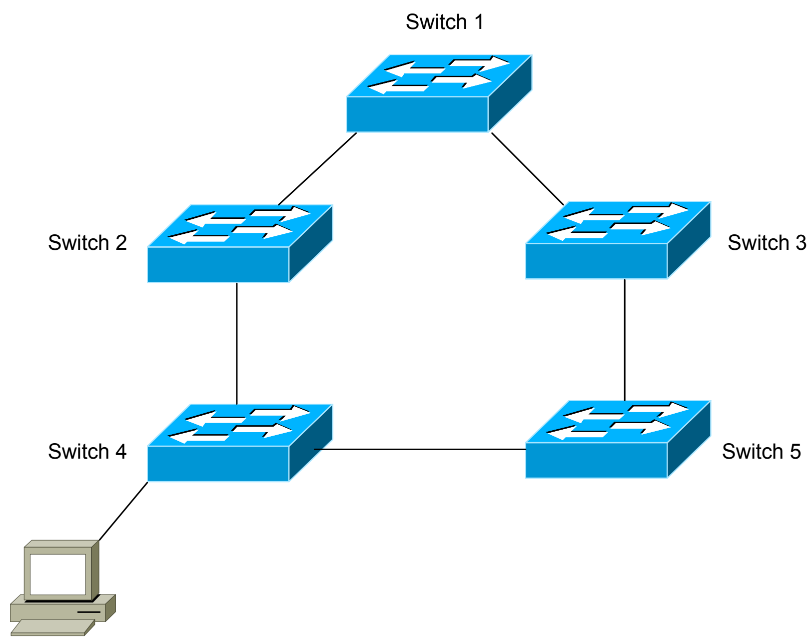
- Spanning Tree Protocol -

Switching Loops

By default, a switch will forward a broadcast or multicast out all ports, excluding the port the broadcast/multicast was sent from.

When a *loop* is introduced into the network, a highly destructive **broadcast storm** can develop within seconds. Broadcast storms occur when broadcasts are endlessly switched through the loop, choking off all other traffic.

Consider the following looped environment:



If the computer connected to Switch 4 sends out a broadcast, the switch will forward the broadcast out all ports, including the ports connecting to Switch 2 and Switch 5. Those switches, likewise, will forward that broadcast out all ports, including to their neighboring switches.

The broadcast will loop around the switches infinitely. In fact, there will be *two* separate broadcast storms cycling in opposite directions through the switching loop. Only powering off the switch or physically removing the loop will stop the storm.

Spanning Tree Protocol (STP)

Switches (and bridges) needed a mechanism to prevent loops from forming, and thus **Spanning Tree Protocol (STP,** or **IEEE 802.1D)** was developed. STP is **enabled** by default on all VLANs on Catalyst switches.

STP-enabled switches communicate to form a topology of the entire switching network, and then shutting down (or *blocking*) a port if a loop exists. The blocked port can be reactivated if another link on the switching network goes down, thus preserving fault-tolerance. Once all switches agree on the topology database, the switches are considered **converged**.

STP switches send **BPDU's (Bridge Protocol Data Units)** to each other to form their topology databases. BPDU's are sent out all ports every **two seconds,** are forwarded to a specific MAC multicast address: 0180.c200.0000.

STP Types

Various flavors of 802.1D STP exist, including:

- **Common Spanning Tree (CST)** – A single STP process is used for all VLANs.
- **Per-VLAN Spanning Tree (PVST) –** Cisco proprietary version of STP, which employs a separate STP process for each VLAN.
- **Per-VLAN Spanning Tree Plus (PVST+)** – Enhanced version of PVST that allows CST-enabled switches and PVST-enabled switches to interoperate. This is *default* on newer Catalyst switches.

The STP Process

To maintain a loop-free environment, STP performs the following functions:

- A **Root Bridge** is elected
- **Root Ports** are identified
- **Designated Ports** are identified
- If a loop exists, a port is placed in **Blocking** state. If the loop is removed the blocked port is activated again.

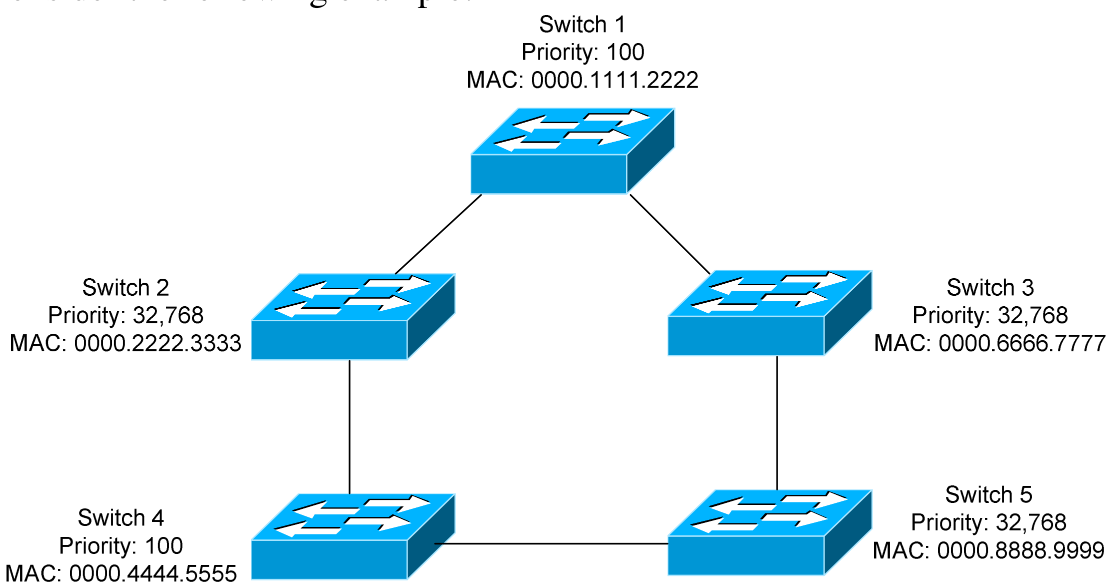
If multiple loops exist in the switching environment, multiple ports will be placed in a blocking state.

Electing an STP Root Bridge

The first step in the STP process is electing a **Root Bridge**, which serves as the centralized point of the STP topology. Good design practice dictates that the Root Bridge be placed closest to the center of the STP topology.

The Root Bridge is determined by a switch's **priority**. The default priority is **32,768**, and the **lowest priority wins**. In case of a tie in priority, the switch with the **lowest MAC address** will be elected root bridge. The combination of a switch's priority and MAC address make up that switch's **Bridge ID.**

Consider the following example:



Remember that the lowest priority determines the Root Bridge. Switches 2, 3, and 5 have the default priority set. Switches 1 and 4 each have a priority of 100 configured. However, Switch 1 will become the root bridge, as it has the lowest MAC address.

Switches exchange BPDU's to perform the election process. By default, all switches "believe" they are the Root Bridge, until a switch with a lower Bridge ID is discovered.

Root Bridge elections are a continuous process. If a new switch with a lower Bridge ID is added to the topology, it will be elected as the new Root Bridge.

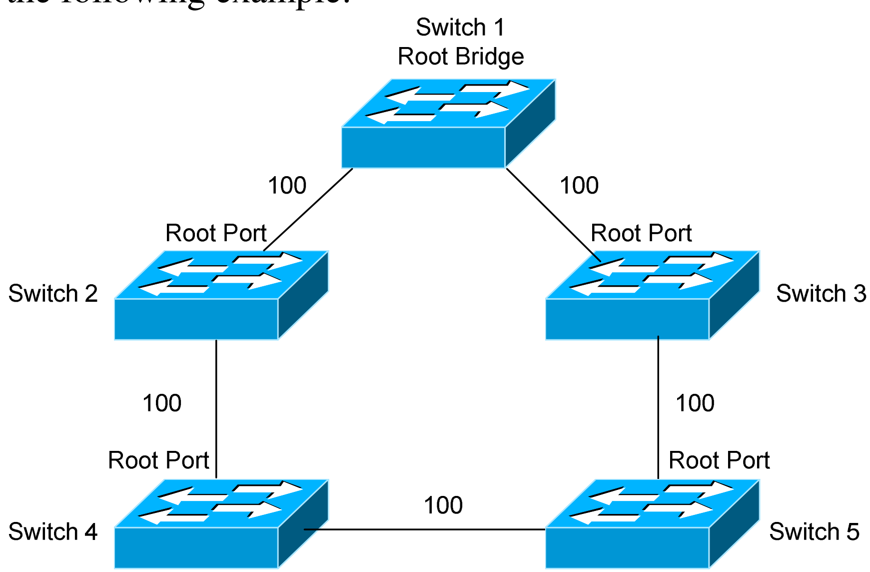
Identifying Root Ports

The second step in the STP process is identifying **Root Ports,** or the port on each switch that has the lowest **path cost** to get to the Root Bridge. Each switch has only **one** Root Port, and the Root Bridge *cannot* have a Root Port.

Path Cost is a cumulative cost based on the bandwidth of the links. The higher the bandwidth, the lower the Path Cost:

Bandwidth	***Cost***
4 Mbps	250
10 Mbps	100
16 Mbps	62
100 Mbps	19
1 Gbps	4

Consider the following example:



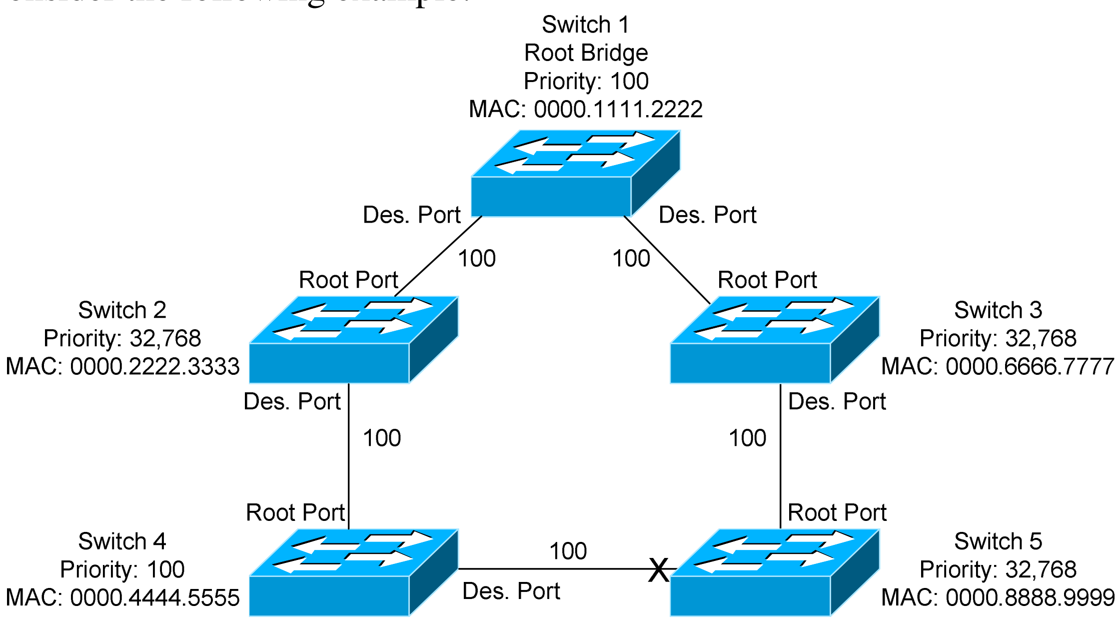
Assume the links between all switches are 10Mbps Ethernet, with a Path Cost of 100. Each switch will identify the port with the least cumulative Path Cost to get to the Root Bridge.

For Switch 4, the port leading up to Switch 2 has a Path Cost of 200, and becomes the Root Port. The port to Switch 5 has a higher Path Cost of 300. The Root Port is said to have received the most **superior BPDU** to the Root Bridge. Likewise, non-Root Ports are said to have received **inferior BPDU's** to the Root Bridge.

Identifying Designated Ports

The third and final step in the STP process is to identify **Designated Ports.** Each *network segment* requires a single Designated Port, which has the lowest path cost leading to the Root Bridge. This port *will not* be placed in a blocking state. A port cannot be both a Designated Port and a Root Port.

Consider the following example:



Ports on the Root Bridge are *never* placed in a blocking state, and thus become Designated Ports for directly attached segments.

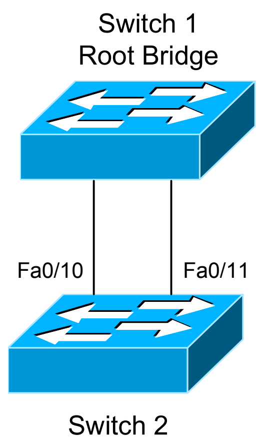
The network segments between Switches 2 and 4, and between Switches 3 and 5, both require a Designated Port. The ports on Switch 2 and Switch 3 have the lowest Path Cost to the Root Bridge for the two respective segments, and thus both become Designated Ports.

The segment between Switch 4 and Switch 5 does not contain a Root Port. One of the ports must be elected the Designated Port for that segment, and the other must be placed in a blocking state.

Normally, Path Cost is used to determine which port is blocked. However, the ports connecting Switches 4 and 5 have the same Path Cost to reach the Root Bridge (*200*). Whichever switch has the **lowest** Bridge ID is awarded the Designated Port. Whichever switch has the **highest** Bridge ID has its port placed in a blocking state. In this example, Switch 4 has the lowest priority, and thus Switch 5's port goes into a blocking state.

Port ID

In certain circumstances, a tie will occur in both Path Cost and Bridge ID. Consider the following example:



If the bandwidth of both links are equal, then both of Switch 2's interfaces have an equal path cost to the Root Bridge. Which interface will become the Root Port? The tiebreaker should be the lowest Bridge ID, but that cannot be used in this circumstance (unless Switch 2 has become schizophrenic).

In this circumstance, **Port ID** will be used as the tiebreaker. An interface's Port ID consists of two parts - a 6-bit **port priority** value, and the MAC address for that port. Whichever interface has the **lowest** Port ID will become the Root Port.

By default, the port priority of an interface is **128**. Lowering this value will ensure a specific interface becomes the Root Port:

Switch(config)# *int fa0/10*
Switch(config-if)# *spanning-tree port-priority 50*

Remember, that port priority is the *last* tiebreaker STP will consider. STP decides Root and Designated Ports based on the following criteria, and *in this order*:

- Lowest Path Cost to the Root Bridge
- Lowest Bridge ID
- Lowest Port ID

Extended System IDs

Normally, a switch's Bridge ID is a 64-bit value that consists of a 16-bit Bridge Priority value, and a 48-bit MAC address.

However, it is possible to include a VLAN ID, called an **extended System ID,** into a Bridge ID. Instead of *adding* bits to the existing Bridge ID, 12 bits of the Bridge Priority value are used for this System ID, which identifies the VLAN this STP process represents.

Because 12 bits have been stolen from the Bridge Priority field, the range of priorities has been reduced. Normally, the Bridge Priority can range from 0 (or *off*) to 65,535, with a default value of 32,768. With extended System ID enabled, the Priority range would be 0 – 61,440, and only in multiples of 4,096.

To enable the extended System ID:

Switch(config)# *spanning-tree extend system-id*

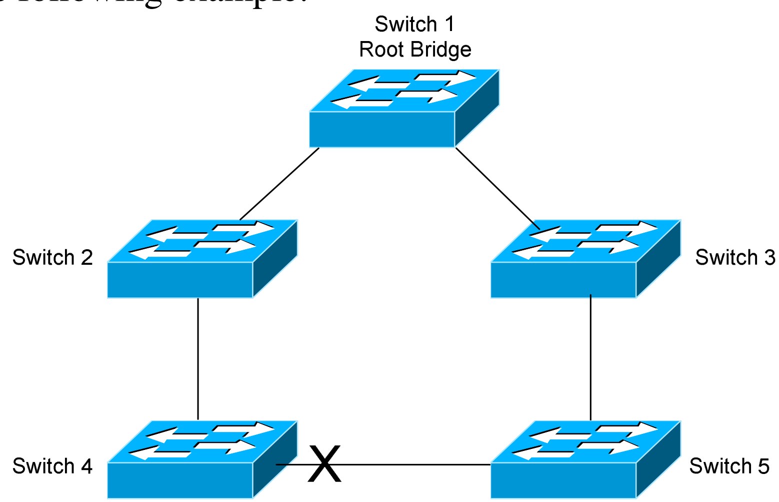
Enabling extended System ID accomplishes two things:
- Increases the amount of supported VLANs on the switch from 1005 to 4094.
- Includes the VLAN ID as part of the Bridge ID.

Thus, when this command is enabled, the 64-bit Bridge ID will consist of the following:
- 4-bit Priority Value
- 12-bit System ID value (VLAN ID)
- 48-bit MAC address

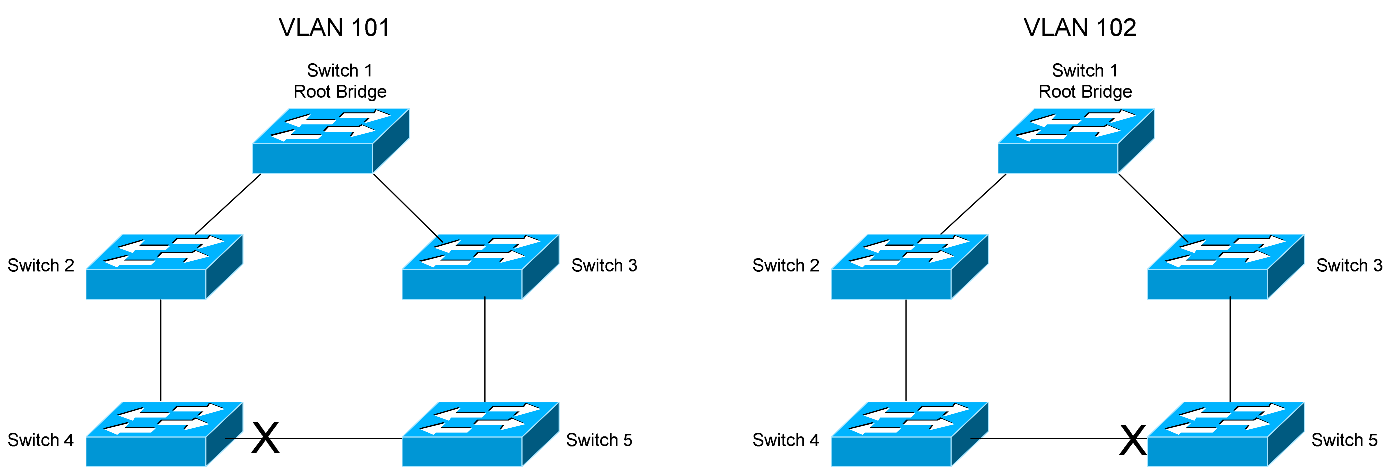
Per-VLAN Spanning Tree (PVST) Example

Remember that PVST+ is the default implementation of STP on Catalyst switches. Thus, each VLAN on the switch is allotted its own STP process.

Consider the following example:



With Common Spanning Tree (CST), all VLANS would belong to the same STP process. Thus, if one Switch 4's ports entered a *blocking* state to eliminate the loop, all VLANs would be blocked out that port. For efficiency purposes, this may not be ideal.



In the above examples, the benefit of PVST becomes apparent. STP runs a separate process for each VLAN, allowing a port to enter a blocking state **only for that specific VLAN.** Thus, it is possible to load balance VLANs, allowing traffic to flow more efficiently.

STP Port States

Switch ports participating in STP progress through five **port states**:

Blocking – The *default* state of an STP port when a switch is powered on, and when a port is shut down to eliminate a loop. Ports in a blocking state do not forward frames or learn MAC addresses. It will still listen for BPDUs from other switches, to learn about changes to the switching topology.

Listening – A port will progress from a Blocking to a Listening state only if the switch believes that the port *will not be shut down* to eliminate a loop. The port will listen for BPDU's to participate in the election of a Root Bridge, Root Ports, and Designated Ports. Ports in a listening state will not forward frames or learn MAC addresses.

Learning – After a brief period of time, called a **Forward Delay**, a port in a listening state will be elected either a Root Port or Designated Port, and placed in a learning state. Ports in a learning state listen for BPDUs, and also begin to learn MAC addresses. However, ports in a learning state will still not forward frames.

(Note: If a port in a listening state *is not* kept as a Root or a Designated Port, it will be placed into a *blocking* state and not a *learning* state.)

Forwarding – After another Forward Delay, a port in learning mode will be placed in forwarding mode. Ports in a forwarding state can send and receive all data frames, and continue to build the MAC address table. All designated, root, and non-uplink ports will eventually be placed in a forwarding state.

Disabled – A port in disabled state has been administratively shut down, and does not participate in STP or forward frames at all.

On average, a port in a blocking state will take 30 to 50 seconds to reach a forwarding state.

To view the current state of a port (such fa0/10):

Switch# *show spanning-tree interface fa0/10*

```
Interface Fa0/10 in Spanning tree 1 is Forwarding
Port path cost 100, Port priority 128

<snip>
```

(Reference: http://www.cisco.com/en/US/docs/switches/lan/catalyst4500/12.1/8aew/configuration/guide/spantree.html#wp1020487)

STP Timers

STP utilizes three timers to ensure all switches remain synchronized, and to allow enough time for the Spanning Tree process to ensure a loop-free environment.

- **Hello Timer –** Default is **2 seconds**. Indicates how often BPDU's are sent by switches.
- **Forward Delay –** Default is **15 seconds**. Indicates a *delay* period in both the *listening* and *learning* states of a port, for a total of **30 seconds**. This delay ensures STP has ample time to detect and eliminate loops.
- **Max Age –** Default is **20 seconds**. Indicates how long a switch will keep BPDU information from a neighboring switch before discarding it. In other words, if a switch fails to receive BPDU's from a neighboring switch for the Max Age period, it will remove that switch's information from the STP topology database.

All timer values can be adjusted, and should *only* be adjusted on the Root Bridge. The Root Bridge will **propagate the changed timers** to all other switches participating in STP. Non-Root switches will ignore their locally configured timers.

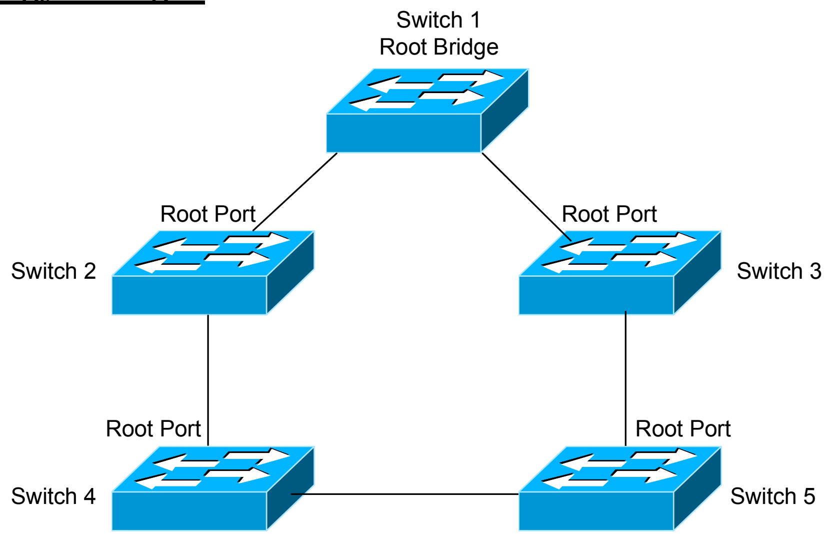
To adjust the three STP timers for VLAN 10:

Switch(config)# *spanning-tree vlan 10 hello-time 10*
Switch(config)# *spanning-tree vlan 10 forward-time 20*
Switch(config)# *spanning-tree vlan 10 max-age 40*

The timers are measured in seconds. The above examples represent the maximum value each timer can be configured to.

Remember that STP is configured on a VLAN by VLAN basis on Catalyst Switches.

STP Topology Changes



An **STP topology change** will occur under two circumstances:

- When an interface is placed into a Forwarding state.
- When an interface *already* in a Forwarding or Learning state is placed into a Blocking state.

The switch recognizing this topology change will send out a **TCN (Topology Change Notification)** BPDU, destined for the Root Bridge. The TCN BPDU does not contain any data about the actual change – it only indicates that a change occurred.

For example, if the interface on Switch 4 connecting to Switch 5 went down, Switch 4 would send a TCN out its Root Port to Switch 2. Switch 2 will acknowledge this TCN by sending a BPDU *back* to Switch 4 with the **Topology Change Acknowledgement (TCA) bit** set. Switch 2 would then forward the TCN out *its* Root Port to Switch 1 (the Root Bridge).

Once the Root Bridge receives the TCN, it will send out a BPDU with the **Topology Change (TC) bit** set to all switches. When a switch receives this Root BPDU, it will temporarily lower its MAC-address Aging Timer from 300 seconds to **15 seconds**, so that any erroneous MAC addresses can be quickly flushed out of the CAM table.

The MAC-Address Aging Timer will stay lowered to 15 seconds for a period of **35 seconds** by default, or one Max Age (20 seconds) plus one Forward Delay (15 seconds) timer.

(Reference: http://www.cisco.com/en/US/tech/tk389/tk621/technologies_tech_note09186a0080094797.shtml)

Basic STP Configuration

To disable STP for a specific VLAN:

Switch(config)# *no spanning-tree vlan 10*

To adjust the Bridge Priority of a switch from its default of 32,768, to increase its chances of being elected Root Bridge of a VLAN:

Switch(config)# *spanning-tree vlan 10 priority 150*

To change an interface's Path Cost from its defaults:

Switch(config)# *int fa0/24*
Switch(config-if)# *spanning-tree cost 42*

To force a switch to become the Root Bridge:

Switch(config)# *spanning-tree vlan 10 root primary*

The *root primary* parameter in the above command automatically lowers the switch's priority to 24,576. If another switch on the network has a lower priority than 24,576, the above command will lower the priority by 4096 less than the priority of the other switch.

It is possible to assign a **Secondary Root Bridge** for redundancy. To force a switch to become a Secondary Root Bridge:

Switch(config)# *spanning-tree vlan 10 root secondary*

The *root secondary* parameter in the above command automatically lowers the switch's priority to 28,672.

To specify the *diameter* of the switching topology:

Switch(config)# *spanning-tree vlan 10 root primary diameter 7*

The *diameter* parameter in the preceding command indicates the length of the STP topology (number of switches). The maximum (and default) value for the diameter is *7*. Note that the switching topology can contain more than seven switches; however, each *branch* of the switching *tree* can only extend seven switches deep, from the Root Bridge.

The *diameter* command will also adjust the Hello, Forward Delay, and Max Age timers. This is the **recommended way** to adjust timers, as the hello timers are tuned specifically to the diameter of the switching network.

STP PortFast

PortFast allows switch ports that connect a host device (such as a printer or a workstation), to bypass the usual progression of STP states. Theoretically, a port connecting to a host device can never create a switching loop. Thus, Port Fast allows the interface to move from a *blocking* state to a *forwarding* state **immediately**, eliminating the normal 30 second STP delay.

To configure PortFast on an interface:

Switch(config)# *int fa0/10*
Switch(config-if)# *spanning-tree portfast*

To enable PortFast globally on all interfaces:

Switch(config)# *spanning-tree portfast default*

PortFast *should not* be enabled on switch ports connecting to another hub/switch, as this may result in a loop. Note that PortFast does *not* disable STP on an interface - it merely speeds up the convergence.

PortFast additionally **reduces unnecessary BPDU traffic**, as TCN BPDU's will not be sent out for state changes on a PortFast-enabled interface.

STP UplinkFast

Switches can have multiple uplinks to other *upstream* switches. If the multiple links are not placed in an EtherChannel, then at least one of the ports is placed into a *blocking* state to eliminate the loop.

If a **directly-connected** interface goes down, STP needs to perform a recalculation to bring the other interface out of a blocking state. As stated earlier, this calculation can take from 30 to 50 seconds.

UplinkFast allows the port in a *blocking* state to be held in standby-mode, and activated immediately if the *forwarding* interface fails. If multiple ports are in a *blocking* state, whichever port has the lowest Root Path Cost will become unblocked. The Root Bridge *cannot* have UplinkFast enabled.

UplinkFast is configured globally for all VLANs on the switch:

Switch(config)# *spanning-tree uplinkfast*

(Reference: http://www.cisco.com/en/US/docs/switches/lan/catalyst3750/software/release/12.2_35_se/configuration/guide/swstpopt.html)

STP BackboneFast

While UplinkFast allows faster convergence if a *directly-connected* interface fails, **BackboneFast** provides the same benefit is an **indirectly-connected** interface fails.

For example, if the Root Bridge fails, another switch will be elected the Root. A switch learning about the new Root Bridge must wait its Max Age timer to flush out the old information, before it will accept the updated info. By default, the Max Age timer is 20 seconds.

BackboneFast allows a switch to bypass the Max Age timer if it detects an **indirect failure** on the network. It will update itself with the new Root info immediately.

BackboneFast is configured globally, and should be implemented on all switches in the network when used:

Switch(config)# *spanning-tree backbonefast*

Protecting STP

STP is vulnerable to attack for two reasons:

- STP builds its topology information by accepting a neighboring switch's BPDU's.
- The Root Bridge is always determined by the lowest Bridge ID.

Switches with a low priority can be maliciously placed on the network, and elected the Root Bridge. This may result in a suboptimal or unstable STP topology.

Cisco implemented three mechanisms to protect the STP topology:

- **Root Guard**
- **BPDU Guard**
- **BPDU Filtering**

All three mechanisms are configured on an individual interface basis, and are *disabled* by default. When enabled, these mechanisms apply to all VLANs for that particular interface.

(Reference: http://www.cisco.com/en/US/docs/switches/lan/catalyst3750/software/release/12.2_35_se/configuration/guide/swstpopt.html)

Root Guard

Root Guard prevents an unauthorized switch from advertising itself as a Root Bridge.

> **Switch(config)#** *interface fa0/10*
> **Switch(config-if)#** *spanning-tree guard root*

The above command will prevents the switch from accepting a new Root Bridge off of the fa0/10 interface. If a Root Bridge advertises itself to this port, the port will enter a *root-inconsistent* state (a pseudo-blocking state):

> **Switch#** *show spanning-tree inconsistentports*

```
Name                 Interface            Inconsistency
-------------------- -------------------- ------------------
VLAN100              FastEthernet0/10     Root Inconsistent
```

BPDU Guard and BPDU Filtering

BPDU Guard is employed on interfaces that are PortFast-enabled. Under normal circumstances, a PortFast-enabled interface connects to a host device, and thus the interface should never receive a BPDU.

If another switch is accidentally or maliciously connected into a PortFast interface, BPDU Guard will place the interface into an **errdisable** state. More accurately, if an interface configured for BPDU Guard receives a BPDU, then the *errdisable* state will occur. To enable BPDU Guard:

> **Switch(config)#** *interface fa0/10*
> **Switch(config-if)#** *spanning-tree bpduguard enable*

To take an interface out of an errdisable state, simply disable and re-enable the interface:

> **Switch(config)#** *interface fa0/10*
> **Switch(config-if)#** *shutdown*
> **Switch(config-if)#** *no shutdown*

BPDU Filtering essentially disables STP on a particular interface, by preventing it from sending or receiving BPDU's:

> **Switch(config)#** *interface fa0/10*
> **Switch(config-if)#** *spanning-tree bpdufilter enable*

Unidirectional Link Detection (UDLD)

Most communication in a switching network is bi-directional. STP requires that switches send BPDU's bi-directionally to build the topology database. If a malfunctioning switch port only allows traffic one way, and the switch still sees that port as *up*, a loop can form without the switch realizing it.

Unidirectional Link Detection (UDLD) periodically tests ports to ensure bi-directional communication is maintained. UDLD sends out **ID frames** on a port, and waits for the remote switch to respond with its own ID frame. If the remote switch does not respond, UDLD assumes the interface has malfunctioned and become unidirectional.

By default, UDLD sends out ID frames every **15 seconds**, and must be enabled on both sides of a link. UDLD can run in two modes:

- **Normal Mode –** If a unidirectional link is detected, the port is *not* shut down, but merely flagged as being in an *undetermined* state
- **Aggressive Mode –** If a unidirectional link is detected, the port is placed in an *errdisable* state

UDLD can be enabled globally (but only for Fiber ports on the switch):

Switch(config)# *udld enable message time 20*
Switch(config)# *udld aggressive message time 20*

The *enable* parameter sets UDLD into normal mode, and the *aggressive* parameter is for aggressive mode (obviously). The *message time* parameter modifies how often ID frames are sent out.

UDLD can be configured on individual interfaces:

Switch(config-if)# *udld enable*
Switch(config-if)# *udld aggressive*
Switch(config-if)# *udld disable*

To view UDLD status on ports, or re-enable UDLD *errdisabled* ports:

Switch# *show udld*
Switch# *udld reset*

STP Troubleshooting Commands

To view STP information for a *specific* VLAN:

Switch# *show spanning-tree vlan 100*

```
VLAN0100
  Spanning tree enabled protocol ieee
  Root ID    Priority    24576
             Address     00a.5678.90ab
             Cost        19
             Port        24 (FastEthernet0/24)
             Hello Time  2 sec  Max Age 20 sec  Forward Delay 15 sec

  Bridge ID  Priority    32768  (priority 32768 sys-id-ext 1)
             Address     000c.1234.abcd
             Hello Time  2 sec  Max Age 20 sec  Forward Delay 15 sec
             Aging Time  300

Interface          Role  Sts   Cost         Prio.Nbr
------------------ ---   ----- ------------ ----------------
Fa0/24             Root  FWD   19           128.24
Fa0/23             Altn  BLK   19           128.23
```

To view STP information for *all* VLANS:

Switch# *show spanning-tree*

To view detailed STP interface information:

Switch# *show spanning-tree detail*

```
VLAN100 is executing the ieee compatible Spanning Tree protocol
Bridge Identifier has priority 32768, address 000c.1234.abcd
Configured hello time 2, max age 20, forward delay 15

<snip>

Port 23 (FastEthernet0/23) of VLAN100 is forwarding
Port path cost 19, Port priority 128, Port Identifier 128.23.
Designated root has priority 24576, address 00a.5678.90ab
Designated bridge has priority 24576, address 00a.5678.90ab
Designated port id is 128.23, designated path cost 0

<snip>
```

(Reference: http://www.cisco.com/en/US/docs/switches/lan/catalyst6500/ios/12.1E/native/command/reference/show4.html#wp1026768)

Rapid Spanning Tree Protocol (RSTP)

To further alleviate the 30 to 50 second convergence delays with STP, enhancements were made to the original IEEE 802.1D standard. The result was **802.1w**, or **Rapid Spanning Tree Protocol (RSTP)**.

RSTP is similar in many respects to STP. BPDU's are forwarded between switches, and a Root Bridge is elected, based on the lowest Bridge ID. Root Ports and Designated Ports are also elected. RSTP defines five **port types**:

- **Root Port –** Switch port on each switch that has the best Path Cost to the Root Bridge (same as STP).
- **Alternate Port –** A backup Root Port, that has a less desirable Path Cost. An Alternate Port is placed in a *discarding* state.
- **Designated Port –** Non-Root port that represents the best Path Cost for each network segment to the Root Bridge (same as STP). Designated ports are also referred to as **Point-to-Point** ports.
- **Backup Port –** A backup Designated Port, that has a less desirable Path Cost. A Backup Port is placed in a *discarding* state.
- **Edge Port –** A port connecting a host device, which is moved to a Forwarding state immediately. If an Edge Port receives a BPDU, it will lose its Edge Port status and participate in RSTP calculations. On Cisco Catalyst switches, any port configured with PortFast becomes an Edge Port.

The key benefit of RSTP is speedier convergence. Switches no longer require artificial Forwarding Delay timers to ensure a loop-free environment.

Switches instead perform a **handshake synchronization** to ensure a consistent topology table. During initial convergence, the Root Bridge and its directly-connected switches will place their interfaces in a *discarding* state. The Root Bridge and those switches will exchange BPDU's, synchronize their topology tables, and then place their interfaces in a forwarding state.

Each switch will then perform the same handshaking process with their downstream neighbors. The result is convergence that completes in a few seconds, as opposed to 30 to 50 seconds.

(Reference: http://www.cisco.com/en/US/tech/tk389/tk621/technologies_white_paper09186a0080094cfa.shtml)

Rapid Spanning Tree Protocol (RSTP) (continued)

Changes to the RSTP topology are also handled more efficiently than 802.1D STP.

Recall in that in 802.1D STP, a switch recognizing a topology change will send out a **TCN (Topology Change Notification)** BPDU, destined for the Root Bridge. Once the Root Bridge receives the TCN, it will send out a BPDU with the **Topology Change (TC) bit** set to all switches. When a switch receives this Root BPDU, it will temporarily lower its MAC-address Aging Timer from 300 seconds to **15 seconds**, so that any erroneous MAC addresses can be quickly flushed out of the CAM table.

In RSTP, a switch recognizing a topology change *does not* have to inform the Root Bridge first. Any switch can generate and forward a **TC BPDU**. A switch receiving a TC BPDU will **flush all MAC addresses learned on all ports**, except for the port that received the TC BPDU.

RSTP incorporates the features of UplinkFast by allowing Alternate and Backup ports to immediately enter a Forwarding state, if the primary Root or Designated port fails. RSTP also inherently employs the principles of BackboneFast, by not requiring an arbitrary Max Age timer for accepting inferior BPDU's if there is an indirect network failure.

802.1w RSTP is backwards-compatible with 802.1D STP. However, when RSTP switches interact with STP switches, RSTP loses its inherent advantages, as will perform according to 802.1D specifications.

Two separate standards of RSTP have been developed:

- **Rapid Per-VLAN Spanning Tree Protocol (RPVST+) –** Cisco's proprietary implementation of RSTP.
- **Multiple Spanning Tree (MST) –** The IEEE 802.1s standard or RSTP.

(Reference: http://www.cisco.com/en/US/tech/tk389/tk621/technologies_white_paper09186a0080094cfa.shtml)

Multiple Spanning Tree (MST)

Earlier in this guide, two types of STP were defined:

- **Common Spanning Tree (CST) –** All VLANs utilize one STP process
- **Per-VLAN Spanning Tree (PVST) –** Each VLAN is allotted its own STP process

PVST allows for more efficient traffic flow throughout the switching network. However, each VLAN must run its own *separate* STP process, often placing an extreme burden on the switch's processor.

Multiple Spanning Tree (MST) allows *groups* of VLANs to be allotted their own STP process. Each STP process is called an **instance.** MST separates the STP topology into **regions** that must contain identical parameters, including:

- **Configuration Name** - a 32-bit value similar to a VTP domain
- **Revision Number –** a 16-bit value that identifies the current MST configuration's revision.
- **VLAN-to-Instance Mappings**

Each **region** runs its own **Internal Spanning Tree (IST)** to eliminate loops within that region. IST is essentially an enhanced form of RSTP that supports MST-specific parameters.

MST is fully compatible with all other implementations of STP.

(Reference: http://www.cisco.com/en/US/docs/switches/lan/catalyst4500/12.2/31sg/configuration/guide/spantree.pdf)

MST Configuration

MST must first be enabled globally on a switch:

Switch(config)# *spanning-tree mode mst*

Most other MST configuration is completed in "MST Configuration" mode:

Switch(config)# *spanning-tree mst configuration*

To configure the switch's MST Configuration Name:

Switch(config-mst)# *name MYMSTNAME*

To configure the switch's Revision Number:

Switch(config-mst)# *revision 10*

To map VLANs to a specific MST instance:

Switch(config-mst)# *instance 2 vlan 1-100*

A maximum of **16 instances** are allowed (0 – 15). By default, all VLANs belong to **instance 0**. Recall that the above three parameters (configuration name, revision number, and mappings) must be identical on all MST switches in a region.

To view the changes to the configuration:

Switch(config-mst)# *show pending*

```
Pending MST configuration
Name [MYMSTNAME]
Revision 10
Instance   Vlans mapped
--------   ------------------------------------------------
0          101-4094
2          1-100
```

All other configuration of MST is identical to standard STP, with two exceptions. The parameter "*mst*" must be used, and all settings are applied to **instances** instead of **VLANs**.

Switch(config)# *spanning-tree mst 2 root primary*
Switch(config)# *spanning-tree mst 2 priority 32000*

The above two configurations are applied to MST Instance 2.