



Configuring STP Features

This chapter describes the STP features supported on the Catalyst 4006 switch with Supervisor Engine III. It also provides guidelines, procedures, and configuration examples.

This chapter consists of the following sections:

- [Root Guard Overview, page 13-1](#)
- [PortFast Overview, page 13-2](#)
- [BPDU Guard Overview, page 13-2](#)
- [UplinkFast Overview, page 13-2](#)
- [BackboneFast Overview, page 13-3](#)
- [Enabling Root Guard, page 13-6](#)
- [Enabling PortFast, page 13-7](#)
- [Enabling BPDU Guard, page 13-8](#)
- [Enabling UplinkFast, page 13-8](#)
- [Enabling BackboneFast, page 13-10](#)



Note

For information on configuring the spanning tree protocol (STP), see [Chapter 12, “Understanding and Configuring STP.”](#)



Note

For complete syntax and usage information for the commands used in this chapter, refer to the *Command Reference for the Catalyst 4006 Switch with Supervisor Engine III* and the publications at <http://www.cisco.com/univercd/cc/td/doc/product/software/ios121/121cgcr/index.htm>

Root Guard Overview

The Spanning tree root guard feature 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(s). If a port goes into the root-inconsistent state, it automatically goes into the listening state.

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

PortFast Overview

Spanning tree PortFast causes an interface configured as a Layer 2 access port to enter the forwarding state immediately, bypassing the listening and learning states. You can use PortFast 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. If the interface receives a bridge protocol data unit (BPDU), which should not happen if the interface is connected to a single workstation or server, spanning tree puts the port into the blocking state.

**Note**

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

BPDU Guard Overview

Spanning tree BPDU guard shuts down PortFast-configured interfaces that receive BPDUs, rather than putting them into the spanning tree blocking state. In a valid configuration, PortFast-configured interfaces do not receive BPDUs. Reception of a BPDU by a PortFast-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 enabled, spanning tree applies the BPDU guard feature to all PortFast-configured interfaces.

UplinkFast Overview

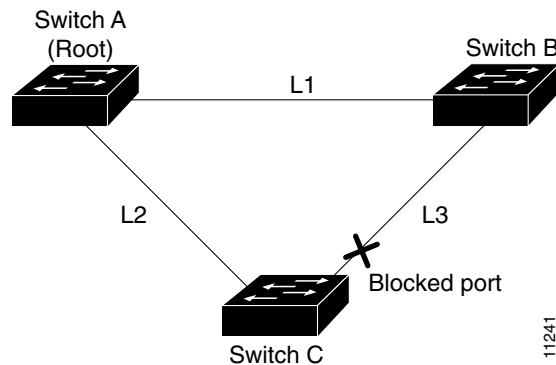
**Note**

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 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.

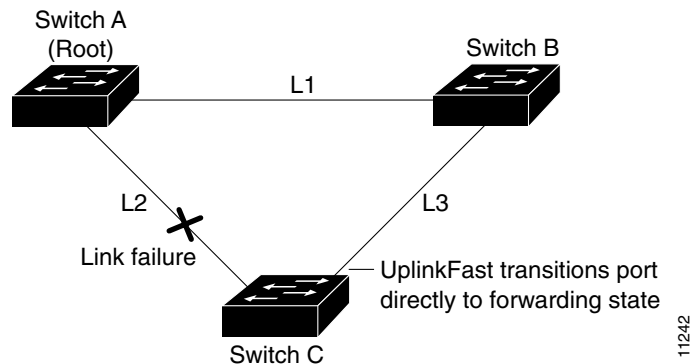
Figure 13-1 shows an example topology with no link failures. Switch A, the root switch, is connected directly to Switch B over link L1 and to Switch C over link L2. The Layer 2 interface on Switch C that is connected directly to Switch B is in the blocking state.

Figure 13-1 UplinkFast Example 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 13-2. This switchover takes approximately one to five seconds.

Figure 13-2 UplinkFast Example After Direct Link Failure



BackboneFast Overview

BackboneFast is a complementary technology to UplinkFast. Whereas UplinkFast is designed to quickly respond to failures on links directly connected to leaf-node switches, it does not help with indirect failures in the backbone core.

BackboneFast is a Max Age optimization. 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 shouldn't 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 will ignore the inferior BPDU for the Max Age time.

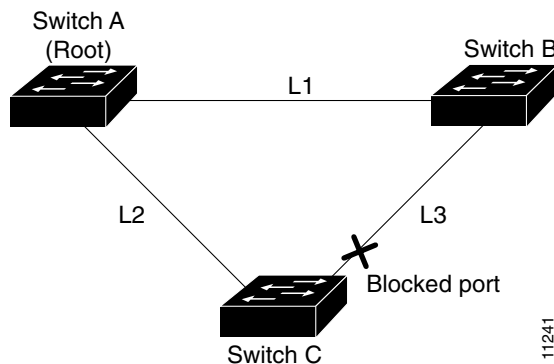
After receiving inferior BPDUs, the receiving switch will try to determine if there is an alternate path to the root bridge.

- 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 there are no other blocking ports on the switch, the receiving switch assumes that the link to the root bridge is down and the Max Age time expires, which turns the switch into the root switch.

If the switch finds an alternate path to the root bridge, it will use this new alternate path. This new path, and any other alternate paths, will be used to send a Root Link Query BPDU. By turning on BackboneFast, the Root Link Query BPDUs are sent out as soon as an inferior BPDU is received. This basically can enable faster convergence in the event of a backbone link failure.

Figure 13-3 shows 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. In this example, because switch B has a lower priority than A but higher than C, switch B becomes the designated bridge for L3. This means that the Layer 2 interface on Switch C that connects directly to Switch B must be in the blocking state.

Figure 13-3 BackboneFast Example 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. To repair the network, it is necessary that blocking interface on Switch C enter the forwarding state. 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 Max Age timer has expired.

You can use BackboneFast to eliminate the Max Age timer (20-second) delay by performing the following tasks:

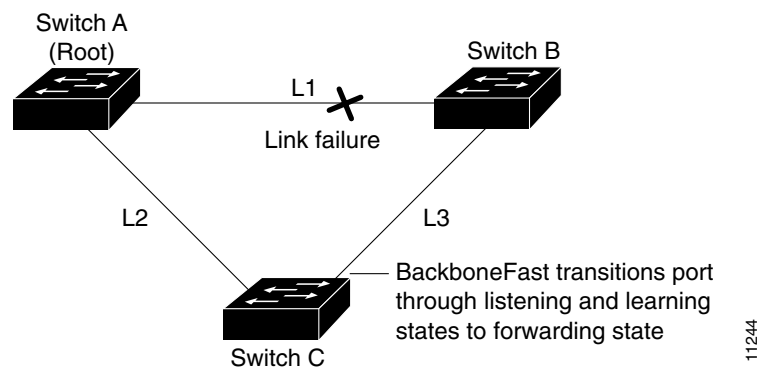
-
- Step 1** L1 fails.
 - Step 2** Switch C cannot detect this failure because it is not connected directly to link L1. However, because Switch B is directly connected to the root switch over L1, Switch B detects the failure and elects itself the root.
 - Step 3** Switch B begins sending configuration BPDUs to Switch C indicating itself as the root. This is part of the normal STP behavior. Steps 4 through 9 are specific to Backbonefast.

- Step 4** When Switch C receives the inferior configuration BPDUs from Switch B, Switch C infers that an indirect failure has occurred.
- Step 5** Switch C then sends out an RLQ Request.
- Step 6** Switch A receives the RLQ Request. Because Switch A is the root bridge, it replies with an RLQ Response listing itself as the root bridge.
- Step 7** 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
- Step 8** BackboneFast allows the blocked port on Switch C to move immediately to the listening state without waiting for the Max Age timer for the port to expire.
- Step 9** 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.

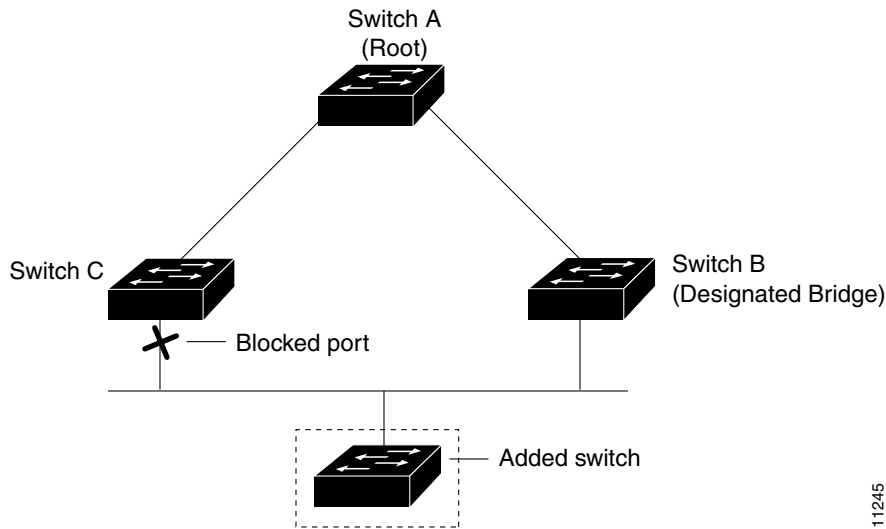
Figure 13-4 shows how BackboneFast reconfigures the topology to account for the failure of link L1.

Figure 13-4 BackboneFast Example After Indirect Link Failure



If a new switch is introduced into a shared-medium topology as shown in Figure 13-5, 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 13-5 Adding a Switch in a Shared-Medium Topology



11245

Enabling Root Guard

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

Task	Command
Step 1 Select an interface to configure.	Switch(config)# interface {{fastethernet gigabitethernet} slot/port}
Step 2 Enable root guard. Use the no keyword to disable root guard.	Switch(config-if)# [no] spanning-tree guard root
Step 3 Exit configuration mode.	Switch(config-if)# end
Step 4 Verify the configuration.	Switch# show spanning-tree

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

```
Switch# configure terminal
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 previous 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 display ports that are in the root-inconsistent state:

```
Switch# show spanning-tree vlan 1 inconsistentports
```

```
Name                Interface                Inconsistency
-----
VLAN1                FastEthernet3/47        Port Type Inconsistent
VLAN1                FastEthernet3/48        Port Type Inconsistent
```

```
Number of inconsistent ports (segments) in VLAN1 :2
```

Enabling PortFast



Caution

Use PortFast *only* when connecting a single end station to a Layer 2 access port. Otherwise, you might create a network loop.

To enable PortFast on a Layer 2 access port to force it to enter the forwarding state immediately, perform this task:

	Task	Command
Step 1	Select an interface to configure.	Switch(config)# interface {{fastethernet gigabitethernet} slot/port} {port-channel port_channel_number}
Step 2	Enable PortFast on a Layer 2 access port connected to a single workstation or server. Use the no keyword to disable PortFast.	Switch(config-if)# [no] spanning-tree portfast
Step 3	Exit configuration mode.	Switch(config-if)# end
Step 4	Verify the configuration.	Switch# show running interface {{fastethernet gigabitethernet} slot/port} {port-channel port_channel_number}

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

```
Switch# configure terminal
Switch(config)# interface fastethernet 5/8
Switch(config-if)# spanning-tree portfast
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:
!
interface FastEthernet5/8
 no ip address
 switchport
 switchport access vlan 200
 switchport mode access
 spanning-tree portfast
```

```
end
Switch#
```

Enabling BPDU Guard

To enable BPDU guard to shut down PortFast-configured interfaces that receive BPDUs, perform this task:

Task	Command
Step 1 Enable BPDU guard on all the switch's PortFast-configured interfaces. Use the no keyword to disable BPDU guard.	Switch(config)# [no] spanning-tree portfast bpduguard
Step 2 Exit configuration mode.	Switch(config)# end
Step 3 Verify the configuration.	Switch# show spanning-tree summary totals

This example shows how to enable BPDU guard:

```
Switch# configure terminal
Switch(config)# spanning-tree portfast bpduguard
Switch(config)# end
Switch#
```

This example shows how to verify the configuration:

```
Switch# show spanning-tree summary totals

Root bridge for: none.
PortFast BPDU Guard is enabled
Etherchannel misconfiguration guard is enabled
UplinkFast is disabled
BackboneFast is disabled
Default pathcost method used is short

Name                               Blocking Listening Learning Forwarding STP Active
-----
34 VLANs 0 0 0 36 36
Switch#
```

Enabling UplinkFast

UplinkFast increases the bridge priority to 49152 and adds 3000 to the spanning tree port cost of all interfaces on the switch, making it unlikely that the switch will become the root switch. However, for spanning tree with non-default bridge priority value and non-default port cost, Uplinkfast does not increase the bridge priority nor increment the port cost. The *max_update_rate* value represents the number of multicast packets transmitted per second (the default is 150 packets per second).

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:

	Task	Command
Step 1	Enable UplinkFast as follows: <ul style="list-style-type: none"> To disable UplinkFast, use the no keyword. To restore the default rate, use the no spanning-tree uplinkfast max-update-rate. 	Switch(config)# [no] spanning-tree uplinkfast [max-update-rate <i>max_update_rate</i>]
Step 2	Exit configuration mode.	Switch(config)# end
Step 3	Verify that UplinkFast is enabled.	Switch# show spanning-tree vlan <i>vlan_ID</i>

This example shows how to enable UplinkFast with an update rate of 400 packets per second:

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

This example shows how to verify that UplinkFast is 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#
```

Enabling BackboneFast


Note

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

To enable BackboneFast, perform this task:

	Task	Command
Step 1	Enable BackboneFast. Use the no keyword to disable BackboneFast.	Switch(config)# [no] spanning-tree backbonefast
Step 2	Exit configuration mode.	Switch(config)# end
Step 3	Verify that BackboneFast is enabled.	Switch# show spanning-tree backbonefast

This example shows how to enable BackboneFast:

```
Switch# configure terminal
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#
```