figure 8: MST BPDU with M-Records for MSTIs***




# MST Configuration Information

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

You can configure the three parameters of the MST configuration as follows:

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

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

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

**Note** When you change the VLAN-to-MSTI mapping, the system reconverges MST.

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

# IST, CIST, and CST

## IST, CIST, and CST Overview

MST establishes and maintains IST, CIST, and CST spanning trees, as follows:

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

- MST establishes and maintains additional spanning trees within each MST region; these spanning trees are called multiple spanning tree instances (MSTIs).
- Instance 0 is a special instance for a region, known as the IST. The IST always exists on all ports; you cannot delete the IST, or Instance 0. By default, all VLANs are assigned to the IST. All other MST instances are numbered from 1 to 4094.
- The IST is the only STP instance that sends and receives BPDUs. All of the other MSTI information is contained in MST records (M-records), which are encapsulated within MST BPDUs.
- All MSTIs within the same region share the same protocol timers, but each MSTI has its own topology parameters, such as the root bridge ID, the root path cost, and so forth.
- An MSTI is local to the region; for example, MSTI 9 in region A is independent of MSTI 9 in region B, even if regions A and B are interconnected. Only CST information crosses region boundaries.
- The CST interconnects the MST regions and any instance of 802.1D and 802.1w STP that may be running on the network. The CST is the one STP instance for the entire bridged network and encompasses all MST regions and 802.1w and 802.1D instances.
- A CIST is a collection of the ISTs in each MST region. The CIST is the same as an IST inside an MST region, and the same as a CST outside an MST region.

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

## Spanning Tree Operation Within an MST Region

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

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

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

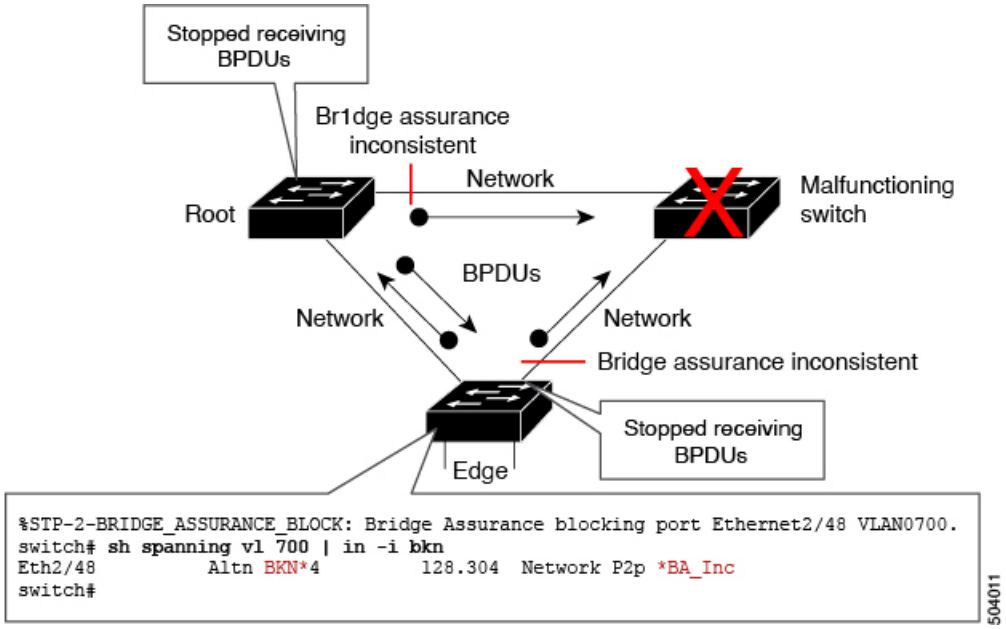
All devices in the MST region must agree on the same CIST regional root. Any two devices in the region will only synchronize their port roles for an MSTI if they converge to a common CIST regional root.

## Spanning Tree Operations Between MST Regions

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

The IST connects all the MST devices in the region and appears as a subtree in the CIST that encompasses the entire switched domain. The root of the subtree is the CIST regional root. The MST region appears as a virtual device to adjacent STP devices and MST regions.

***Figure 9: MST Regions, CIST Regional Roots, and CST Root***




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

MST devices use Version 3 BPDUs. If the MST device falls back to 802.1D STP, the device uses only 802.1D BPDUs to communicate with 802.1D-only devices. MST devices use MST BPDUs to communicate with MST devices.

## MST Terminology

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

- The CIST root is the root bridge for the CIST, which is the unique instance that spans the whole network.
- The CIST external root path cost is the cost to the CIST root. This cost is left unchanged within an MST region. An MST region looks like a single device to the CIST. The CIST external root path cost is the root path cost 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 device to the CIST root in the region. The CIST regional root acts as a root bridge for the IST.

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

# Hop Count

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

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

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

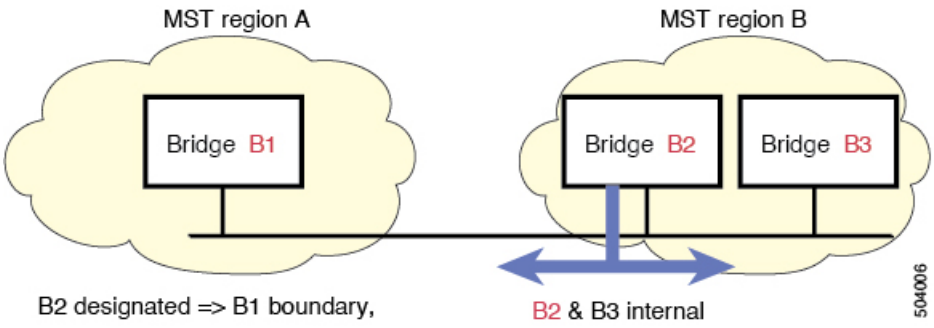
The message-age and maximum-age information in the 802.1w portion of the BPDU remain the same throughout the region (only on the IST), and the same values are propagated by the region-designated ports at the boundary.

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

# Boundary Ports

A boundary port is a port that connects to a LAN, the designated bridge of a bridge with a different MST configuration (and so, a separate MST region) 802.1D STP bridge. A designated port knows that it is on the boundary if it detects an STP bridge or receives an agreement proposal from an MST bridge with a different configuration. This definition allows two ports that are internal to a region to share a segment with a port that belongs to a different region, creating the possibility of receiving both internal and external messages on a port.

***Figure 10: MST Boundary Ports***




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

# Port Cost and Port Priority

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

- 1 Gigabit Ethernet—20,000
- 10 Gigabit Ethernet—2,000
- 40 Gigabit Ethernet—500

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

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

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

# Interoperability with IEEE 802.1D

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

However, the device does not automatically revert to the MST mode if it no longer receives 802.1D BPDUs because it cannot detect whether the 802.1D device has been removed from the link unless the 802.1D device 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), enter the **clear spanning-tree detected-protocols** command.

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

MST interoperates with the Cisco prestandard MSTP whenever it receives prestandard MSTP on an MST port; no explicit configuration is necessary.

You can also configure the interface to proactively send prestandard MSTP messages.

# Prerequisites for MST

MST has the following prerequisites:

- You must be logged onto the device.

# Guidelines and Limitations for Configuring MST

**Note** When you change the VLAN-to-MSTI mapping, the system reconverges MST.

MST has the following configuration guidelines and limitations:

- For MST configuration limits, see the *Cisco Nexus® 3550-T Verified Scalability Guide*.
- **show** commands with the **internal** keyword are not supported.
- Beginning with Cisco NX-OS Release 10.2(3t), RSTP (Rapid Spanning Tree Protocol) is the default spanning tree mode.
- You can assign a VLAN to only one MST instance in Cisco Nexus® 3550-T switches.
- By default, all VLANs are mapped to MSTI 0 or the IST.
- You can load balance only within the MST region.
- Ensure that trunks within an MST region carry all of the VLANs that are mapped to an MSTI or exclude all those VLANs that are mapped to an MSTI.
- Always leave STP enabled.
- Do not change timers because you can adversely affect your network stability.
- Keep user traffic off the management VLAN; keep the management VLAN separate from user data.
- Choose the distribution and core layers as the location of the primary and secondary root switches.
- Port channeling—The port channel bundle is considered as a single port. The port cost is the aggregation of all the configured port costs assigned to that channel.
- When you map a VLAN to an MSTI, the system automatically removes that VLAN from its previous MSTI.
- You can map any number of VLANs to an MSTI.
- Do not partition the network into a large number of regions. However, if this situation is unavoidable, we recommend that you partition the switched LAN into smaller LANs interconnected by non-Layer 2 devices.
- When you are in the MST configuration submode, the following guidelines apply:
  - Each command reference line creates its pending regional configuration.
  - The pending region configuration starts with the current region configuration.
  - To leave the MST configuration submode without committing any changes, enter the **abort** command.
  - To leave the MST configuration submode and commit all the changes that you made before you left the submode, enter the **exit** or **end** commands, or press **Ctrl + Z**.

# Default Settings for MST

This table lists the default settings for MST parameters.

*Table 8: Default MST Parameters*

| Parameters | Default |
|---|---|
| Spanning tree | Enabled |
| Name | Empty string |
| VLAN mapping | All VLANs mapped to a CIST instance |
| Revision | 0 |
| Instance ID | Instance 0; VLANs 1 to 3967 are mapped to Instance 0 by default |
| MSTI per MST region | Only single instance of MST is permitted in the Cisco Nexus® 3550-T switches |
| Bridge priority (configurable per CIST port) | 32768 |
| Spanning tree port priority (configurable per CIST port) | 128 |
| Spanning tree port cost (configurable per CIST port) | Auto<br>The default port cost is determined by the port speed as follows:<br>• 1 Gigabit Ethernet: 20,000<br>• 10 Gigabit Ethernet: 2,000<br>• 40 Gigabit Ethernet: 500 |
| Hello time | 2 seconds |
| Forward-delay time | 15 seconds |
| Maximum-aging time | 20 seconds |
| Maximum hop count | 20 hops |
| Link type | Auto<br>The default link type is determined by the duplex, as follows:<br>• Full duplex: point-to-point link<br>• Half duplex: shared link |

# Configuring MST

**Note** If you are familiar with the Cisco IOS CLI, be aware that the Cisco software commands for this feature might differ from the Cisco IOS commands that you would use.

## Enabling MST - CLI Version

**Note** When you change the spanning tree mode, traffic is disrupted because all spanning tree instances are stopped for the previous mode and started for the new mode.

### Procedure

| | Command or Action | Purpose |
|---|---|---|
| **Step 1** | **config t**<br>**Example:**<br>`switch# config t`<br>`switch(config)#` | Enters configuration mode. |
| **Step 2** | **spanning-tree mode mst** or **no spanning-tree mode mst**.<br>**Example:**<br>`switch(config)# spanning-tree mode mst` | • **spanning-tree mode mst**<br>Enables MST on the device.<br>• **no spanning-tree mode mst**<br>Disables MST on the device. |
| **Step 3** | **exit**<br>**Example:**<br>`switch(config)# exit`<br>`switch#` | Exits configuration mode. |
| **Step 4** | (Optional) **show running-config spanning-tree all**<br>**Example:**<br>`switch# show running-config spanning-tree all` | Displays the currently running STP configuration. |
| **Step 5** | (Optional) **copy running-config startup-config**<br>**Example:**<br>`switch# copy running-config startup-config` | Copies the running configuration to the startup configuration. |

### Example

This example shows how to enable MST on the device:

```
switch# config t
switch(config)# spanning-tree mode mst
switch(config)# exit
switch#
```

# Entering MST Configuration Mode

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

If two or more devices are in the same MST region, they must have the identical MST name, VLAN-to-instance mapping, and MST revision number.

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

### Procedure

| | Command or Action | Purpose |
|---|---|---|
| **Step 1** | **config t**<br>**Example:**<br>`switch# config t`<br>`switch(config)#` | Enters configuration mode. |
| **Step 2** | **spanning-tree mst configuration** or **no spanning-tree mst configuration**<br>**Example:**<br>`switch(config)# spanning-tree mst configuration`<br>`switch(config-mst)#` | • **spanning-tree mst configuration**<br>Enters MST configuration submode on the system. You must be in the MST configuration submode to assign the MST configuration parameters, as follows:<br>• MST name<br>• VLAN-to-MST instance mapping<br>• MST revision number<br>• **no spanning-tree mst configuration**<br>Returns the MST region configuration to the following default values:<br>• The region name is an empty string. |

| | Command or Action | Purpose |
|---|---|---|
| | | • No VLANs are mapped to any MST instance (all VLANs are mapped to the CIST instance).<br>• The revision number is 0. |
| **Step 3** | **exit** or **abort**<br>**Example:**<br>`switch(config-mst)# exit`<br>`switch(config)#` | • **exit**<br>Commits all the changes and exits MST configuration submode.<br>• **abort**<br>Exits the MST configuration submode without committing any of the changes. |
| **Step 4** | (Optional) **copy running-config startup-config**<br>**Example:**<br>`switch(config)# copy running-config startup-config` | Copies the running configuration to the startup configuration. |

### Example

This example shows how to enter the MST configuration submode on the device:

```
switch# config t
switch(config)# spanning-tree mst configuration
switch(config-mst)# exit
switch(config)#
```

# Specifying the MST Name

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

### Procedure

| | Command or Action | Purpose |
|---|---|---|
| **Step 1** | **config t**<br>**Example:**<br>`switch# config t`<br>`switch(config)#` | Enters configuration mode. |
| **Step 2** | **spanning-tree mst configuration**<br>**Example:** | Enters MST configuration submode. |

| | Command or Action | Purpose |
|---|---|---|
| | `switch(config)# spanning-tree mst configuration`<br>`switch(config-mst)#` | |
| **Step 3** | **name** *name*<br>**Example:**<br>`switch(config-mst)# name accounting` | Specifies the name for the MST region. The *name* string has a maximum length of 32 characters and is case sensitive. The default is an empty string. |
| **Step 4** | **exit** or **abort**<br>**Example:**<br>`switch(config-mst)# exit`<br>`switch(config)#` | • **exit**<br>Commits all the changes and exits MST configuration submode.<br>• **abort**<br>Exits the MST configuration submode without committing any of the changes. |
| **Step 5** | (Optional) **show spanning-tree mst configuration**<br>**Example:**<br>`switch# show spanning-tree mst configuration` | Displays the MST configuration. |
| **Step 6** | (Optional) **copy running-config startup-config**<br>**Example:**<br>`switch(config)# copy running-config startup-config` | Copies the running configuration to the startup configuration. |

### Example

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

```
switch# config t
switch(config)#  spanning-tree mst configuration
switch(config-mst)# name accounting
switch(config-mst)# exit
switch(config)#
```

# Specifying the MST Configuration Revision Number

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

### Procedure

| | Command or Action | Purpose |
|---|---|---|
| **Step 1** | **config t**<br>**Example:**<br>`switch# config t`<br>`switch(config)#` | Enters configuration mode. |
| **Step 2** | **spanning-tree mst configuration**<br>**Example:**<br>`switch(config)# spanning-tree mst configuration`<br>`switch(config-mst)#` | Enters MST configuration submode. |
| **Step 3** | **revision** *version*<br>**Example:**<br>`switch(config-mst)# revision 5` | Specifies the revision number for the MST region. The range is from 0 to 65535, and the default value is 0. |
| **Step 4** | **exit** or **abort**<br>**Example:**<br>`switch(config-mst)# exit`<br>`switch(config)#` | • **exit**<br>Commits all the changes and exits MST configuration submode.<br>• **abort**<br>Exits the MST configuration submode without committing any of the changes. |
| **Step 5** | (Optional) **show spanning-tree mst configuration**<br>**Example:**<br>`switch# show spanning-tree mst configuration` | Displays the MST configuration. |
| **Step 6** | (Optional) **copy running-config startup-config**<br>**Example:**<br>`switch(config)# copy running-config startup-config` | Copies the running configuration to the startup configuration. |

### Example

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

```
switch# config t
switch(config)# spanning-tree mst configuration
switch(config-mst)# revision 5
switch(config-mst)#
```

# Configuring the Root Bridge

You can configure the device to become the MST root bridge.

The **spanning-tree vlan** *vlan_ID* **primary root** command fails if the value required to be the root bridge is less than 4096. If the software cannot lower the bridge priority any lower, the device returns the following message:

```
Error: Failed to set root bridge for VLAN 1
It may be possible to make the bridge root by setting the priority
for some (or all) of these instances to zero.
```

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

Enter the **diameter** keyword, which is available only for MSTI 0 (or the IST), to specify the Layer 2 network diameter (that is, the maximum number of Layer 2 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 enter the **hello** keyword to override the automatically calculated hello time.

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

### Procedure

| | Command or Action | Purpose |
|---|---|---|
| **Step 1** | **config t**<br>**Example:**<br>`switch# config t`<br>`switch(config)#` | Enters configuration mode. |
| **Step 2** | **spanning-tree mst** *instance-id* **root** {**primary** \| **secondary**} [**diameter** *dia* [**hello-time** *hello-time*]] or **no spanning-tree mst** *instance-id* **root**<br>**Example:**<br>`switch(config)# spanning-tree mst 5 root primary` | • **spanning-tree mst** *instance-id* **root** {**primary** \| **secondary**} [**diameter** *dia* [**hello-time** *hello-time*]]<br>Configures a device as the root bridge as follows:<br>• For *instance-id*, specify a single instance, a range of instances separated by a hyphen, or a series of instances separated by a comma. The range is from 1 to 4094.<br>• For **diameter** *net-diameter*, specify the maximum number of Layer 2 |

| | Command or Action | Purpose |
|---|---|---|
| | | hops between any two end stations. The default is 7. This keyword is available only for MST instance 0.<br>• For **hello-time** *seconds*, specify the interval in seconds between the generation of configuration messages by the root bridge. The range is from 1 to 10 seconds; the default is 2 seconds.<br>• **no spanning-tree mst** *instance-id* **root**<br>Returns the switch priority, diameter, and hello time to default values. |
| **Step 3** | **exit** or **abort**<br>**Example:**<br>`switch(config)# exit`<br>`switch#` | • **exit**<br>Commits all the changes and exits MST configuration submode.<br>• **abort**<br>Exits the MST configuration submode without committing any of the changes. |
| **Step 4** | (Optional) **show spanning-tree mst**<br>**Example:**<br>`switch# show spanning-tree mst` | Displays the MST configuration. |
| **Step 5** | (Optional) **copy running-config startup-config**<br>**Example:**<br>`switch(config)# copy running-config startup-config` | Copies the running configuration to the startup configuration. |

### Example

This example shows how to configure the device as the root switch for MSTI 5:

```
switch# config t
switch(config)# spanning-tree mst 5 root primary
switch(config)# exit
switch(config)#
```

# Configuring an MST Secondary Root Bridge

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

### Procedure

| | Command or Action | Purpose |
|---|---|---|
| **Step 1** | **config t**<br>**Example:**<br>`switch# config t`<br>`switch(config)#` | Enters configuration mode. |
| **Step 2** | **spanning-tree mst** *instance-id* **root** {**primary** \| **secondary**} [**diameter** *dia*[**hello-time** *hello-time*]] or **no spanning-tree mst** *instance-id* **root**<br>**Example:**<br>`switch(config)# spanning-tree mst 0 root secondary` | • **spanning-tree mst** *instance-id* **root** {**primary** \| **secondary**} [**diameter** *dia*[**hello-time** *hello-time*]]<br>Configures a device as the secondary root bridge as follows:<br>• For *instance-id*, specify the single MSTI ID.<br>• For **diameter** *net-diameter*, specify the maximum number of Layer 2 hops between any two end stations. The default is 7. This keyword is available only for MST instance 0.<br>• For **hello-time** *seconds*, specify the interval in seconds between the generation of configuration messages by the root bridge. The range is from 1 to 10 seconds; the default is 2 seconds.<br>• **no spanning-tree mst** *instance-id* **root**<br>Returns the switch priority, diameter, and hello-time to default values. |
| **Step 3** | **exit**<br>**Example:**<br>`switch# exit`<br>`switch(config)#` | Exits configuration mode. |
| **Step 4** | (Optional) **show spanning-tree mst**<br>**Example:**<br>`switch# show spanning-tree mst` | Displays the MST configuration. |

| | Command or Action | Purpose |
|---|---|---|
| **Step 5** | (Optional) **copy running-config startup-config**<br>**Example:**<br>`switch(config)# copy running-config startup-config` | Copies the running configuration to the startup configuration. |

### Example

This example shows how to configure the device as the secondary root switch for MSTI 0:

```
switch# config t
switch(config)# spanning-tree mst 0 root secondary
switch(config)# exit
switch#
```

# Configuring the MST Switch Priority

You can configure the switch priority for an MST instance so that it is more likely that the specified device is chosen as the root bridge.

### Procedure

| | Command or Action | Purpose |
|---|---|---|
| **Step 1** | **config t**<br>**Example:**<br>`switch# config t`<br>`switch(config)#` | Enters configuration mode. |
| **Step 2** | **spanning-tree mst** *instance-id* **priority** *priority-value*<br>**Example:**<br>`switch(config)# spanning-tree mst 0 priority 4096` | Configures a device priority as follows:<br>• For *instance-id*, specify the single MSTI ID.<br>• For *priority-value* the range is from 0 to 61440 in increments of 4096; the default is 32768. A lower number indicates that the device will most likely be chosen as the root bridge.<br>Priority values are 0, 4096, 8192, 12288, 16384, 20480, 24576, 28672, 32768, 36864, 40960, 45056, 49152, 53248, 57344, and 61440. The system rejects all other values. |
| **Step 3** | **exit**<br>**Example:** | Exits configuration mode. |

| | Command or Action | Purpose |
|---|---|---|
| | `switch(config)# exit`<br>`switch#` | |
| **Step 4** | (Optional) **show spanning-tree mst**<br>**Example:**<br>`switch# show spanning-tree mst` | Displays the MST configuration. |
| **Step 5** | (Optional) **copy running-config startup-config**<br>**Example:**<br>`switch(config)# copy running-config startup-config` | Copies the running configuration to the startup configuration. |

### Example

This example shows how to configure the priority of the bridge to 4096 for MSTI 0:

```
switch# config t
switch(config)# spanning-tree mst 0 priority 4096
switch(config)# exit
switch#
```

# Configuring the MST Port Priority

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

### Procedure

| | Command or Action | Purpose |
|---|---|---|
| **Step 1** | **config t**<br>**Example:**<br>`switch# config t`<br>`switch(config)#` | Enters configuration mode. |
| **Step 2** | **interface** {{*type slot/port*} \| {**port-channel** *number*}}<br>**Example:**<br>`switch(config)# interface ethernet 1/1`<br>`switch(config-if)#` | Specifies an interface to configure, and enters interface configuration mode. |
| **Step 3** | **spanning-tree mst** *instance-id* **port-priority** *priority*<br>**Example:** | Configures the port priority as follows:<br>• For *instance-id*, specify the single MSTI ID. |

| | Command or Action | Purpose |
|---|---|---|
| | `switch(config-if)# spanning-tree mst 3 port-priority 64` | • For *priority*, the range is from 0 to 224 in increments of 32. The default is 128. A lower number indicates a higher priority.<br>The priority values are 0, 32, 64, 96, 128, 160, 192, and 224. The system rejects all other values. |
| **Step 4** | **exit**<br>**Example:**<br>`switch(config-if)# exit`<br>`switch(config)#` | Exits interface mode. |
| **Step 5** | (Optional) **show spanning-tree mst**<br>**Example:**<br>`switch# show spanning-tree mst` | Displays the MST configuration. |
| **Step 6** | (Optional) **copy running-config startup-config**<br>**Example:**<br>`switch(config)# copy running-config startup-config` | Copies the running configuration to the startup configuration. |

### Example

This example shows how to set the MST interface port priority for MSTI 0 on Ethernet port 1/1 to 64:

```
switch# config t
switch(config)# interface ethernet 1/1
switch(config-if)# spanning-tree mst 0 port-priority 64
switch(config-if)# exit
switch(config)#
```

# Configuring the MST Port Cost

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

**Note** MST uses the long path-cost calculation method.

### Procedure

| | Command or Action | Purpose |
|---|---|---|
| **Step 1** | **config t**<br>**Example:**<br>`switch# config t`<br>`switch(config)#` | Enters configuration mode. |
| **Step 2** | **interface** {{*type slot/port*} \| {**port-channel** *number*}}<br>**Example:**<br>`switch# config t`<br>`switch(config)# interface ethernet 1/1`<br>`switch(config-if)#` | Specifies an interface to configure, and enters interface configuration mode. |
| **Step 3** | **spanning-tree mst** *instance-id* **cost** {*cost* \| *auto*}<br>**Example:**<br>`switch(config-if)# spanning-tree mst 4 cost 17031970` | Configures the cost.<br>If a loop occurs, MST uses the path cost when selecting an interface to place into the forwarding state. A lower path cost represents higher-speed transmission as follows:<br>• For *instance-id*, specify the single MSTI ID.<br>• For *cost*, the range is from 1 to 200000000. The default value is **auto**, which is derived from the media speed of the interface. |
| **Step 4** | **exit**<br>**Example:**<br>`switch(config-if)# exit`<br>`switch(config)#` | Exits interface mode. |
| **Step 5** | (Optional) **show spanning-tree mst**<br>**Example:**<br>`switch# show spanning-tree mst` | Displays the MST configuration. |
| **Step 6** | (Optional) **copy running-config startup-config**<br>**Example:**<br>`switch(config)# copy running-config startup-config` | Copies the running configuration to the startup configuration. |

### Example

This example shows how to set the MST interface port cost on Ethernet 1/1 for MSTI 0:

```
switch# config t
switch(config)# interface ethernet 1/1
switch(config-if)# spanning-tree mst 0 cost 17031970
switch(config-if)# exit
switch(config)#
```

# Configuring the MST Hello Time

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

**Note** Be careful when using the **spanning-tree mst hello-time** command. For most situations, we recommend that you enter the **spanning-tree mst** *instance-id* **root primary** and the **spanning-tree mst** *instance-id* **root secondary** global configuration commands to modify the hello time.

### Procedure

| | Command or Action | Purpose |
|---|---|---|
| **Step 1** | **config t**<br>**Example:**<br>`switch# config t`<br>`switch(config)#` | Enters configuration mode. |
| **Step 2** | **spanning-tree mst hello-time** *seconds*<br>**Example:**<br>`switch(config)# spanning-tree mst hello-time 1` | Configures the hello time for the MST instance. The hello time is the interval between the generation of configuration messages by the root bridge. These messages mean that the device is alive. For *seconds*, the range is from 1 to 10, and the default is 2 seconds. |
| **Step 3** | **exit**<br>**Example:**<br>`switch(config)# exit`<br>`switch#` | Exits configuration mode. |
| **Step 4** | (Optional) **show spanning-tree mst**<br>**Example:**<br>`switch# show spanning-tree mst` | Displays the MST configuration. |
| **Step 5** | (Optional) **copy running-config startup-config**<br>**Example:**<br>`switch(config)# copy running-config startup-config` | Copies the running configuration to the startup configuration. |

### Example

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

```
switch# config t
switch(config)# spanning-tree mst hello-time 1
switch(config)# exit
switch#
```

# Configuring the MST Forwarding-Delay Time

You can set the forward delay timer for the MST instance on the device with one command.

### Procedure

| | Command or Action | Purpose |
|---|---|---|
| **Step 1** | **config t**<br>**Example:**<br>`switch# config t`<br>`switch(config)#` | Enters configuration mode. |
| **Step 2** | **spanning-tree mst forward-time** *seconds*<br>**Example:**<br>`switch(config)# spanning-tree mst forward-time 10` | Configures the forward time for the MST instance. The forward delay is the number of seconds that a port waits before changing from its spanning tree blocking and learning states to the forwarding state. For *seconds*, the range is from 4 to 30, and the default is 15 seconds. |
| **Step 3** | **exit**<br>**Example:**<br>`switch(config)# exit`<br>`switch#` | Exits configuration mode. |
| **Step 4** | (Optional) **show spanning-tree mst**<br>**Example:**<br>`switch# show spanning-tree mst` | Displays the MST configuration. |
| **Step 5** | (Optional) **copy running-config startup-config**<br>**Example:**<br>`switch(config)# copy running-config startup-config` | Copies the running configuration to the startup configuration. |

### Example

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

```
switch# config t
switch(config)# spanning-time mst forward-time 10
switch(config)# exit
switch#
```

# Configuring the MST Maximum-Aging Time

You can set the maximum-aging timer for the MST instance on the device with one command (the maximum age time only applies to the IST).

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

### Procedure

| | Command or Action | Purpose |
|---|---|---|
| **Step 1** | **config t**<br>**Example:**<br>`switch# config t`<br>`switch(config)#` | Enters configuration mode. |
| **Step 2** | **spanning-tree mst max-age** *seconds*<br>**Example:**<br>`switch(config)# spanning-tree mst max-age 40` | Configures the maximum-aging time for the MST instance. The maximum-aging time is the number of seconds that a device waits without receiving spanning tree configuration messages before attempting a reconfiguration. For *seconds*, the range is from 6 to 40, and the default is 20 seconds. |
| **Step 3** | **exit**<br>**Example:**<br>`switch(config)# exit`<br>`switch#` | Exits configuration mode. |
| **Step 4** | (Optional) **show spanning-tree mst**<br>**Example:**<br>`switch# show spanning-tree mst` | Displays the MST configuration. |
| **Step 5** | (Optional) **copy running-config startup-config**<br>**Example:**<br>`switch(config)# copy running-config startup-config` | Copies the running configuration to the startup configuration. |

### Example

This example shows how to configure the maximum-aging timer of the device to 40 seconds:

```
switch# config t
switch(config)# spanning-tree mst max-age 40
switch(config)# exit
switch#
```

# Configuring the MST Maximum-Hop Count

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

### Procedure

| | Command or Action | Purpose |
|---|---|---|
| **Step 1** | **config t**<br>**Example:**<br>`switch# config t`<br>`switch(config)#` | Enters configuration mode. |
| **Step 2** | **spanning-tree mst max-hops** *hop-count*<br>**Example:**<br>`switch(config)# spanning-tree mst max-hops 40` | Specifies the number of hops in a region before the BPDU is discarded and the information held for a port is aged. For *hop-count*, the range is from 1 to 255, and the default value is 20 hops. |
| **Step 3** | **exit**<br>**Example:**<br>`switch(config-mst)# exit`<br>`switch#` | Exits configuration mode. |
| **Step 4** | (Optional) **show spanning-tree mst**<br>**Example:**<br>`switch# show spanning-tree mst` | Displays the MST configuration. |
| **Step 5** | (Optional) **copy running-config startup-config**<br>**Example:**<br>`switch(config)# copy running-config startup-config` | Copies the running configuration to the startup configuration. |

### Example

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

```
switch# config t
switch(config)# spanning-tree mst max-hops 40
switch(config)# exit
switch#
```

# Configuring an Interface to Proactively Send Prestandard MSTP Messages - CLI Version

By default, interfaces on a device running MST send prestandard, rather than standard, MSTP messages after they receive a prestandard MSTP message from another interface. You can configure the interface to proactively send prestandard MSTP messages. That is, the specified interface would not have to wait to receive a prestandard MSTP message; the interface with this configuration always sends prestandard MSTP messages.

### Procedure

| | Command or Action | Purpose |
|---|---|---|
| **Step 1** | **config t**<br>**Example:**<br>`switch# config t`<br>`switch(config)#` | Enters configuration mode. |
| **Step 2** | **interface** *type slot/port*<br>**Example:**<br>`switch(config)# interface ethernet 1/4`<br>`switch(config-if)#` | Specifies the interface to configure and enters the interface configuration mode. |
| **Step 3** | **spanning-tree mst pre-standard**<br>**Example:**<br>`switch(config-if)# spanning-tree mst pre-standard` | Specifies that the interface always sends MSTP messages in the prestandard format, rather than in the MSTP standard format. |
| **Step 4** | **exit**<br>**Example:**<br>`switch(config-if)# exit`<br>`switch(config)#` | Exits interface mode. |
| **Step 5** | (Optional) **show spanning-tree mst**<br>**Example:**<br>`switch# show spanning-tree mst` | Displays the MST configuration. |
| **Step 6** | (Optional) **copy running-config startup-config**<br>**Example:**<br>`switch(config)# copy running-config startup-config` | Copies the running configuration to the startup configuration. |

### Example

This example shows how to set the MST interface so that it always sends MSTP messages in the prestandard format:

```
switch# config t
switch (config)# interface ethernet 1/4
switch(config-if)# spanning-tree mst pre-standard
switch(config-if)# exit
switch(config)#
```

# Specifying the Link Type for MST - CLI Version

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

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

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

## Procedure

| | Command or Action | Purpose |
|---|---|---|
| **Step 1** | **config t**<br>**Example:**<br>`switch# config t`<br>`switch(config)#` | Enters configuration mode. |
| **Step 2** | **interface** *type slot/port*<br>**Example:**<br>`switch(config)# interface ethernet 1/4`<br>`switch(config-if)#` | Specifies the interface to configure and enters the interface configuration mode. |
| **Step 3** | **spanning-tree link-type** {*auto* \| *point-to-point* \| *shared*}<br>**Example:**<br>`switch(config-if)# spanning-tree link-type point-to-point` | Configures the link type to be either a point-to-point link or shared link. The system reads the default value from the device connection, as follows: half duplex links are shared and full-duplex links are point to point. If the link type is shared, the STP falls back to 802.1D. The default is auto, which sets the link type based on the duplex setting of the interface. |
| **Step 4** | **exit**<br>**Example:**<br>`switch(config-if)# exit`<br>`switch(config)#` | Exits interface mode. |
| **Step 5** | (Optional) **show spanning-tree**<br>**Example:**<br>`switch# show spanning-tree` | Displays the STP configuration. |

| | Command or Action | Purpose |
|---|---|---|
| **Step 6** | (Optional) **copy running-config startup-config**<br>**Example:**<br>`switch(config)# copy running-config startup-config` | Copies the running configuration to the startup configuration. |

### Example

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

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

# Reinitializing the Protocol for MST

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

### Procedure

| | Command or Action | Purpose |
|---|---|---|
| **Step 1** | **clear spanning-tree detected-protocol** [**interface** *interface* [*interface-num* \| *port-channel*]]<br>**Example:**<br>`switch# clear spanning-tree detected-protocol` | Reintializes MST on an entire device or specified interfaces. |

### Example

This example shows how to reinitialize MST on the Ethernet interface on slot 1, port 8:

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

# Verifying the MST Configuration

To display MST configuration information, perform one of the following tasks:

| Command | Purpose |
|---|---|
| **show running-config spanning-tree** [**all**] | Displays STP information. |
| **show spanning-tree mst configuration** | Displays MST information. |
| **show spanning-tree mst** [**detail**] | Displays information about MST instances. |
| **show spanning-tree mst** *instance-id* [**detail**] | Displays information about the specified MST instance. |
| **show spanning-tree mst** *instance-id* **interface** {**ethernet** *slot/port* \| **port-channel** *channel-number*} [**detail**] | Displays MST information for the specified interface and instance. |
| **show spanning-tree summary** | Displays summary STP information. |
| **show spanning-tree detail** | Displays detailed STP information. |
| **show spanning-tree** {**vlan** *vlan-id* \| **interface** {[**ethernet** *slot/port*] \| [**port-channel** *channel-number*]}} [**detail**] | Displays STP information per VLAN and interface. |
| **show spanning-tree vlan** *vlan-id* **bridge** | Displays information on the STP bridge. |

# Displaying and Clearing MST Statistics -- CLI Version

To display MST configuration information, perform one of the following tasks:

| Command | Purpose |
|---|---|
| **clear spanning-tree counters** [ **interface** *type slot/port* \| **vlan***vlan-id*] | Clears the counters for STP. |
| **show spanning-tree** {**vlan** *vlan-id* \| **interface** {[**ethernet** *slot/port*] \| [**port-channel***channel-number*]}} **detail** | Displays information about STP by interface or VLAN including BPDUs sent and received. |

# MST Example Configuration

The following example shows how to configure MST:

```
switch# configure terminal
switch(config)# spanning-tree mode mst
switch(config)# spanning-tree port type edge bpduguard default
switch(config)# spanning-tree port type edge bpdufilter default
switch(config)# spanning-tree port type network default
switch(config)# spanning-tree mst 0-64 priority 24576
```

```
switch(config)# spanning-tree mst configuration
switch(config-mst)# name cisco_region_1
switch(config-mst)# revision 2
switch(config-mst)# instance 1 vlan 1-21
```

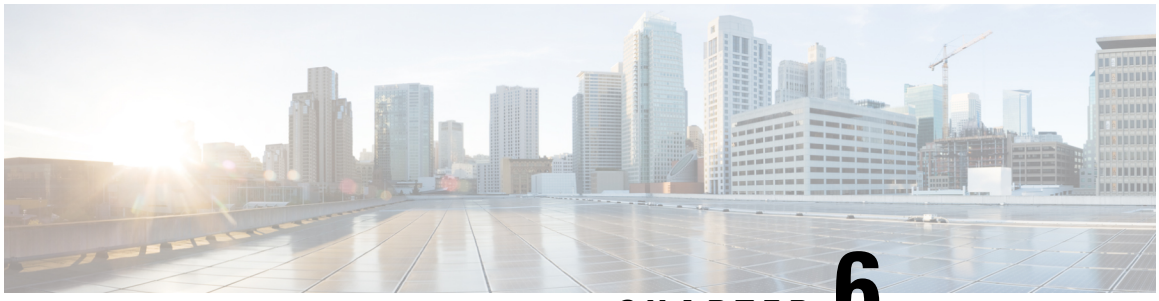
# Additional References for MST -- CLI Version

### Related Documents

| Related Topic | Document Title |
|---|---|
| Layer 2 interfaces | *Cisco Nexus® 3550-T Interfaces Configuration Guide* |
| System management | *Cisco Nexus® 3550-T System Management Configuration Guide* |

### Standards

| Standards | Title |
|---|---|
| IEEE 802.1Q-2006 (formerly known as IEEE 802.1s), IEEE 802.1D-2004 (formerly known as IEEE 802.1w), IEEE 802.1D, IEEE 802.1t | — |



CHAPTER 6

# Configuring STP Extensions Using Cisco NX-OS



## Information About STP Extensions

**Note** See the *Cisco Nexus® 3550-T Interfaces Configuration Guide*, for information on creating Layer 2 interfaces.

Cisco has added extensions to STP that enhances loop prevention, protects against some possible user misconfigurations, and provides better control over the protocol parameters. Although, in some cases, similar functionality may be incorporated into the IEEE 802.1w Rapid Spanning Tree Protocol (RSTP) standard, we recommend using these extensions. All of these extensions, except PVST Simulation, can be used with MST. You use PVST Simulation only with MST.

The available extensions are spanning tree edge ports (which supply the functionality previously known as PortFast), Bridge Assurance, BPDU Guard, BPDU Filtering, Loop Guard, Root Guard, and PVT Simulation. Many of these features can be applied either globally or on specified interfaces.

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

# STP Port Types

You can configure a spanning tree port as an edge port, a network port, or a normal port. A port can be in only one of these states at a given time. The default spanning tree port type is normal.

Edge ports, which are connected to Layer 2 hosts, can be either an access port or a trunk port.

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

Network ports are connected only to Layer 2 switches or bridges.

**Note** If you mistakenly configure ports that are connected to Layer 2 hosts, or edge devices, as spanning tree network ports, those ports will automatically move into the blocking state.

## STP Edge Ports

You connect STP edge ports only to Layer 2 hosts. The edge port interface immediately transitions to the forwarding state, without moving through the blocking or learning states. (This immediate transition was previously configured as the Cisco-proprietary feature PortFast.)

Interfaces that are connected to Layer 2 hosts should not receive STP bridge protocol data units (BPDUs).

# Bridge Assurance

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

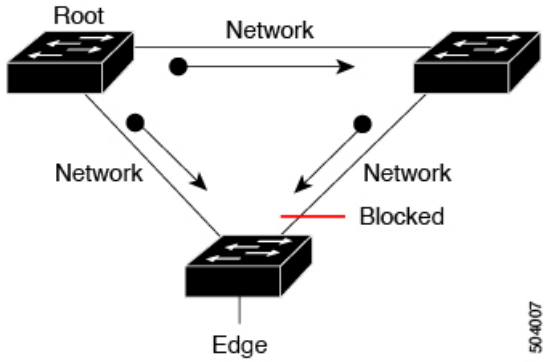
**Note** Bridge Assurance is supported only by MST.

Bridge Assurance is enabled by default and can only be disabled globally. Also, Bridge Assurance can be enabled only on spanning tree network ports that are point-to-point links. Finally, both ends of the link must have Bridge Assurance enabled. If the device on one side of the link has Bridge Assurance enabled and the device on the other side either does not support Bridge Assurance or does not have this feature enabled, the connecting port is blocked.

With Bridge Assurance enabled, BPDUs are sent out on all operational network ports, including alternate and backup ports, for each hello time period. If the port does not receive a BPDU for a specified period, the port

moves into the blocking state and is not used in the root port calculation. Once that port receives a BPDU, it resumes the normal spanning tree transitions.

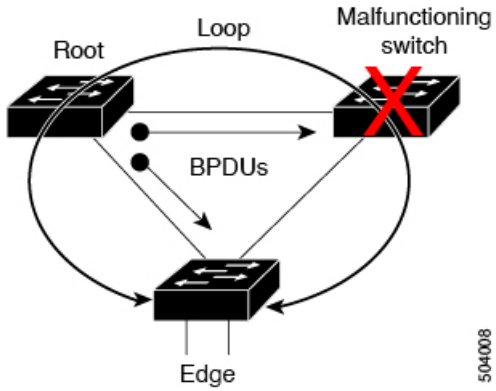
***Figure 11: Network with Normal STP Topology***




This figure shows a normal STP topology.

***Figure 12: Network Problem without Running Bridge Assurance***

This figure demonstrates a potential network problem when the device fails and you are not running Bridge




Assurance.

***Figure 13: Network STP Topology Running Bridge Assurance***

This figure shows the network with Bridge Assurance enabled, and the STP topology progressing normally with bidirectional BPDUs issuing from every STP network port.

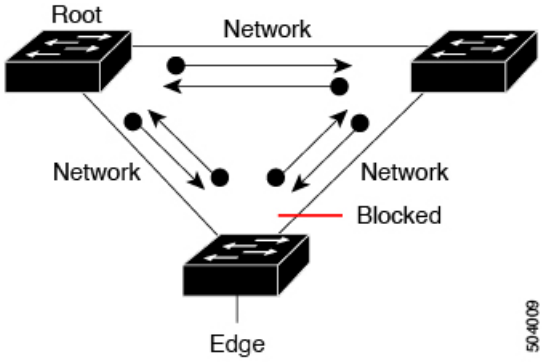
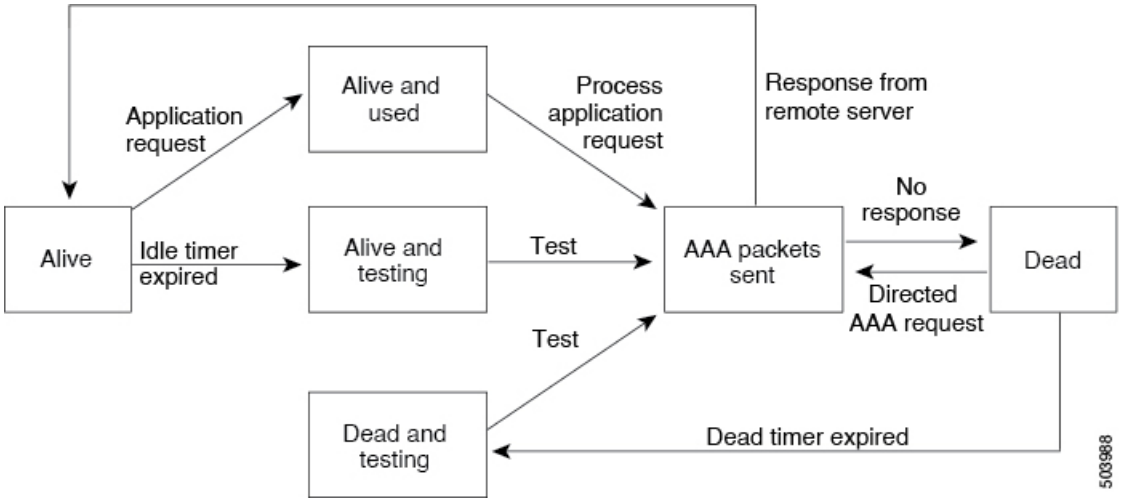


*Figure 14: Network Problem Averted with Bridge Assurance Enabled*

This figure shows how the potential network problem does not happen when you have Bridge Assurance enabled on your network.




# BPDU Guard

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

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

When you configure BPDU Guard globally, it is effective only on operational spanning tree edge ports. In a valid configuration, Layer 2 LAN edge interfaces do not receive BPDUs. A BPDU that is received by an edge Layer 2 LAN interface signals an invalid configuration, such as the connection of an unauthorized device. BPDU Guard, when enabled globally, shuts down all spanning tree edge ports when they receive a BPDU.

BPDU Guard provides a secure response to invalid configurations, because you must manually put the Layer 2 LAN interface back in service after an invalid configuration.

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

# BPDU Filtering

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

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

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

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

This table lists all the BPDU Filtering combinations.

***Table 9: BPDU Filtering Configurations***

| BPDU Filtering Per Port Configuration | BPDU Filtering Global Configuration | STP Edge Port Configuration | BPDU Filtering State |
|---|---|---|---|
| Default [1] | Enable | Enable | Enable [2] |
| Default | Enable | Disable | Disable |
| Default | Disable | Not applicable | Disable |
| Disable | Not applicable | Not applicable | Disable |
| Enable | Not applicable | Not applicable | Enable |

1 No explicit port configuration.
2 The port transmits at least 10 BPDUs. If this port receives any BPDUs, the port returns to the spanning tree normal port state and BPDU filtering is disabled.

# Loop Guard

Loop Guard helps prevent bridging loops that could occur because of a unidirectional link failure on a point-to-point link.

An STP loop occurs when a blocking port in a redundant topology erroneously transitions to the forwarding state. Transitions are usually caused by a port in a physically redundant topology (not necessarily the blocking port) that stops receiving BPDUs.

When you enable Loop Guard globally, it is useful only in switched networks where devices are connected by point-to-point links. On a point-to-point link, a designated bridge cannot disappear unless it sends an inferior BPDU or brings the link down. However, you can enable Loop Guard on shared links per interface.

You can use Loop Guard to determine if a root port or an alternate/backup root port receives BPDUs. If the port that was previously receiving BPDUs is no longer receiving BPDUs, Loop Guard puts the port into an inconsistent state (blocking) until the port starts to receive BPDUs again. If such a port receives BPDUs again, the port—and link—is deemed viable again. The protocol removes the loop-inconsistent condition from the port, and the STP determines the port state because the recovery is automatic.

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

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

Enabling Loop Guard on a root device has no effect but provides protection when a root device becomes a nonroot device.

# Root Guard

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

When you enable Root Guard on an interface, this functionality applies to all VLANs to which that interface belongs.

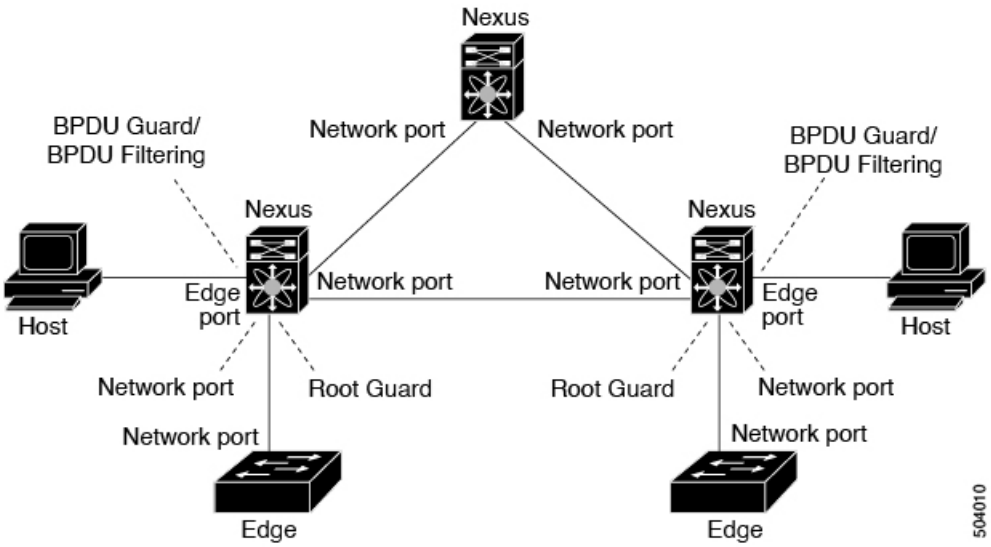
You can use Root Guard to enforce the root bridge placement in the network. Root Guard ensures that the port on which Root Guard is enabled is the designated port. Normally, root bridge ports are all designated ports, unless two or more of the ports of the root bridge are connected. If the bridge receives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

You cannot configure Root Guard globally.

# Applying STP Extension Features

***Figure 15: Network with STP Extensions Correctly Deployed***

We recommend that you configure the various STP extension features through your network as shown in this figure. Bridge Assurance is enabled on the entire network. You should enable either BPDU Guard or BPDU Filtering on the host interface.



# PVST Simulation

MST operates with no need for user configuration. The PVST simulation feature enables this interoperability.

**Note** PVST simulation is enabled by default when you enable MST. By default, all interfaces on the device operate on MST.

The root bridge for all STP instances must all be in the MST region. If the root bridge for all STP instances are not on MST, the software moves the port into a PVST simulation-inconsistent state.

**Note** We recommend that you put the root bridge for all STP instances in the MST region.

# High Availability for STP

The software supports high availability for STP. However, the statistics and timers are not restored when STP restarts. The timers start again and the statistics begin from 0.

# Prerequisites for STP Extensions

STP has the following prerequisites:

- You must be logged onto the device.
- You must have STP configured already.

# Guidelines and Limitations for Configuring STP Extensions

STP extensions have the following configuration guidelines and limitations:

- **show** commands with the **internal** keyword are not supported.
- Connect STP network ports only to switches.
- You should configure host ports as STP edge ports and not as network ports.
- If you enable STP network port types globally, ensure that you manually configure all ports connected to hosts as STP edge ports.
- You should configure all access and trunk ports connected to Layer 2 hosts as edge ports.
- Bridge Assurance runs only on point-to-point spanning tree network ports. You must configure each side of the link for this feature.
- We recommend that you enable Bridge Assurance throughout your network.

- We recommend that you enable BPDU Guard on all edge ports.
- Enabling Loop Guard globally works only on point-to-point links.
- Enabling Loop Guard per interface works on both shared and point-to-point links.
- Root Guard forces a port to always be a designated port; it does not allow a port to become a root port. Loop Guard is effective only if the port is a root port or an alternate port. You cannot enable Loop Guard and Root Guard on a port at the same time.
- Loop Guard has no effect on a disabled spanning tree instance or a VLAN.
- Spanning tree always chooses the first operational port in the channel to send the BPDUs. If that link becomes unidirectional, Loop Guard blocks the channel, even if other links in the channel are functioning properly.
- If you group together a set of ports that are already blocked by Loop Guard to form a channel, spanning tree loses all the state information for those ports and the new channel port may obtain the forwarding state with a designated role.
- If a channel is blocked by Loop Guard and the channel members go back to an individual link status, spanning tree loses all the state information. The individual physical ports may obtain the forwarding state with the designated role, even if one or more of the links that formed the channel are unidirectional.

**Note** You can enable UniDirectional Link Detection (UDLD) aggressive mode to isolate the link failure. A loop may occur until UDLD detects the failure, but Loop Guard will not be able to detect it. See the *Cisco NX-OS Series NX-OS Interfaces Configuration Guide*, for information on UDLD.

- You should enable Loop Guard globally on a switch network with physical loops.
- You should enable Root Guard on ports that connect to network devices that are not under direct administrative control.

# Default Settings for STP Extensions

This table lists the default settings for STP extensions.

***Table 10: Default STP Extension Parameters***

| Parameters | Default |
|---|---|
| Port type | Normal |
| Bridge Assurance | Enabled (on STP network ports only) |
| Global BPDU Guard | Disabled |
| BPDU Guard per interface | Disabled |
| Global BPDU Filtering | Disabled |

| Parameters | Default |
|---|---|
| BPDU Filtering per interface | Disabled |
| Global Loop Guard | Disabled |
| Loop Guard per interface | Disabled |
| Root Guard per interface | Disabled |

# Configuring STP Extensions Steps

**Note** If you are familiar with the Cisco IOS CLI, be aware that the Cisco NX-OS commands for this feature might differ from the Cisco IOS commands that you would use.

You can enable Loop Guard per interface on either shared or point-to-point links.

## Configuring Spanning Tree Port Types Globally

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

- Edge—Edge ports are connected to Layer 2 hosts and are access ports.
- Network—Network ports are connected only to Layer 2 switches or bridges and can be either access or trunk ports.
- Normal—Normal ports are neither edge ports nor network ports; they are normal spanning tree ports. These ports can be connected to any device.

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

### Before you begin

Before you configure the spanning port type, you should do the following:

- Ensure that STP is configured.
- Ensure that you are configuring the ports correctly as to the device to which the port is connected.

### Procedure

| | Command or Action | Purpose |
|---|---|---|
| **Step 1** | **config t**<br>**Example:**<br>`switch# config t`<br>`switch(config)#` | Enters configuration mode. |

| | Command or Action | Purpose |
|---|---|---|
| **Step 2** | **spanning-tree port type edge default** or **spanning-tree port type network default**<br>**Example:**<br>`switch(config)# spanning-tree port type edge default` | • **spanning-tree port type edge default**<br>Configures all access ports connected to Layer 2 hosts as edge ports. Edge ports immediately transition to the forwarding state without passing through the blocking or learning state at linkup. By default, spanning tree ports are normal port types.<br>• **spanning-tree port type network default**<br>Configures all interfaces connected to Layer 2 switches and bridges as spanning tree network ports. If you enable Bridge Assurance, it automatically runs on network ports. By default, spanning tree ports are normal port types.<br>**Note**<br>If you configure interfaces connected to Layer 2 hosts as network ports, those ports automatically move into the blocking state. |
| **Step 3** | **exit**<br>**Example:**<br>`switch(config)# exit`<br>`switch#` | Exits configuration mode. |
| **Step 4** | (Optional) **show spanning-tree summary**<br>**Example:**<br>`switch# show spanning-tree summary` | Displays the STP configuration including STP port types if configured. |
| **Step 5** | (Optional) **copy running-config startup-config**<br>**Example:**<br>`switch# copy running-config startup-config` | Copies the running configuration to the startup configuration. |

### Example

This example shows how to configure all access ports connected to Layer 2 hosts as spanning tree edge ports:

```
switch# config t
switch(config)# spanning-tree port type edge default
switch(config)# exit
switch#
```

This example shows how to configure all ports connected to Layer 2 switches or bridges as spanning tree network ports:

```
switch# config t
switch(config)# spanning-tree port type network default
switch(config)# exit
switch#
```

# Configuring Spanning Tree Edge Ports on Specified Interfaces

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

This command has four states:

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

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

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

### Before you begin

Before you configure the spanning port type, you should do the following:

- Ensure that STP is configured.
- Ensure that you are configuring the ports correctly as to the device to which the port is connected.

### Procedure

| | Command or Action | Purpose |
|---|---|---|
| **Step 1** | **config t**<br>**Example:**<br>`switch# config t`<br>`switch(config)#` | Enters configuration mode. |

| | Command or Action | Purpose |
|---|---|---|
| **Step 2** | **interface** *type slot/port*<br>**Example:**<br>`switch(config)# interface ethernet 1/4`<br>`switch(config-if)#` | Specifies the interface to configure, and enters the interface configuration mode. |
| **Step 3** | **spanning-tree port type edge**<br>**Example:**<br>`switch(config-if)# spanning-tree port type edge` | Configures the specified access interfaces to be spanning edge ports. Edge ports immediately transition to the forwarding state without passing through the blocking or learning state at linkup. By default, spanning tree ports are normal port types. |
| **Step 4** | **exit**<br>**Example:**<br>`switch(config-if)# exit`<br>`switch(config)#` | Exits interface configuration mode. |
| **Step 5** | (Optional) **show spanning-tree interface** *type slot/port* **ethernet** *x/y*<br>**Example:**<br>`switch# show spanning-tree ethernet 1/4` | Displays the STP configuration including the STP port type if configured. |
| **Step 6** | (Optional) **copy running-config startup-config**<br>**Example:**<br>`switch# copy running-config startup-config` | Copies the running configuration to the startup configuration. |

### Example

This example shows how to configure the Ethernet access interface 1/4 to be a spanning tree edge port:

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

# Configuring Spanning Tree Network Ports on Specified Interfaces

You can configure spanning tree network ports on specified interfaces.

Bridge Assurance runs only on spanning tree network ports.

This command has three states:

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

- **spanning-tree port type normal** —This command explicitly configures the port as a normal spanning tree port and Bridge Assurance cannot run on this interface.
- **no spanning-tree port type**—This command implicitly enables the port as a spanning tree network port if you define the **spanning-tree port type network default** command in global configuration mode. If you enable Bridge Assurance globally, it automatically runs on this port.

**Note** A port connected to Layer 2 hosts that is configured as a network ports automatically moves into the blocking state.

### Before you begin

Before you configure the spanning port type, you should do the following:

- Ensure that STP is configured.
- Ensure that you are configuring the ports correctly as to the device to which the port is connected.

## Procedure

| | Command or Action | Purpose |
|---|---|---|
| **Step 1** | **config t**<br>**Example:**<br>`switch# config t`<br>`switch(config)#` | Enters configuration mode. |
| **Step 2** | **interface** *type slot/port*<br>**Example:**<br>`switch(config)# interface ethernet 1/4`<br>`switch(config-if)#` | Specifies the interface to configure, and enters the interface configuration mode. |
| **Step 3** | **spanning-tree port type network**<br>**Example:**<br>`switch(config-if)# spanning-tree port type network` | Configures the specified interfaces to be spanning network ports. If you enable Bridge Assurance, it automatically runs on network ports. By default, spanning tree ports are normal port types. |
| **Step 4** | **exit**<br>**Example:**<br>`switch(config-if)# exit`<br>`switch(config)#` | Exits interface configuration mode. |
| **Step 5** | (Optional) **show spanning-tree interface** *type slot/port*<br>**Example:** | Displays the STP configuration including the STP port type if configured. |

| | Command or Action | Purpose |
|---|---|---|
| | `switch# show spanning-tree interface ethernet 1/4` | |
| **Step 6** | (Optional) **copy running-config startup-config**<br>**Example:**<br>`switch# copy running-config startup-config` | Copies the running configuration to the startup configuration. |

### Example

This example shows how to configure the Ethernet interface 1/4 to be a spanning tree network port:

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

# Enabling BPDU Guard Globally

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

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

### Before you begin

Before you configure the spanning port type, you should do the following:

- Ensure that STP is configured.
- Ensure that you are configuring the ports correctly as to the device to which the port is connected.

### Procedure

| | Command or Action | Purpose |
|---|---|---|
| **Step 1** | **config t**<br>**Example:**<br>`switch# config t`<br>`switch(config)#` | Enters configuration mode. |
| **Step 2** | **spanning-tree port type edge bpduguard default**<br>**Example:** | Enables BPDU Guard by default on all spanning tree edge ports. By default, global BPDU Guard is disabled. |

| | Command or Action | Purpose |
|---|---|---|
| | `switch(config)# spanning-tree port type edge bpduguard default` | |
| **Step 3** | **exit**<br>**Example:**<br>`switch(config)# exit`<br>`switch#` | Exits configuration mode. |
| **Step 4** | (Optional) **show spanning-tree summary**<br>**Example:**<br>`switch# show spanning-tree summary` | Displays summary STP information. |
| **Step 5** | (Optional) **copy running-config startup-config**<br>**Example:**<br>`switch# copy running-config startup-config` | Copies the running configuration to the startup configuration. |

### Example

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

```
switch# config t
switch(config)# spanning-tree port type edge bpduguard default
switch(config)# exit
switch#
```

# Enabling BPDU Guard on Specified Interfaces

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

You can configure BPDU Guard on specified interfaces as follows:

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

### Before you begin

Before you configure this feature, you should do the following:

- Ensure that STP is configured.

### Procedure

| | Command or Action | Purpose |
|---|---|---|
| **Step 1** | **config t**<br>**Example:**<br>`switch# config t`<br>`switch(config)#` | Enters configuration mode. |
| **Step 2** | **interface** *type slot/port*<br>**Example:**<br>`switch(config)# interface ethernet 1/4`<br>`switch(config-if)#` | Specifies the interface to configure, and enters the interface configuration mode. |
| **Step 3** | **spanning-tree bpduguard** {**enable** \| **disable**} or **no spanning-tree bpduguard**<br>**Example:**<br>`switch(config-if)# spanning-tree bpduguard enable` | • **spanning-tree bpduguard** {**enable** \| **disable**}<br>Enables or disables BPDU Guard for the specified spanning tree edge interface. By default, BPDU Guard is disabled on the interfaces.<br>• **no spanning-tree bpduguard**<br>Falls back to the default BPDU Guard global setting that you set for the interfaces by entering the **spanning-tree port type edge bpduguard default** command. |
| **Step 4** | **exit**<br>**Example:**<br>`switch(config-if)# exit`<br>`switch(config)#` | Exits interface mode. |
| **Step 5** | (Optional) **show spanning-tree interface** *type slot/port* **detail**<br>**Example:**<br>`switch# show spanning-tree interface ethernet detail` | Displays summary STP information. |
| **Step 6** | (Optional) **copy running-config startup-config**<br>**Example:**<br>`switch(config)# copy running-config startup-config` | Copies the running configuration to the startup configuration. |

### Example

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

```
switch# config t
switch(config)# interface ethernet 1/4
switch(config-if)# spanning-tree bpduguard enable
switch(config-if)# exit
switch(config)#
```

# Enabling BPDU Filtering Globally

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

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

**Caution** Be careful when using this command. Using this command incorrectly can cause bridging loops.

### Before you begin

Before you configure this feature, you should do the following:

- Ensure that STP is configured.
- Ensure that you have configured some spanning tree edge ports.

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

### Procedure

| | Command or Action | Purpose |
|---|---|---|
| **Step 1** | **config t**<br>**Example:**<br>`switch# config t`<br>`switch(config)#` | Enters configuration mode. |
| **Step 2** | **spanning-tree port type edge bpdufilter default**<br>**Example:**<br>`switch(config)# spanning-tree port type edge bpdufilter default` | Enables BPDU Filtering by default on all operational spanning tree edge ports. Global BPDU Filtering is disabled by default. |
| **Step 3** | **exit**<br>**Example:** | Exits configuration mode. |

| | Command or Action | Purpose |
|---|---|---|
| | `switch(config)# exit`<br>`switch#` | |
| **Step 4** | (Optional) **show spanning-tree summary**<br>**Example:**<br>`switch# show spanning-tree summary` | Displays summary STP information. |
| **Step 5** | (Optional) **copy running-config startup-config**<br>**Example:**<br>`switch# copy running-config startup-config` | Copies the running configuration to the startup configuration. |

### Example

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

```
switch# config t
switch(config)# spanning-tree port type edge bpdufilter default
switch(config)# exit
switch#
```

# Enabling BPDU Filtering on Specified Interfaces

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

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

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

This command has three states:

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

### Before you begin

Before you configure this feature, you should do the following:

- Ensure that STP is configured.

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

### Procedure

| | Command or Action | Purpose |
|---|---|---|
| **Step 1** | **config t**<br>**Example:**<br>`switch# config t`<br>`switch(config)#` | Enters configuration mode. |
| **Step 2** | **interface** *type slot/port*<br>**Example:**<br>`switch(config)# interface ethernet 1/4`<br>`switch(config-if)#` | Specifies the interface to configure, and enters the interface configuration mode. |
| **Step 3** | **spanning-tree bpdufilter** {**enable** \| **disable**} or **no spanning-tree bpdufilter**<br>**Example:**<br>`switch(config-if)# spanning-tree bpdufilter enable` | • **spanning-tree bpdufilter** {**enable** \| **disable**}<br>Enables or disables BPDU Filtering for the specified spanning tree edge interface. By default, BPDU Filtering is disabled.<br>• **no spanning-tree bpdufilter**<br>Enables BPDU Filtering on the interface if the interface is an operational spanning tree edge port and if you enter the **spanning-tree port type edge bpdufilter default** command. |
| **Step 4** | **exit**<br>**Example:**<br>`switch(config-if)# exit`<br>`switch(config)#` | Exits interface mode. |
| **Step 5** | (Optional) **show spanning-tree summary**<br>**Example:**<br>`switch# show spanning-tree summary` | Displays summary STP information. |
| **Step 6** | (Optional) **copy running-config startup-config**<br>**Example:**<br>`switch(config)# copy running-config startup-config` | Copies the running configuration to the startup configuration. |

### Example

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

```
switch# config t
switch(config)# interface ethernet 1/4
switch(config-if)# spanning-tree bpdufilter enable
switch(config-if)# exit
switch(config)#
```

# Enabling Loop Guard Globally

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

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

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

### Before you begin

Before you configure this feature, you should do the following:

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

### Procedure

| | Command or Action | Purpose |
|---|---|---|
| **Step 1** | **config t**<br>**Example:**<br>`switch# config t`<br>`switch(config)#` | Enters configuration mode. |
| **Step 2** | **spanning-tree loopguard default**<br>**Example:**<br>`switch(config)# spanning-tree loopguard default` | Enables Loop Guard by default on all spanning tree normal and network ports. By default, global Loop Guard is disabled. |
| **Step 3** | **exit**<br>**Example:**<br>`switch(config)# exit`<br>`switch#` | Exits configuration mode. |

| | Command or Action | Purpose |
|---|---|---|
| **Step 4** | (Optional) **show spanning-tree summary**<br>**Example:**<br>`switch# show spanning-tree summary` | Displays summary STP information. |
| **Step 5** | (Optional) **copy running-config startup-config**<br>**Example:**<br>`switch# copy running-config startup-config` | Copies the running configuration to the startup configuration. |

### Example

This example shows how to enable Loop Guard on all spanning tree normal or network ports:

```
switch# config t
switch(config)# spanning-tree loopguard default
switch(config)# exit
switch#
```

# Enabling Loop Guard or Root Guard on Specified Interfaces

**Note** You can run Loop Guard on spanning tree normal or network ports. You can run Root Guard on all spanning tree ports: normal, edge, or network.

You can enable either Loop Guard or Root Guard on specified interfaces.

Enabling Root Guard on a port means that port cannot become a root port, and Loop Guard prevents alternate or root ports from becoming the designated port because of a failure that could lead to a unidirectional link.

Both Loop Guard and Root Guard enabled on an interface apply to all VLANs to which that interface belongs.

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

### Before you begin

Before you configure this feature, you should do the following:

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

### Procedure

| | Command or Action | Purpose |
|---|---|---|
| **Step 1** | **config t**<br>**Example:**<br>`switch# config t`<br>`switch(config)#` | Enters configuration mode. |
| **Step 2** | **interface** *type slot/port*<br>**Example:**<br>`switch(config)# interface ethernet 1/4`<br>`switch(config-if)#` | Specifies the interface to configure, and enters the interface configuration mode. |
| **Step 3** | **spanning-tree guard** {**loop** \| **root** \| **none**}<br>**Example:**<br>`switch(config-if)# spanning-tree guard loop` | Enables or disables either Loop Guard or Root Guard for the specified interface. By default, Root Guard is disabled by default, and Loop Guard on specified ports is also disabled.<br>**Note**<br>Loop Guard runs only on spanning tree normal and network interfaces. This example shows Loop Guard is enabled on the specified interface. |
| **Step 4** | **exit**<br>**Example:**<br>`switch(config-if)# exit`<br>`switch(config)#` | Exits interface mode. |
| **Step 5** | **interface** *type slot/port*<br>**Example:**<br>`switch(config)# interface ethernet 1/10`<br>`switch(config-if)#` | Specifies the interface to configure, and enters the interface configuration mode. |
| **Step 6** | **spanning-tree guard** {**loop** \| **root** \| **none**}<br>**Example:**<br>`switch(config-if)# spanning-tree guard root` | Enables or disables either Loop Guard or Root Guard for the specified interface. By default, Root Guard is disabled by default, and Loop Guard on specified ports is also disabled.<br>The example shows Root Guard is enabled on a different interface. |
| **Step 7** | **exit**<br>**Example:**<br>`switch(config-if)# exit`<br>`switch(config)#` | Exits interface mode. |
| **Step 8** | (Optional) **show spanning-tree interface** *type slot/port* **detail** | Displays summary STP information. |

| | Command or Action | Purpose |
|---|---|---|
| | **Example:**<br>`switch# show spanning-tree interface ethernet 1/4 detail` | |
| **Step 9** | (Optional) **copy running-config startup-config**<br>**Example:**<br>`switch(config)# copy running-config startup-config` | Copies the running configuration to the startup configuration. |

### Example

This example shows how to enable Root Guard on Ethernet port 1/4:

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

# Verifying the STP Extension Configuration

To display the configuration information for the STP extensions, perform one of the following tasks:

| Command | Purpose |
|---|---|
| **show running-config spanning-tree** [**all**] | Displays information about STP. |
| **show spanning-tree summary** | Displays summary information on STP. |
| **show spanning-tree mst** *instance-id* **interface** {**ethernet** *slot/port* \| **port-channel** *channel-number*} [**detail**] | Displays MST information for the specified interface and instance. |

# Configuration Examples for STP Extension

The following example shows how to configure the STP extensions:

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

switch(config)# interface ethernet 1/1
switch(config-if)# spanning-tree port type edge
switch(config-if)# exit

switch(config)# interface ethernet 1/2
switch(config-if)# spanning-tree port type edge
```

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

# Additional References for STP Extensions -- CLI Version

### Related Documents

| Related Topic | Document Title |
|---|---|
| Layer 2 interfaces | *Cisco Nexus® 3550-T Interfaces Configuration Guide* |
| System management | *Cisco Nexus® 3550-T System Management Configuration Guide* |
| | *Cisco Nexus 3550-T NX-OS Smart Licensing Using Policy User Guide* |

### Standards

| Standards | Title |
|---|---|
| IEEE 802.1Q-2006 (formerly known as IEEE 802.1s), IEEE 802.1D-2004 (formerly known as IEEE 802.1w), IEEE 802.1D, IEEE 802.1t | — |