Configuring Optional STP Features

This chapter describes the Spanning Tree Protocol (STP) features supported on the switch. It also provides guidelines, procedures, and configuration examples. To configure STP, see Chapter 22, "Configuring STP and MST."

This chapter includes the following major sections:

- About Root Guard, page 25-2
- Enabling Root Guard, page 25-2
- About Loop Guard, page 25-3
- Enabling Loop Guard, page 25-5
- About EtherChannel Guard, page 25-6
- Enabling EtherChannel Guard (Optional), page 25-6
- About STP PortFast Port Types, page 25-7
- Enabling PortFast Port Types, page 25-8
- About Bridge Assurance, page 25-11
- Configuring Bridge Assurance, page 25-13
- About BPDU Guard, page 25-15
- Enabling BPDU Guard, page 25-15
- About PortFast Edge BPDU Filtering, page 25-16
- Enabling PortFast Edge BPDU Filtering, page 25-17
- About UplinkFast, page 25-19
- Enabling UplinkFast, page 25-20
- About BackboneFast, page 25-21
- Enabling BackboneFast, page 25-23

Note For complete syntax and usage information for the switch commands used in this chapter, see the Cisco IOS Command Reference Guides for the Catalyst 4500 Series Switch.

If a command is not in the *Cisco Catalyst 4500 Series Switch Command Reference* , you can locate it in the Cisco IOS Master Command List, All Releases.

About Root Guard

Spanning Tree root guard forces an interface to become a designated port, to protect the current root status and prevent surrounding switches from becoming the root switch.

When you enable root guard on a per-port basis, it is automatically applied to all of the active VLANs to which that port belongs. When you disable root guard, it is disabled for the specified port and the port automatically goes into the listening state.

When a switch that has ports with root guard enabled detects a new root, the ports enter the root-inconsistent state. The switch no longer detects a new root and its ports automatically go into the listening state.

Enabling Root Guard

To enable root guard on a Layer 2 access port (to force it to become a designated port), perform this task:

	Command	Purpose
Step 1	`Switch(config)# interface {{fastethernet | gigabitethernet | tengigabitethernet} slot/port}`	Specifies an interface to configure.
Step 2	`Switch(config-if)# [no] spanning-tree guard root`	Enables root guard. Use the **no** keyword to disable root guard.
Step 3	`Switch(config-if)# end`	Exits configuration mode.
Step 4	`Switch# show spanning-tree`	Verifies the configuration.

This example shows how to enable root guard on Fast Ethernet interface 5/8:

```
Switch(config)# interface fastethernet 5/8
Switch(config-if)# spanning-tree guard root
Switch(config-if)# end
Switch#
```

This example shows how to verify the configuration:

```
Switch# show running-config interface fastethernet 5/8
Building configuration...

Current configuration: 67 bytes
!
interface FastEthernet5/8
 switchport mode access
 spanning-tree guard root
end

Switch#
```

This example shows how to determine whether any ports are in root inconsistent state:

```
Switch# show spanning-tree inconsistentports

Name                 Interface              Inconsistency
-------------------- ---------------------- ------------------
VLAN0001             FastEthernet3/1        Root Inconsistent
VLAN0001             FastEthernet3/2        Root Inconsistent
VLAN1002             FastEthernet3/1        Root Inconsistent
VLAN1002             FastEthernet3/2        Root Inconsistent
VLAN1003             FastEthernet3/1        Root Inconsistent
VLAN1003             FastEthernet3/2        Root Inconsistent
VLAN1004             FastEthernet3/1        Root Inconsistent
VLAN1004             FastEthernet3/2        Root Inconsistent
VLAN1005             FastEthernet3/1        Root Inconsistent
VLAN1005             FastEthernet3/2        Root Inconsistent

Number of inconsistent ports (segments) in the system :10
```

About Loop Guard

Loop guard helps prevent bridging loops that could occur because of a unidirectional link failure on a point-to-point link. When enabled globally, loop guard applies to all point-to-point ports on the system. Loop guard detects root ports and blocked ports and ensures that they keep receiving BPDUs from their designated port on the segment. If a loop-guard-enabled root or blocked port stop receiving BPDUs from its designated port, it transitions to the blocking state, assuming there is a physical link error on this port. The port recovers from this state as soon as it receives a BPDU.

You can enable loop guard on a per-port basis. When you enable loop guard, 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. Disabling loop guard moves all loop-inconsistent ports to the listening state.

If you enable loop guard on a channel and the first link becomes unidirectional, loop guard blocks the entire channel until the affected port is removed from the channel. Figure 25-1 shows loop guard in a triangular switch configuration.

Figure 25-1 Triangular Switch Configuration with Loop Guard

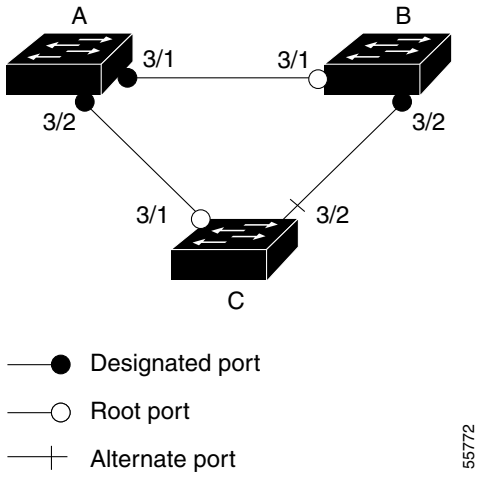


Figure 25-1 illustrates the following configuration:

- Switches A and B are distribution switches.
- Switch C is an access switch.
- Loop guard is enabled on ports 3/1 and 3/2 on Switches A, B, and C.

Enabling loop guard on a root switch has no effect but provides protection when a root switch becomes a nonroot switch.

Follow these guidelines when using loop guard:

- Do not enable loop guard on PortFast edge-enabled or dynamic VLAN ports.
- Do not enable loop guard if root guard is enabled.

Loop guard interacts with other features as follows:

- Loop guard does not affect the functionality of UplinkFast or BackboneFast.
- Enabling loop guard on ports that are not connected to a point-to-point link does not work.
- Root guard forces a port to always be the 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 uses the ports known to spanning tree. Loop guard can take advantage of logical ports provided by the Port Aggregation Protocol (PAgP). However, to form a channel, all the physical ports grouped in the channel must have compatible configurations. PAgP enforces uniform configurations of root guard or loop guard on all the physical ports to form a channel.
  - 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 a set of ports that are already blocked by loop guard are grouped together 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 breaks, 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) to help isolate the link failure. A loop may occur until UDLD detects the failure, but loop guard is not able to detect it.

- Loop guard has no effect on a disabled spanning tree instance or a VLAN.

Enabling Loop Guard

You can enable loop guard globally or per-port.

Loop Guard can be enabled only on network and normal spanning tree port types.

To enable loop guard globally on the switch, perform this task:

	Command	Purpose
Step 1	`Switch(config)# spanning-tree loopguard default`	Enables loop guard globally on the switch.
Step 2	`Switch(config)# end`	Exits configuration mode.
Step 3	`Switch# show spanning tree interface 4/4 detail`	Verifies the configuration impact on a port.

This example shows how to enable loop guard globally:

```
Switch(config)# spanning-tree loopguard default
Switch(config)# Ctrl-Z
```

This example shows how to verify the previous configuration of port 4/4:

```
Switch# show spanning-tree interface fastethernet 4/4 detail
 Port 196 (FastEthernet4/4) of VLAN0010 is forwarding
   Port path cost 1000, Port priority 160, Port Identifier 160.196.
   Designated root has priority 32768, address 00d0.00b8.140a
   Designated bridge has priority 32768, address 00d0.00b8.140a
   Designated port id is 160.196, designated path cost 0
   Timers:message age 0, forward delay 0, hold 0
   Number of transitions to forwarding state:1
   The port is in the portfast mode by portfast trunk configuration
   Link type is point-to-point by default
   Bpdu filter is enabled
   Loop guard is enabled by default on the port
   BPDU:sent 0, received 0
```

To enable loop guard on an interface, perform this task:

	Command	Purpose
Step 1	`Switch(config)# interface` {*type slot/port*} | {**port-channel** *port_channel_number*}	Selects an interface to configure.
Step 2	`Switch(config-if)# spanning-tree guard loop`	Configures loop guard.
Step 3	`Switch(config)# end`	Exits configuration mode.
Step 4	`Switch# show spanning tree interface 4/4 detail`	Verifies the configuration impact on that port.

This example shows how to enable loop guard on port 4/4:

```
Switch(config)# interface fastEthernet 4/4
Switch(config-if)# spanning-tree guard loop
Switch(config-if)# ^Z
```

This example shows how to verify the configuration impact on port 4/4:

```
Switch# show spanning-tree interface fastEthernet 4/4 detail
 Port 196 (FastEthernet4/4) of VLAN0010 is forwarding
   Port path cost 1000, Port priority 160, Port Identifier 160.196.
   Designated root has priority 32768, address 00d0.00b8.140a
   Designated bridge has priority 32768, address 00d0.00b8.140a
   Designated port id is 160.196, designated path cost 0
   Timers:message age 0, forward delay 0, hold 0
   Number of transitions to forwarding state:1
   The port is in the portfast mode by portfast trunk configuration
   Link type is point-to-point by default
   Bpdu filter is enabled
   Loop guard is enabled on the port
   BPDU:sent 0, received 0
Switch#
```

About EtherChannel Guard

EtherChannel guard allows you to detect an EtherChannel misconfiguration between the switch and a connected device. A misconfiguration can occur if the interfaces of a switch are manually configured in an EtherChannel, and one or more interfaces on the other device are not. For EtherChannel configuration guidelines, see the "EtherChannel Configuration Guidelines and Restrictions" section on page 26-6.

Note EtherChannel guard applies only to EtherChannels in forced mode (that is, manually configured) rather than through PAgP or LACP.

If the switch detects a misconfiguration on the other device, EtherChannel guard error-disables all interfaces in the EtherChannel bundle, and displays an error message.

You can enable this feature with the **spanning-tree etherchannel guard misconfig** global configuration command.

Enabling EtherChannel Guard (Optional)

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

To enable EtherChannel guard, perform this task:

	Command	Purpose
Step 1	`Switch(config)# configure terminal`	Enters global configuration mode.
Step 2	`Switch(config)# spanning-tree etherchannel guard misconfig`	Enables EtherChannel guard.
Step 3	`Switch(config)# end`	Returns to privileged EXEC mode.
Step 4	`Switch(config)# show spanning-tree summary`	Verifies your entries.
Step 5	`Switch(config)# copy running-config startup-config`	(Optional) Saves your entries in the configuration file.

To disable the EtherChannel guard feature, use the **no spanning-tree etherchannel guard misconfig** global configuration command.

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.

About 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, see About Bridge Assurance, page 25-11.

  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].

Enabling PortFast Port Types

- Configuring the PortFast Default State Globally, page 25-8
- Configuring a PortFast Edge Port on a Specified Interface, page 25-8
- Configuring a PortFast Network Port on a Specified Interface, page 25-10

Configuring the PortFast Default State Globally

To configure the default PortFast state, perform this task:

	Command	Purpose
Step 1	`Switch# configure terminal`	Enters global configuration mode.
Step 1	`Switch(config)# spanning-tree portfast [edge | network | normal] 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. • (Optional) **normal**—Configures all interfaces as normal spanning tree ports. These ports can be connected to any type of device. • **default**—The default port type is **normal**.
Step 2	`Switch(config)# end`	Exits configuration mode.

Configuring a PortFast Edge Port on a Specified Interface

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

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

	Command	Purpose
Step 1	`Switch# configure terminal`	Enters global configuration mode.
Step 2	`Switch(config)# interface {{fastethernet | gigabitethernet | tengigabitethernet} slot/port} | {port-channel port_channel_number}`	Specifies an interface to configure.

	Command	Purpose
Step 3	`Switch(config-if)# spanning-tree portfast edge [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 4	`Switch(config-if)# end`	Exits global configuration mode
Step 5	`Switch# show running interface {{fastethernet | gigabitethernet | tengigabitethernet} slot/port} | {port-channel port_channel_number}`	Verifies the configuration.
Step 6	`Switch# show spanning-tree interface {{fastethernet | gigabitethernet | tengigabitethernet} slot/port} | {port-channel port_channel_number} portfast edge`	Displays spanning-tree PortFast information for the specified interface.

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

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

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

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

```
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
------------------- ---- --- --------- -------- --------------------------------
Gi5/7    Desg FWD 4 128.1 P2p Edge
```

Configuring a PortFast Network Port on a Specified Interface

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

Note Bridge Assurance is enabled only on PortFast network ports. For more information, see About Bridge Assurance, page 25-11

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

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

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

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

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

About 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.

This example shows how Bridge Assurance protects your network from bridging loops. Here, Figure 25-2 shows a normal STP topology, and Figure 25-3 demonstrates a potential network problem when the device fails (brain dead) and Bridge Assurance is not enabled on the network.

Figure 25-2 Network with Normal STP Topology

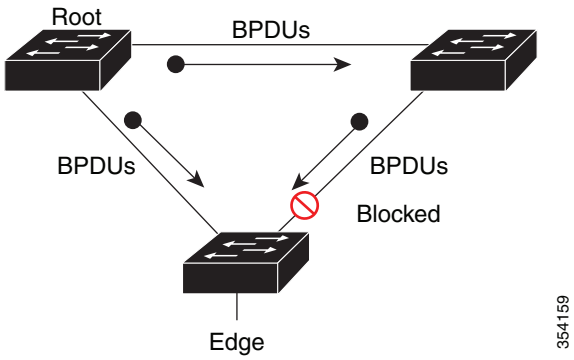


Figure 25-3 Network Loop Due to a Malfunctioning Switch

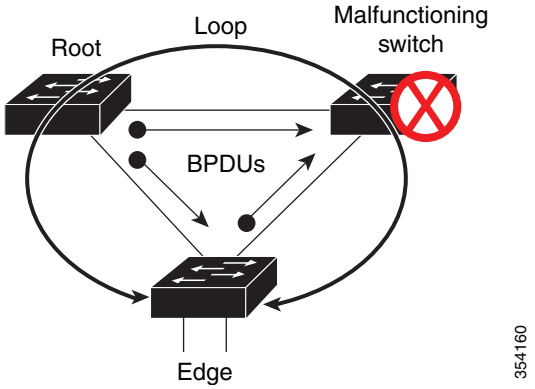


Figure 25-4 shows the network with Bridge Assurance enabled, and the STP topology progressing normally with bidirectional BDPUs issuing from every STP network port. Figure 25-5 shows how the potential network problem shown in Figure 25-3 does not occur when you have Bridge Assurance enabled on your network.

Figure 25-4 Network with STP Topology Running Bridge Assurance

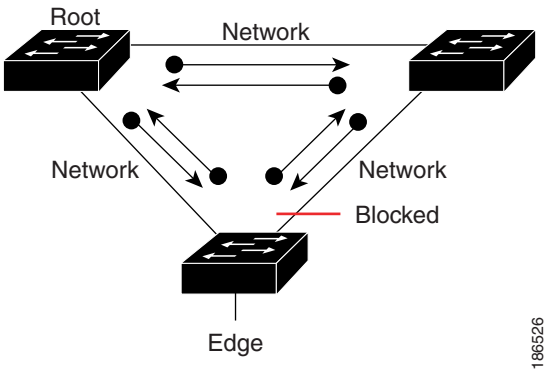
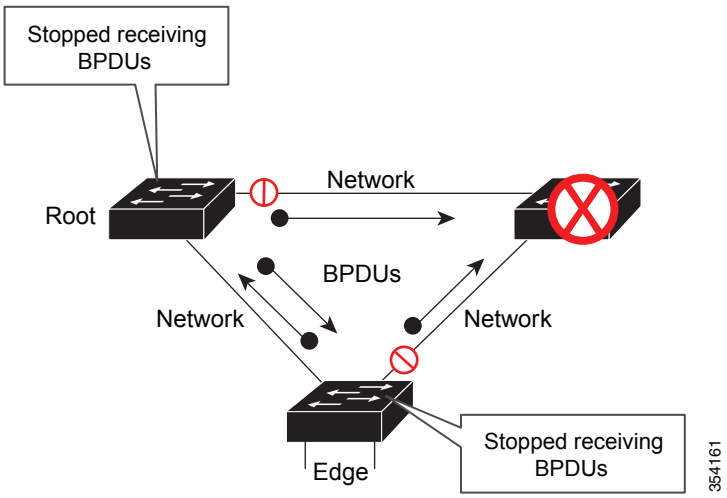


Figure 25-5 Network Problem Averted with Bridge Assurance Enabled



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

Blocked port:

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

Unblocked Port:

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

Observe these guidelines when configuring Bridge Assurance:

- It can 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, then the connecting port is blocked (a Bridge Assurance inconsistent state). We recommend that you enable Bridge Assurance throughout your network.
- To enable Bridge Assurance on a port, BPDU filtering and BPDU Guard must be disabled.
- You can enable Bridge Assurance 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.

Configuring Bridge Assurance

	Command	Purpose
Step 1	`Switch # configure terminal`	Enters the global configuration mode.
Step 2	`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 3	`Switch(config-if)# end`	Exits configuration mode.
Step 4	`Switch# show spanning-tree summary`	Displays spanning tree information and shows if Bridge Assurance is enabled

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

- Portfast Default—Network
- Bridge Assurance—Enabled

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

```
Configured Pathcost method used is short

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

This example shows how to verify if GigabitEthernet 5/8 (configured as a network port), is in a normal state. (From the **show spanning-tree summary** output above, we know that Bridge Assurance is enabled on GigabitEthernet 5/8).

```
Switch# show spanning-tree vlan
Sep 17 09:51:36.370 PDT: %SYS-5-CONFIG_I: Configured from console by console2
VLAN0200
  Spanning tree enabled protocol rstp
  Root ID    Priority    2
             Address     7010.5c9c.5200
             This bridge is the root
             Hello Time  2 sec  Max Age 20 sec  Forward Delay 15 sec

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

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

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

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

```
Switch# show spanning-tree vlan
VLAN200
  Spanning tree enabled protocol rstp
  Root ID    Priority    32778
             Address     0002.172c.f400
             This bridge is the root
             Hello Time   2 sec  Max Age 20 sec  Forward Delay 15 sec

  Bridge ID  Priority    32778  (priority 32768 sys-id-ext 10)
             Address     0002.172c.f400
             Hello Time   2 sec  Max Age 20 sec  Forward Delay 15 sec
             Aging Time 300

Interface        Role Sts Cost      Prio.Nbr Type
---------------- ---- --- --------- -------- --------------------------------
Gia5/8            Desg BKN*4         128.270  Network, P2p *BA_Inc
```

About BPDU Guard

Spanning Tree BPDU guard shuts down PortFast edge-configured interfaces that receive BPDUs, rather than putting them into the spanning tree blocking state.

When configured globally, BPDU Guard is only effective on ports in the operational PortFast edge state. In a valid configuration, PortFast edge-configured interfaces do not receive BPDUs. Reception of a BPDU by a PortFast edge-configured interface signals an invalid configuration, such as connection of an unauthorized device.

BPDU guard provides a secure response to invalid configurations, because the administrator must manually put the interface back in service.

Note When the BPDU guard feature is enabled, spanning tree applies the BPDU guard feature to all PortFast-configured interfaces. BPDU Guard shuts down that interface if a BPDU is received, regardless of the PortFast port type configuration.

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

Enabling BPDU Guard

Enabling BPDU Guard Globally

To globally enable BPDU guard on edge ports that receive BPDUs, perform this task:

	Command	Purpose
Step 1	`Switch# configure terminal`	
Step 2	`Switch(config)# spanning-tree portfast edge bpduguard default`	Enables BPDU Guard globally by default on all edge ports of the switch. Use the **no** version of the command to disable BPDU guard.
Step 3	`Switch(config)# end`	Exits configuration mode.
Step 4	`Switch# show spanning-tree summary`	Verifies the BPDU configuration.

This example shows how to enable BPDU guard:

```
Switch(config)# spanning-tree portfast edge bpduguard default
Switch(config)# end
Switch#
```

This example shows how to verify the configuration:

```
Switch# show spanning-tree summary
Root bridge for: Bridge VLAN0025
EtherChannel misconfiguration guard is enabled
```

```
Extended system ID                  is enabled
PortFast Edge BPDU Guard Default    is enabled
Portfast Edge BPDU Filter Default   is disabled
Portfast Default                    is edge
Bridge Assurance                    is enabled
Loopguard                           is disabled
UplinkFast                          is disabled
BackboneFast                        is disabled
Pathcost method used is short

Name                   Blocking Listening Learning Forwarding STP Active

---------------------- -------- --------- -------- ---------- ----------

2 vlans                     0        0         0        3          3
```

Enabling BPDU Guard on a Specified Interface

	Command	Purpose
Step 1	`Switch# configure terminal`	Enters global configuration mode.
Step 2	`Switch(config)# interface {{fastethernet | gigabitethernet | tengigabitethernet} slot/port} | {port-channel port_channel_number}`	Specifies an interface to configure.
Step 3	`Switch(config-if)# spanning-tree portfast edge bpduguard default`	Enables BPDU Guard on the specified edge port. Use the **no** keyword to disable BPDU guard.
Step 4	`Switch(config)# end`	Exits configuration mode.
Step 5	`Switch# show spanning-tree summary`	Verifies the BPDU configuration.

About PortFast Edge BPDU Filtering

Cisco IOS Release 12.2(31)SGA and later support PortFast edge BPDU filtering, which allows the administrator to prevent the system from sending or even receiving BPDUs on specified ports.

When configured globally, PortFast edge BPDU filtering applies to all operational PortFast edge ports. Ports in an operational PortFast edge state are supposed to be connected to hosts that typically drop BPDUs. If an operational PortFast edge port receives a BPDU, it immediately loses its operational PortFast edge status. In that case, PortFast edge BPDU filtering is disabled on this port and STP resumes sending BPDUs on this port.

PortFast edge BPDU filtering can also be configured on a per-port basis. When PortFast edge BPDU filtering is explicitly configured on a port, it does not send any BPDUs and drops all BPDUs it receives.

Caution Explicitly configuring PortFast edge BPDU filtering on a port that is not connected to a host can result in bridging loops, because the port ignores any BPDU it receives and goes to the forwarding state.

When you enable PortFast edge BPDU filtering globally and set this port configuration as the default for PortFast edge BPDU filtering (see the "Enabling BackboneFast" section on page 25-23), PortFast enables or disables PortFast edge BPDU filtering.

If the port configuration is not set to default, then the PortFast edge configuration does not affect PortFast edge BPDU filtering. Table 25-1 lists all the possible PortFast edge BPDU filtering combinations. PortFast edge BPDU filtering allows access ports to move directly to the forwarding state as soon as the end hosts are connected.

Table 25-1 PortFast Edge BPDU Filtering Port Configurations

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

1. The port transmits at least 10 BPDUs. If this port receives any BPDUs, then PortFast edge and PortFast edge BPDU filtering are disabled.

Enabling PortFast Edge BPDU Filtering

Enabling PortFast Edge BPDU Filtering Globally

	Command	Purpose
Step 1	`Switch# configure terminal`	Enters global configuration mode.
Step 2	`Switch(config)# spanning-tree portfast edge bpdufilter default`	Enables BPDU filtering globally by default on all edge ports of the switch. Use the no prefix to disable BPDU filtering by default on all edge ports of the switch.
Step 3	`Switch# show spanning-tree summary totals`	Verifies the BPDU configuration.

This example shows how to enable PortFast edge BPDU filtering as default on all edge ports and verify the configuration in PVST+ mode :

```
Switch(config)# spanning-tree portfast edge bpdufilter default
Switch(config)# exit

Switch# show spanning-tree summary totals
Root bridge for: Bridge VLAN0025
EtherChannel misconfiguration guard is enabled
Extended system ID                  is enabled
PortFast Edge BPDU Guard Default    is disabled
Portfast Edge BPDU Filter Default   is enabled
Portfast Default                    is edge
Bridge Assurance                    is enabled
Loopguard                           is disabled
UplinkFast                          is disabled
BackboneFast                        is disabled
Pathcost method used is long

Name                   Blocking Listening Learning Forwarding STP Active
```

```
---------------------- -------- --------- -------- ---------- ----------
2 vlans                     0        0         0        3          3
```

Note For PVST+ information, see Chapter 22, "Configuring STP and MST."

Enabling PortFast Edge BPDU Filtering on a Specified Interface

	Command	Purpose
Step 1	`Switch# configure terminal`	Enters global configuration mode.
Step 2	`Switch(config)# interface {{fastethernet | gigabitethernet | tengigabitethernet} slot/port} | {port-channel port_channel_number}`	Specifies an interface to configure.
Step 3	`Switch(config-if)# spanning-tree bpdufilter [enable | disable]`	Enables or Disables BPDU filtering.
Step 4	`Switch# show spanning-tree interface {type slot/port}`	Verifies the configuration.

This example shows how to enable PortFast edge BPDU filtering on port 4/4:

```
Switch(config)# interface fastethernet 4/4
Switch(config-if)# spanning-tree bpdufilter enable
Switch(config-if)# end
```

This example shows how to verify that PortFast edge BPDU filtering is enabled:

```
Switch# show spanning-tree interface fastethernet 4/4

Vlan             Role Sts Cost      Prio.Nbr Status
---------------- ---- --- --------- -------- --------------------------------
VLAN0010         Desg FWD 1000      160.196  Edge P2p
```

This example shows more detail on the port:

```
Switch# show spanning-tree interface fastEthernet 4/4 detail
   Port 196 (FastEthernet4/4) of VLAN0002 is forwarding
   Port path cost 4, Port priority 128, Port Identifier 128.269.
   Designated root has priority 32770, address 0002.172c.f400
   Designated bridge has priority 32770, address 0002.172c.f400
   Designated port id is 128.269, designated path cost 0
   Timers: message age 0, forward delay 0, hold 0
   Number of transitions to forwarding state: 1
   Link type is point-to-point by default
   Loop guard is enabled by default on the port
   The port is in portfast edge trunk mode
   Link type is point-to-point by default
   BPDU:sent 2183, received 0
Switch#
```

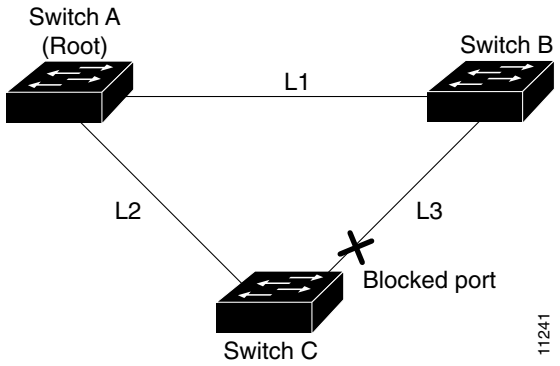
About UplinkFast

Note UplinkFast is most useful in wiring-closet switches. This feature might not be useful for other types of applications.

Spanning Tree UplinkFast provides fast convergence after a direct link failure and uses uplink groups to achieve load balancing between redundant Layer 2 links. Convergence is the speed and ability of a group of internetworking devices running a specific routing protocol to agree on the topology of an internetwork after a change in that topology. 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.

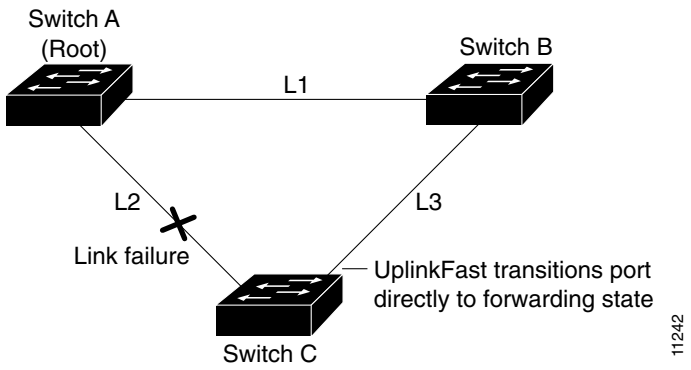
Figure 25-6 shows an example of a topology with no link failures. Switch A, the root switch, is connected directly to Switch B over link L1 and to Switch C over link L2. The Layer 2 interface on Switch C that is connected directly to Switch B is in the blocking state.

Figure 25-6 UplinkFast Before 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 port on Switch C and transitions it to the forwarding state without going through the listening and learning states, as shown in Figure 25-7. This switchover takes approximately one to five seconds.

Figure 25-7 UplinkFast After Direct Link Failure



Enabling UplinkFast

UplinkFast increases the bridge priority to 49,152 and adds 3000 to the spanning tree port cost of all interfaces on the switch, making it unlikely that the switch becomes the root switch. The *max_update_rate* value represents the number of multicast packets transmitted per second (the default is 150 packets per second [pps]).

UplinkFast cannot be enabled on VLANs that have been configured for bridge priority. To enable UplinkFast on a VLAN with bridge priority configured, restore the bridge priority on the VLAN to the default value by entering a **no spanning-tree vlan** *vlan_ID* **priority** command in global configuration mode.

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

To enable UplinkFast, perform this task:

	Command	Purpose
Step 1	`Switch(config)# [no] spanning-tree uplinkfast [max-update-rate max_update_rate]`	Enables UplinkFast. Use the **no** keyword to disable UplinkFast and restore the default rate, use the command.
Step 2	`Switch(config)# end`	Exits configuration mode.
Step 3	`Switch# show spanning-tree vlan vlan_ID`	Verifies that UplinkFast is enabled on that VLAN.

This example shows how to enable UplinkFast with a maximum update rate of 400 pps:

```
Switch(config)# spanning-tree uplinkfast max-update-rate 400
Switch(config)# exit
Switch#
```

This example shows how to verify which VLANS have UplinkFast enabled:

```
Switch# show spanning-tree uplinkfast
UplinkFast is enabled

Station update rate set to 150 packets/sec.

UplinkFast statistics
-----------------------
Number of transitions via uplinkFast (all VLANs)            :14
Number of proxy multicast addresses transmitted (all VLANs) :5308

Name                 Interface List
-------------------- ------------------------------------
VLAN1                Fa6/9(fwd), Gi5/7
VLAN2                Gi5/7(fwd)
VLAN3                Gi5/7(fwd)
VLAN4
VLAN5
VLAN6
VLAN7
VLAN8
VLAN10
VLAN15
VLAN1002             Gi5/7(fwd)
```

```
VLAN1003            Gi5/7(fwd)
VLAN1004            Gi5/7(fwd)
VLAN1005            Gi5/7(fwd)
Switch#
```

About BackboneFast

BackboneFast is a complementary technology to UplinkFast. UplinkFast is designed to quickly respond to failures on links directly connected to leaf-node switches, but it does not help with indirect failures in the backbone core. BackboneFast optimizes the topology based on the Max Age setting. It allows the default convergence time for indirect failures to be reduced from 50 seconds to 30 seconds. However, it never eliminates forward delays and offers no assistance for direct failures.

Note BackboneFast should be enabled on every switch in your network.

Sometimes a switch receives a BPDU from a designated switch that identifies the root bridge and the designated bridge as the same switch. Because this should not happen, the BPDU is considered inferior.

BPDUs are considered inferior when a link from the designated switch has lost its link to the root bridge. The designated switch transmits the BPDUs with the information that it is now the root bridge as well as the designated bridge. The receiving switch ignores the inferior BPDU for the time defined by the Max Age setting.

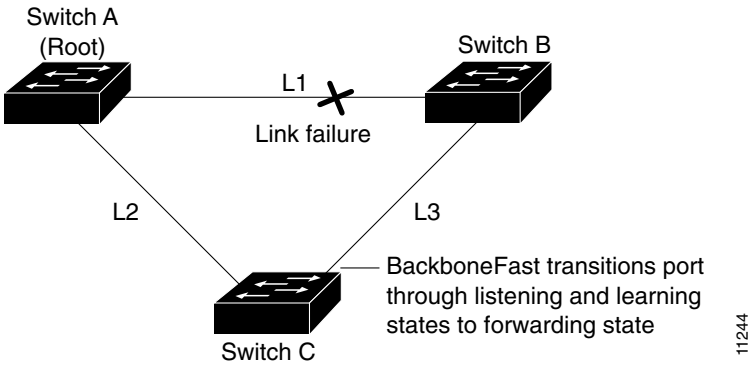
After receiving inferior BPDUs, the receiving switch tries to determine if there is an alternate path to the root bridge.

- If the port that the inferior BPDUs are received on is already in blocking mode, then the root port and other blocked ports on the switch become alternate paths to the root bridge.
- If the inferior BPDUs are received on a root port, then all presently blocking ports become the alternate paths to the root bridge. Also, if the inferior BPDUs are received on a root port and no other blocking ports exist on the switch, the receiving switch assumes that the link to the root bridge is down and the time defined by the Max Age setting expires, which turns the switch into the root switch.

If the switch finds an alternate path to the root bridge, it uses this new alternate path. This new path, and any other alternate paths, are used to send a Root Link Query (RLQ) BPDU. When BackboneFast is enabled, the RLQ BPDUs are sent out as soon as an inferior BPDU is received. This process can enable faster convergence in the event of a backbone link failure.

Figure 25-8 shows an example of a 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. In this example, because switch B has a lower priority than A but higher than C, switch B becomes the designated bridge for L3. Consequently, the Layer 2 interface on Switch C that connects directly to Switch B must be in the blocking state.

Figure 25-8 BackboneFast Before Indirect Link Failure



Next, assume that L1 fails. Switch A and Switch B, the switches directly connected to this segment, instantly know that the link is down. The blocking interface on Switch C must enter the forwarding state for the network to recover. However, because L1 is not directly connected to Switch C, Switch C does not start sending any BPDUs on L3 under the normal rules of STP until the time defined by the Max Age setting has expired.

In an STP environment without BackboneFast, if L1 should fail, 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, Switch B detects the failure and elects itself the root. Switch B begins sending configuration BDPUs to Switch C, listing itself as the root.

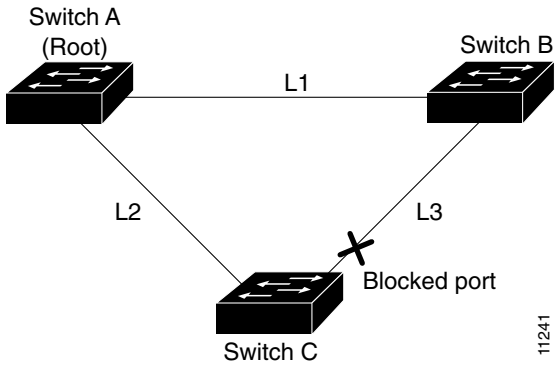
The following actions also occur when you use BackboneFast to eliminate the time defined by the Max Age setting (20-second) delay:

1. When Switch C receives the inferior configuration BPDUs from Switch B, Switch C infers that an indirect failure has occurred.
2. Switch C then sends out an RLQ.
3. Switch A receives the RLQ. Because Switch A is the root bridge, it replies with an RLQ response, listing itself as the root bridge.
4. When Switch C receives the RLQ response on its existing root port, it knows that it still has a stable connection to the root bridge. Because Switch C originated the RLQ request, it does not need to forward the RLQ response on to other switches.
5. BackboneFast allows the blocked port on Switch C to move immediately to the listening state without waiting for the time defined by the Max Age setting for the port to expire.
6. BackboneFast transitions the Layer 2 interface on Switch C to the forwarding state, providing a path from Switch B to Switch A.

This switchover takes approximately 30 seconds, twice the Forward Delay time if the default forward delay time of 15 seconds is set.

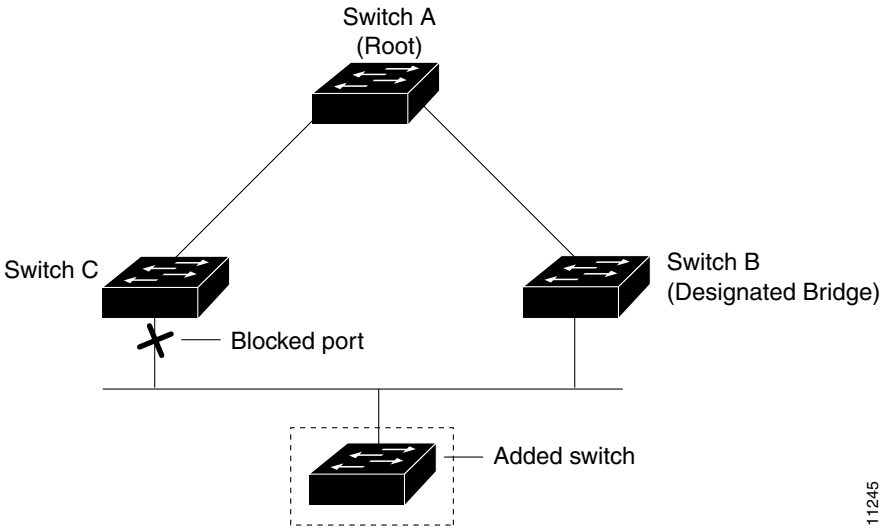
Figure 25-9 shows how BackboneFast reconfigures the topology to account for the failure of link L1.

Figure 25-9 BackboneFast after Indirect Link Failure



If a new switch is introduced into a shared-medium topology as shown in Figure 25-10, BackboneFast is not activated, because the inferior BPDUs did not come from the recognized designated bridge (Switch B). The new switch begins sending inferior BPDUs that say it is the root switch. However, the other switches ignore these inferior BPDUs, and the new switch learns that Switch B is the designated bridge to Switch A, the root switch.

Figure 25-10 Adding a Switch in a Shared-Medium Topology



Enabling BackboneFast

Note For BackboneFast to work, you must enable it on all switches in the network. BackboneFast is supported for use with third-party switches but it is not supported on Token Ring VLANs.

To enable BackboneFast, perform this task:

	Command	Purpose
Step 1	`Switch(config)# [no] spanning-tree backbonefast`	Enables BackboneFast. Use You can use the **no** keyword to disable BackboneFast.
Step 2	`Switch(config)# end`	Exits configuration mode.
Step 3	`Switch# show spanning-tree backbonefast`	Verifies that BackboneFast is enabled.

This example shows how to enable BackboneFast:

```
Switch(config)# spanning-tree backbonefast
Switch(config)# end
Switch#
```

This example shows how to verify that BackboneFast is enabled:

```
Switch# show spanning-tree backbonefast
BackboneFast is enabled

BackboneFast statistics
-----------------------
Number of transition via backboneFast (all VLANs) : 0
Number of inferior BPDUs received (all VLANs)     : 0
Number of RLQ request PDUs received (all VLANs)   : 0
Number of RLQ response PDUs received (all VLANs)  : 0
Number of RLQ request PDUs sent (all VLANs)       : 0
Number of RLQ response PDUs sent (all VLANs)      : 0
Switch#
```

This example shows how to display a summary of port states:

```
Switch# show spanning-tree summary
Switch is in rapid-pvst mode
Root bridge for:VLAN0001, VLAN1002-VLAN1005
EtherChannel misconfiguration guard is enabled
Extended system ID     is enabled
PortFast Edge BPDU Guard Defaultis disabled
Portfast Edge BPDU Filter Default is disabled
Portfast Default       is disabled
Bridge Assurance       is enabled
Loopguard Default      is disabled
UplinkFast                is disabled
BackboneFast              is disabled
Pathcost method used is short

Name                   Blocking Listening Learning Forwarding STP Active
---------------------- -------- --------- -------- ---------- ----------
VLAN0001                  0        0         0        3          3
VLAN1002                  0        0         0        2          2
VLAN1003                  0        0         0        2          2
VLAN1004                  0        0         0        2          2
VLAN1005                  0        0         0        2          2
---------------------- -------- --------- -------- ---------- ----------
5 vlans                   0        0         0        11         11

BackboneFast statistics
-----------------------
Number of transition via backboneFast (all VLANs)           :0
Number of inferior BPDUs received (all VLANs)               :0
Number of RLQ request PDUs received (all VLANs)             :0
```

```
Number of RLQ response PDUs received (all VLANs)            :0
Number of RLQ request PDUs sent (all VLANs)                 :0
Number of RLQ response PDUs sent (all VLANs)                :0
Switch#
```

This example shows how to display the total lines of the spanning tree state section:

```
Switch# show spanning-tree summary totals
Root bridge for:VLAN0001, VLAN1002-VLAN1005
Extended system ID   is disabled
PortFast Edge BPDU Guard Default    is disabled
Portfast Edge BPDU Filter Default   is enabled
Portfast Default                    is network
Bridge Assurance                    is enabled
Loopguard            is disabled by default
EtherChannel misconfiguration guard is enabled
UplinkFast           is enabled
BackboneFast         is enabled
Pathcost method used is short

Name                   Blocking Listening Learning Forwarding STP Active
---------------------- -------- --------- -------- ---------- ----------
5 vlans                   0        0         0        11         11

BackboneFast statistics
-----------------------
Number of transition via backboneFast (all VLANs)           :0
Number of inferior BPDUs received (all VLANs)               :0
Number of RLQ request PDUs received (all VLANs)             :0
Number of RLQ response PDUs received (all VLANs)            :0
Number of RLQ request PDUs sent (all VLANs)                 :0
Number of RLQ response PDUs sent (all VLANs)                :0
Switch#
```