Configuring Optional Spanning-Tree Features

This chapter describes how to configure optional spanning-tree features on the Catalyst switches. You can configure all of these features when your switch is running the per-VLAN spanning-tree plus (PVST+). You can configure only the noted features when your switch or switch stack is running the Multiple Spanning Tree Protocol (MSTP) or the rapid per-VLAN spanning-tree plus (rapid-PVST+) protocol.

Finding Feature Information

Your software release may not support all the features documented in this module. For the latest caveats and feature information, see Bug Search Tool and the release notes for your platform and software release. To find information about the features documented in this module, and to see a list of the releases in which each feature is supported, see the feature information table at the end of this module.

Use Cisco Feature Navigator to find information about platform support and Cisco software image support. To access Cisco Feature Navigator, go to https://cfnng.cisco.com/. An account on Cisco.com is not required.

Restriction for Optional Spanning-Tree Features

- PortFast minimizes the time that interfaces must wait for spanning tree to converge, so it is effective only when used on interfaces connected to end stations. If you enable PortFast on an interface connecting to another switch, you risk creating a spanning-tree loop.

Information About Optional Spanning-Tree Features

PortFast

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

Figure 1: PortFast-Enabled Interfaces

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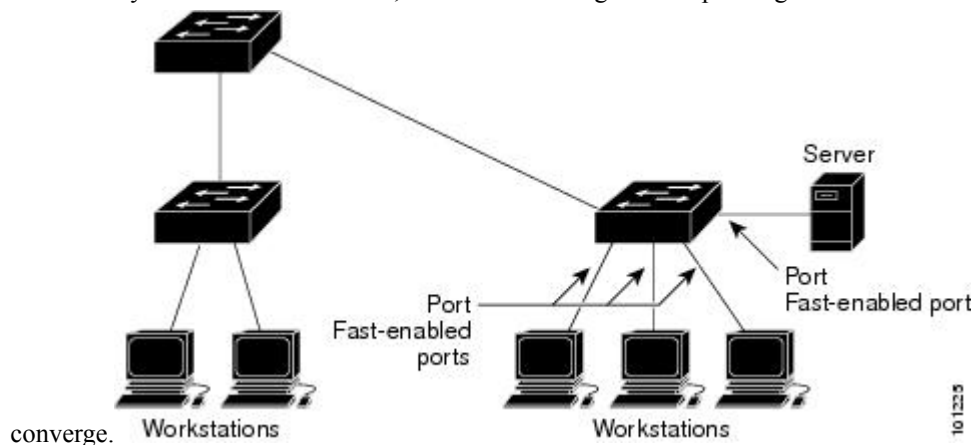
converge.

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

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

BPDU Guard

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

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

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

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

BPDU Filtering

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

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

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

⚠️

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

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

UplinkFast

Figure 2: Switches in a Hierarchical Network

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

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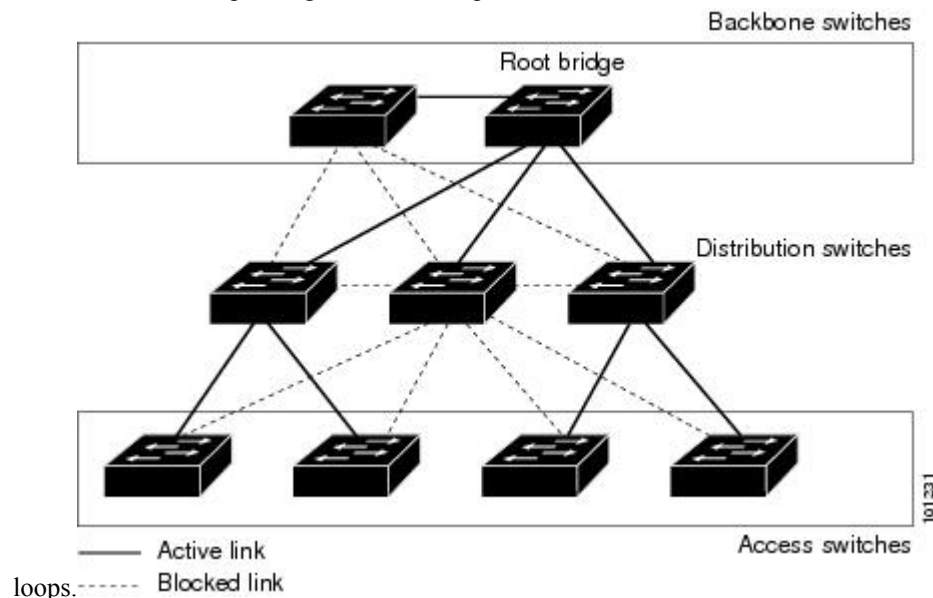
If a switch loses connectivity, it begins using the alternate paths as soon as the spanning tree selects a new root port. You can accelerate the choice of a new root port when a link or switch fails or when the spanning tree reconfigures itself by enabling UplinkFast. The root port transitions to the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

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

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

UplinkFast provides fast convergence after a direct link failure and achieves load-balancing between redundant Layer 2 links using uplink groups. An uplink group is a set of Layer 2 interfaces (per VLAN), only one of which is forwarding at any given time. Specifically, an uplink group consists of the root port (which is forwarding) and a set of blocked ports, except for self-looping ports. The uplink group provides an alternate path in case the currently forwarding link fails.

Figure 3: UplinkFast Example Before Direct Link Failure

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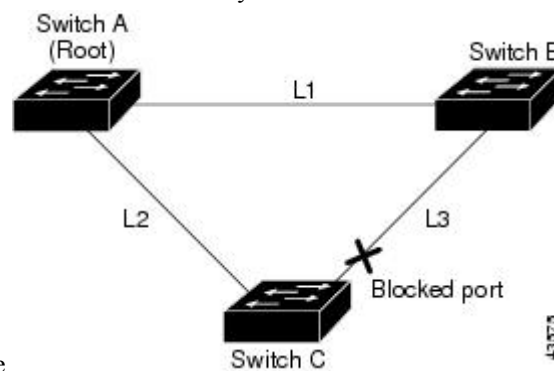
a blocking state.

Figure 4: UplinkFast Example After Direct Link Failure

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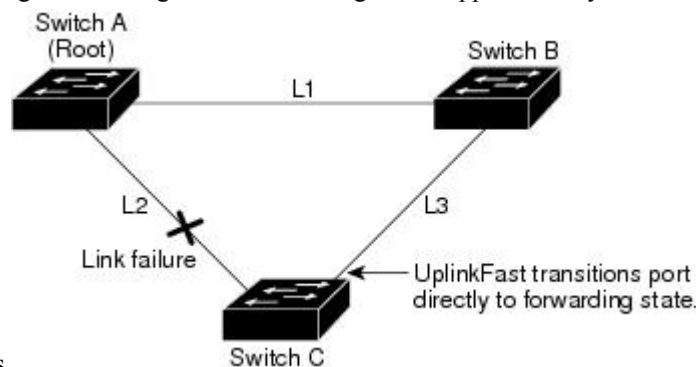
5 seconds.

Cross-Stack UplinkFast

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

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

How Cross-Stack UplinkFast Works

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

Figure 5: Cross-Stack UplinkFast Topology

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

Link 1, the root link, is in the spanning-tree forwarding state. Links 2 and 3 are alternate redundant links that are in the spanning-tree blocking state. If Switch 1 fails, if its stack-root port fails, or if Link 1 fails, CSUF selects either the alternate stack-root port on Switch 2 or Switch 3 and puts it into the forwarding state in less than 1 second.
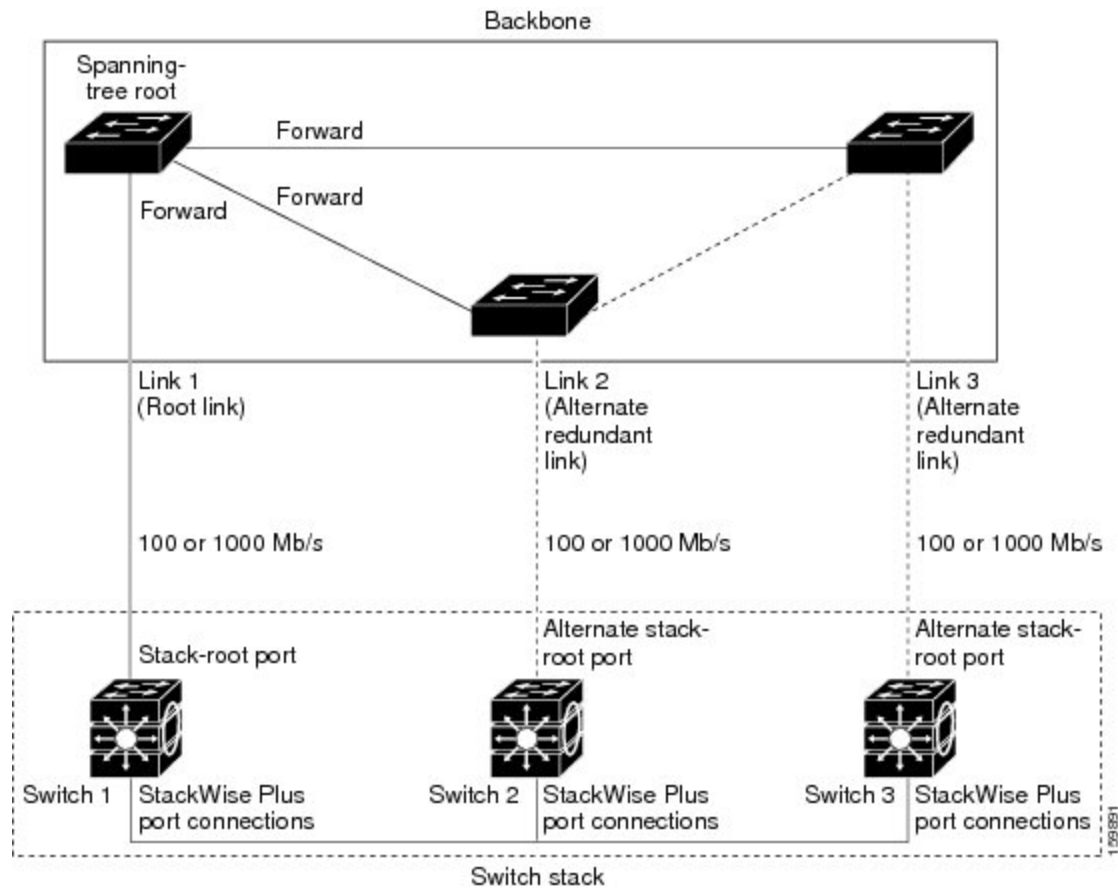
When certain link loss or spanning-tree events occur (described in the following topic), the Fast Uplink Transition Protocol uses the neighbor list to send fast-transition requests to stack members.

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

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

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

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

Events That Cause Fast Convergence

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

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

- The stack-root port link fails.

   If two switches in the stack have alternate paths to the root, only one of the switches performs the fast transition.

- The failed link, which connects the stack root to the spanning-tree root, recovers.

- A network reconfiguration causes a new stack-root switch to be selected.

- A network reconfiguration causes a new port on the current stack-root switch to be chosen as the stack-root port.

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

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

- The stack-root switch is powered off, or the software failed.

- The stack-root switch, which was powered off or failed, is powered on.

- A new switch, which might become the stack root, is added to the stack.

BackboneFast

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

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

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

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

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

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

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

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

Figure 6: BackboneFast Example Before Indirect Link Failure

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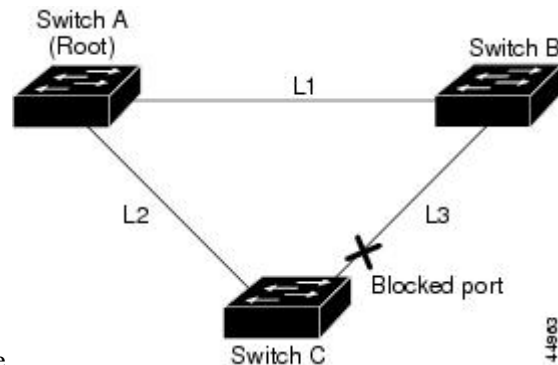
B is in the blocking state.

Figure 7: BackboneFast Example After Indirect Link Failure

If link L1 fails, Switch C cannot detect this failure because it is not connected directly to link L1. However, because Switch B is directly connected to the root switch over L1, it detects the failure, elects itself the root, and begins sending BPDUs to Switch C, identifying itself as the root. When Switch C receives the inferior BPDUs from Switch B, Switch C assumes that an indirect failure has occurred. At that point, BackboneFast allows the blocked interface on Switch C to move immediately to the listening state without waiting for the maximum aging time for the interface to expire. BackboneFast then transitions the Layer 2 interface on Switch C to the forwarding state, providing a path from Switch B to Switch A. The root-switch election takes approximately 30 seconds, twice the Forward Delay time if the default Forward Delay time of 15 seconds is

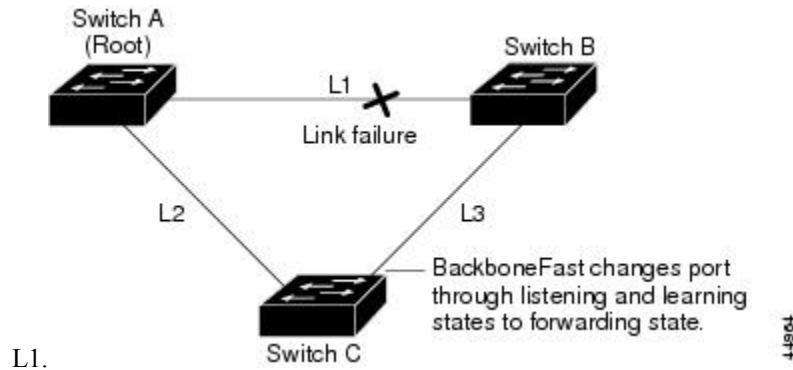set. BackboneFast reconfigures the topology to account for the failure of link

L1.

Figure 8: Adding a Switch in a Shared-Medium Topology

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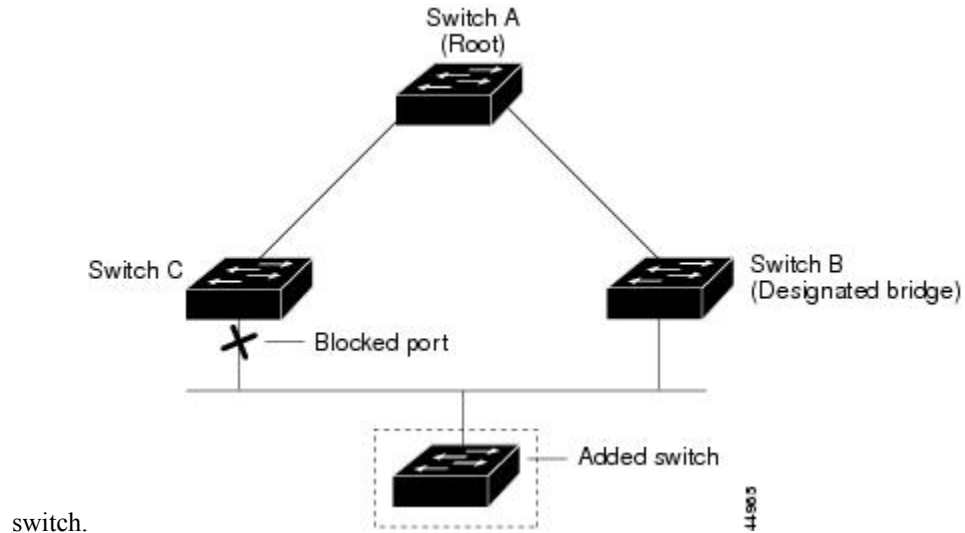
switch.

EtherChannel Guard

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

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

Root Guard

Figure 9: Root Guard in a Service-Provider Network

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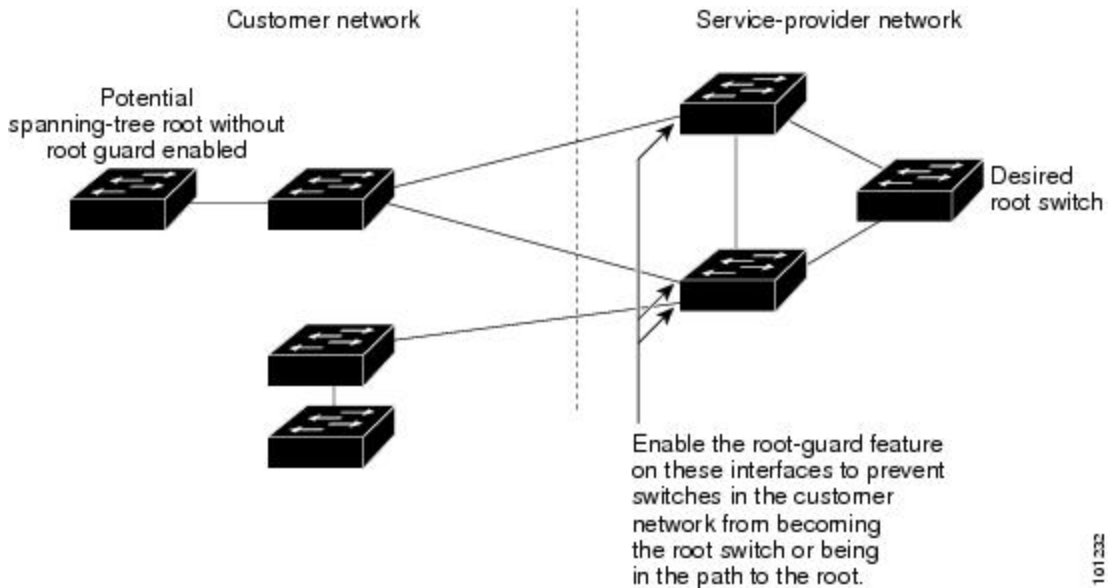
If a switch outside the SP network becomes the root switch, the interface is blocked (root-inconsistent state), and spanning tree selects a new root switch. The customer's switch does not become the root switch and is not in the path to the root.

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

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

⚠️

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

Loop Guard

You can use loop guard to prevent alternate or root ports from becoming designated ports because of a failure that leads to a unidirectional link. This feature is most effective when it is enabled on the entire switched

network. Loop guard prevents alternate and root ports from becoming designated ports, and spanning tree does not send BPDUs on root or alternate ports.

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

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

STP PortFast 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. You can configure the port type either globally or per interface.

Depending on the type of device to which the interface is connected, you can configure a spanning tree port as one of these port types:

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

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

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

- A PortFast network port—is connected only to a Layer 2 switch or bridge. Bridge Assurance is enabled only on PortFast network ports. For more information, refer to *Bridge Assurance*.

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

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

  Note   Beginning with Cisco IOS Release 15.2(4)E, or IOS XE 3.8.0E, if you enter the **spanning-tree portfast** [trunk] command in the global or interface configuration mode, the system automatically saves it as **spanning-tree portfast edge** [trunk].

Bridge Assurance

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

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

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

The following example shows how Bridge Assurance protects your network from bridging loops.

The following figure shows a network with normal STP topology.

Figure 10: Network with Normal STP Topology
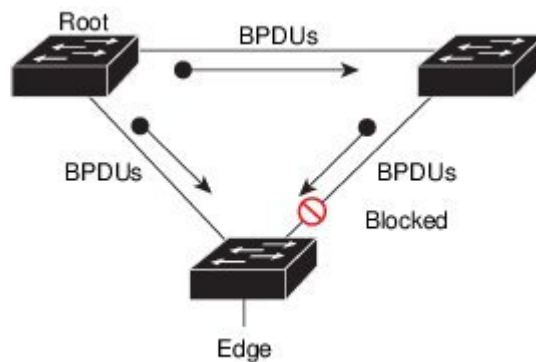
The following figure demonstrates a potential network problem when the device fails (brain dead) and Bridge Assurance is not enabled on the network.

Figure 11: Network Loop Due to a Malfunctioning Switch
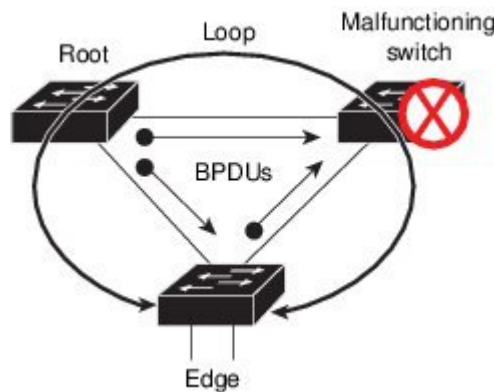
The following figure shows the network with Bridge Assurance enabled, and the STP topology progressing normally with bidirectional BDPUs issuing from every STP network port.

Figure 12: Network with STP Topology Running Bridge Assurance
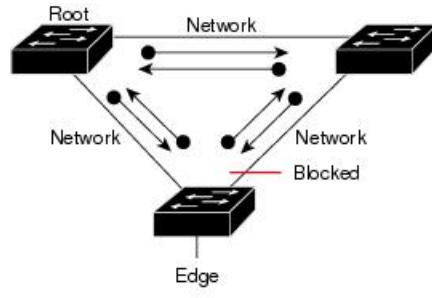
The following figure shows how the potential network problem shown in figure *Network Loop Due to a Malfunctioning Switch* does not occur when you have Bridge Assurance enabled on your network.

Figure 13: Network Problem Averted with Bridge Assurance Enabled
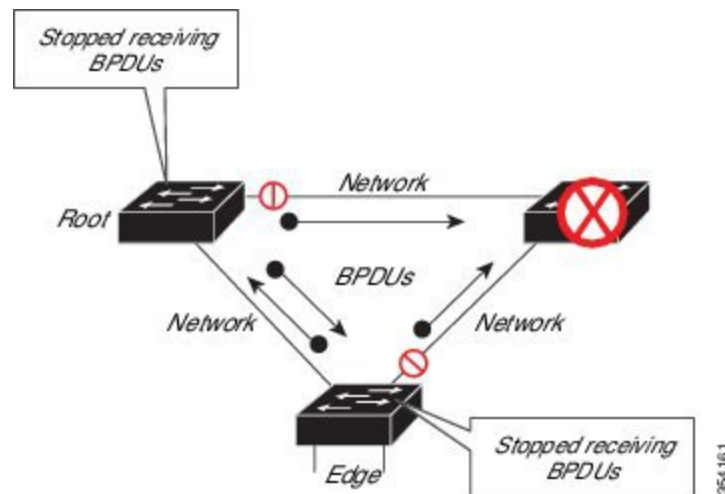
The system generates syslog messages when a port is block and unblocked. The following sample output shows the log that is generated for each of these states:

BRIDGE_ASSURANCE_BLOCK

```
Sep 17 09:48:16.249 PDT: %SPANTREE-2-BRIDGE_ASSURANCE_BLOCK: Bridge Assurance blocking port
 GigabitEthernet0/1 on VLAN0001.
```

BRIDGE_ASSURANCE_UNBLOCK

```
Sep 17 09:48:58.426 PDT: %SPANTREE-2-BRIDGE_ASSURANCE_UNBLOCK: Bridge Assurance unblocking
 port GigabitEthernet0/1 on VLAN0001.
```

Follow these guidelines when enabling Bridge Assurance:

   • It can only be enabled or disabled globally.

   • It applies to all operational network ports, including alternate and backup ports.

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

• For Bridge Assurance to work properly, it must be supported and configured on both ends of a point-to-point link. If the device on one side of the link has Bridge Assurance enabled and the device on the other side does not, the connecting port is blocked and in a Bridge Assurance inconsistent state. We recommend that you enable Bridge Assurance throughout your network.

• To enable Bridge Assurance on a port, BPDU filtering and BPDU Guard must be disabled.

• You can enable Bridge Assurance in conjunction with Loop Guard.

• You can enable Bridge Assurance in conjunction with Root Guard. The latter is designed to provide a way to enforce the root bridge placement in the network.

How to Configure Optional Spanning-Tree Features

Enabling PortFast

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

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

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

⚠

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

This procedure is optional.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **interface** *interface-id*
4. **spanning-tree portfast** [**trunk**]
5. **end**

DETAILED STEPS

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

	Command or Action	**Purpose**
	`Switch> enable`	
Step 2	**configure terminal** **Example:** `Switch# configure terminal`	Enters global configuration mode.
Step 3	**interface** *interface-id* **Example:** `Switch(config)# interface gigabitethernet 1/0/2`	Specifies an interface to configure, and enters interface configuration mode.
Step 4	**spanning-tree portfast** [**trunk**] **Example:** `Switch(config-if)# spanning-tree portfast trunk`	Enables PortFast on an access port connected to a single workstation or server. By specifying the **trunk** keyword, you can enable PortFast on a trunk port. **Note** To enable PortFast on trunk ports, you must use the **spanning-tree portfast trunk** interface configuration command. The **spanning-tree portfast** command will not work on trunk ports. Make sure that there are no loops in the network between the trunk port and the workstation or server before you enable PortFast on a trunk port. By default, PortFast is disabled on all interfaces.
Step 5	**end** **Example:** `Switch(config-if)# end`	Returns to privileged EXEC mode.

What to do next

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

Enabling BPDU Guard

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

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

This procedure is optional.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **interface** *interface-id*
4. **spanning-tree portfast edge**
5. **end**

DETAILED STEPS

	Command or Action	**Purpose**
Step 1	**enable** Example: Switch> **enable**	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** Example: Switch# **configure terminal**	Enters global configuration mode.
Step 3	**interface** *interface-id* Example: Switch(config)# **interface gigabitethernet 1/0/2**	Specifies the interface connected to an end station, and enters interface configuration mode.
Step 4	**spanning-tree portfast edge** Example: Switch(config-if)# **spanning-tree portfast edge**	Enables the PortFast edge feature.
Step 5	**end** Example: Switch(config-if)# **end**	Returns to privileged EXEC mode.

What to do next

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

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

Enabling BPDU Filtering

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

⚠️

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

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

⚠️

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

This procedure is optional.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **spanning-tree portfast edge bpdufilter default**
4. **interface** *interface-id*
5. **spanning-tree portfast edge**
6. **end**

DETAILED STEPS

	Command or Action	**Purpose**
Step 1	**enable** **Example:** Switch> **enable**	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** Switch# **configure terminal**	Enters global configuration mode.
Step 3	**spanning-tree portfast edge bpdufilter default** **Example:**	Globally enables BPDU filtering. By default, BPDU filtering is disabled.

	Command or Action	Purpose
	Switch(config)# **spanning-tree portfast edge bpdufilter default**	
Step 4	**interface** *interface-id* **Example:** Switch(config)# **interface gigabitethernet 1/0/2**	Specifies the interface connected to an end station, and enters interface configuration mode.
Step 5	**spanning-tree portfast edge** **Example:** Switch(config-if)# **spanning-tree portfast edge**	Enables the PortFast edge feature on the specified interface.
Step 6	**end** **Example:** Switch(config-if)# **end**	Returns to privileged EXEC mode.

Enabling UplinkFast for Use with Redundant Links

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

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

This procedure is optional. Follow these steps to enable UplinkFast and CSUF.

Before you begin

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

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **spanning-tree uplinkfast** [**max-update-rate** *pkts-per-second*]
4. **end**

DETAILED STEPS

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

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

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

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

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

Disabling UplinkFast

This procedure is optional.

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

Before you begin

UplinkFast must be enabled.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **no spanning-tree uplinkfast**
4. **end**

DETAILED STEPS

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

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

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

Enabling BackboneFast

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

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

This procedure is optional. Follow these steps to enable BackboneFast on the switch.

Before you begin

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

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **spanning-tree backbonefast**
4. **end**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** Switch> **enable**	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** Switch# **configure terminal**	Enters global configuration mode.
Step 3	**spanning-tree backbonefast** **Example:** Switch(config)# **spanning-tree backbonefast**	Enables BackboneFast.
Step 4	**end** **Example:** Switch(config)# **end**	Returns to privileged EXEC mode.

Enabling EtherChannel Guard

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

This procedure is optional.

Follow these steps to enable EtherChannel Guard on the switch.

SUMMARY STEPS

1. **enable**

2. **configure terminal**
3. **spanning-tree etherchannel guard misconfig**
4. **end**

DETAILED STEPS

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

What to do next

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

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

Enabling Root Guard

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

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

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

This procedure is optional.

Follow these steps to enable root guard on the switch.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **interface** *interface-id*
4. **spanning-tree guard root**
5. **end**

DETAILED STEPS

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

Enabling Loop Guard

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

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

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

This procedure is optional. Follow these steps to enable loop guard on the switch.

SUMMARY STEPS

1. Enter one of the following commands:
   - **show spanning-tree active**
   - **show spanning-tree mst**

2. **configure terminal**
3. **spanning-tree loopguard default**
4. **end**

DETAILED STEPS

	Command or Action	**Purpose**
Step 1	Enter one of the following commands: • **show spanning-tree active** • **show spanning-tree mst** **Example:** Switch# **show spanning-tree active** or Switch# **show spanning-tree mst**	Verifies which interfaces are alternate or root ports.
Step 2	**configure terminal** **Example:** Switch# **configure terminal**	Enters global configuration mode.
Step 3	**spanning-tree loopguard default** **Example:** Switch(config)# **spanning-tree loopguard default**	Enables loop guard. By default, loop guard is disabled.

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

Enabling PortFast Port Types

This section describes the different steps to enable Portfast Port types.

Configuring the Default Port State Globally

To configure the default PortFast state, perform this task:

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **spanning-tree portfast** [**edge** | **network** | **normal**] **default**
4. **end**

DETAILED STEPS

	Command or Action	**Purpose**
Step 1	**enable** **Example:** `Switch> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Switch# configure terminal`	Enters global configuration mode.
Step 3	**spanning-tree portfast** [**edge** | **network** | **normal**] **default** **Example:** `Switch(config)# spanning-tree portfast default`	Configures the default state for all interfaces on the switch. You have these options: • (Optional) **edge**—Configures all interfaces as edge ports. This assumes all ports are connected to hosts/servers. • (Optional) **network**—Configures all interfaces as spanning tree network ports. This assumes all ports are connected to switches and bridges. Bridge Assurance is enabled on all network ports by default.

	Command or Action	Purpose
		• (Optional) **normal**—Configures all interfaces normal spanning tree ports. These ports can be connected to any type of device.
		• **default**—The default port type is normal.
Step 4	**end** **Example:** Switch(config)# **end**	Returns to privileged EXEC mode.

Configuring PortFast Edge on a Specified Interface

Interfaces configured as edge ports immediately transition to the forwarding state, without passing through the blocking or learning states, on linkup.

✎

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

To configure an edge port on a specified interface, perform this task:

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **interface** *interface-id* | **port-channel** *port_channel_number*
4. **spanning-tree portfast edge** [**trunk**]
5. end
6. **show running interface** *interface-id* | **port-channel** *port_channel_number*

DETAILED STEPS

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

	Command or Action	**Purpose**
Step 3	**interface** *interface-id* | **port-channel** *port_channel_number* **Example:** Switch(config)# **interface gigabitethernet 0/1 | port-channel** *port_channel_number*	Specifies an interface to configure.
Step 4	**spanning-tree portfast edge** [**trunk**] **Example:** Switch(config-if)# **spanning-tree portfast trunk**	Enables edge behavior on a Layer 2 access port connected to an end workstation or server. • (Optional) **trunk**—Enables edge behavior on a trunk port. Use this keyword if the link is a trunk. Use this command only on ports that are connected to end host devices that terminate VLANs and from which the port should never receive STP BPDUs. Such end host devices include workstations, servers, and ports on routers that are not configured to support bridging. • Use the **no** version of the command to disable PortFast edge.
Step 5	end **Example:** Switch(config-if)# **end**	Exits configuration mode.
Step 6	**show running interface** *interface-id* | **port-channel** *port_channel_number* **Example:** Switch# **show running interface gigabitethernet 0/1| port-channel** *port_channel_number*	Verifies the configuration.

Configuring a PortFast Network Port on a Specified Interface

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

✎

Note Bridge Assurance is enabled only on PortFast network ports. For more information, refer to *Bridge Assurance*.

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

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **interface** *interface-id* \| **port-channel** *port_channel_number*
4. **spanning-tree portfast network**
5. end

6. **show running interface** *interface-id* | **port-channel** *port_channel_number*

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** Switch> **enable**	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** Switch# **configure terminal**	Enters global configuration mode.
Step 3	**interface** *interface-id*	**port-channel** *port_channel_number* **Example:** Switch(config)# **interface gigabitethernet 0/1**| **port-channel** *port_channel_number*	Specifies an interface to configure.
Step 4	**spanning-tree portfast network** **Example:** Switch(config-if)# **spanning-tree portfast network**	Enables edge behavior on a Layer 2 access port connected to an end workstation or server. • Configures the port as a network port. If you have enabled Bridge Assurance globally, it automatically runs on a spanning tree network port. • Use the **no** version of the command to disable PortFast.
Step 5	end **Example:** Switch(config-if)# **end**	Exits configuration mode.
Step 6	**show running interface** *interface-id*	**port-channel** *port_channel_number* **Example:** Switch# **show running interface gigabitethernet 0/1** **| port-channel** *port_channel_number*	Verifies the configuration.

Enabling Bridge Assurance

To configure the Bridge Assurance, perform the steps given below:

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **spanning-tree bridge assurance**
4. **end**
5. **show spanning-tree summary**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** Switch> **enable**	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** Switch# **configure terminal**	Enters global configuration mode.
Step 3	**spanning-tree bridge assurance** **Example:** Switch(config)# **spanning-tree bridge assurance**	Enables Bridge Assurance on all network ports on the switch. Bridge Assurance is enabled by default. Use the **no** version of the command to disable the feature. Disabling Bridge Assurance causes all configured network ports to behave as normal spanning tree ports.
Step 4	**end** **Example:** Switch(config)# **end**	Returns to privileged EXEC mode.
Step 5	**show spanning-tree summary** **Example:** Switch# **show spanning-tree summary**	Displays spanning tree information and shows if Bridge Assurance is enabled.

Examples

Examples: Configuring PortFast Edge on a Specified Interface

This example shows how to enable edge behavior on GigabitEthernet interface **0/1**:

```
Switch# configure terminal
Switch(config)# interface gigabitethernet0/1
Switch(config-if)# spanning-tree portfast edge
Switch(config-if)# end
Switch#
```

This example shows how to verify the configuration:

```
Switch# show running-config interface gigabitethernet0/1
Building configuration...
Current configuration:
!
interface GigabitEthernet0/1
no ip address
switchport
switchport access vlan 200
switchport mode access
spanning-tree portfast edge
end
```

This example shows how you can display that port GigabitEthernet **0/1** is currently in the edge state:

```
Switch# show spanning-tree vlan 200
VLAN0200
Spanning tree enabled protocol rstp
Root ID Priority 2
Address 001b.2a68.5fc0
Cost 3
Port 125 (GigabitEthernet5/9)
Hello Time 2 sec Max Age 20 sec Forward Delay 15 sec
Bridge ID Priority 2 (priority 0 sys-id-ext 2)
Address 7010.5c9c.5200
Hello Time 2 sec Max Age 20 sec Forward Delay 15 sec
Aging Time 0 sec
Interface Role Sts Cost Prio.Nbr Type
------------------ ---- --- -------- ------- --------------------------------
Gi0/1 Desg FWD 4 128.1 P2p Edge
```

Examples: Configuring a PortFast Network Port on a Specified Interface

This example shows how to configure GigabitEthernet interface **0/1** as a network port:

```
Switch# configure terminal
Switch(config)# interface gigabitethernet0/1
Switch(config-if)# spanning-tree portfast network
Switch(config-if)# end
Switch#
```

This example shows how to verify the configuration:

```
Switch# show running-config interface gigabitethernet0/1
Building configuration...
Current configuration:
!
interface GigabitEthernet0/1
no ip address
switchport
switchport access vlan 200
switchport mode access
spanning-tree portfast network
end
```

This example shows the output for show spanning-tree vlan

```
Switch# show spanning-tree vlan
Sep 17 09:51:36.370 PDT: %SYS-5-CONFIG_I: Configured from console by console2

VLAN0002
  Spanning tree enabled protocol rstp
  Root ID    Priority    2
             Address    7010.5c9c.5200
             This bridge is the root
             Hello Time   2 sec  Max Age 20 sec  Forward Delay 15 sec

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

Interface          Role Sts Cost      Prio.Nbr Type
------------------ ---- --- --------- -------- -------------------------------
Gi1/0/1            Desg FWD 4          128.1    P2p Edge
Po4                Desg FWD 3          128.480  P2p Network
Gi4/0/1            Desg FWD 4          128.169  P2p Edge
Gi4/0/47           Desg FWD 4          128.215  P2p Network

Switch#
```

Example: Configuring Bridge Assurance

This output shows port GigabitEthernet **0/1** has been configured as a network port and it is currently in the Bridge Assurance inconsistent state.

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

```
Switch# show spanning-tree
VLAN0010
Spanning tree enabled protocol rstp
Root ID Priority 32778
Address 0002.172c.f400
This bridge is the root
Hello Time 2 sec Max Age 20 sec Forward Delay 15 sec
Bridge ID Priority 32778 (priority 32768 sys-id-ext 10)
Address 0002.172c.f400
Hello Time 2 sec Max Age 20 sec Forward Delay 15 sec
Aging Time 300
Interface Role Sts   Cost    Prio.  Nbr    Type
--------------- ---- --- --------- -------- --------------------------------
Gi0/1    Desg BKN*4 128.270 Network, P2p *BA_Inc
```

The example shows the output for show spanning-tree summary.

```
Switch#sh spanning-tree summary
Switch is in rapid-pvst mode
Root bridge for: VLAN0001-VLAN0002, VLAN0128
EtherChannel misconfig guard         is enabled
Extended system ID                   is enabled
Portfast Default                     is network
Portfast Edge BPDU Guard Default     is disabled
Portfast Edge BPDU Filter Default    is disabled
Loopguard Default                    is enabled
```

```
PVST Simulation Default                 is enabled but inactive in rapid-pvst mode
Bridge Assurance                        is enabled
UplinkFast                              is disabled
BackboneFast                            is disabled
Configured Pathcost method used is short

Name                   Blocking Listening Learning Forwarding STP Active
---------------------- -------- --------- -------- ---------- ----------
VLAN0001                      0         0        0          5          5
VLAN0002                      0         0        0          4          4
VLAN0128                      0         0        0          4          4
---------------------- -------- --------- -------- ---------- ----------
3 vlans                       0         0        0         13         13

Switch#
```

Monitoring the Spanning-Tree Status

Table 1: Commands for Monitoring the Spanning-Tree Status

Command	Purpose
show spanning-tree active	Displays spanning-tree information on active interfaces only.
show spanning-tree detail	Displays a detailed summary of interface information.
show spanning-tree interface *interface-id*	Displays spanning-tree information for the specified interface.
show spanning-tree mst interface *interface-id*	Displays MST information for the specified interface.
show spanning-tree summary [**totals**]	Displays a summary of interface states or displays the total lines of the spanning-tree state section.
show spanning-tree mst interface *interface-id* **portfast edge**	Displays spanning-tree portfast information for the specified interface.

Additional References for Optional Spanning Tree Features

Related Documents

Related Topic	Document Title
Configuring STP Configuring MSTP Configuring Voice VLAN Commands reference	Software Configuration Guide, Cisco IOS Release 15.2(2)E (Catalyst 2960, 2960-S, 2960-SF and 2960-Plus Switches)

Standards and RFCs

Standard/RFC	Title
None	—

MIBs

MIB	MIBs Link
All the supported MIBs for this release.	To locate and download MIBs for selected platforms, Cisco IOS releases, and feature sets, use Cisco MIB Locator found at the following URL: http://www.cisco.com/go/mibs

Technical Assistance

Description	Link
The Cisco Support website provides extensive online resources, including documentation and tools for troubleshooting and resolving technical issues with Cisco products and technologies. To receive security and technical information about your products, you can subscribe to various services, such as the Product Alert Tool (accessed from Field Notices), the Cisco Technical Services Newsletter, and Really Simple Syndication (RSS) Feeds. Access to most tools on the Cisco Support website requires a Cisco.com user ID and password.	http://www.cisco.com/support