

# Understanding and Designing Networks Using Spanning Tree and UplinkFast Groups

## Introduction

With the increased deployment of Layer 2 switches in the wiring closet and Layer 2/Layer 3 switches in the distribution layers of campus and enterprise networks, network managers find themselves more and more relying on the Spanning Tree Protocol. Spanning Tree, developed by Radia Perlman for use with early bridges, has also been used in switches, since they themselves are Layer 2 bridging devices. The Spanning Tree Protocol is an IEEE/ISO standard since 1990 (IEEE 802.1D). Spanning Tree is a protocol designed to allow bridges to map a redundant topology without causing loops in the network. Loops, in turn, can create broadcast storms, crippling a network and destroying network performance.

However, as network traffic has increased across the backbone, network managers have become more and more disenchanted with Spanning Tree due to its relatively slow convergence time in the event of a link failure. Many vendors have come up with a proprietary protocol designed to enable faster convergence and less perceived network down time while a convergence takes place. However, these protocols typically are completely proprietary in nature and do not interoperate with Spanning Tree (for example, Cabletron's Virtual Link State Protocol, VLSP).

Cisco has addressed this concern by developing the Spanning Tree UplinkFast feature. This feature is based completely on the IEEE 802.1D Spanning Tree and is fully standards-based. The benefit of Spanning Tree UplinkFast is rapid spanning tree convergence around a failed link within 5 seconds delay. This differs markedly from the typical spanning tree convergence of 30 seconds or more.

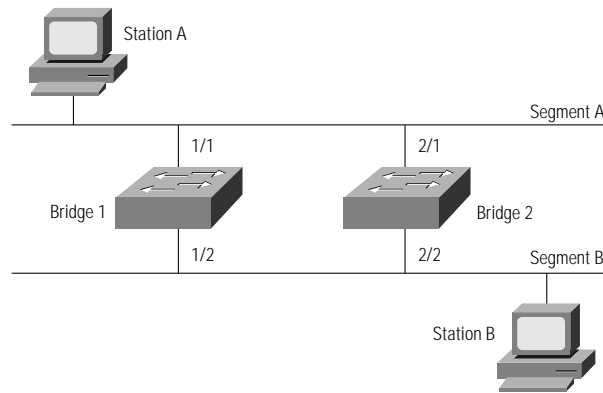
The purpose of this paper is to describe Spanning Tree UplinkFast by first providing an introduction to Spanning Tree itself and then by explaining the Spanning Tree UplinkFast technology.

## Introduction to Spanning Tree

### Transparent Bridge Operation

To understand Spanning Tree, it is first necessary to review how bridges and switches function (remember, a switch is nothing more than a multiport bridge). The bridging of a frame is a two step process. In the first step a transparent spanning tree bridge automatically learns the addressees of the source station in its filtering database. In the second step the bridge forwards the frame. If the destination address is known, the bridge will forward that frame only to its destination segment. If the destination address is not known, the bridge will flood that frame over all the segments with the exception of the receiving one. However, in redundant configurations, such as that in Figure 1, a problem can be realized.

Figure 1 Redundant Bridge Topology



As you can see, Station A has two potential paths to Station B, each via one of the bridges. Now, what happens if Station A sends to Station B, but neither bridge has Station B in its address table.

1. Station A sends a frame out on the wire. Both Bridge 1 and Bridge 2 see that frame and populate their respective address tables indicating that Station A resides on Segment A on ports 1/1 and 2/1, respectively.
2. The frame is forwarded by both bridges onto Segment B. However, each bridge will see that packet again since it is being sent by the other bridge. Each bridge learns Station A as residing on Ports 1/2 and 2/2.
3. The packet is then forwarded again to Segment A, which is the segment where the frame originated. Now the network is seeing the beginning of a loop. Since neither of the bridges are aware of each other, and each bridge continually forwards the frame on the other port, this loop will go on forever.

#### Spanning Tree Operation


The purpose of Spanning Tree is to avoid and eliminate loops in the network by negotiating a loop free path. In the above example, this means that one of the ports will need to be “disabled” or put into such a mode where it is not forwarding packets, but it is still aware of the topology of the network. This section will briefly describe how Spanning Tree operates.

To avoid the formation of loops, devices, such as switches or bridges, execute the Spanning Tree Algorithm. This algorithm effectively “severs” the redundant links within the network. In order to determine which links to “sever”, switches exchange special messages, called bridge protocol data unit (BPDU) frames, that allow them to calculate a Spanning Tree and hence the active topology. This topology will be a subset of the network that is loop-free (for example, the branch of a tree), yet still connects every pair of stations in the network. Using information contained in the BPDU frames, the switches calculate the Spanning Tree in accordance with the algorithm and typically elect to block all of the redundant links, leaving a single communications path.

#### Selection of Root Bridge and Root Port

At the commencement of a Spanning Tree operation, the execution of the Spanning Tree Algorithm causes the bridges to elect a single bridge, among all the bridges within each network, to be the “root” bridge. In selection which bridge will function as the root, the algorithm uses the bridge’s unique numerical identifier, the Bridge ID. The bridge selected as the root is the bridge having the lowest Bridge ID numeric value. In addition, for each segment coupled to more than one bridge, a single “designated bridge” is elected that will forward frames from the segment toward the root. The designated bridge is typically the one closest to the root bridge. By establishing designated bridges, connectivity to all segments, is assured.

Each bridge within the network also selects one port, known as its “root port.” The port selected as the root port is the one which provides the fewest number of hops, assuming all links have the same cost from the bridge to the root. The root ports and designated bridge ports are selected for inclusion in the Spanning Tree and are placed in a forwarding state such that data frames may be forwarded to and from these ports and thus onto the corresponding paths or links. Ports not included within the spanning



are placed in a blocked state. When a port is in the blocked state, data frames will not be forwarded to or received from the port. At the root bridge, all ports are designated ports and are therefore placed in the forwarding state, except for some self-looping ports, if any. A self-looping port is a port coupled to another port at the same bridge.

Each BPDU frame typically includes the following information:

- the identifier of the bridge assumed to be the root (by the bridge transmitting the BPDU),
- the root path cost to the assumed root and
- the identifier of the bridge transmitting the BPDU.

Upon receipt of a BPDU, its contents are examined and compared with similar information (such as the assumed root ID, lowest root path cost and bridge ID) stored by the receiving bridge. If the information from the received BPDU is “better” than the stored information, the bridge adopts the better information and begins transmitting it (adding the cost associated with the receiving port to the root path cost) through its ports, except for the port on which the “better” information was received. This perpetuates the “improved” information to all bridges downstream from the root and the root port of the bridge. Eventually, all bridges will agree on the root and each will be able to identify which of its ports presents the lowest cost path to the root (i.e., its root port).

Depending on the configuration of a given network, the location of the root can significantly affect the distance that messages must travel. For example, many networks include a plurality of switches designated as access switches that provide connectivity to segments and end stations, as well as plurality of backbone switches that, in turn, interconnect the various access switches. If the root bridge is located at an access switch and the principal server utilized by the end stations (for example, network clients) is coupled to a backbone switch, the average distance between end stations and the primary server may be quite high, resulting in inefficient network operation. In addition, the backbone switches may become partitioned as ports between them are blocked. To reduce the average distance and avoid partitioning of the backbone switches, it is desirable to locate the root bridge at a backbone switch. Bridge IDs, moreover, include a fixed portion and a settable portion. By substantially decreasing the value of the settable portion of the identifier for a selected bridge, a network administrator may “force” the network to choose the selected switch as the root.

To identify which bridge should be the designated bridge, bridges again compare information in received BPDUs with their stored information. If the root path cost stored by a first bridge is lower than the root path cost contained in BPDUs received from a second bridge, then the first bridge is the designated bridge. If the root path cost for both the first and second bridges is the same, the first bridge compares the next informational element in the BPDU, the Bridge IDs. If the Bridge ID of the first bridge is less than the ID of the second bridge, then the first bridge is the designated bridge, otherwise the second bridge is the designated bridge.

In accordance with the Spanning Tree Algorithm, the root bridge generates and transmits BPDUs from its ports every hello time which is a settable parameter (the default hello time is two seconds). In response to receiving BPDUs, bridges transmit their own BPDUs. Thus every two seconds BPDUs are propagated through the network. BPDU information is subject to being aged out and discarded. Typically, a timer is associated with the BPDU information stored for each port of a bridge. The timer is set to a value referred to as the maximum age, which is placed into BPDU generated by the root bridge and copied by the other bridges. (A typical value for maximum age value is twenty seconds.) As BPDUs are received, their contents are examined. If the contents match the information already stored for that port, the timer is reset back to zero. Accordingly, by receiving consistent BPDUs every hello time, which is significantly less than the maximum age, the current BPDU information is maintained and the accuracy of the Spanning Tree or active topology is confirmed.

#### Spanning Tree Recalculations

If a bridge stops receiving BPDUs on its root port, indicating a possible link or device failure, the corresponding maximum age timer will expire and the information will be discarded. In response, the bridge will select a new root port based upon the next best information it has, and begin transmitting BPDUs through its other ports. Similarly, as links or devices are repaired or added, a bridge may receive BPDUs containing better information than that stored for a particular port, thereby causing the bridge to replace the previously stored information, as described above.

As BPDU information is updated and/or timed-out, the Spanning Tree is recalculated and ports may transition from the blocked state to the forwarding state and vice versa. That is, as a result of new BPDU information, a previously blocked port may learn that it is now the root port or the designated port for a given segment. Rather than transition directly from the blocked state to the forwarding state, ports transition through two intermediate states: a listening state and a learning state. The bridge will remain in each state for a preset period of time, called the forwarding delay. In the listening state, a port waits for information indicating that it should return to the blocked state. If, by the end of the forwarding delay time, no such information is received, the port transitions to the learning state. In the learning state, a port still blocks the receiving and forwarding of frames, but received frames are examined and the corresponding location information is stored, as described above. At the end of a second forwarding delay time, the port transitions from the learning state to the forwarding state, thereby allowing frames to be forwarded and received at the port.

As ports transition between the blocked and forwarding states, end-station MAC addresses may appear to move from one port to another. To prevent switches from distributing messages based upon incorrect information, switches quickly age-out and discard the “old” information in their filtering databases. More specifically, upon detection of a change in the Spanning Tree, switches transmit Topology Change Notification Protocol Data Unit (TCN-PDU) frames toward the root. The TCN-PDU is propagated hop-by-hop until it reaches the root which confirms receipt of the TCN-PDU by setting a Topology Change Flag in all BPDUs subsequently transmitted by the root for a period of time. Other switches, receiving these BPDUs, note that the Topology Change Flag has been set, thereby alerting them to the change in the active topology. In response, switches significantly lower the aging time associated with their filtering databases which, as described above, contain destination information corresponding to the entities within the network. Specifically, switches replace the default aging time of five minutes with the forwarding delay time, which is generally fifteen seconds. Information contained in the filtering databases is thus quickly discarded.

Although the Spanning Tree Algorithm is able to maintain a loop-free tree despite network changes, recalculation of the Spanning Tree is a relatively time consuming process. Standard Spanning Tree values for the maximum age of BPDUs (which is the length of time that BPDU information is kept) is typically twenty seconds. The forwarding delay time (which is the length of time that ports are to remain in each of the listening and learning states) is fifteen seconds. As a result, recalculation of the Spanning Tree following a network change takes approximately fifty seconds: twenty seconds for BPDU information to time out, fifteen seconds in the listening state and another fifteen seconds in the learning state.

During this recalculation period, message delivery is often delayed as ports transition between states. That is, ports in the listening and learning states do not forward or receive frames. To the network users, these delays are perceived as service interruptions, which may present significant problems, especially on networks requiring high availability. In addition, certain applications, protocols or processes may time-out and shut down during the reconfiguration process, resulting in even greater disruption to the network. Another disadvantage relates to subsequent message distribution. Following the reconfiguration process, messages are flooded across the network until the “new” destination ports are learned and the aging time returned to five minutes. Such flooding of messages often consumes substantial communications and processor resources.

## **Theory of Spanning Tree Optimization**

Spanning Tree optimization can, in theory, be achieved in many ways. When Spanning Tree was first developed, bridges forwarded BPDUs and data frame in software, resulting in a long propagation and processing delay of BPDUs. For that reason, the designers of Spanning Tree were very conservative in the values chosen for spanning Tree convergence parameters. Back when the protocol was developed, this was considered acceptable in order to achieve graceful topology reconvergence. Today, however, these are viewed as unacceptably long convergence times.

### **Root Bridge Location Selection**

The Max Age, Forward Delay and Hello Time parameters are common to all the switches in a spanning tree. They are set at the root bridge and they are propagated by BPDUs to all other bridges in the spanning tree. Therefore, to tune these parameters, it is necessary to set them at the root bridge and at the backup root bridge. Although simple tuning of some Spanning Tree parameters may optimize Spanning Tree operation, another important consideration is the distance from the root bridge. Since

the root bridge is responsible for generating configuration and management BPDUs, it is important to place it in the optimum place in the network. This logically implies that the root bridge should reside at the center of the topology, approximately equidistant from all of its Spanning Tree leaves. This allows Topology Change Notifications and Acknowledgments to rapidly propagate through the bridged network.

#### Tuning Spanning Tree Parameters

Changing Spanning Tree parameters, while resulting in fast convergence, may have unexpected and possibly detrimental results. Tuning Max Age down to the minimum value of 6 seconds can result in BPDUs being discarded too early. During normal operation, the root bridge periodically transmits configuration BPDUs every hello time (by default every 2 seconds). These BPDUs are, in turn, propagated to bridges downstream from the root. If BPDU frames are inadvertently dropped, a low value for Max Age may cause the bridge to trash its current configuration and recompute the Spanning Tree.

Tuning Forward Delay down to the minimum value of 4 seconds means that when transitioning from Blocking to Forwarding a port will spend 4 seconds in the listening state and 4 seconds in learning state. This means that there is only a 8 second window in the listening and learning states in which a BPDU indicating a loop must be received and processed. Thus setting this value too low may not account for the fact that BPDUs may be inadvertently dropped and can also result in a loop and possibly in a broadcast storm.

Table 1 below provides descriptions of each parameter and what their tunable ranges are in the Catalyst LAN switching product family.

Table 1 Summary of Spanning Tree Optimization Parameters

Parameter	Purpose	Default Value	Tunable range
Max Age	Maximum time a BPDU received on a port is considered valid	All Catalyst products: <b>20 seconds</b>	Catalyst 5000 Series: <b>4-30 seconds</b> Catalyst 3000 Series: <b>4-30 seconds</b> Catalyst 1900/2820/2900XL: <b>6-40 seconds</b>
Forward Delay	Amount of time spent in Listening and Learning states. The total delay prior to forwarding, therefore, is $FWDDelay * 2$	All Catalyst products: <b>15 seconds</b>	All Catalyst products: <b>4-30 seconds</b>
Hello Time	Interval of time when root bridge generates BPDU frames	All Catalyst products: <b>2 seconds</b>	All Catalyst products: <b>1-10 seconds</b>

Table 1 Summary of Spanning Tree Optimization Parameters

Parameter	Purpose	Default Value	Tunable range
Bridge Priority	The settable portion of the Bridge ID. The bridge with the lowest Bridge ID will be selected as the root	All Catalyst Products: <b>32,768 (same value per VLAN)</b>	All Catalyst Products <b>0-65,535</b>
Port priority	Settable portion of the Port ID. Determines, between two equal cost paths, which one will be selected to be in forwarding mode, putting the other link in blocking	Catalyst 5000 Series: <b>32</b> Catalyst 3000 Series: <b>128</b> Catalyst 1900/2820/2900XL: <b>128</b>	Catalyst 5000 Series: <b>(0-63)</b> Catalyst 3000 Series: <b>(1-255)</b> Catalyst 1900/2820/2900XL: <b>(1-255)</b>
Port priority by VLAN	Same function as Port Priority, only this is set on a per-VLAN basis with the command: <code>set spantree vlanportpri [mod/port] vlan</code>	Catalyst 5000 Series: <b>32 (per VLAN)</b> Catalyst 3000 Series: <b>128 (80 Hex)</b> Catalyst 1900/2820/2900XL: <b>Not applicable</b>	Catalyst 5000 Series: <b>(0-63)</b> Catalyst 3000 Series: <b>(1-255)</b> Catalyst 1900/2820/2900XL: <b>Not applicable</b>
Port cost by VLAN	Allows configuration of the port path cost based on media type.	All Catalyst Products: defaults depend on media type, only settable by VLAN in 2.4 release ATM OC3 = 6 10 Mbps = 100, 100 Mbps = 10, 10/100 Mbps = 100)	All Catalyst Products <b>0-65,535</b>

### Spanning Tree UplinkFast Groups

As discussed in the earlier sections, the Spanning Tree Protocol (STP) configures bridges linked as a partially-meshed network into a loop-free tree topology. STP ensures that this loop-free topology is maintained even in the face of changes either due to machines being added or removed. A problem with STP however, is that the convergence to a loop-free topology can be time-consuming since it does not address how faults are detected and what action can be taken to optimize convergence. During the time it takes for Spanning Tree to converge, some stations may become inaccessible depending on the Spanning Tree states of their bridge ports. Customers view this as a major disruption in service for many reasons. Thus, speeding up this lengthy convergence time would be a major factor in customer decisions to deploy additional redundancy. In recognition of this clear business demand for fast convergence without compromising load balancing, the UplinkFast convergence feature is available starting the 3.1.1 software release for the Catalyst 5000 series.

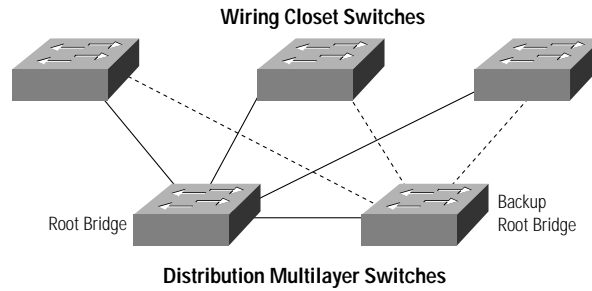
UplinkFast is a solution which in a redundantly connected topology aims to address the problem of fast convergence of Spanning Tree while still adhering to the standard. It does this by grouping uplink ports together in a specific way. These groupings of specific uplink ports are called “uplink groups”. Along with the Uplink Groups feature the tuning of Spanning Tree has also been enhanced with the ability to now tune the path cost by VLAN in the 3.1.1 software release.

The Spanning Tree UplinkFast feature has been designed specifically for use with access layer wiring closet switches, which are directly connecting users to the network, and distribution layer switches, which are interconnecting multiple wiring closets. This feature is not designed for use within network cores. It is strongly recommended that, in order to achieve better scalability of the network, Layer 3 routing protocols be used in the network core.

#### Overview of UplinkFast Convergence

Figure 2 displays a typical network design, utilizing both an access layer, which uses switches to connect users to the network, and a distribution layer, which connects multiple access switches together.

Figure 2 Hierarchical Network Design with Spanning Tree



Every wiring closet switch in Figure 2 has two uplinks: one to the root bridge, the other to the backup root bridge. One of the two uplinks is redundant (the redundant links are marked with the dotted lines). Redundant links have been blocked by STP to avoid loops with uplink fast, as soon as a switch detects a link down condition on its currently active link, it immediately move the blocked port to the forwarding state, without going via the listening and learning phases. In this way reconfiguration occurs within 3–4 seconds without going through the usual convergence time of STP.

The conditions for UplinkFast to trigger a fast reconfiguration are:

- UplinkFast has to be enable on the switch;
- the switch must have at least one blocked port;
- the failure must be related to the active path toward the root.

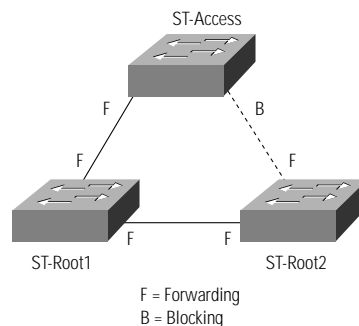
UplinkFast will initiate the reconfiguration as soon as a link down on the root port has been detected. This allows convergence to begin immediately without waiting for the MAX-AGE timer to expire.

This concept for a single VLAN can be applied across as many VLANs that the wiring closet switch supports. If the root bridges are placed in such a way so as to divide the VLANs between the two uplinks, we could achieve both VLAN load distribution as well as fast convergence. One link in the uplink group is forwarding, for example, VLANs 1, 3, and 5 (blocking for VLANs 2, 4, and 6) while the second link is forwarding for VLANs 2, 4, and 6 (and blocking for 1, 3, and 5).

#### Sample UplinkFast Configuration and Guidelines

The following section is provided in order to show an example of how UplinkFast can be activated on multiple switches. In this example, three switches have been configured in a redundant topology. It is important to understand that UplinkFast runs on the Access Layer switch (refer to Figure 3 for a diagram of the topology). UplinkFast is enabled only after the Spanning Tree has been calculated and the topology has “settled down.”

Figure 3 Test Network Topology



ST-Root1 has been designated as the root bridge for VLAN-1. The primary forwarding path is always the Spanning Tree determined root port. Influencing this selection has to be achieved, either through placement of the VLAN root bridge (by adjusting the bridge priority settings), through the VLAN port priority (by adjusting portvlanpri settings on parallel paths to the same bridge), or by adjusting the VLAN port costs (by adjusting the portvlancost settings). In the 3.1(1) software, a command has been added to allow easy configuration of the root.

To activate that bridge as the root, use the following command:

```
set spantree root [secondary] [<VLAN_list>] [dia <network diameter>] [hello <hello_time>]
```

where

*secondary*—determines whether this is the primary or secondary root bridge

*VLAN\_list*—the VLANs to which this bridge is to function as the root for

*dia*—the maximum number of bridges between two end stations

*hello*—the hello timer by which BPDUs are transmitted.

```
ST-Root1> (enable)
ST-Root1> (enable) set spantree root 1
VLAN 1 bridge priority set to 8192.
VLAN 1 bridge max aging time set to 20.
VLAN 1 bridge hello time set to 2.
VLAN 1 bridge forward delay set to 15.
Switch is now the root switch for active VLAN 1.
ST-Root1> (enable)
```

In the event that ST-Root1 fails or “goes away” as far as Spanning Tree is concerned, it is important to ensure that a distribution layer switch is still the root. For that reason, configure ST-Root2 as the secondary root bridge. This is accomplished using the same command as

```
ST-Root2> (enable)
ST-Root2> (enable) set spantree root secondary 1
VLAN 1 bridge priority set to 16384.
VLAN 1 bridge max aging time set to 20.
VLAN 1 bridge hello time set to 2.
VLAN 1 bridge forward delay set to 15.
ST-Root2> (enable)
```

Should it be necessary to clear the designation of ST-Root1 or ST-Root2 as the root bridge and restore the original Spanning Tree settings, use the command:

```
clear spantree root [<VLAN_list>]
```

Since the root priorities have been determined, it is now possible to activate UplinkFast on as many access layer switches as desired. To enable this feature, use the following command set.

```
ST-Access> (enable)
ST-Access> (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 enabled for bridge.
ST-Access> (enable)
```

The backup UplinkFast port that will be switched into forwarding after detection of a failure on the primary (root port) is the non-designated port that Spanning Tree had initially put into blocking state while laying itself out initially. Note that there is nothing to preclude the primary and backup links being dissimilar media types or speeds. Any combination of Ethernet, Fast Ethernet, Fast EtherChannel, or ATM ports can be used.





**Caveat:** It is important to note that UplinkFast should only be configured on Access Layer switches, because quick transitions of the root port to its non-designated backup(s) can only be deterministically achieved at the end leaf-node switches in a Spanning Tree topology. If UplinkFast is not being used, it is strongly recommended to use the Catalyst (and the standards) default settings. This has been conservatively designed in the original specification to take into account BPDU propagation delay across multiple bridge (switch) hops as well as CAM table re-learning down converged paths. The UplinkFast feature does check if a switch is itself the root bridge and in that case disables this feature.

To show the status of UplinkFast and to determine which of the ports is forwarding and which is blocking, use the following command.

```
ST-Access> (enable) sh spantree uplinkfast
Station update rate set to 15 packets/100ms.
VLAN      port list
-----
1          3/3(fwd),3/1
ST-Access> (enable)
```

The failover time for UplinkFast is demonstrated by this ping sequence. Note that link 3/3 is the forwarding link. Link 3/1 is blocking.

```
ST-Access> (enable) sh spantree uplinkfast
Station update rate set to 15 packets/100ms.
VLAN      port list
-----
1          3/3(fwd),3/1
```

convergence section will be modified to use two endstations.

```
ST-Access> (enable) ping -s 10.1.1.3
PING 10.1.1.3: 56 data bytes
64 bytes from 10.1.1.3: icmp_seq=0. time=2 ms
64 bytes from 10.1.1.3: icmp_seq=1. time=1 ms
64 bytes from 10.1.1.3: icmp_seq=2. time=1 ms
64 bytes from 10.1.1.3: icmp_seq=3. time=1 ms
64 bytes from 10.1.1.3: icmp_seq=4. time=1 ms
64 bytes from 10.1.1.3: icmp_seq=8. time=1 ms
64 bytes from 10.1.1.3: icmp_seq=9. time=1 ms
64 bytes from 10.1.1.3: icmp_seq=10. time=1 ms
64 bytes from 10.1.1.3: icmp_seq=11. time=1 ms
64 bytes from 10.1.1.3: icmp_seq=12. time=1 ms
64 bytes from 10.1.1.3: icmp_seq=13. time=1 ms
64 bytes from 10.1.1.3: icmp_seq=14. time=1 ms
64 bytes from 10.1.1.3: icmp_seq=15. time=1 ms
64 bytes from 10.1.1.3: icmp_seq=16. time=1 ms
64 bytes from 10.1.1.3: icmp_seq=17. time=1 ms
64 bytes from 10.1.1.3: icmp_seq=18. time=1 ms
64 bytes from 10.1.1.3: icmp_seq=19. time=1 ms
64 bytes from 10.1.1.3: icmp_seq=20. time=1 ms
^C
----10.1.1.3 PING Statistics----
21 packets transmitted, 18 packets received, 14% packet loss
round-trip (ms) min/avg/max = 1/1/2
ST-Access> (enable)
```

(At this point, link 3/3 was pulled)  
(Note the loss of only 3 pings)

As shown above, the Spanning Tree failover time, previously as high as 50 seconds, has been reduced to 3 seconds, resulting in fast network resiliency, ensuring application and network integrity.

## Conclusion

Spanning Tree provides a means of redundancy between multiple bridges or switches in a network. As switches are being deployed more and more for desktop connectivity, Spanning Tree has become increasingly crucial. However, Spanning Tree introduces an unacceptably long convergence in the event of network failure. It also requires a good understanding in order to implement a well-designed network. Cisco has addressed these concerns by developing the Spanning Tree UplinkFast feature. This feature, based completely on the standard IEEE 802.1D Spanning Tree Protocol, provides the same level of redundancy with significantly faster convergence times. This feature is designed to be used to scale bandwidth and convergence between access layer wiring closets and building distribution or core networks. In addition, with the simplicity of implementing UplinkFast, a major understanding of Spanning Tree, while valuable, is not necessary to deploying scalable, multilayer switched networks.



### Corporate Headquarters

Cisco Systems, Inc.  
170 West Tasman Drive  
San Jose, CA 95134-1706  
USA

<http://www.cisco.com>  
Tel: 408 526-4000  
800 553-NETS (6387)  
Fax: 408 526-4100

### European Headquarters

Cisco Systems Europe s.a.r.l.  
Parc Evolic, Batiment L1/L2  
16 Avenue du Quebec  
Villebon, BP 706  
91961 Courtaboeuf Cedex  
France

<http://www-europe.cisco.com>  
Tel: 33 1 6918 61 00  
Fax: 33 1 6928 83 26

### Americas Headquarters

Cisco Systems, Inc.  
170 West Tasman Drive  
San Jose, CA 95134-1706  
USA

<http://www.cisco.com>  
Tel: 408 526-7660  
Fax: 408 527-0883

### Asia Headquarters

Nihon Cisco Systems K.K.  
Fuji Building, 9th Floor  
3-2-3 Marunouchi  
Chiyoda-ku, Tokyo 100  
Japan

<http://www.cisco.com>  
Tel: 81 3 5219 6250  
Fax: 81 3 5219 6001

**Cisco Systems has more than 200 offices in the following countries. Addresses, phone numbers, and fax numbers are listed on the Cisco Connection Online Web site at <http://www.cisco.com>.**

Argentina • Australia • Austria • Belgium • Brazil • Canada • Chile • China (PRC) • Colombia • Costa Rica • Czech Republic • Denmark  
England • France • Germany • Greece • Hungary • India • Indonesia • Ireland • Israel • Italy • Japan • Korea • Luxembourg • Malaysia  
Mexico • The Netherlands • New Zealand • Norway • Peru • Philippines • Poland • Portugal • Russia • Saudi Arabia • Scotland • Singapore