Troubleshooting Spanning−Tree Protocol and Related Design C

Table of Contents

Troubleshooting Spanning−Tree Protocol and Related Design Considerations

Contents

Introduction

The primary function of the spanning−tree algorithm (STA) is to cut loops created by redundant links in bridged networks. The Spanning−Tree Protocol (STP) operates at Layer 2 of the OSI model and, by the means of bridge protocol data units (BPDUs) exchanged between bridges, elects the ports that will eventually forward or block traffic. This protocol can fail in some specific cases and troubleshooting the resulting situation can be very difficult, depending on the design of the network. We can even say that in this particular area, the most important part of the troubleshooting is done before the problem occurs.

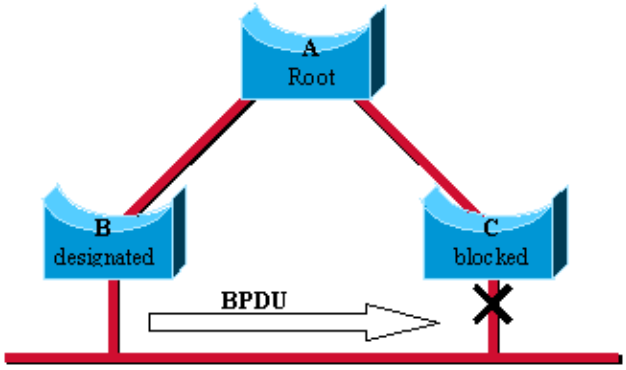
This document is not intended to be a complete LAN design guide, but just a list of recommendations that will help in implementing a safe network as far as bridging is concerned. Assuming knowledge of the protocol itself, we will introduce:

- The reasons that can cause the STP to fail.
- What information to look for in order to identify the source of the problem.
- What kind of design minimizes spanning tree risks and is easy to troubleshoot.

Spanning−Tree Protocol Failure

A failure in the STA generally leads to a bridging loop (not a spanning tree loop as you don't need STP to have a loop). Most customers calling the TAC for spanning tree problems are suspecting a bug, but experience proves that it is seldom the case. Even if the software is at stake, a bridging loop in a STP environment necessarily comes from a port that should block, but that is forwarding traffic.

What can cause a blocked port to go to forwarding? Let's rather recall why a port ends up in a blocking state. Each LAN has a single designated bridge. This bridge is responsible for the connectivity of the LAN towards the Root Bridge.



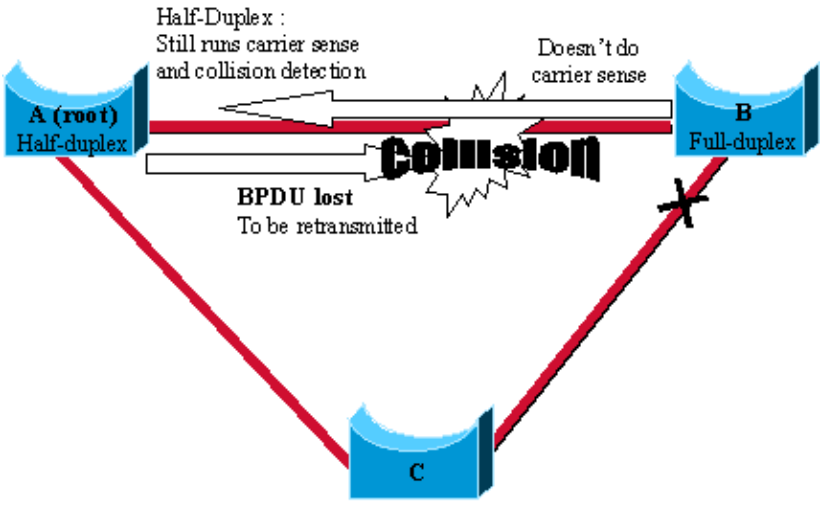
In this example, bridge B has been elected as designated bridge and bridge C is blocking because it is only providing an alternate path to the root. Why is C blocking and not B? This is determined practically by the BPDUs that B and C exchange on the LAN. Here, B had a better BPDU than C. B keeps sending BPDUs advertising its superiority over the other bridges on this LAN. Should C fail to receive these BPDUs for a certain period of time (called max age, 20 seconds by default), it would start a transition to the forwarding mode.

Important note: A port must keep receiving superior BPDUs to stay in blocking mode.

The rest of this document lists the different situations that can lead the STA to fail. Most of these failures are in fact related to a massive loss of BPDUs, causing blocked ports to transition to forwarding mode.

Duplex Mismatch

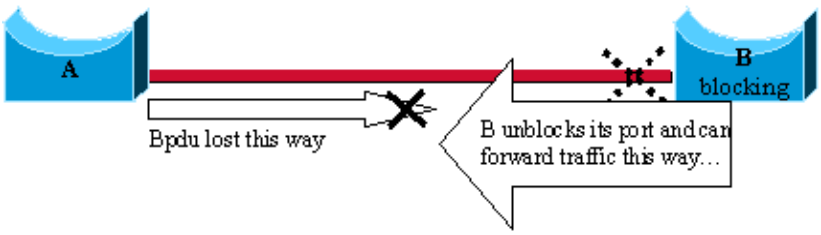
Duplex mismatch on a point−to−point link is a very common configuration error. This occurs especially when one side of the link is hardcoded full duplex. If you leave the other side in auto−negotiation mode, it will end up in half−duplex (a port with duplex hardcoded does not negotiate any more).



The worst case scenario is when a bridge sending BPDUs is configured for half−duplex on a link whereas its peer is configured full duplex. In the above example, the duplex mismatch on the link between bridge A and B can easily lead to a bridging loop. Because B is configured for full duplex, it does not perform carrier sense when accessing the link. B will then start sending frames even if A is already using the link. This is a problem for A that detects a collision and runs the backoff algorithm before attempting another transmission of its frame. The result is that, if there is enough traffic from B to A, every single packet (including the BPDUs) sent by A will be deferred or collisioned and eventually dropped. From a STP point of view, because it does not receive BPDUs from A any more, bridge B lost its root. This leads B to unblock its port to C, hence creating the loop.

Unidirectional Link

This is a very frequent cause for a bridging loop. Unidirectional links are often caused by a failure not detected on a fiber link for instance, or a problem with a transceiver. Anything that can lead a link to stay up while providing a one−way communication is very dangerous as far as STP is concerned. The example is very straightforward :



Here, let's suppose the link between A and B is unidirectional and drops traffic from A to B while transmitting traffic from B to A. Suppose that B should be blocking. We already mentioned that a port can only block if it receives BPDUs from a bridge that has a better priority. In this case, all these BPDUs coming from A are lost and bridge B eventually forwards traffic, creating a loop. Note that in this case, if the failure exists at startup, the STP will not converge correctly. It means that rebooting the bridges will have absolutely no effect (whereas it could temporarily help in the previous case).

Cisco introduced the Uni−Directional Link Detection (UDLD) protocol on high−end switches. This feature is able to detect wrong cabling or uni−directional links on Layer 2 and will automatically break resulting loops by disabling some ports. It is really worth running wherever possible in a bridged environment.

Packet Corruption

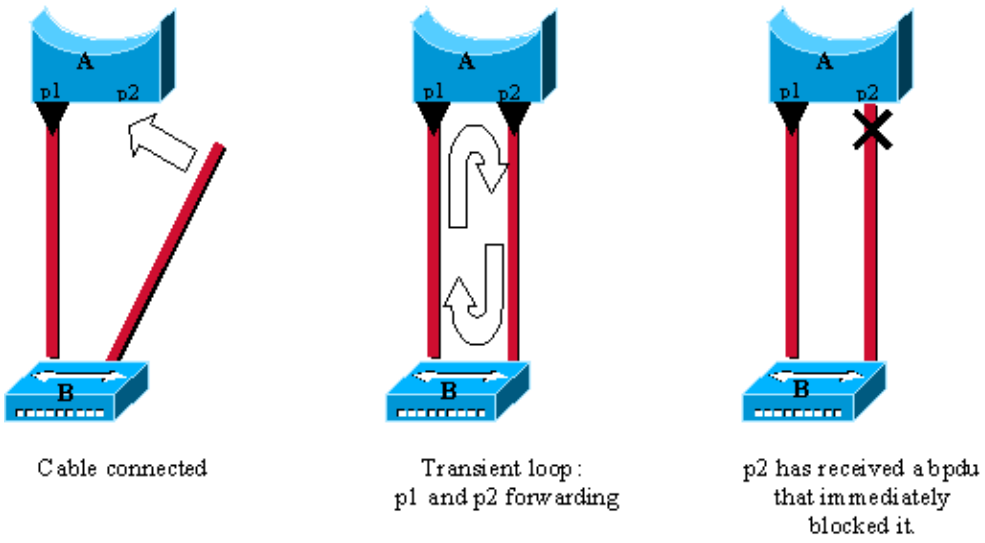
Packet corruption can also lead to the same kind of failure. If a link is experiencing a high rate of physical errors, a certain number of consecutive BPDUs could be lost, leading a blocking port to transition to forwarding. This case is rather seldom, because STP default parameters are very conservative. The blocking port would need to miss its BPDUs for 50 seconds before transitioning to forwarding, and a single BPDU successfully transmitted would break the loop. This case specially occurs when STP parameters have been adjusted without care (max age reduced for instance).

Resource Errors

Even on high−end switches that perform most of their switching functions in hardware using specialized Asics, STP is implemented in software. This means that if the CPU of the bridge is over−utilized for any reason, it is possible that it lacks resources to send out BPDUs. The STA is generally not very processor intensive and has priority over other processes. We will see in the Look for Resource Errors section below that there are some guidelines on the number of instances of STP that a particular platform can handle.

Portfast Configuration Error

Portfast is a feature that one typically wants to enable for a port connected to a host. When the link comes up on this port, the first stages of the STA are skipped and the port directly transitions to the forwarding mode. This can obviously be dangerous when not used correctly. Loops occurs then when moving a cable and **should** be transient only.



In this example, A is a bridge with port p1 already forwarding and port p2 configured for Portfast. B is a hub. As soon as the second cable is plugged into A, port p2 goes to forwarding and create a loop between p1 and p2. This will stop as soon as p1 or p2 receives a BPDU that will put one of these two ports in blocking mode. The problem with that kind of transient loop is that if the looping traffic is very intensive, the bridge may have trouble successfully sending the BPDU that will stop the loop. This can delay the convergence considerably. The latest high−end Catalyst software implement a feature called BPDU guard that will even disable the port if it is configured for portfast and receives a BPDU.

Awkward STP Parameter Tuning and Diameter Issues

We already saw that an aggressive value for the max−age parameter and the forward−delay could lead to a very unstable STP. The loss of some BPDUs can then cause a loop to appear. Another issue, not very known, is related to the diameter of the bridged network. The conservative default values for the STP impose a maximum network diameter of seven. This means that two distinct bridges in the network should not be more than seven hops away the one to the other. Part of this restriction is coming from the age field BPDU carry: when a BPDU is propagated from the root bridge towards the leaves of the tree, the age field is incremented each time it goes though a bridge. Eventually, when the age field of a BPDU goes beyond max age, it is discarded. Typically, this will occur if the root is too far away from some bridges of the network. This issue will impact convergence of the spanning tree.

Software Errors

As mentioned in the introduction, the STP is one of the very first features that was implemented in Cisco products. You can expect this feature to be very stable. Only interaction with new features, such as EtherChanneling, caused STP to fail in some very specific cases that have been addressed now. A software bug can be anything, so there is no way of really describing the issue it could introduce. Let's simply state again that the most dangerous situation would be to ignore some BPDUs or, generally speaking, having a blocking port transitioning to forwarding.

Troubleshooting a Failure

Unfortunately, there is no systematic procedure to troubleshoot a STP issue. This section will rather look like a checklist, recapitulating some of the actions available to you. Most of the indications given here rather apply to a bridging loop troubleshooting. Other failures of the STP leading to a loss of connectivity can be identified using a more conventional way (by exploring the path taken by traffic experiencing a problem).

Note that most of these troubleshooting steps assume connectivity to the different devices of the bridge network. This means having console access. During a bridging loop for example, you will probably not be able to Telnet.

Use the Diagram of the Network

You need to know some basic things about your network before troubleshooting a bridging loop.

You need to know at least:

- The topology of the bridged network.
- Where the root bridge is located.
- Where the blocked ports (and so the redundant links) are located.

This knowledge is essential at least for two reasons:

- How could you know what to fix in the network if you don't know how it should look like when it is working?

- Most of the troubleshooting steps are simply using **show** commands to try to identify error conditions. The knowledge of the network helps focusing on the critical ports on the key devices.

Identify a Bridging Loop

It used to be that a broadcast storm could have the same effect on the network. Nowadays, with high−speed links and devices providing switching at hardware level, it is nearly impossible that, for instance, a single server brings down a network by broadcasting. The real way of identifying for sure a bridging loop is to capture the traffic on a saturated link and to check that similar packets are seen multiple times.

But practically, if all users in a certain bridging domain have connectivity issue at the same time, you can already suspect a bridging loop.

Check the port utilization on your devices and look for abnormal values. See the Check Port Utilization section below.

On the Catalyst switches running a CatOS, you can easily check the overall backplane usage using the show system command. This command is very useful because it will give you not only the current usage of the switch backplane, but also specifies the peak usage (and its date). An unusual peak utilization shows you whether there has ever been a bridging loop or not on this device.

Restore Connectivity Quickly and Be Ready for Another Time

Break the Loop Disabling Ports

Bridging loops have extremely severe consequences on a bridged network. Administrators generally don't have time to look for the reason of the loop and prefer restoring connectivity as soon as possible. The easy way out in this case is to disable manually every single port that is providing redundancy in the network. If you have been able to identify a part of the network that is more affected, start disabling ports in this area. Even better, if possible, start with disabling ports that should be blocking. Each time you disable a port, check if connectivity is restored in the network as if you are hit by a bridging loop, its effect should stop immediately after you broke it. Knowing which port disabled stopped the loop, you are sure that the failure was located on a redundant path where this port was located. If this port should have been blocking, you have probably found the link on which the failure appeared.

Log STP Events on Devices Hosting Blocked Ports

If you couldn't precisely identify the source of the problem or if, for instance, the problem is only transient, enable the logging of STP event on the bridges and switches of the network experiencing the failure. If you want to limit the number of devices to configure, enable this logging at least on devices hosting blocked ports, as this is always the transition of a blocked port that creates a loop.

- IOS: Enter the exec command **debug spantree events** to enable STP debugging information being generated. Use the general config mode command logging buffered to capture this debug information in the device's buffers.

- CatOS: The command set logging level spantree 7 default increase the default level of STP related event to debugging. Be sure that you are logging a maximum amount of messages in the switch's buffers, using the set logging buffer 500 command.

You can also try to send these output to a syslog device. Unfortunately, when a bridging loop occurs, it is seldom that you can keep connectivity to a syslog server.

Check Ports

As mentioned before, the critical ports to be investigated first are the blocking ports. Here is a list of what you can look for on the different ports, with a quick description of the commands to enter for both IOS−based machines and CatOS−based switches.

Check that Blocked Ports Receive BPDUs

Especially on blocked ports and root ports, check you keep receiving BPDUs periodically. Several issues can lead to a port not receiving packets/BPDUs.

- IOS: If you are running an IOS release 12.0 or greater, the command show spanning−tree <bridge−group #> has a field named BPDU that will show you the number of BPDUs you received for each interface. Issuing the command once or twice more will quickly tell you if the device is receiving BPDUs.
- If you don't have the field BPDU on the output of the show spanning−tree command, then the easiest way of checking if you are receiving BPDUs is to simply enable STP debug with the **debug spantree tree** command.

CatOS: The show mac <module/port> command will tell you the number of multicast packets that a specific port receives. But the simplest is to use show spantree statistic <module#/port#> <vlan#>. This command displays the exact number of configuration BPDUs received for the specified port, on the specified VLAN (a port can belong to several VLANs if trunking). See the An Additional CatOS Command section below.

Check for Duplex−Mismatch

To look for a duplex−mismatch you obviously have to check on each side of the point−to−point link.

- IOS: Simply use the show interface command to check the speed and duplex status of the specified port.

- CatOS: The very first lines of the output of a show port <module#/port#> will give you the speed and duplex for which the port is configured.

Check Port Utilization

We have seen that an interface overloaded can fail to transmit vital BPDUs. A very loaded link is also an indication of a possible bridging loop.

- ◊ IOS: Use the command show interface to determine an interface utilization. Several fields will help you here (load, packets input/output and so on).

- ◊ CatOS: The command used to display statistics about packets received and sent on a port is show mac <module#/port#>. The command show top will automatically evaluate the port utilization over a 30 second period of time and display the result classified by percentage bandwidth utilization (other options available). Also, the show system command will give an indication on the backplane utilization, even if it does not point out to a specific port.

Check Packet Corruption

- ♦ IOS: Look for increasing figures in the input errors fields of the show interface command.

- ♦ CatOS: The command show port <module#/port#> will give you some details with the Aling−Err, FCS−Err, Xmit−Err, Rcv−Err, and Undersize fields. You will get even more detailed statistics using the show counters <module#/port#> command.

An Additional CatOS Command

The Catalyst specific software is richer than the IOS as far as STP troubleshooting is concerned. The command show spantree statistics <module#/port#> <vlan#> gives very accurate information on a specific port. On suspected ports, run this command and pay special attention to the fields:

- ◊ Forward trans count: this counter remembers how many times a port transition from learning to forwarding. In a stable topology, this counter should always show 1. This counter is reset to 0 is the corresponding port is going down and up. So, if the value is higher than 1, it means that the transition this port experienced is the result of a STP recalculation, not of a direct link failure.

- ◊ Max age expiry count: this counter tracks the number of times the max age expired on this link. Basically, a port expecting BPDUs will wait for max age (default 20 seconds) before considering its designated bridge as lost. Each time this event occurs, the counter is incremented. When the value is not zero, you know that for whatever reason, the designated bridge for this LAN is unstable or has problem transmitting its BPDUs.

Look for Resource Errors

We have seen that a high CPU utilization can be dangerous for a system running the STA. Here is how to

check that the device is not running short of CPU resource.

- IOS: Use the show processes cpu command. Check that the CPU utilization is not getting too close to 100%.

- CatOS: Look for the field RsrcErrors (resource error) in the output of a show inband (on some supervisors, this command is hidden under the name show biga). Basically, this counter is incremented when the processor was too overloaded to perform some of its tasks. There is a limitation on the number of different instances of STP a supervisor engine can handle. Check the release notes of the software you are running for this.

Here is a summary of the restrictions that apply to the Catalyst 4000/5000/6000 series:

Ensure that the total number of logical ports across all instances of STP for different VLANs does not exceed the maximum number supported for each supervisor engine type and memory configuration. You can use the **show spantree summary** command and this formula to compute the sum of logical ports on the switch:

(number of non−ATM trunks * number of active Vlans on that trunk)

+ J*(number of ATM trunks * number of active Vlans on that trunk)

+ number of non−trunking ports.

The sum of all logical ports, as calculated with the formula above, should be less than or equal to:

For the Catalyst 4000 series:

1500 for the Catalyst 4000 family Supervisor Engine I and II.

For the Catalyst 5000 series:

200 for Supervisor Engine I (with 8−MB DRAM)

400 for Supervisor Engine I (with 20−MB DRAM)

1500 for Supervisor Engine II and III F

1800 for Supervisor Engine II G and III G

4000 for Supervisor Engine III

For the Catalyst 6000 series:

4000 for Supervisor

Disable Unneeded Features

Troubleshooting is a matter of identifying what is currently wrong in the network. In this regard, disabling as many features as possible helps simplifying the network structure and eases the identification of the problem. EtherChanneling, for instance, is an advanced feature that needs STP to logically bundle several different links into a single one. It makes sense to disable this feature during a troubleshooting period. Again, this is just an example, but in a general matter, going to a configuration as simple as possible reduces the troubleshooting effort.

Useful Commands

Catalyst IOS Commands

- show interface
- show spanning−tree
- show bridge
- show processes cpu
- debug spantree
- logging buffered

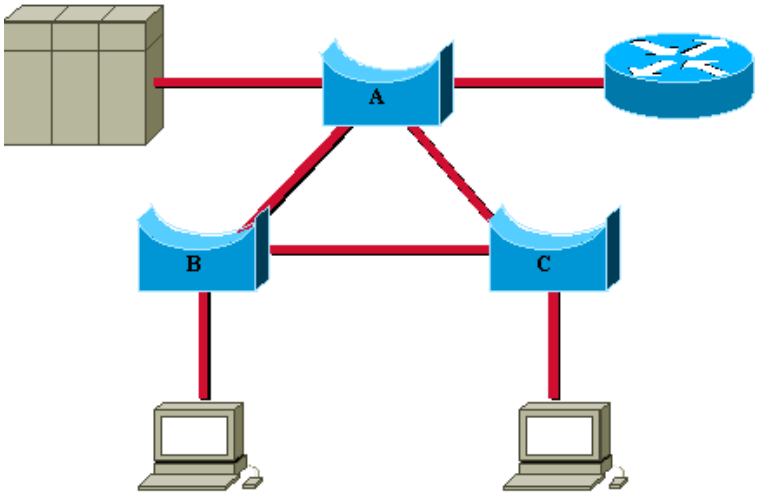
Catalyst OS Commands

- show port
- show mac
- show spantree
- show spantree statistics
- show spantree blockedports
- show spantree summary
- show top
- show inband/show biga
- show system
- show counters
- set spantree root [secondary]
- set spantree uplinkfast
- set logging level
- set logging buffered

Designing STP for Trouble Avoidance

Know Where The Root Is

It sounds trivial, but very often the information is not available at troubleshooting time. Don't leave the STP to decide which bridge will be root. Depending on the design of the network, you should be able to identify for each VLAN which switch is well suited to be root. Generally speaking, it is good to choose a powerful bridge in the middle of the network. Putting the Root Bridge in the center of the network, directly connected to the servers and routers, generally reduces the average distance from the clients to the servers and routers.



In this simple diagram, we clearly see that:

- If bridge B is root, link A to C will be blocked on A or C. In this case, hosts connected to switch B can access the server and the router in 2 hops, hosts connected to bridge C in 3 hops. That makes an average of 2.5 hops.
- If bridge A is root, the router and the server are reachable in two hops for both hosts connected on B and C. The average distance to them is now 2 hops.

This example is obvious, but it is the same kind of reasoning that is needed in more complex topologies.

Important Note: For each VLAN, hardcode the Root Bridge and the backup root bridge by reducing the value of the STP priority parameter (or using the set spantree root macro).

Know Where Redundancy Is

Plan the way your redundant links are organized. Here again, forget about the plug−and−play feature of the STP. Decide which ports will be blocking by turning the cost parameter of the STP. Hopefully, this is usually not necessary if you have a hierarchical design and a well located Root Bridge.

Important Note: For each VLAN, know which ports should be blocking in the stable network. Have a network diagram that clearly shows each physical loop in the network, and which blocked ports break the loops.

In case of accidental bridging loops, knowing exactly where the redundant links are helps identifying the loop and its cause. Knowing where the blocked ports should be also help finding where the error is coming from (by simple comparison).

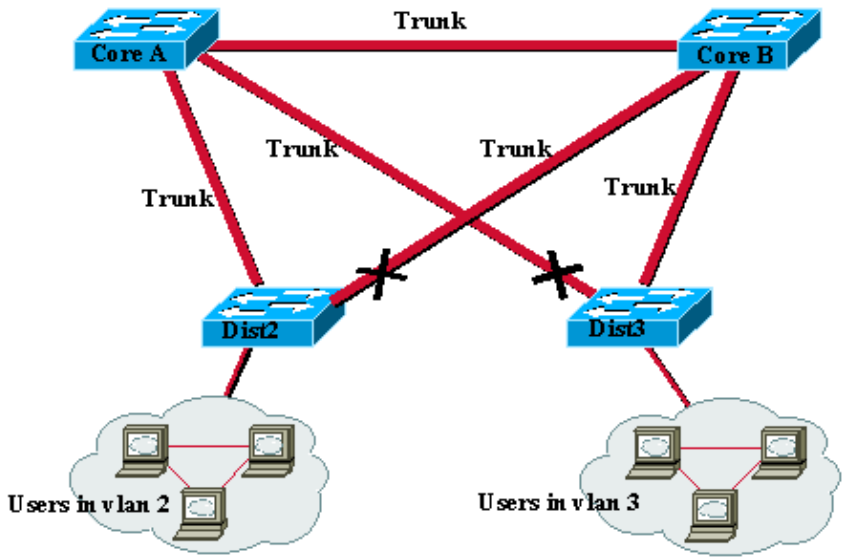
Minimize the Number of Blocked Ports

The only critical action taken by STP is blocking ports. A single blocking port transitioning to forwarding by error can meltdown a big part of the network. A good way of limiting the risk implied by the use of the STP

is to reduce as much as possible the number of blocked port.

Prune VLANs That Are Not Used

You don't need more than two redundant links between two nodes in a bridged network. Such configuration is however frequently seen:

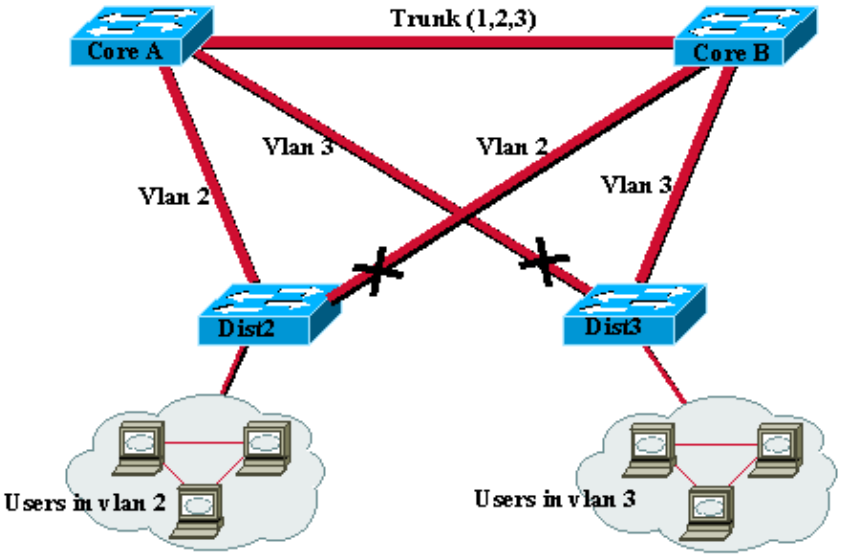


This is a very common design. Distribution switches are dual−attached to two core switches. Users connected on distribution switches are only in a subset of the VLANs available in the network (here, users connected on Dist2 are all in VLAN 2, Dist3 only connects users in VLAN 3). By default, trunks carry all the VLANs defined in the VTP domain. Now only Dist2 is receiving unnecessary broadcast and multicast traffic for VLAN 3, but it is also blocking one of its ports for VLAN 3. The result is that there are three redundant paths between Core A and Core B. This means more blocked ports and increase "chances" for a loop.

Important Note: Prune any VLAN not needed off your trunks.

VTP pruning can help doing this, but that kind of plug−and−play feature is not really needed in the core of the network.

Let's take the same example as previously. This time, we just use an access VLAN to connect the distribution switches to the core.



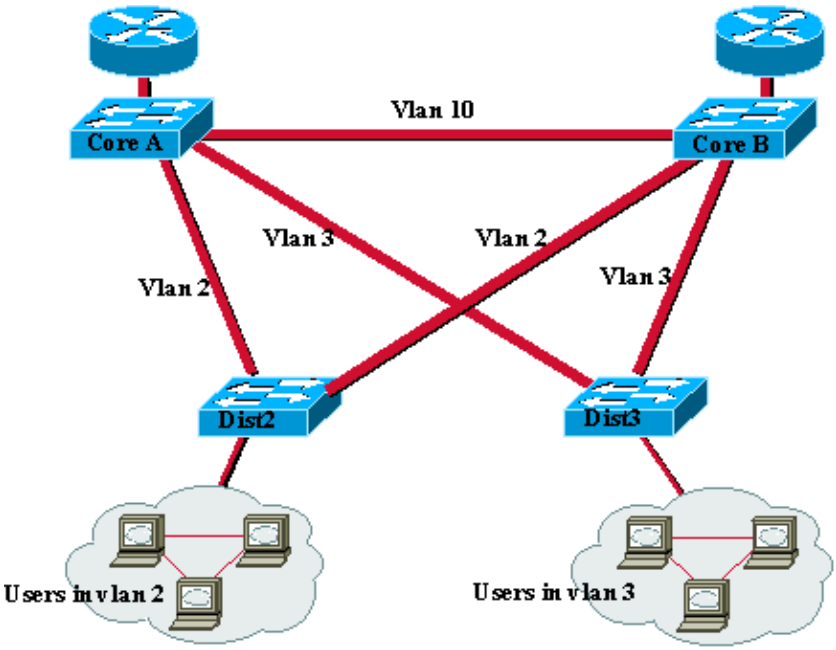
In this design, we only have one port blocked per VLAN. Note also that with this design, it is possible to remove all redundant links in just one step by shutting down Core A or Core B.

Use Layer 3 Switching

Layer 3 switching means approximately routing at the speed of switching. A router performs two main functions:

1. It builds a forwarding table, generally exchanging information with its peers by the way of routing protocols.
2. It receives packets and forwards them to the correct interface based on their destination address.

High−end Cisco Layer 3 switches are now able to perform this second function, at the same speed as Layer 2 switching function. There is no speed penalty in introducing a routing hop and creating an additional segmentation of the network. Using again the same diagram:



In this example, Core A and Core B are now some Layer 3 switches. Note that we are not bridging any more VLAN 2 and VLAN 3 between Core A and Core B, thus, we no longer have a loop to cut by the ways of the STP.

- Redundancy is still there, relying on Layer 3 routing protocols (and ensuring a reconvergence even faster that with STP).
- There is no longer any single port blocked by the STP. This removes all the potentiality of a bridging loop.
- There is no speed penalty as leaving the VLAN via Layer 3 switching is as fast as bridging inside the VLAN.

The only drawback is that migrating to that kind of design generally implies a re−work of the addressing scheme.

Keep STP Even if it is Not Needed

Even if you have succeeded in removing all the blocked ports of your network and even if you don't have any physical redundancy, it is safer to keep STP enabled. STP is generally not too processor intensive (and anyway, CPU is not involved in packet switching in most Cisco switches), and the few BPDUs sent on each link do not significantly reduce the available bandwidth. On the other end, a bridged network without STP can be melt down in a fraction of second by an operator who makes an error on a patch panel for instance.

Generally speaking, disabling the STP in a bridged network is not worth the risk.

Keep Traffic Off the Administrative VLAN and Avoid Having a Single VLAN Spanning the Entire Network

These two points are related.

A Cisco switch typically has a single IP address bound to a VLAN (which is often called the administrative VLAN). In this VLAN, the switch is behaving like a generic IP host. In particular, every single broadcast/multicast packet will be forwarded to the CPU. Having a high rate of broadcast/multicast on the administrative VLAN can hit badly the CPU and impact its ability of processing vital BPDUs. Therefore, it is always a good idea to keep off user traffic from the administrative VLAN.

Until recently, in Cisco implementation, there was no way of removing VLAN 1 from a trunk. This VLAN is generally used as an administrative VLAN, where all switches are accessible in the same IP subnet. Though useful, this may be dangerous because a bridging loop on VLAN 1 will affect all trunks and will probably bring the whole network down. Of course, the same problem exists whatever the VLAN. If possible, try to segment the bridging domains using high−speed Layer 3 switches.

As of version 5.4, the CatOS software allows the clearing of VLAN 1 on trunks (in fact, VLAN 1 is still existing but blocks traffic, thus preventing any loop possibility).

Avoid Tuning STP Parameters

Take special care if you plan to change STP timers from their default value. (Another option is to use CatOS macros.) Trying to get faster re−convergence from this, for instance, is very dangerous. It has implication on the diameter of the network and the stability of the STP. The only parameters you may want to change are the bridge priority (to select the Root Bridge) and the port cost or priority (to control redundancy and load balancing).

Cisco Catalyst software provides you with macros that will finely tune most important STP parameters for you:

- The set spantree root [secondary] command: this macro decreases the bridge priority so that it becomes root (or alternate root). You have an additional option that helps you tuning the STP timers by specifying the diameter of your network. Even when correctly done, timer tuning does not significantly improve the convergence time (specially compared to features like uplink fast or backbone fast or a good Layer 3 switching design) and introduces some instability risks in the network. That kind of tuning has to be updated each time a device is added into the network. It is better to keep the conservative default values, familiar to network engineers.

- The set spantree uplinkfast command increases the switch priority so that it cannot be root. You typically want to use that command on a distribution switch, at least dually attached to some core switches. Read the uplink fast feature documentation to learn more about the impact of this command.

Configure UDLD When Possible

In case of a unidirectional link occurring on a link having a blocked port, you have 50% chances of getting a bridging loop. This is the most dangerous possibility of STP failure, because the algorithm is not able to

handle this situation at all. Latest Catalyst software implements the Uni−Directional Link Detection (UDLD) feature that helps detecting this dangerous condition. This works on point−to−point links between Cisco devices only.

Related Information

- **LAN Technologies Technical Tips**