

Table of Contents

<u>Understanding and Configuring the Cisco UplinkFast Feature</u>	1
<u>Document ID: 10575</u>	1
<u>Introduction</u>	1
<u>Before You Begin</u>	1
<u>Conventions</u>	2
<u>Prerequisites</u>	2
<u>Components Used</u>	2
<u>Background Information</u>	2
<u>Uplink Failure Without Uplink Fast Enabled</u>	2
<u>Uplink Fast Theory of Operation</u>	3
<u>Uplink Failure With Uplink Fast Enabled</u>	4
<u>Immediate Switch Over to the Alternate Uplink</u>	4
<u>CAM Table Update</u>	5
<u>New Uplink Added</u>	7
<u>Uplink Failure Repeated After Primary Uplink is Brought Back Up</u>	8
<u>Changes Implied by Uplink Fast</u>	8
<u>Uplink Fast Feature Limitations and Interfacing with Other Features</u>	8
<u>Uplink Fast Configuration</u>	9
<u>Viewing the STP Parameter Default</u>	10
<u>Configure Uplink Fast and Check the Changes In the STP Parameters</u>	10
<u>Increase the Logging Level on Switch A In Order to See the STP Debugging Information</u>	11
<u>Unplug the Primary Uplink Between A and D1</u>	11
<u>Plug Back the Primary Uplink</u>	11
<u>Disable and Clear the Uplink Fast Feature From the Switch</u>	12
<u>Uplink Fast Configuration on Other Switches</u>	13
<u>Conclusion</u>	13
<u>Command Reference</u>	14
<u>Related Information</u>	14

Understanding and Configuring the Cisco UplinkFast Feature

Document ID: 10575

Introduction

Before You Begin

- Conventions

- Prerequisites

- Components Used

- Background Information

Uplink Failure Without Uplink Fast Enabled

Uplink Fast Theory of Operation

Uplink Failure With Uplink Fast Enabled

- Immediate Switch Over to the Alternate Uplink

- CAM Table Update

New Uplink Added

Uplink Failure Repeated After Primary Uplink is Brought Back Up

Changes Implied by Uplink Fast

Uplink Fast Feature Limitations and Interfacing with Other Features

Uplink Fast Configuration

- Viewing the STP Parameter Default

- Configure Uplink Fast and Check the Changes In the STP Parameters

- Increase the Logging Level on Switch A In Order to See the STP Debugging Information

- Unplug the Primary Uplink Between A and D1

- Plug Back the Primary Uplink

- Disable and Clear the Uplink Fast Feature From the Switch

Uplink Fast Configuration on Other Switches

Conclusion

Command Reference

Related Information

Introduction

UplinkFast is a Cisco specific feature that improves the convergence time of the Spanning-Tree Protocol (STP) in the event of the failure of an uplink. The UplinkFast feature is supported on Cisco Catalyst 4000, 5000, and 6000 series switches running CatOS. This feature is also supported on Catalyst 4000 and 6000 switches running Cisco IOS® System Software and 2900 XL/3500 XL, 2950, and 3550 series switches. The UplinkFast feature is designed to run in a switched environment when the switch has at least one alternate/backup root port (port in blocking state), that is why Cisco recommends that UplinkFast be enabled only for switches with blocked ports, typically at the access-layer. Do not use on switches without the implied topology knowledge of a alternative/backup root link typically to distribution and core switches in Cisco multilayer design.

Before You Begin

Conventions

For more information on document conventions, see the Cisco Technical Tips Conventions.

Prerequisites

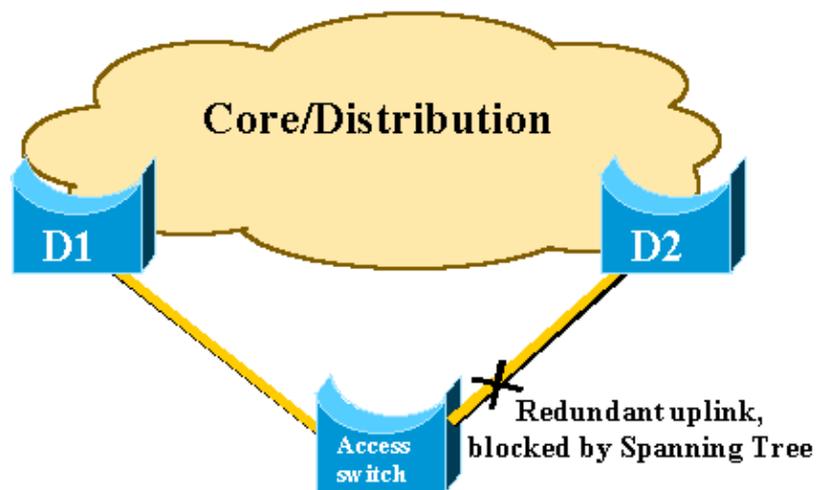
There are no specific prerequisites for this document.

Components Used

This document is not restricted to specific software and hardware versions.

Background Information

The following diagram illustrates a typical redundant network design. Users are connected to an access switch. The access switch is dually attached to two core, or distribution, switches. As the redundant uplink introduces a loop in the physical topology of the network, the Spanning-Tree Algorithm (STA) blocks it.



In the event of failure of the primary uplink to core switch D1, the STP recalculates and eventually unblocks the second uplink to switch D2, hence restoring connectivity. With the default STP parameters, the recovery takes up to 30 seconds, and with aggressive timer tuning, this lapse of time can be reduced to 14 seconds. The UplinkFast feature is a Cisco proprietary technique that reduces the recovery time further down to the order of one second.

This document details how the standard STP performs when the primary uplink fails, how UplinkFast achieves faster reconvergence than the standard reconvergence procedure, and how to configure UplinkFast. This document does not cover the basic knowledge of STP operation. To learn more about STP operation and configuration, refer to the following link:

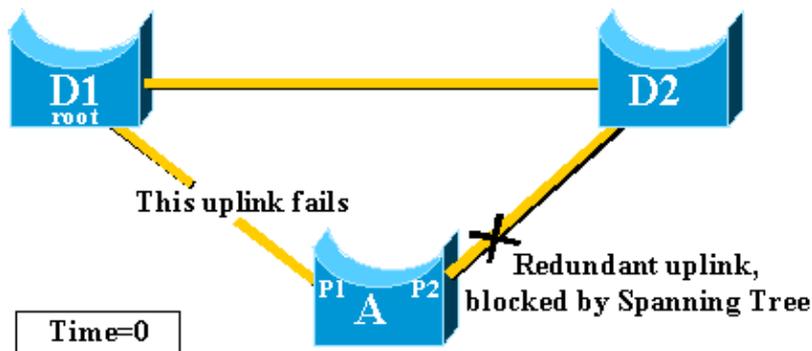
- [Understanding and Configuring Spanning Tree Protocol \(STP\) on Catalyst Switches](#)

Uplink Failure Without Uplink Fast Enabled

In this section, we look at the diagram above, which uses a minimal backbone. We inspect step-by-step the behavior of the STP in the event of uplink failure. Each step is followed with a diagram.

D1 and D2 are core switches. D1 is configured as the root bridge of the network. A is an access switch with one of its uplinks in blocking mode

1. Assume that the primary uplink from A to D1 fails.

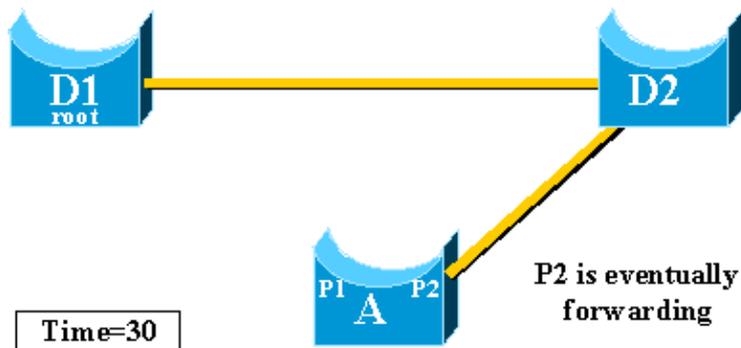


2. Port P1 goes down immediately and switch A declares its uplink to D1 as down.

Switch A considers its link to D2, which still receives BPDUs from the root, as an alternate root port. Bridge A can start transitioning port P2 from the blocking state to the forwarding state. In order to achieve this, it has to go through the listening and learning stages. Each of these stages last forward_delay (15 seconds by default), holding port P2 blocking for 30 seconds.

3. Once port P2 reaches the forwarding state, the network connectivity is re-established for hosts attached to switch A.

The network outage lasted 30 seconds.

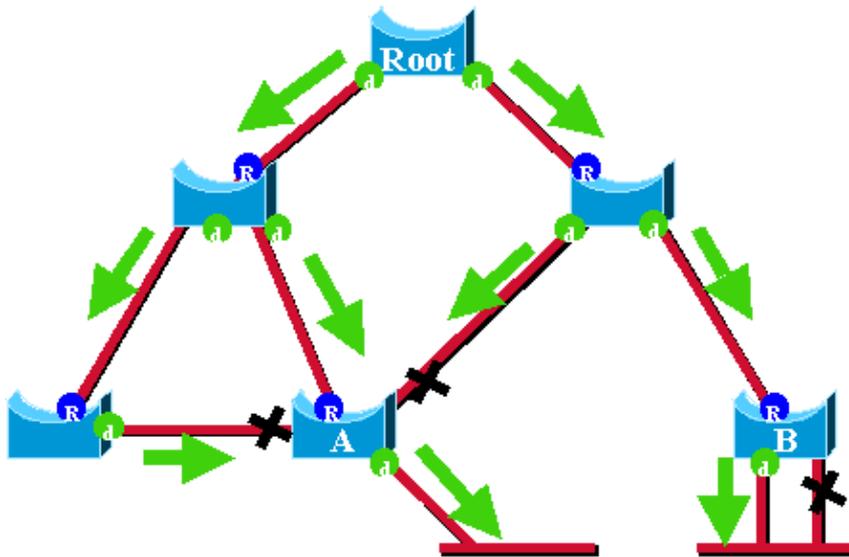


The minimum value allowed for the forward_delay timer is seven seconds. Tuning the STP parameters can lead to a recovery time of 14 seconds. This is still a noticeable delay for a user, and this kind of tuning should be done with caution. The following section of this document will show how UplinkFast dramatically reduces the downtime.

Uplink Fast Theory of Operation

The UplinkFast feature is based on the definition of an uplink group. On a given switch, the uplink group consists in the root port and all the ports that provide an alternate connection to the root bridge. If the root port is failing (that is, if the primary uplink fails), a port with next lowest cost from the uplink group is selected to immediately replace it.

The following diagram helps to explain what the UplinkFast feature is based on:



In this diagram, root ports are represented with a blue R and designated ports are represented with a green d. The green arrows represent the BPDUs generated by the root bridge and retransmitted by the bridges on their designated ports. Without entering a formal demonstration, we can determine the following about BPDUs and ports in a stable network:

- When a port is receiving a BPDU, it has a path to the root bridge. This is because BPDUs are originated from the root bridge. In this diagram, check switch A: three of its ports are receiving BPDUs, and three of its ports lead to the root bridge. The port on A that is sending BPDU is designated and not leading to the root bridge.
- On any given bridge, all ports receiving BPDUs are blocking, except the root port. A port receiving a BPDU is leading to the root bridge. If we had a bridge with two ports leading to the root bridge, we would have a bridging loop.
- A self-looped port does not provide an alternate path to the root bridge. See switch B in the diagram. Switch B's blocked port is self-looped, which means that it cannot receive its own BPDUs. In this case, the blocked port is not providing an alternate path to the root.

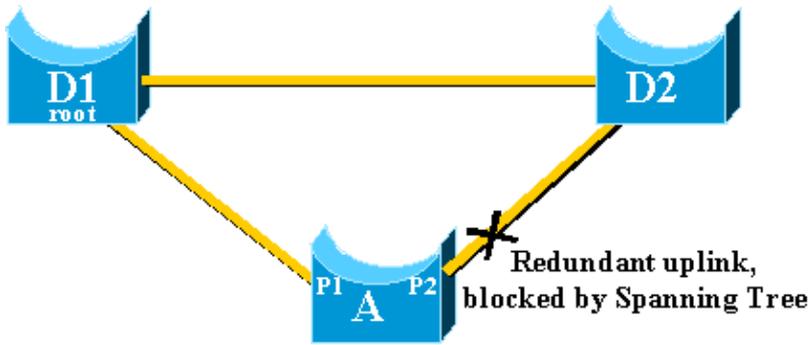
On a given bridge, the root port and all blocked ports that are not self-looped form the uplink group. The following section describes step-by-step how UplinkFast achieves fast convergence using an alternate port from this uplink group.

Note: UplinkFast is only working when the switch has blocked ports. The feature is typically designed for an access switch having redundant blocked uplinks. When you enable UplinkFast, it is enabled for the entire switch and cannot be enabled for individual VLANs.

Uplink Failure With Uplink Fast Enabled

This section details the steps for UplinkFast recovery. We will use the network diagram that was introduced at the beginning of the document.

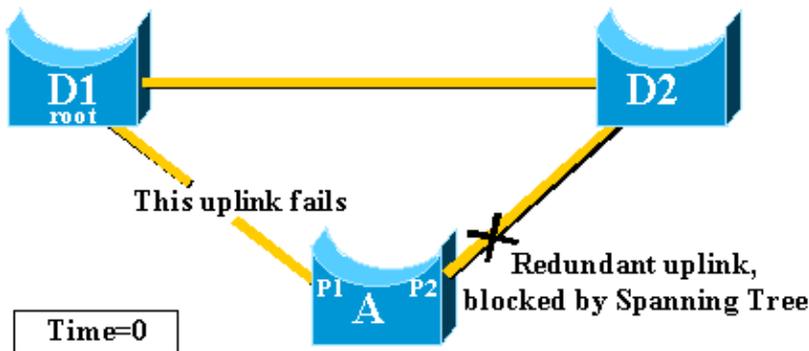
Immediate Switch Over to the Alternate Uplink



Follow these steps for an immediate switch over to the alternate uplink:

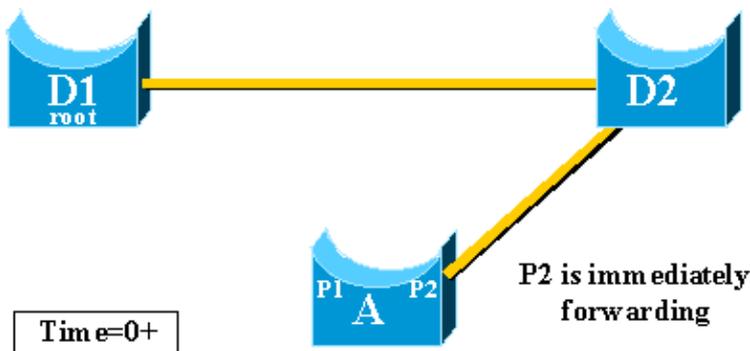
1. A's uplink group consists of P1 and its nonself-looped blocked port, P2.
2. When the link between D1 and A fails, A detects a link down on port P1.

It knows immediately that its unique path to the root bridge is lost (other paths are via the uplink group, for example, port P2, which is blocked).



3. A places port P2 in forwarding mode immediately, thus violating the standard STP procedures.

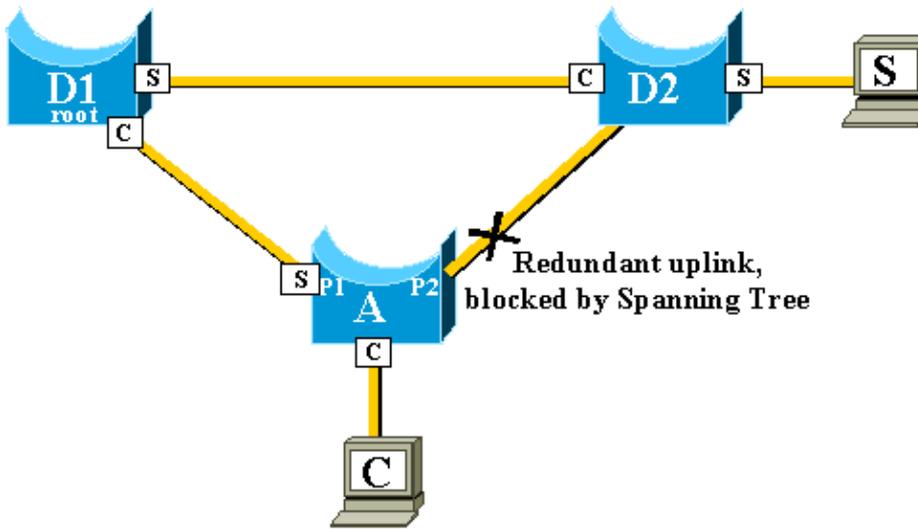
We know that there will be no loop in the network, as the only path to the root bridge is currently down. Therefore, recovery is almost immediate.



CAM Table Update

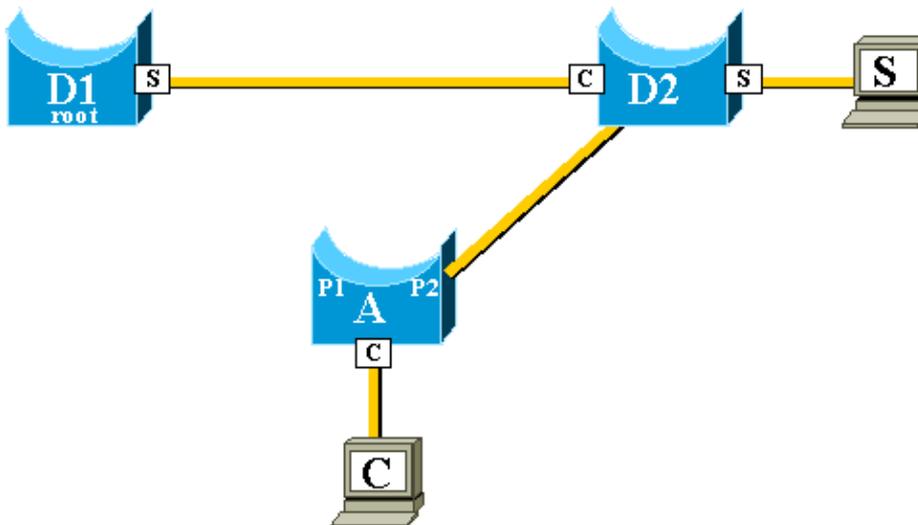
Once UplinkFast has achieved a fast-switchover between two uplinks, the Content-Addressable Memory (CAM) table in the different switches of the network can be momentarily invalid and slow down the actual convergence time.

In order to illustrate this, we have added two hosts, named S and C, to our example:



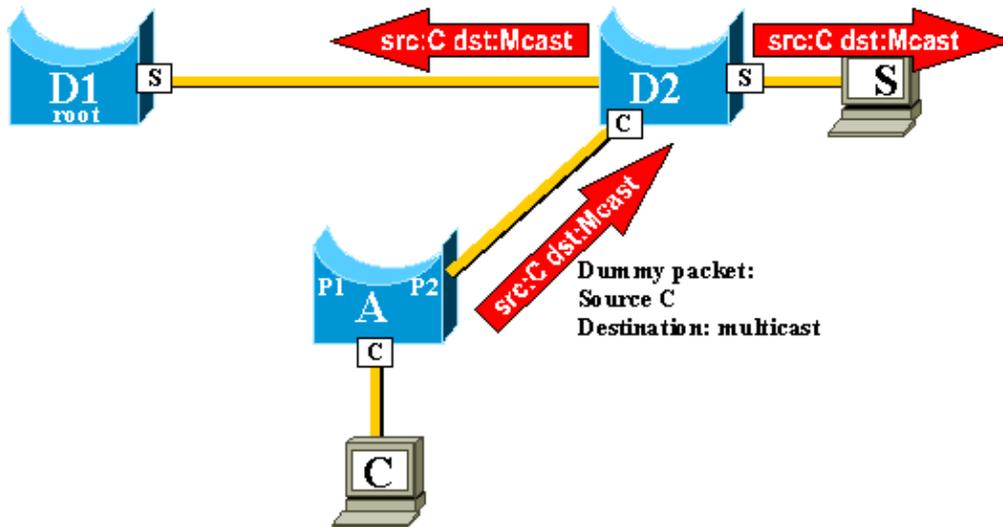
The CAM tables of the different switches are represented in the diagram. We can see that, in order to reach C, packets originated from S have to go through D2, D1, and then A.

As shown in the following diagram, the backup link has been brought up:



The backup link has been brought up so quickly, however, that the CAM tables are no longer accurate. If S sends a packet to C, it is forwarded to D1, where it is dropped. Communication between S and C is interrupted as long as the CAM table is incorrect. Even with the topology change mechanism, it can take up to 15 seconds before the problem is solved.

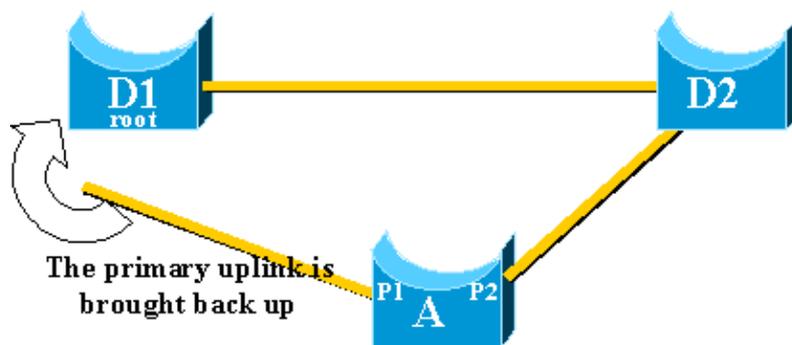
In order to solve this problem, switch A will begin to flood dummy packets with the different MAC addresses that it has in its CAM table as a source. In this case, a packet with C as a source address is generated by A. Its destination is a Cisco proprietary multicast MAC address that ensures that the packet will be flooded on the whole network and will update the necessary CAM tables on the other switches.



The rate at which the dummy multicasts are sent can be configured.

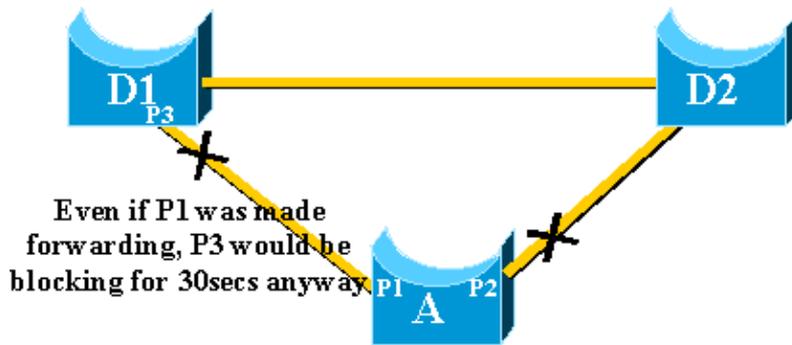
New Uplink Added

In the event of failure of the primary uplink, a replacement is immediately selected within the uplink group. What happens when a new port comes up, and this port, according to STP rules, should rightfully become the new primary uplink (root port)? An example of this is when the original root port P1 on switch A goes down, port P2 takes over, but then port P1 on switch A comes back up. Port P1 has the right to regain the root port function. Should UplinkFast immediately allow port P1 to take over and put P2 back in blocking mode?



No. An immediate switchover to port P1, which would immediately block port P2 and put port P1 in forwarding mode, is not wanted, for the following reasons:

- **Stability:** if the primary uplink is flapping, it is better to not introduce instability in the network by re-enabling it immediately. We can afford to keep our existing uplink temporarily.
- The only thing UplinkFast could do is to move port P1 in forwarding mode as soon as it is up. The problem is that the remote port on D1 is also going up and will follow the usual STP rules.



Immediately blocking port P2 and moving port P1 to forwarding will not help in this case. Port P3 would not forward before going through the listening and learning stages, which take 15 seconds each by default.

The best solution is to keep the current uplink active and hold port P1 blocked until port P3 begins forwarding. The switchover between port P1 and port P2 is then delayed by $2 * \text{forward_delay} + 5$ seconds (which is 35 seconds by default). The five seconds leave time for other protocols to negotiate (DTP of Etherchannel, for example).

Uplink Failure Repeated After Primary Uplink is Brought Back Up

When the primary uplink comes back up, it is first kept blocked for about 35 seconds by uplinkfast, before being immediately switched to a forwarding state, as was explained previously. This port will not be able to do another uplinkfast transition for roughly the same period of time. The idea is to protect against a flapping uplink that would keep triggering UplinkFast to often, and would cause too many dummy multicasts to be flooded through the network

Changes Implied by Uplink Fast

In order to be effective, the feature needs to have blocked ports providing redundant connectivity to the root. As soon as Uplink Fast is configured on a switch, switch automatically adjusts some STP parameters are adjusted to help achieve this:

- The bridge priority of the switch is increased to a significantly higher value than the default. This ensures that the switch is not likely to be elected root bridge, which does not have any root ports (all ports are designated).
- All the ports of the switch have their cost increased by 3000. This ensures that switch ports are not likely be elected designated ports.



Warning: Be careful before configuring Uplink Fast feature because the automatic changes of STP parameters can change the current STP topology.

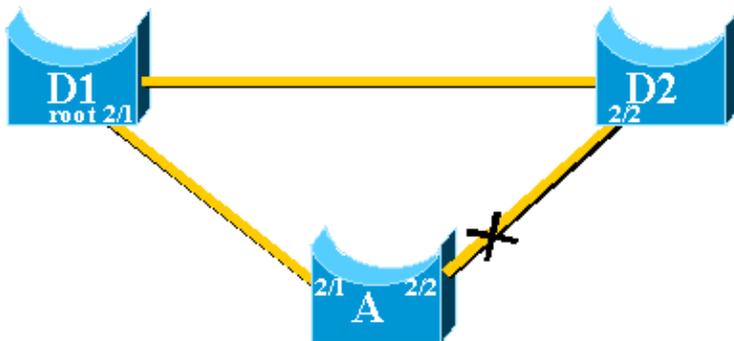
Uplink Fast Feature Limitations and Interfacing with Other Features

Sometimes a Switching hardware or software feature will cause the UplinkFast feature not to function properly. Below are some examples these limitations.

- Uplink fast does not do the fast transition during a High Availability supervisor switchover on 6500/6000 switches running CatOS. When the root port is lost on failed–resetting supervisor, the situation after a switchover is similar to when the switch boots up the first time because we do not sync the root port information between Supervisors. High Availability (HA) maintains only spanning tree port state, not the root port information, so when the HA switchover occurs, the new sup has no idea that it has lost a port on one of the uplink ports of the failed supervisor. A common workaround is the use of a port channel (EtherChannel). Root port status is maintained when a Port Channel is built across both supervisors (1/1–2/1 or 1/2–2/2, for example) or root port is on the port of any Line Card. As no spanning tree topology change occurs when failing–resetting the active supervisor, no UplinkFast transition is necessary.
- Uplink fast does not do the fast transition during an RPR or RPR+ switchover on a 6500/6000 switch running Cisco IOS System Software. There is no workaround because L2 port must go through spanning tree convergence states (listening, learning, forwarding) .
- Cisco IOS Software on 6500/6000 does not implement generation of dummy multicast packets after uplinkfast transition for updating CAM tables, refer to Cisco bug ID CSCdm65881.
- Uplink fast implementation on gigastack of 2900/3500XL/2950/3550 is called CSUF (Cross Stack Uplink Fast Feature), general UplinkFast feature on gigastack setup is not supported. CSUF does not implement generation of dummy multicast packets after UplinkFast transition for updating CAM tables.
- Do not change spanning tree priority on the switch when UplinkFast is enabled because, depending on platform, it can cause UplinkFast feature to be disabled, or it can cause a loop as the UplinkFast feature automatically changes the priority to a higher value to prevent the switch from becoming Root Bridge.

Uplink Fast Configuration

This section gives a step–by–step example of UplinkFast configuration and operation. We will use the following network diagram:



Switches A, D1, and D2 are all Catalyst 5000s, running 5.5 CatOS software. We will focus on switch A, while performing the following:

- Viewing the STP Parameter Default
- Configure UplinkFast and Check the Changes In the STP Parameters
- Increase the Logging Level on Switch A In Order to See the STP Debugging Information
- Unplug the Primary Uplink Between A and D1
- Plug Back the Primary Uplink
- Disable and Clear the UplinkFast Feature From the Switch

Viewing the STP Parameter Default

Below are the default parameters that are set for the STP on our access switch A:

Note: Port 2/2 is correctly blocking, the current cost value for the ports depends on the bandwidth (100 for an Ethernet port, 19 for a Fast Ethernet port), and the priority of the bridge is the default 32768.

```
>(enable) show spantree
VLAN 1
Spanning tree enabled
Spanning tree type          ieee

Designated Root             00-40-0b-cd-b4-09
Designated Root Priority     8192
Designated Root Cost        100
Designated Root Port        2/1
Root Max Age 20 sec  Hello Time 2 sec  Forward Delay 15 sec

Bridge ID MAC ADDR          00-90-d9-5a-a8-00
Bridge ID Priority          32768
Bridge Max Age 20 sec  Hello Time 2 sec  Forward Delay 15 sec

Port              Vlan Port-State      Cost  Priority  Portfast  Channel_id
-----
1/1                1  not-connected      19    32 disabled  0
1/2                1  not-connected      19    32 disabled  0
2/1                1  forwarding         100   32 disabled  0
2/2                1  blocking           100   32 disabled  0
2/3                1  not-connected      100   32 disabled  0
2/4                1  not-connected      100   32 disabled  0
2/5                1  not-connected      100   32 disabled  0
<snip>
```

Configure Uplink Fast and Check the Changes In the STP Parameters

We will enable UplinkFast on A using the command `set spantree uplink fast enable`. The following parameters are set:

```
A>(enable) set spantree uplinkfast enable
VLANs 1-1005 bridge priority set to 49152.
The port cost and portvlancost of all ports set to above 3000.
Station update rate set to 15 packets/100ms.
uplinkfast all-protocols field set to off.
uplinkfast enabled for bridge.
```

Using the `show spantree` command, we can see the main changes: the priority of the bridge has increased to 49152, and the cost of the ports has increased by 3000:

```
A>(enable) show spantree
VLAN 1
Spanning tree enabled
Spanning tree type          ieee

Designated Root             00-40-0b-cd-b4-09
Designated Root Priority     8192
Designated Root Cost        3100
Designated Root Port        2/1
Root Max Age 20 sec  Hello Time 2 sec  Forward Delay 15 sec

Bridge ID MAC ADDR          00-90-d9-5a-a8-00
```

```

Bridge ID Priority          49152
Bridge Max Age 20 sec      Hello Time 2 sec      Forward Delay 15 sec

Port                      Vlan Port-State      Cost  Priority Portfast  Channel_id
-----
1/1                       1    not-connected      3019    32 disabled  0
1/2                       1    not-connected      3019    32 disabled  0
2/1                       1    forwarding         3100    32 disabled  0
2/2                       1    blocking           3100    32 disabled  0
<snip>

```

Increase the Logging Level on Switch A In Order to See the STP Debugging Information

Using the **set logging level** command, we increase the logging level for the STP, so that we can have detailed information displayed on the screen during our test:

```

A>(enable) set logging level spantree 7
System logging facility for this session set to severity 7(debugging)
A>(enable)

```

Unplug the Primary Uplink Between A and D1

At this stage, we unplug the cable between A and D1. In the same second, you can see port 2/1 going down and port 2/2 being transferred immediately into forwarding mode:

```

2000 Nov 21 01:34:55 %SPANTREE-5-UFAST_PORTFWD: Port 2/2 in vlan 1 moved to
forwarding(UplinkFast)
2000 Nov 21 01:34:55 %SPANTREE-6-PORTFWD: Port 2/2 state in vlan 1 changed to forwarding
2000 Nov 21 01:34:55 %SPANTREE-7-PORTDEL_SUCCESS:2/1 deleted from vlan 1 (LinkUpdPrs)

```

Using the **show spantree** command, we can check that we have immediately updated our STP:

```

A>(enable) sh spantree
<snip>
Port                      Vlan Port-State      Cost  Priority Portfast  Channel_id
-----
1/1                       1    not-connected      3019    32 disabled  0
1/2                       1    not-connected      3019    32 disabled  0
2/1                       1    not-connected      3100    32 disabled  0
2/2                       1    forwarding         3100    32 disabled  0
<snip>

```

Plug Back the Primary Uplink

At this point, the primary uplink is manually plugged in and put back up. We can see that the UplinkFast feature forces the port into a blocking mode, whereas usual STP rules would have put it in listening mode. At the same time, port 2/2, which should go immediately into blocking mode according to the standard STP, is kept in forwarding mode. UplinkFast forces the current uplink to stay up until the new one is fully operational:

```

A>(enable) 2000 Nov 21 01:35:38 %SPANTREE-6-PORTBLK: Port 2/1
state in vlan 1 changed to blocking
2000 Nov 21 01:35:39 %SPANTREE-5-PORTLISTEN: Port 2/1 state in vlan 1 changed to listening
A>(enable) sh2000 Nov 21 01:35:41 %SPANTREE-6-PORTBLK: Port 2/1 state in vlan 1 changed to

```

blocking

A>(enable) **sho spantree**

<snip>

Port	Vlan	Port-State	Cost	Priority	Portfast	Channel_id
------	------	------------	------	----------	----------	------------

<snip>

2/1	1	blocking	3100	32	disabled	0
2/2	1	forwarding	3100	32	disabled	0

<snip>

A>(enable)

35 seconds after port 2/1 has been brought up, UplinkFast switches the uplinks, blocking 2/2 and moving 2/1 directly into forwarding mode:

2000 Nov 21 01:36:15 %SPANTREE-6-PORTBLK: Port 2/2

state in vlan 1 changed to blocking

2000 Nov 21 01:36:15 %SPANTREE-5-UFAST_PORTFWD: Port 2/1 in vlan 1 moved to forwarding(UplinkFast)

2000 Nov 21 01:36:15 %SPANTREE-6-PORTFWD: Port 2/1 state in vlan 1 changed to forwarding

A>(enable) **sho spantree**

<snip>

Port	Vlan	Port-State	Cost	Priority	Portfast	Channel_id
------	------	------------	------	----------	----------	------------

<snip>

2/1	1	forwarding	3100	32	disabled	0
2/2	1	blocking	3100	32	disabled	0

<snip>

Disable and Clear the Uplink Fast Feature From the Switch

We will now disable UplinkFast, using the command **set spantree uplinkfast disable**. Only the feature is disabled by issuing this command. All the tuning that has been done on the port cost and switch priority remains unchanged:

A>(enable) **set spantree uplinkfast disable**

uplinkfast disabled for bridge.

Use clear spantree uplinkfast to return stp parameters to default.

A>(enable) **show spantree**

VLAN 1

Spanning tree enabled

Spanning tree type ieee

Designated Root 00-40-0b-cd-b4-09

Designated Root Priority 8192

Designated Root Cost 3100

Designated Root Port 2/1

Root Max Age 20 sec Hello Time 2 sec Forward Delay 15 sec

Bridge ID MAC ADDR 00-90-d9-5a-a8-00

Bridge ID Priority 49152

Bridge Max Age 20 sec Hello Time 2 sec Forward Delay 15 sec

Port	Vlan	Port-State	Cost	Priority	Portfast	Channel_id
------	------	------------	------	----------	----------	------------

1/1	1	not-connected	3019	32	disabled	0
1/2	1	not-connected	3019	32	disabled	0
2/1	1	forwarding	3100	32	disabled	0
2/2	1	blocking	3100	32	disabled	0

<snip>

We can now use the command **clear spantree uplinkfast** . This command not only disables the feature, but also resets the parameters:

```
A>(enable) clear spantree uplinkfast
This command will cause all portcosts, portvlancosts, and the
bridge priority on all vlans to be set to default.
Do you want to continue (y/n) [n]? y
VLANs 1-1005 bridge priority set to 32768.
The port cost of all bridge ports set to default value.
The portvlancost of all bridge ports set to default value.
uplinkfast all-protocols field set to off.
uplinkfast disabled for bridge.
A>(enable) show spantree
VLAN 1
Spanning tree enabled
Spanning tree type          ieee

Designated Root            00-40-0b-cd-b4-09
Designated Root Priority    8192
Designated Root Cost       100
Designated Root Port       2/1
Root Max Age 20 sec      Hello Time 2 sec      Forward Delay 15 sec

Bridge ID MAC ADDR         00-90-d9-5a-a8-00
Bridge ID Priority         32768
Bridge Max Age 20 sec      Hello Time 2 sec      Forward Delay 15 sec

Port                        Vlan Port-State      Cost  Priority Portfast  Channel_id
-----
1/1                        1    not-connected      19   32 disabled  0
1/2                        1    not-connected      19   32 disabled  0
2/1                        1    forwarding         100  32 disabled  0
2/2                        1    blocking           100  32 disabled  0
<snip>
```

Uplink Fast Configuration on Other Switches

The UplinkFast feature can be enabled/disabled on Catalyst 4000 and 6000 switches running Cisco IOS Software and Catalyst 2900 XL/3500 XL, 2950, and 3550 series switches by issuing the following commands:

- **spanning-tree uplinkfast** – use this command in global configuration mode to enable the UplinkFast feature.
- **no spanning-tree uplinkfast** – use the no form of this command to disable UplinkFast feature.
- **show spanning-tree** – use this command to display STP state information.

Conclusion

The UplinkFast feature dramatically decreases the convergence time of the STP in the event of the failure of an uplink on an access switch. UplinkFast interacts with other switches that would have a strict standard STP. UplinkFast is only effective when the configured switch has some non-self-looped blocked ports. To increase the chances of having blocked ports, the port cost and bridge priority of the switch are modified. This tuning is consistent for an access switch, but would not be useful on a core switch.

UplinkFast only reacts to direct link failure. A port on the access switch must physically go down in order to trigger the feature. Another Cisco proprietary feature, Backbone Fast, can help to improve convergence time of a bridged network in case of indirect link failure.

Command Reference

- **clear spantree uplinkfast** (CatOS)
 - **set spantree uplinkfast** (CatOS)
 - **show spantree** (CatOS)
 - **set logging level** (CatOS)
 - **spanning-tree uplinkfast** (Cisco IOS)
 - **show spanning-tree** (Cisco IOS)
-

Related Information

- **Understanding and Configuring Backbone Fast on Catalyst Switches**
 - **Understanding and Configuring Spanning-Tree Protocol (STP) on Catalyst Switches**
 - **Spanning Tree Protocol Problems and Related Design Considerations**
 - **Spanning-Tree (Technology Support Pages)**
 - **UplinkFast Configuration Guide**
 - **Technical Support – Cisco Systems**
-

All contents are Copyright © 1992–2005 Cisco Systems, Inc. All rights reserved. Important Notices and Privacy Statement.

Updated: May 10, 2005

Document ID: 10575
