CHAPTER 23

Troubleshooting LAN Switching Environments

The sections in this chapter describe common LAN switch features and offer solutions to some of the most common LAN switching problems. The following items will be covered:

- LAN Switching Introduction
- General Switch Troubleshooting Suggestions
- Troubleshooting Port Connectivity Problems
- Troubleshooting Ethernet 10/100-Mb Half-/Full-Duplex Autonegotiation
- ISL Trunking on Catalyst 5000 and 6000 Family Switches
- Example of Configuring and Troubleshooting Ethernet 10/100-Mb Autonegotiation
- Configuring EtherChannel Switch-to-Switch on Catalyst 4000/5000/6000 Switches
- Using PortFast and Other Commands to Fix End-Station Startup Connectivity Problems
- Configuring and Troubleshooting IP Multilayer Switching
- Troubleshooting Spanning Tree Protocol and Related Design Considerations

LAN Switching Introduction

If you are new to LAN switching, then the following sections will take you through some of the main concepts related to switches. One of the prerequisites to troubleshooting any device is to know the rules under which it operates. Switches have become much more complex over the last few years as they have gained popularity and sophistication. The next few paragraphs describe some of the key concepts to know about switches.

Hubs and Switches

Because of the great demand placed on local-area networks, we have seen a shift from a shared-bandwidth network, using hubs and coaxial cable, to a dedicated bandwidth network, using switches. A hub allows multiple devices to be connected to the same network segment. The devices on that segment share the bandwidth with each other. If it is a 10-Mb hub and six devices are connected to six different ports on the hub, all six devices would share the 10 Mb of bandwidth with each other. A 100-Mb hub would share 100 Mb of bandwidth among the connected devices. In terms of the OSI model, a hub would be considered a Layer 1 (physical layer) device. It hears an electrical signal on the wire and passes it along to the other ports.

A switch can physically replace a hub in your network. A switch allows multiple devices to be connected to the same network, just like a hub does, but this is where the similarity ends. A switch allows each connected device to have dedicated bandwidth instead of shared bandwidth. The bandwidth between the switch and the device is reserved for communication to and from that device alone. Six devices connected to six different ports on a 10-Mb switch would each have 10 Mb of bandwidth to work with, instead of sharing that bandwidth with the other devices. A switch can greatly increase the available bandwidth in your network, which can lead to improved network performance.

Bridges and Switches

A basic switch would be considered a Layer 2 device. When we use the word *layer,* we are referring to the seven-layer OSI model. A switch does not just pass electrical signals along, like a hub does; instead, it assembles the signals into a frame (Layer 2) and then decides what to do with the frame. A switch determines what to do with a frame by borrowing an algorithm from another common networking device, a transparent bridge. Logically, a switch acts just like a transparent bridge would, but it can handle frames much faster than a transparent bridge (because of special hardware and architecture). When a switch decides where the frame should be sent, it passes the frame out the appropriate port (or ports). You can think of a switch as a device creating instantaneous connections between various ports, on a frame-by-frame basis.

VLANs

Because the switch decides on a frame-by-frame basis which ports should exchange data, it is a natural extension to put logic inside the switch to allow it to select ports for special groupings. This grouping of ports is called a virtual local-area network (VLAN). The switch makes sure that traffic from one group of ports never gets sent to other groups of ports (which would be routing). These port groups (VLANs) can each be considered an individual LAN segment.

VLANs are also described as being broadcast domains. This is because of the transparent bridging algorithm, which says that broadcast packets (packets destined for the “all devices” address) should be sent out all ports that are in the same group (that is, in the same VLAN). Therefore, all ports that are in the same VLAN are also in the same broadcast domain.

Transparent Bridging Algorithm

The transparent bridging algorithm and the Spanning-Tree Protocol are covered in more detail elsewhere (see Chapter 20, “Troubleshooting Transparent Bridging Environments”). When a switch receives a frame, it must decide what to do with that frame. It could ignore the frame, it could pass the frame out one other port, or it could pass the frame out many other ports.

To know what to do with the frame, the switch learns the location of all devices on the segment. This location information is placed in a CAM table (Content Addressable Memory, named for the type of memory used to store these tables). The CAM table shows, for each device, the device’s MAC address, out which port that MAC address can be found, and which VLAN this port is associated with. The switch continually does this *learning* process as frames are received into the switch. The switch’s CAM table is continually being updated.

This information in the CAM table is used to decide how a received frame should be handled. To decide where to send a frame, the switch looks at the *destination* MAC address in a received frame and then looks up that destination MAC address in the CAM table. The CAM table shows which port the frame should be sent out for that frame to reach the specified destination MAC address.

These are the basic rules that a switch will use in carrying out the frame forwarding responsibility:

If the destination MAC address is found in the CAM table, then the switch will send the frame out the port that is associated with that destination MAC address in the CAM table. This is called *forwarding*.

If the associated port to send the frame out is the same port on which the frame originally came in, then there is no need to send the frame back out that same port, and the frame is ignored. This is called *filtering*.

If the destination MAC address is not in the CAM table (the address is unknown), then the switch will send the frame out *all* other ports that are in the *same* VLAN as the received frame. This is called *flooding*. It will not flood the frame out the same port on which the frame was received.

If the destination MAC address of the received frame is the broadcast address (FFFF.FFFF.FFFF), then the frame is sent out all ports that are in the same VLAN as the received frame. This is also called *flooding*. The frame will not be sent out the same port on which the frame it was received.

Spanning-Tree Protocol

As we have seen, the transparent bridging algorithm floods unknown and broadcast frames out all the ports that are in the same VLAN as the received frame. This causes a potential problem. If the network devices running this algorithm are connected in a physical loop, then flooded frames (such as broadcasts) will be passed from switch to switch, around and around the loop forever. Depending on the physical connections involved, the frames may actually multiply exponentially as a result of the flooding algorithm, which can cause serious network problems.

There is a benefit to having a physical loop in your network: It can provide redundancy. If one link fails, there is still another way for the traffic to reach its destination. To allow the benefits derived from redundancy, without breaking the network because of flooding, a protocol called the Spanning-Tree Protocol was created. It was standardized in the IEEE 802.1d specification.

The purpose of the Spanning-Tree Protocol is to identify and temporarily block the loops in a network segment or VLAN. The switches run the Spanning-Tree Protocol, which involves electing a root bridge or switch. The other switches measure their distance from the root switch. If there is more than one way to get to the root switch, then there is a loop. The switches follow the algorithm to determine which ports should be blocked to break the loop. STP is dynamic; if a link in the segment fails, then ports that were originally blocking may possibly be changed to forwarding mode.

Trunking

Trunking is a mechanism that is most often used to allow multiple VLANs to function independently across multiple switches. Routers and servers may use trunking as well, which allows them to live simultaneously on multiple VLANs. If your network has only one VLAN in it, then you may never need trunking; if your network has more than one VLAN, however, you will probably want to take advantage of the benefits of trunking.

A port on a switch normally belongs to only one VLAN; any traffic received or sent on this port is assumed to belong to the configured VLAN. A trunk port, on the other hand, is a port that can be configured to send and receive traffic for many VLANs. It accomplishes this by attaching VLAN information to each frame, a process called “tagging” the frame. Also, trunking must be active on both sides of the link; the other side must be expecting frames that include VLAN information for proper communication to occur.

Different methods of trunking exist, depending on the media being used. Trunking methods for Fast Ethernet or Gigabit Ethernet are Inter-Switch Link (ISL) or 802.1q. Trunking over ATM uses LANE. Trunking over FDDI uses 802.10.

EtherChannel

EtherChannel is a technique that can be used when you have multiple connections to the same device. Instead of having each link function independently, EtherChannel groups the ports together to work as one unit. It distributes traffic across all the links and provides redundancy in case one or more links fail. EtherChannel settings must be the same on both sides of the links involved in the channel. Normally, the Spanning-Tree Protocol would block all these parallel connections between devices because they are loops; however, EtherChannel runs "underneath" Spanning-Tree Protocol so that the protocol thinks that all the ports within a given EtherChannel are only a single port.

Multilayer Switching

Multilayer switching (MLS) refers to the capability of a switch to forward frames based on information in the Layer 3 (and sometimes Layer 4) header. This usually applies to IP packets, but now it also can occur for IPX packets. The switch learns how to handle these packets by communicating with one or more routers. Using a simplified explanation, the switch watches how the router processes a packet, and then the switch takes over processing future packets in this same flow. Traditionally, switches have been much faster at switching frames than routers, so to have them offload traffic from the router can result in significant speed improvements. If something changes in the network, the router can tell the switch to erase its Layer 3 cache and build it from scratch again as the situation evolves. The protocol used to communicate with the routers is called Multilayer Switching Protocol (MLSP).

How to Learn About These Features

These are just some of the basic features that switches support. More are being added every day. It is important to understand how your switches work, which features you are using, and how those features should work. One of the best places to learn this information about Cisco switches is on Cisco's web site.

Go to www.cisco.com; under the section "Service & Support," select Technical Documents. From here, select Documentation Home Page to find documentation sets for all Cisco products. The "Multilayer LAN Switches" link will lead you to documentation for all Cisco LAN switches. To learn about the features of a switch, read the "Software Configuration Guide" for the particular release of software that you use. The software configuration guides give you background information about what the feature does and what commands to use to configure it on your switch. All this information is free on the web; you do not even need an account for this documentation because it is available to anyone. Some of these configuration guides can be read in an afternoon and are well worth the time spent.

Another part of Cisco's web site is populated by Cisco's Technical Assistance Center (TAC). It is filled with information designed to help you implement, maintain, and troubleshoot your network. Go to the TAC web site at: www.cisco.com/tac; from here, you can select Products Home Page to get detailed support information organized by specific products, or you can go to the Technologies Home Page to get support information on technology (Fast Ethernet, Spanning-Tree Protocol, trunking, and so on). TAC documents and online tools specific to LAN Technologies are here: www.cisco.com/warp/customer/473/. Some of the material on the TAC web site, and, in particular, the online tools, are accessible only to users with a Cisco support contract.

General Switch Troubleshooting Suggestions

Many ways exist by which to troubleshoot a switch. As the features of switches grow, the possible things that can break also increase. If you develop an approach or test plan for troubleshooting, you will be better off in the long run than if you just try a hit-and-miss approach. Here are some general suggestions for making your troubleshooting more effective:

- Take the time to become familiar with normal switch operation. Cisco's web site has a tremendous amount of technical information describing how Cisco switches work, as mentioned in the previous section. The configuration guides, in particular, are very helpful. Many cases opened with Cisco's Technical Assistance Center (TAC) are solved with information from the product configuration guides.
- For the more complex situations, have an accurate physical and logical map of your network. A physical map shows how the devices and cables are connected. A logical map shows what segments (VLANs) exist in your network and which routers provide routing services to these segments. A spanning-tree map is highly useful for troubleshooting complex issues. Because of a switch's capability to create different segments by implementing VLANs, the physical connections alone do not tell the whole story; you must know how the switches are configured to determine which segments (VLANs) exist and to know how they are logically connected.
- Have a plan. Some problems and solutions are obvious; some are not. The symptoms that you see in your network may be the result of problems in another area or layer. Before jumping to conclusions, try to verify in a structured way what is working and what is not. Because networks can be complex, it is helpful to isolate possible problem domains. One way of doing this is by using the OSI seven-layer model. For example, check the physical connections involved (Layer 1), check connectivity issues within the VLAN (Layer 2), check connectivity issues across different VLANs (Layer 3), and so on. Assuming a correct configuration on the switch, many of the problems that you encounter will be related to physical layer issues (physical ports and cabling). Today, switches are involved in Layer 3 and Layer 4 issues, incorporating intelligence to switch packets based on information derived from routers, or by actually having routers living inside the switch (Layer 3 or Layer 4 switching).
- Do not assume that a component is working without checking it first. This can save you a lot of wasted time. For example, if a PC is not capable of logging into a server across your network, many things could be wrong. Don't skip the basic things and assume that something works—someone might have changed something without telling you. It takes only a minute to check some of the basic things (for example, that the ports involved are connected to the right place and are active), which could save you many wasted hours.

Troubleshooting Port Connectivity Problems

If the port doesn't work, nothing works! Ports are the foundation of your switching network. Some ports have special significance because of their location in the network and the amount of traffic that they carry. These ports would include connections to other switches, routers, and servers. These ports can be more complicated to troubleshoot because they often take advantage of special features such as trunking and EtherChannel. The rest of the ports are significant as well because they connect the actual users of the network.

Many things can cause a port to be nonfunctional: hardware issues, configuration issues, and traffic issues. Let's look at these categories a little deeper.

Hardware Issues

This section discusses issues related to general hardware requirements, copper, and fiber.

General

Port functionality requires two working ports connected by a working cable (assuming that it is of the correct type). Most Cisco switches default to having a port in notconnect state, which means that it is currently not connected to anything but is willing to connect. If you connect a good cable to two switch ports in the notconnect state, the link light should become green for both ports, and the port status should be "connected," which means that the port is up as far as Layer 1 is concerned. The following paragraphs point out items to check if Layer 1 is not up.

Check the port status for both ports involved. Make sure that neither port involved in the link is shut down. The administrator could have manually shut down one or both ports. Software inside the switch could have shut down the port because of configuration error conditions (we will expand on this later). If one side is shut down and the other is not, the status on the enabled side will be notconnect (because it does not sense a neighbor on the other side of the wire). The status on the shut-down side would say something like "disable" or "errDisable" (depending on what actually shut down the port). The link will not come up unless both ports are enabled.

When you hook up a good cable (again, assuming that it is of the correct type) between two enabled ports, both ports should show a green link light within a few seconds. Also, the port state should show "connected" in the command-line interface (CLI). At this point, if you do not have link, your problem is limited to three things: the port on one side, the port on the other side, or the cable in the middle. In some cases, other devices are involved: media converters (fiber-to-copper, and so on), or, on Gigabit links, you may have gigabit interface connectors (GBICs). Still, this is a reasonably limited area to search.

Media converters can add noise to a connection or weaken the signal if they are not functioning correctly. They also add extra connectors that can cause problems, so this is another component to debug.

Check for loose connections. Sometimes a cable appears to be seated in the jack, but it actually isn't; unplug the cable and re-insert it. You should also look for dirt or broken or missing pins. Do this for both ports involved in the connection.

The cable could be plugged into the wrong port, which commonly happens. Make sure that both ends of the cable are plugged into the ports where you really want them.

You also can have a link on one side and not on the other. Check both sides for link. A single broken wire can cause this type of problem.

A link light does not guarantee that the cable is fully functional. It may have encountered physical stress that causes it to be functional at a marginal level. Usually you will notice this if the port has lots of packet errors.

To determine whether the cable is the problem, swap it with a known good cable. Don't just swap it with any other cable; make sure that you swap it with a cable that you know is good and is of the correct type.

If this is a very long cable run (underground, across a large campus, for example), then it would be nice to have a sophisticated cable tester. If you do not have a cable tester, you might consider the following:

- Trying different ports to see if they come up using this long cable
- Connecting the port in question to another port in the same switch, just to see if the port will link up locally
- Temporarily relocating the switches near each other so that you can try out a known good cable

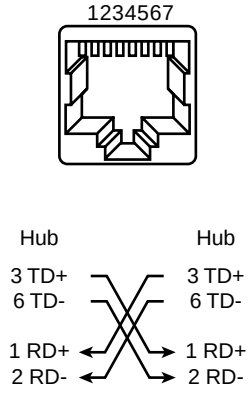
Copper

Make sure that you have the correct cable for the type of connection you are making. Category 3 cable can be used for 10 MB UTP connections, but Category 5 should be used for 10/100 connections.

A straight-through RJ-45 cable is used for end stations, routers, or servers to connect to a switch or hub. An Ethernet crossover cable is used for switch-to-switch or hub-to-switch connections. Below is the pin-out for an Ethernet crossover cable. Maximum distances for Ethernet or Fast Ethernet copper wires are 100 meters. A good general rule of thumb is that when crossing an OSI layer, such as between a switch and a router, use a straight-through cable; when connecting two devices in the same OSI layer, such as between two routers or two switches, use a crossover cable. For purposes of this rule only, treat a workstation like a router.

Figure 23-1 shows the pinouts required for a switch-to-switch crossover cable.

Figure 23-1 Illustration of the Pinouts Required for a Switch-to-Switch Crossover Cable



Fiber

For fiber, make sure that you have the correct cable for the distances involved and the type of fiber ports being used (single mode, multimode). Make sure that the ports being connected are both single-mode or both multimode ports. Single-mode fiber generally reaches 10 km, and multimode fiber can usually reach 2 km, but the special case of 100BaseFX multimode used in half-duplex mode can go only 400 meters.

For fiber connections, make sure that the transmit lead of one port is connected to the receive lead of the other port, and vice versa; transmit-to-transmit and receive-to-receive will not work.

For gigabit connections, GBICs must be matched on each side of the connection. There are different types of GBICs, depending on the cable and distances involved: short wavelength (SX), long wavelength/long haul (LX/LH), and extended distance (ZX). An SX GBIC needs to connect with an SX GBIC; an SX GBIC will not link with an LX GBIC. Also, some gigabit connections require conditioning cables, depending on the lengths involved. Refer to the GBIC installation notes (for examples, see www.cisco.com/univercd/cc/td/doc/product/lan/cat5000/cnfg_nts/ethernet/5399_01.htm).

If your gigabit link will not come up, check to make sure that the flow control and port negotiation settings are consistent on both sides of the link. There could be incompatibilities in the implementation of these features if the switches being connected are from different vendors. If in doubt, turn off these features on both switches.

Configuration Issues

Another cause of port connectivity issues is incorrect software configuration of the switch. If a port has a solid orange light, it means that software inside the switch shut down the port, either by way of the user interface or by internal processes.

Make sure that the administrator has not shut down the ports involved (as mentioned earlier). The administrator could have manually shut down the port on one side of the link. This link will not come up until you re-enable the port; check the port status.

Some switches, such as the Catalyst 4000/5000/6000, may shut down the port if software processes inside the switch detect an error. When you look at the port status, it will read "errDisable." You must fix the configuration problem and then manually take the port out of errDisable state. Some newer software versions—CatOS 5.4(1) and later—have the capability to automatically re-enable a port after a configurable amount of time spent in the errDisable state. Some of the causes for this errDisable state are listed here:

- **EtherChannel misconfiguration**—If one side is configured for EtherChannel and the other is not, it can cause the spanning-tree process to shut down the port on the side configured for EtherChannel. If you try to configure EtherChannel but the ports involved do not have the same settings (speed, duplex, trunking mode, and so on) as their neighbor ports across the link, then it could cause the errDisable state. It is best to set each side for the EtherChannel desirable mode if you want to use EtherChannel. The section "Configuring EtherChannel Switch-to-Switch Connections on Catalyst 4000/5000/6000 Switches" talks in depth about configuring EtherChannel.
- **Duplex mismatch**—If the switch port receives a lot of late collisions, this usually indicates a duplex mismatch problem. There are other causes for late collisions—such as a bad NIC or cable segments that are too long—but the most common reason today is a duplex mismatch. The full-duplex side thinks that it can send whenever it wants to, but the half-duplex side expects packets only at certain times, not at any time.
- **BPDU port guard**—Some newer versions of switch software can monitor whether PortFast is enabled on a port. A port using PortFast should be connected to an end station, not to devices that generate spanning-tree packets called BPDUs. If the switch notices a BPDU coming into a port that has PortFast enabled, it will put the port in errDisable mode.
- **Unidirectional Link Detection**—Unidirectional Link Detection (UDLD) is a protocol on some new versions of software that discovers whether communication over a link is one-way only. A broken fiber cable or other cabling/port issues could cause this one-way only communication. These partially functional links can cause problems when the switches involved do not know that the link is partially broken. Spanning-tree loops can occur with this problem. UDLD can be configured to put a port in errDisable state when it detects a unidirectional link.
- **Native VLAN mismatch**—Before a port has trunking turned on, it belongs to a single VLAN. When trunking is turned on, the port can carry traffic for many VLANs. The port will still remember the VLAN that it was in before trunking was turned on, which is called the native VLAN. The native VLAN is central to 802.1q trunking. If the native VLAN on each end of the link does not match, a port will go into the errDisable state.
- **Other**—Any process within the switch that recognizes a problem with the port can place it in the errDisable state.

Another cause of inactive ports occurs when the VLAN to which the ports belong disappears. Each port in a switch belongs to a VLAN. If that VLAN is deleted, then the port will become inactive. Some switches show a steady orange light on each port in which this has happened. If you come to work one day and see hundreds of orange lights, don't panic; it could be that all the ports belonged to the same

VLAN and someone accidentally deleted the VLAN to which the ports belong. When you add the VLAN back into the VLAN table, the ports will become active again because a port remembers its assigned VLAN.

If you have a link and the ports show that they are connected, but you cannot communicate with another device, this can be particularly perplexing. It usually indicates a problem above the physical layer: Layer 2 or Layer 3. Try the actions suggested in the next paragraphs.

Check the trunking mode on each side of the link. Make sure that both sides are in the same mode. If you turn the trunking mode to on (as opposed to auto or desirable) for one port, and the other port has the trunking mode set to off, the ports will not be capable of communicating. Trunking changes the formatting of the packet; the ports must be in agreement as to what format they are using on the link, or they will not understand each other.

Make sure that all devices are in the same VLAN. If they are not in the same VLAN, then a router must be configured to allow the devices to communicate.

Make sure that your Layer 3 addressing is correctly configured.

Traffic Issues

In this section, we describe some of the things you can learn by looking at a port's traffic information. Most switches have some way to track the packets going in and out of a port. Commands that generate this type of output on the Catalyst 4000/5000/6000 switches are **show port** and **show mac**. Output from these commands on the 4000/5000/6000 switches is described in the switch command references.

Some of these port traffic fields show how much data is being transmitted and received on the port. Other fields show how many error frames are being encountered on the port. If you have a large amount of alignment errors, FCS errors, or late collisions, this may indicate a duplex mismatch on the wire. Other causes for these types of errors may be bad network interface cards or cable problems. If you have a large number of deferred frames, it is a sign that your segment has too much traffic; the switch is not capable of sending enough traffic on the wire to empty its buffers. Consider removing some devices to another segment.

Switch Hardware Failure

If you have tried everything you can think of and the port will not work, there might be faulty hardware.

Sometimes ports are damaged by electrostatic discharge (ESD). You may or may not see any indication of this.

Look at the power-on self-test (POST) results from the switch to see whether any failures are indicated for any part of the switch.

If you see behavior that can be considered "strange," this could indicate hardware problems, but it could also indicate software problems. It is usually easier to reload the software than it is to get new hardware. Try working with the switch software first.

The operating system might have a bug. Loading a newer operating system could fix this. You can research known bugs by reading the release notes for the version of code that you are using or by using Cisco's Bug Navigator tool (www.cisco.com/support/bugtools).

The operating system could have somehow become corrupted. Reloading the same version of the operating system could fix the problem.

If the status light on the switch is flashing orange, this usually means that there is some kind of hardware problem with the port or the module or the switch. The same thing is true if the port or module status indicates "faulty."

Before exchanging the switch hardware, you might try a few things:

- Reseat the module in the switch. If you do this with the power on, make sure that the module is hot-swappable. If in doubt, turn off the switch before reseating the module, or refer to the hardware installation guide. If the port is built into the switch, ignore this step.
- Reboot the switch. Sometimes this causes the problem to disappear; this is a workaround, not a fix.
- Check the switch software. If this is a new installation, remember that some components may work with only certain releases of software. Check the release notes or the hardware installation and configuration guide for the component you are installing.
- If you are reasonably certain that you have a hardware problem, then replace the faulty component.

Troubleshooting Ethernet 10/100-Mb Half-/Full-Duplex Autonegotiation

This section presents general troubleshooting information and a discussion of techniques for troubleshooting Ethernet autonegotiation.

- This section shows how to determine the current behavior of a link. It goes on to show how users can control the behavior, and it also explains situations when autonegotiation will fail.
- Many different Cisco Catalyst switches and Cisco routers support autonegotiation. This section focuses on autonegotiation between Catalyst 5000 switches. However, the concepts explained here can be applied to the other types of devices.

Introduction

Autonegotiation is an optional function of the IEEE 802.3u Fast Ethernet standard that enables devices to automatically exchange information over a link about speed and duplex capabilities.

Autonegotiation is targeted at ports, which are allocated to areas where transient users or devices connect to a network. For example, many companies provide shared offices or cubes for account managers and system engineers to use when they are in the office rather than on the road. Each office or cube will have an Ethernet port permanently connected to the office's network. Because it may not be possible to ensure that every user has either a 10-Mb, a 100-Mb Ethernet, or a 10/100-Mb card in their laptops, the switch ports that handle these connections must be capable of negotiating their speed and duplex mode. The alternative would be to provide both a 10-Mb and a 100-Mb port in each office or cube and then label them accordingly.

Autonegotiation should not be used for ports that support network infrastructure devices such as switches and routers, or other nontransient end systems such as servers and printers. Although autonegotiation for speed and duplex is normally the default behavior on switch ports that are capable of it, ports connected to fixed devices should always be configured for the correct behavior rather than allowed to negotiate it. This eliminates any potential negotiation issues and ensures that you always know exactly how the ports should be operating. For example, a 10/100BaseTX Ethernet switch-to-switch link that has been configured for 100 Mb full-duplex will operate only at that speed and mode. There is no possibility for the ports to downgrade the link to a slower speed during a port reset or a switch reset. If the ports cannot operate as configured, they should stop passing any traffic. On the other

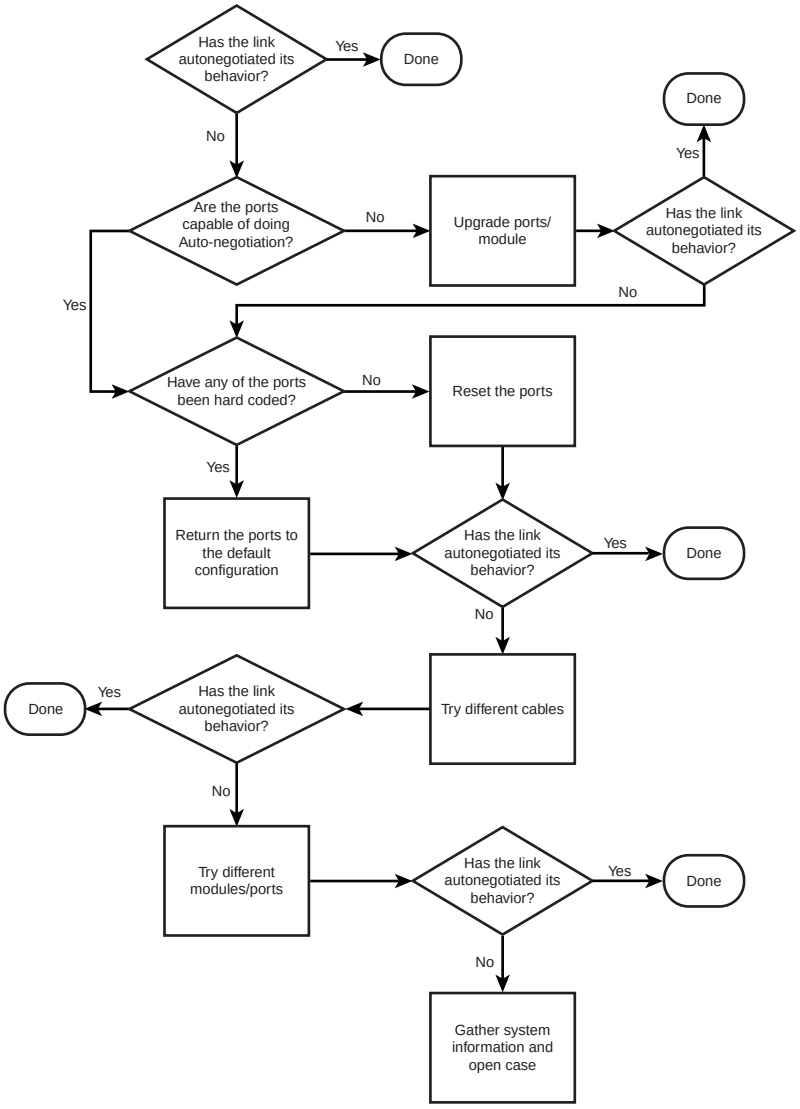
hand, a switch-to-switch link that has been allowed to negotiate its behavior could end up operating at 10 Mb half-duplex. A nonfunctional link is usually easier to discover than a link that is operational but not operating at the expected speed or mode.

One of the most common causes of performance issues on 10/100-Mb Ethernet links is one port on the link operating at half-duplex mode while the other port is operating at full-duplex mode. This occasionally happens when one or both ports on a link are reset and the autonegotiation process doesn't result in the same configuration for both link partners. It also happens when users reconfigure one side of a link and forget to reconfigure the other side. Many performance-related support calls will be avoided by creating a policy that requires ports for all nontransient devices to be configured for their required behavior and enforcing the policy with adequate change control measures.

Troubleshooting Ethernet Autonegotiation Between Network Infrastructure Devices

Figure 23-2 show the process you should follow in troubleshooting Ethernet autonegotiation between network infrastructure devices.

Figure 23-2 Troubleshooting Ethernet Autonegotiation



Procedures and Scenarios

Figure 23-3 shows a scenario using Cat 5k to Cat 5k, using Fast Ethernet.

Figure 23-3 Scenario 1: Cat 5K to Cat 5K, Using Fast Ethernet

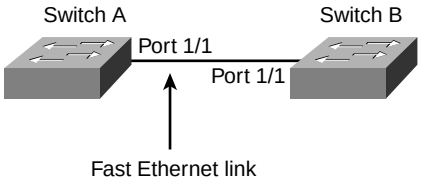


Table 23-1 Autonegotiation Connectivity Issues

Possible Problem	Solution
Was the current behavior of the link autonegotiated?	1. Use the **show port mod_num/port_num** command to determine the current behavior of the link. If both link partners (interfaces at either end of the link) have an "a-" prefix on their Duplex and Speed status fields, autonegotiation was probably successful.
Autonegotiation is not supported.	1. Issue the **show port capabilities mod_num/port_num** command to verify that your modules support autonegotiation.
Autonegotiation is not working on Catalyst switches.	1. Use the **set port speed mod_num/port_num auto** command on a Catalyst to configure autonegotiation. 2. Try different ports or modules. 3. Try resetting the ports. 4. Try different patch cables. 5. Turn the devices off and back on again.
Autonegotiation is not working on Cisco routers.	1. Issue the correct IOS command to enable autonegotiation (if available). 2. Try different interfaces. 3. Try resetting the interfaces. 4. Try different patch cables. 5. Turn the devices off and back on again.

Example of Configuring and Troubleshooting Ethernet 10/100-Mb Autonegotiation

This section walks you through examining the behavior of a 10/100-Mb Ethernet port that supports autonegotiation. It will also show how to make changes to its default behavior and how to restore it to the default behavior.

Tasks That Will Be Performed

In this section, you'll perform these tasks:

- Examine the capabilities of the ports.

- Configure autonegotiation for port 1/1 on both switches.
- Determine whether the speed and duplex mode are set to autonegotiate.
- Change the speed on port 1/1 in Switch A to 10 Mb.
- Understand the meaning of the "a-" prefix on the Duplex and Speed status fields.
- View the duplex status of port 1/1 on Switch B.
- Understand the duplex mismatch error.
- Understand the spanning-tree error messages.
- Change the duplex mode to half on port 1/1 on Switch A.
- Set the duplex mode and speed of port 1/1 on Switch B.
- Restore the default duplex mode and speed to ports 1/1 on both switches.
- View the changes of the port status on both switches.

The following steps are performed on the console of a Catalyst 5K switch.

Step 1 The **show port capabilities 1/1** command displays the capabilities of a Ethernet 10/100BaseTX 1/1 port on Switch A.

Enter this command for both of the ports that you are troubleshooting. Both ports must support the speed and duplex capabilities shown here if they are supposed to be using autonegotiation.

The *italic* text in the output shows where the information on the speed and duplex mode capabilities will be found.

```
Switch-A> (enable) show port capabilities 1/1
Model WS-X5530
Port 1/1
Type 10/100BaseTX
Speed auto,10,100
Duplex half,full
```

Step 2 Autonegotiation is configured for both speed and duplex mode on port 1/1 of both switches by entering the **set port speed 1/1 auto** command (auto is the default for ports that support autonegotiation).

```
Switch-A> (enable) set port speed 1/1 auto
Port(s) 1/1 speed set to auto detect.
Switch-A (enable)
```

Note The **set port speed {mod_num/port_num} auto** command also sets the duplex mode to auto. There is no **set port duplex {mod_num/port_num} auto** command.

Step 3 The **show port 1/1** command below displays the status of ports 1/1 on switches A and B.

```
Switch-A> (enable) show port 1/1
Port  Name               Status     Vlan       Level  Duplex Speed Type
----- ------------------ ---------- ---------- ------ ------ ----- ------------
 1/1                     connected  1          normal a-full a-100 10/100BaseTX
Switch-B> (enable) show port 1/1
Port  Name               Status     Vlan       Level  Duplex Speed Type
----- ------------------ ---------- ---------- ------ ------ ----- -----------
 1/1                     connected  1          normal a-full a-100 10/100BaseTX
```

The *italic* text in this output shows where the information on the current status of a port can be found. Note that most of the normal output from the **show port {mod_num/port_num}** command has been omitted.

The "a-" prefixes on the "full" and "100" indicate that this port has not been hard-coded (configured) for a specific duplex mode or speed. Therefore, it is willing to autonegotiate its duplex mode and speed if the device it is connected to (its link partner) is also willing to autonegotiate its duplex mode and speed.

Also note that the status shows "connected" on both ports, which means that a link pulse has been detected from the other port. The status can show "connected" even if duplex has been incorrectly negotiated or misconfigured.

Step 4 To demonstrate what happens when one link partner is autonegotiating and the other link partner is not, the speed on port 1/1 in Switch A will be set to 10 Mb by using the **set port speed 1/1 10** command.

```
Switch-A> (enable) set port speed 1/1 10
Port(s) 1/1 speed set to 10Mbps.
Switch-A> (enable)
```

Note Hard-coding the speed on a port disables all autonegotiation functionality on the port for speed and duplex.

When a port has been configured for a speed, its duplex mode will automatically be configured for the mode that it had previously negotiated—in this case, full duplex. Therefore, entering the **set port speed 1/1 10** command caused the duplex mode on port 1/1 to be configured as if the command **set port duplex 1/1 full** had also been entered. This is explained in the next step.

Step 5 Now you must understand the meaning of the "a-" prefix in the Duplex and Speed status fields.

The absence of the "a-" prefix in the status fields of the output from the **show port 1/1** command on Switch A shows that the duplex mode is now configured for full-duplex operation, and the speed is now configured for 10 Mb.

```
Switch-A> (enable) show port 1/1
Port  Name               Status     Vlan       Level  Duplex Speed Type
----- ------------------ ---------- ---------- ------ ------ ----- ------------
 1/1                     connected  1          normal  full  10    10/100BaseTX
```

Step 6 The **show port 1/1** command on Switch B indicates that the port is now operating at half-duplex and 10 Mb.

```
Switch-B> (enable) show port 1/1
Port  Name               Status     Vlan       Level  Duplex Speed Type
----- ------------------ ---------- ---------- ------ ------ ----- ------------
 1/1          connected  1       normal a-half a-10  10/100BaseTX
```

This step shows that it is possible for a link partner to detect the speed at which the other link partner is operating, even though the other link partner is not configured for autonegotiation. Sensing the type of electrical signal that is arriving to see if it is 10 Mb or 100 Mb does this. This is how Switch B determined that port 1/1 should be operating at 10 Mb.

It is not possible to detect the correct duplex mode in the same way that the correct speed can be detected. In this case, where Switch B's 1/1 port is configured for autonegotiation and Switch A's is not, Switch B's 1/1 port was forced to select the default duplex mode. On Catalyst Ethernet ports, the default mode is autonegotiate and, if autonegotiation fails, then is half-duplex.

This example also shows that a link can be successfully connected when there is a mismatch in the duplex modes. Port 1/1 on Switch A is configured for full-duplex operation, while port 1/1 on Switch B has defaulted to half-duplex operation. To avoid this, always configure both link partners.

The "a-" prefix on the Duplex and Speed status fields does not always mean that the current behavior was negotiated. Sometimes it means only that the port has not been configured for a speed or duplex mode.

The previous output from Switch B shows the Duplex field as "a-half" and the Speed field as "a-10," which indicates that the port is operating at 10 Mb in half-duplex mode. In this example, however, the link partner on this port (port 1/1 on Switch A) is configured for full-duplex mode and 10 Mb. Therefore, it was not possible for port 1/1 on Switch B to have autonegotiated its current behavior. This proves that the "a-" prefix indicates only a willingness to perform autonegotiation, not that autonegotiation actually took place.

Step 7 The following message about a duplex mode mismatch is displayed on Switch A after the speed on port 1/1 was changed to 10 Mb. The mismatch was caused by Switch B's 1/1 port defaulting to half-duplex mode because it sensed that its link partner was no longer performing autonegotiation.

%CDP-4-DUPLEXMISMATCH:Full/half duplex mismatch detected o1

It is important to note that this message is created by the Cisco Discovery Protocol (CDP), not the 802.3 autonegotiation protocol. CDP can report problems that it discovers, but it typically doesn't automatically fix them.

A duplex mismatch may or may not result in an error message. Another indication of a duplex mismatch is rapidly increasing FCS and alignment errors on the half-duplex side, and "runts" on the full-duplex port (as seen in a **sh port {mod_num/port_num}**).

Step 8 In addition to the duplex mismatch error message, you may also see the following spanning-tree messages when you change the speed on a link. A discussion of the Spanning-Tree Protocol is beyond the scope of this document; see the section found later in this chapter "Troubleshooting Spanning-Tree Protocol and Related Design Considerations," for more information.

```
%PAGP-5-PORTFROMSTP:Port 1/1 left bridge port 1/1
%PAGP-5-PORTTOSTP:Port 1/1 joined bridge port 1/1
```

Step 9 To demonstrate what happens when the duplex mode has been configured, the mode on port 1/1 in Switch A will be set to half-duplex mode using the **set port duplex 1/1 half** command.

```
Switch-A> (enable) set port duplex 1/1 half
Port(s) 1/1 set to half-duplex.
Switch-A> (enable)
```

The **show port 1/1** command shows the change in the duplex mode on this port.

```
Switch-A> (enable) sh port 1/1
Port  Name               Status     Vlan       Level  Duplex Speed Type
----- ------------------ ---------- ---------- ------ ------ ----- ------------
 1/1                     connected  1          normal half   10    10/100BaseTX
```

At this point, ports 1/1 on both switches are operating at half-duplex mode. Port 1/1 on Switch B, however, is still configured to autonegotiate, as shown in the following output of the **show port 1/1** command.

```
Switch-B> (enable) show port 1/1
Port  Name               Status     Vlan       Level  Duplex Speed Type
----- ------------------ ---------- ---------- ------ ------ ----- ------------
 1/1                     connected  1          normal a-half a-10  10/100BaseTX
```

The next step shows how to configure the duplex mode on port 1/1 in Switch B to half-duplex mode. This is in keeping with the recommended policy of always configuring both link partners in the same way.

Step 10 To implement the policy of always configuring both link partners for the same behavior, this step now sets the duplex mode to half-duplex and the speed to 10 on port 1/1 in Switch B.

Here is the output of entering the **set port duplex 1/1 half** command on Switch B:

```
Switch-B> (enable) set port duplex 1/1 half
Port 1/1 is in auto-sensing mode.
Switch-B> (enable)
```

The **set port duplex 1/1 half** command failed because this command won't work if autonegotiation is enabled. This also means that this command will not disable autonegotiation. Autonegotiation can be disabled only by using the **set port speed {mod_num/port_num** {**10** | **100**}**}** command.

Here is the output of entering the **set port speed 1/1 10** command on Switch B:

```
Switch-B> (enable) set port speed 1/1 10
Port(s) 1/1 speed set to 10 Mbps.
Switch-B> (enable)
```

Now the **set port duplex 1/1 half** command on Switch B will work:

```
Switch-A> (enable) set port duplex 1/1 half
Port(s) 1/1 set to half-duplex.
Switch-A> (enable)
```

The **show port 1/1** command on Switch B shows that the ports is now configured for half-duplex mode and 10 Mb.

```
Switch-B> (enable) show port 1/1
Port  Name               Status     Vlan       Level  Duplex Speed Type
----- ------------------ ---------- ---------- ------ ------ ----- ------------
 1/1                     connected  1          normal half   10    10/100BaseTX
```

Note The **set port duplex {mod_num/port_num** {**half** | **full** }**}** command is dependent on the **set port speed {mod_num/port_num** {**10** | **100** }**}** command. In other words, you must set the speed before you can set the duplex mode.

Step 11 Configure ports 1/1 on both switches to autonegotiate with the **set port speed 1/1 auto** command.

```
Switch-A> (enable) set port speed 1/1 auto
Port(s) 1/1 speed set to auto detect.
Switch-A> (enable)
```

Note When a port's duplex mode has been configured to something other than auto, the only way to configure the port to autosense its duplex mode is to issue the **set port speed {mod_num/port_num} auto** command. There is no **set port duplex {mod_num/port_num} auto** command. In other words, issuing the **set port speed {mod_num/port_num} auto** command has the effect of resetting both port speed sensing and duplex mode sensing to auto.

Step 12 Examine the status of ports 1/1 on both switches by using the **show port 1/1** command.

```
Switch-A> (enable) show port 1/1
Port  Name               Status     Vlan       Level  Duplex Speed Type
----- ------------------ ---------- ---------- ------ ------ ----- ------------
 1/1                     connected  1          normal a-full a-100 10/100BaseTX
Switch-B> (enable) show port 1/1
Port  Name               Status     Vlan       Level  Duplex Speed Type
----- ------------------ ---------- ---------- ------ ------ ----- ------------
 1/1                     connected  1          normal a-full a-100 10/100BaseTX
```

Both ports are now set to their default behavior of autonegotiation. Both ports have negotiated full-duplex mode and 100 Mb.

Before Calling Cisco Systems' TAC Team

Before calling Cisco Systems' Technical Assistance Center (TAC), make sure that you have read through this chapter and completed the actions suggested for your system's problem.

Additionally, do the following and document the results so that we can better assist you:

- Capture the output of **show version** from all the affected devices.
- Capture the output of **show port mod_num/port_num** from all the affected ports.
- Capture the output of **show port mod_num/port_num capabilities** from all the affected ports.

Additional Sources

- IEEE web site: www.ieee.org/

ISL Trunking on Catalyst 5000 and 6000 Family Switches

This section illustrates how to create a switch-to-switch Inter-Switch Link (ISL) trunk. Trunk ports enable connections between switches to carry traffic from more than one virtual local-area network (VLAN). Without trunking, a link between two switches can carry only traffic from one VLAN.

- This section shows how to determine the current behavior of a link. It goes on to show how users can control the behavior, and it also explains situations when autonegotiation will fail.
- Many different Cisco Catalyst switches and Cisco routers support autonegotiation. This section focuses on autonegotiation between Catalyst 5000 switches. However, the concepts explained here can be applied to the other types of devices.

Introduction

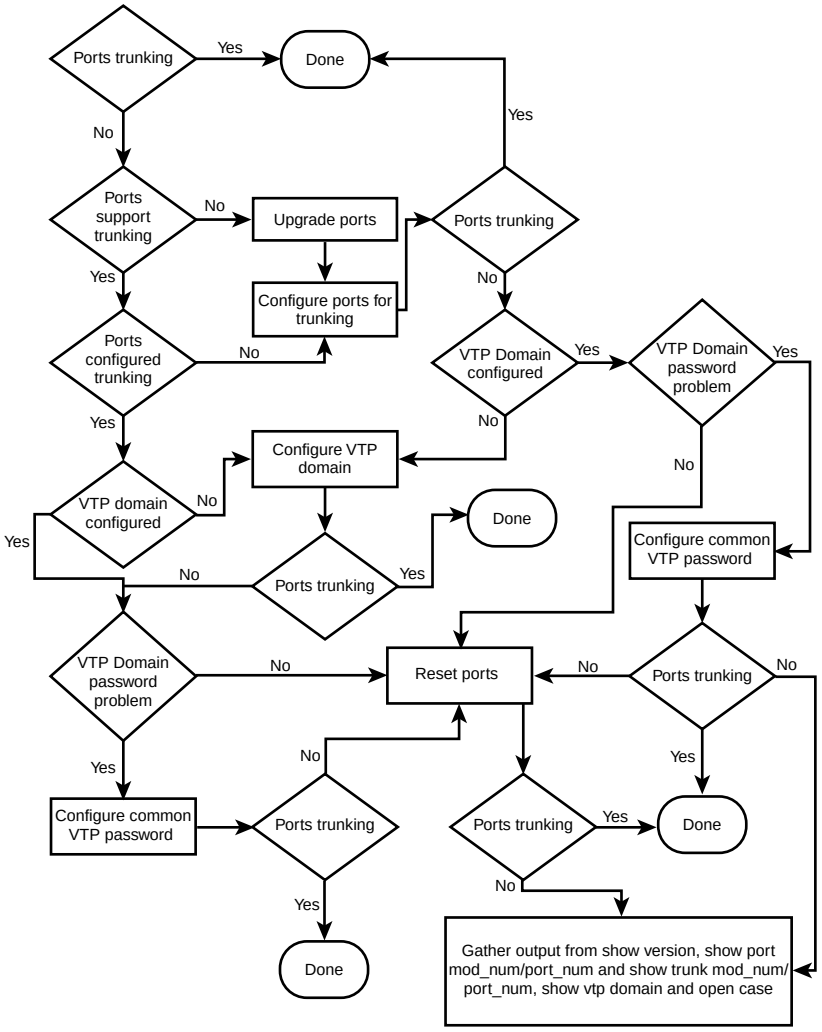
Trunking is not required in very simple switched networks with only one VLAN (broadcast domain). In most LANs, a small portion of traffic is made up of special protocols used for managing the network (Cisco Discovery Protocol, Virtual Trunking Protocol, Dynamic Trunking Protocol, Spanning-Tree Protocol, and Port Aggregation Protocol, to name a few examples). The management VLAN (VLAN 1) is also used when you **ping** or Telnet directly to or from the switch (this VLAN and the IP address of the switch are defined by configuring the sc0 interface, explained later). In a multi-VLAN environment, many network administrators advocate restricting this management traffic to its own VLAN, normally VLAN 1. User traffic is then configured to flow in VLANs other than this default VLAN.

ISL (Cisco proprietary) is one of two possible trunking protocols for Ethernet. The other protocol is the IEEE 802.1q standard.

Troubleshooting ISL Trunking on Catalyst 5000 and 6000 Family Switches

This section will walk the reader through some basic ISL trunking scenarios. The reader will learn basic ISL trunking configuration and troubleshooting skills (Figure 23-4).

Figure 23-4 Troubleshooting ISL Trunking on Catalyst 5000 and 6000 Family Switches



Procedures and/or Scenarios

Scenario: Cat 5K to Cat 5K, using Fast Ethernet for ISL trunking (Figure 23-5).

Figure 23-5 Cat 5K to Cat 5K, Using Fast Ethernet for ISL Trunking

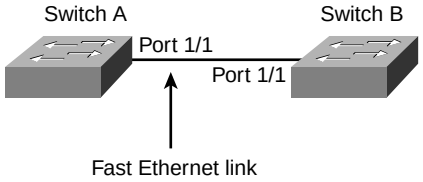


Table 23-2 ISL Trunking Issues

Possible Problem	Solution
Are the ports trunking?	1. Use the **show trunk mod_num/port_num** command to determine whether the ports are trunking.
Trunking is not supported.	1. Issue the **show port capabilities mod_num/port_num** command to verify that your modules support trunking.
Trunking is not configured.	1. Use the **set trunk mod_num/port_num desirable** on a Catalyst to configure trunking.
Verify that the VTP domain name has been configured.	1. Use the **show vtp domain** command to determine whether a domain name has been configured. All switches in a domain must have the same name. The name is case-sensitive.
VTP domain name is not configured.	1. Set **vtp domain domain_name**.
A VTP domain password problem has occurred.	1. Use the **show vtp domain** command to determine whether a password has been established. All switches must have the same password. The passwords are case-sensitive.

Example of Configuring and Troubleshooting ISL Trunking on Catalyst 5000 and 6000 Family Switches

This section walks you through configuring and troubleshooting ISL trunking.

Tasks That Will Be Performed

In this section, you will perform these tasks:

- Verify ISL support on the ports.
- Connect the switches.
- Verify that the ports are operational.
- Assign IP addresses to the management ports.
- Verify that the switches are not trunking over their link.
- Ping from switch to switch.
- Create a VLAN 2 in each switch.
- Move the management interface (sc0) to VLAN 2.

- Verify that you cannot **ping** from switch to switch.
- Configure the same VTP domain name in each switch.
- Enable trunking between the switches.
- Verify that the switches are trunking over their link.
- Ping from switch to switch.

Step-by-Step

The following steps are performed on the console of a Catalyst 5K switch.

Step 1 Make certain that the ports you have decided to use support ISL trunking. Several types of Ethernet interfaces support ISL trunking. 10BaseT (common Ethernet) ports do not support trunking; most 100BaseT (Fast Ethernet) ports do.

Use the **show port capabilities {module_number}|{module_number/port_number}** command on both switches to determine whether the ports that you are using support ISL. In this example, note that the port designator 1/1 has been specified at the end of the command. This limits the response to the information directly applicable to port 1/1.

```
Switch-A> show port capabilities 1/1
Model                     WS-X5530
Port                      1/1
Type                      10/100BaseTX
Speed                     auto,10,100
Duplex                    half,full
Trunk encap type          ISL
Trunk mode                on,off,desirable,auto,nonegotiate
Channel                   1/1-2
Broadcast suppression     percentage(0-100)
Flow control              no
Security                  yes
Membership                static,dynamic
Fast start                yes
QOS                       n/a
Rewrite                   no
UDLD                      Not capable
Switch-A>
```

Step 2 Connect the two switch ports using the Ethernet crossover cable. In this example, Switch A's 1/1 port is connected to Switch B's 1/1 port.

Step 3 Verify that the ports are operational by entering the **show port 1/1** command on Switch A. You should see that the status shows "connected."

```
Switch-A> (enable) show port 1/1
Port  Name               Status     Vlan       Level  Duplex Speed Type
----- ------------------ ---------- ---------- ------ ------ ----- ------------
1/1                      connected          1  normal a-full a-100 10/100BaseTX

Switch-A> (enable)
```

Step 4 Use the **set interface sc0 172.16.84.17 255.255.255.0 172.16.84.255** command on Switch A and the **set interface sc0 172.16.84.18 255.255.255.0 172.16.84.255** command on Switch B to assign IP addresses from the same subnet to the management ports on both switches. The VLAN for sc0 (the management VLAN) must also be specified in this command if it is different than the default of VLAN 1.

```
Switch-A> (enable) set int sc0 172.16.84.17 255.255.255.0 172.16.84.255
Interface sc0 IP address, netmask, and broadcast set.
Switch-A> (enable)
```

Step 5 Verify that the link between switches A and B is not trunking by entering the **show trunk 1/1** command on Switch A.

```
Switch-A> (enable) show trunk 1/1
Port      Mode         Encapsulation  Status        Native vlan
--------  -----------  -------------  ------------  -----------
 1/1      auto         isl            not-trunking  1

Port      Vlans allowed on trunk
--------  --------------------------------------------------------------------
 1/1      1-1005

Port      Vlans allowed and active in management domain
--------  --------------------------------------------------------------------
 1/1      1

Port      Vlans in spanning tree forwarding state and not pruned
--------  --------------------------------------------------------------------
 1/1      1
Switch-A> (enable)
```

Note The term **Native vlan** in the output indicates the VLAN in which this port will be placed when it is not in trunking mode. If the port is instead configured for 802.1q trunking, **Native vlan** also indicates the VLAN for whose frames will be untagged; all others will be tagged (conversely, with ISL trunking, every data frame is tagged with the appropriate VLAN identifier).

The trunking status should read “not-trunking” because the default mode for the Dynamic Trunking Protocol (DTP) is Auto. DTP is the strategic replacement for Dynamic ISL (DISL) because it incorporates support for 802.1q trunking negotiation. DTP is available as of version 4.x Catalyst software and certain hardware modules. The following bullets describe the five different states for which DTP can be configured.

- **Auto**—The port listens for DTP frames from the neighboring switch. If the neighboring switch indicates that it would like to be a trunk, or if it is a trunk, then Auto state creates the trunk with the neighboring switch. Auto does not propagate any intent to become an trunk; it depends solely on the neighboring switch to make the trunking decision.
- **Desirable**—DTP is spoken to the neighboring switch. This communicates to the neighboring switch that it is capable of being an ISL trunk and would like the neighboring switch to also be an ISL trunk.
- **On**—DTP is spoken to the neighboring switch. This automatically enables ISL trunking on its port, regardless of the state of its neighboring switch. It remains an ISL trunk unless it receives an ISL packet that explicitly disables the ISL trunk.
- **Nonegotiate**—DTP is not spoken to the neighboring switch. This automatically enables ISL trunking on its port, regardless of the state of its neighboring switch.
- **Off**—ISL is not allowed on this port, regardless of the DTP mode configured on the other switch.

On an individual trunk link, Cisco generally recommends configuring desirable trunking mode on the port nearest the network core, and auto trunking mode on the other side of the link. To hard-code trunking to be enabled, set the trunking mode on both sides to on; you will need to manually set the VLANs to be forwarded across the trunk link (use the full **set trunk** command) and ensure that the trunk settings are consistent on either side.

Step 6 **ping** Switch B from Switch A to verify that the switches can talk to each other over the link.

```
Switch-A> ping 172.16.84.18
172.16.84.18 is alive
Switch-A>
```

Step 7 Create VLAN 2 in Switch A by entering the **set vlan 2** command on Switch A. Switch B will learn about VLAN 2 after the DTP domain is established in Step 11.

```
Switch-A> (enable) set vlan 2
Vlan 2 configuration successful
Switch-A> (enable)
```

Step 8 Move the management interface in switches A and B to VLAN 2, which you created previously. Use the **set interface sc0 2** command to do this. The following output shows this being done on Switch A.

```
Switch-A> (enable) set int sc0 2
Interface sc0 vlan set.
Switch-A> (enable)
```

Use the **show interface** command to view the change that you just made. The following output shows this being done on Switch A.

```
Switch-A> (enable) sh int
sl0: flags=51<UP,POINTOPOINT,RUNNING>
        slip 0.0.0.0 dest 0.0.0.0
sc0: flags=63<UP,BROADCAST,RUNNING>
        vlan 2 inet 172.16.84.17 netmask 255.255.255.0 broadcast 172.16.84.255
Switch-A> (enable)
```

Step 9 Attempt to **ping** Switch B from Switch A. This should fail because the management ports are now in VLAN 2, while the link between the switches is in VLAN 1.

```
Switch-A> (enable) ping 172.16.84.18
no answer from 172.16.84.18
Switch-A> (enable)
```

Step 10 Establish the same VTP domain, named Cookbook, for both switches by entering the **set vtp domain Cookbook** command on both switches.

```
Switch-A> (enable) set vtp domain Cookbook
VTP domain Cookbook modified
Switch-A> (enable)
```

Step 11 Turn on trunking between the switches by configuring port 1/1 on Switch A for desirable mode by entering the **set trunk 1/1desirable** command on Switch A. Switch B will place its 1/1 port into trunking mode after the DTP negotiation between the two switches is complete.

```
Switch-A> (enable) set trunk 1/1 desirable
Port(s) 1/1 trunk mode set to desirable.
Switch-A> (enable)
```

You should see the following message as ISL becomes active:

```
1999 Aug 10 15:33:10 %DTP-5-TRUNKPORTON:Port 1/1 has become isl trunk
```

Possible Combinations of DTP Configurations

Table 23-3 shows the 15 possible unique combinations of DTP modes and indicates whether they will result in an active bidirectional trunk. Although it is theoretically possible to trunk in one direction on a link and not the other, it is not recommended.

Table 23-3 The Fifteen Possible Unique Combinations of DTP Modes

Switch A Port 1/1	Switch B Port 1/1	ISL Trunk Status
DTP Mode Auto	DTP Mode Auto	Not-trunking
DTP Mode Desirable	DTP Mode Auto	Trunking
DTP Mode ON	DTP Mode Auto	Trunking
DTP Mode Nonegotiate	DTP Mode Auto	Not-trunking
DTP Mode Off	DTP Mode Auto	Not-trunking
DTP Mode Desirable	DTP Mode Desirable	Trunking
DTP Mode On	DTP Mode Desirable	Not-trunking
DTP Mode Nonegotiate	DTP Mode Desirable	Not-trunking
DTP Mode Off	DTP Mode Desirable	Not-trunking
DTP Mode On	DTP Mode On	Trunking
DTP Mode Nonegotiate	DTP Mode On	Trunking
DTP Mode Off	DTP Mode On	Not-trunking
DTP Mode Nonegotiate	DTP Mode Nonegotiate	Trunking
DTP Mode Off	DTP Mode Nonegotiate	Not-trunking
DTP Mode Off	DTP Mode Off	Not-trunking

Before Calling Cisco Systems' TAC Team

Before calling Cisco Systems's Technical Assistance Center (TAC), make sure that you have read through this chapter and completed the actions suggested for your system's problem.

Additionally, do the following and document the results so that we can better assist you:

- Capture the output of **show version** from all the affected switches.
- Capture the output of **show vtp domain** from all the affected switches.
- Capture the output of **show trunk mod_num/port_num** from all the affected ports.
- Capture the output of **show port mod_num/port_num capabilities** from all the affected ports.

Configuring EtherChannel Switch-to-Switch Connections on Catalyst 4000/5000/6000 Switches

EtherChannel allows multiple physical Fast Ethernet or Gigabit Ethernet links to be combined into one logical channel. This allows load-sharing of traffic among the links in the channel, as well as redundancy in case one or more links in the channel fail. EtherChannel can be used to interconnect LAN switches, routers, servers, and clients via unshielded twisted-pair (UTP) wiring or single-mode and multimode fiber.

EtherChannel is an easy way to aggregate bandwidth between critical networking devices. On the Catalyst 5000, a channel can be created from two ports, making it a 200-Mbps link (400 Mbps full-duplex), or four ports, making it a 400-Mbps link (800 Mbps full-duplex). Some cards and platforms also support Gigabit EtherChannel and have the capability to use from two to eight ports in an EtherChannel. The concept is the same, no matter what speeds or number of links are involved. Normally, the Spanning-Tree Protocol would consider these redundant links between two devices to be loops and would cause the redundant links to be in blocking mode, effectively making these links inactive (providing only backup capabilities in case the main link fails). When using IOS 3.1.1 or greater, Spanning-Tree Protocol treats the channel as one big link, so all the ports in the channel can be active at the same time.

This section takes you through the steps for configuring EtherChannel between two Catalyst 5000 switches and shows you the results of the commands as they are executed. Catalyst 4000 and 6000 switches could have been used in the scenarios presented in this document to obtain the same results. For the Catalyst 2900XL and 1900/2820, the command syntax is different, but the EtherChannel concepts are the same.

EtherChannel may be configured manually by typing in the appropriate commands, or it may be configured automatically by having the switch negotiate the channel with the other side using the Port Aggregation Protocol (PAgP). It is recommended to use PAgP desirable mode to configure EtherChannel whenever possible because manually configuring EtherChannel can create some complications. This section gives examples of configuring EtherChannel manually and examples of configuring EtherChannel by using PAgP. Also included is how to troubleshoot EtherChannel and how to use trunking with EtherChannel. In this chapter, the terms *EtherChannel*, *Fast EtherChannel*, *Gigabit EtherChannel*, and *channel* will all refer to EtherChannel.

Contents

The following topics will be covered in this section:

- Tasks for manually configuring EtherChannel
- Verifying the EtherChannel configuration
- Using PAgP to automatically configure EtherChannel (preferred method)
- Trunking and EtherChannel
- Troubleshooting EtherChannel
- Commands used in this section

Figure 23-6 illustrates our test environment. The configuration of the switches has been cleared using the **clear config all** command. Then the prompt was changed using **set system name**. An IP address and mask were assigned to the switch for management purposes using **set int sc0 172.16.84.6 255.255.255.0** for Switch A and **set int sc0 172.16.84.17 255.255.255.0** for Switch B. A default gateway was assigned to both switches using **set ip route default 172.16.84.1**.

The switch configurations were cleared so that we could start from the default conditions. The switches were given names so that we can identify them from the prompt on the command line. The IP addresses were assigned so that we can **ping** between the switches for testing. The default gateway was not used.

Figure 23-6 Test Environment

Many of the commands display more output than is needed for our discussion. Extraneous output will be deleted.

Tasks for Manually Configuring EtherChannel

The following tasks will be performed to manually configure EtherChannel:

- Show the IOS version and modules that we are using in this chapter.
- Verify that EtherChannel is supported on the ports.
- Verify that the ports are connected and operational.
- Verify that the ports to be grouped have the same settings.
- Identify valid port groups.
- Create the channel.

Step-by-Step

The following steps will be done from the console of Switch-A and Switch-B:

Step 1 The **show version** command displays the software version that the switch is running. The **show module** command lists which modules are installed in the switch.

```
Switch-A show version
WS-C5505 Software, Version McpSW: 4.5(1) NmpSW: 4.5(1)
Copyright (c) 1995-1999 by Cisco Systems
…

Switch-A show module
Mod Module-Name          Ports Module-Type           Model    Serial-Num Status
--- ------------------- ----- --------------------- --------- --------- -------
1                        0     Supervisor III        WS-X5530  006841805 ok
2                        24    10/100BaseTX Ethernet WS-X5225R 012785227 ok
…
```

Step 2 Verify that EtherChannel is supported on the ports. The **show port capabilities** command appears in versions 4.x and greater. If you have an IOS earlier than 4.x, you must skip this step. Not every Fast Ethernet module supports EtherChannel. Some of the original EtherChannel modules have “Fast

EtherChannel" written on the bottom-left corner of the module (as you face it in the switch), which tells you that the feature is supported. However, this convention was abandoned on later modules. The modules in this test do not have "Fast EtherChannel" printed on them, but they do support the feature.

```
Switch-A show port capabilities
Model                     WS-X5225R
Port                      2/1
Type                      10/100BaseTX
Speed                     auto,10,100
Duplex                    half,full
Trunk encap type          802.1Q,ISL
Trunk mode                on,off,desirable,auto,nonegotiate
Channel                   2/1-2,2/1-4
Broadcast suppression     percentage(0-100)
Flow control              receive-(off,on),send-(off,on)
Security                  yes
Membership                static,dynamic
Fast start                yes
Rewrite                   yes
Switch-B show port capabilities
Model                     WS-X5234
Port                      2/1
Type                      10/100BaseTX
Speed                     auto,10,100
Duplex                    half,full
Trunk encap type          802.1Q,ISL
Trunk mode                on,off,desirable,auto,nonegotiate
Channel                   2/1-2,2/1-4
Broadcast suppression     percentage(0-100)
Flow control              receive-(off,on),send-(off,on)
Security                  yes
Membership                static,dynamic
Fast start                yes
Rewrite                   no
A port that does not support EtherChannel would look like this:
Switch show port capabilities
Model                     WS-X5213A
Port                      2/1
Type                      10/100BaseTX
Speed                     10,100,auto
Duplex                    half,full
Trunk encap type          ISL
Trunk mode                on,off,desirable,auto,nonegotiate
Channel                   no
Broadcast suppression     pps(0-150000)
Flow control              no
Security                  yes
Membership                static,dynamic
Fast start                yes
```

Step 3 Verify that the ports are connected and operational. Before connecting the cables, the port status is as follows:

```
Switch-A show port
Port  Name               Status     Vlan       Level  Duplex Speed Type
----- ------------------ ---------- ---------- ------ ------ ----- ------------
 2/1                     notconnect 1          normal   auto  auto 10/100BaseTX
 2/2                     notconnect 1          normal   auto  auto 10/100BaseTX
 2/3                     notconnect 1          normal   auto  auto 10/100BaseTX
 2/4                     notconnect 1          normal   auto  auto 10/100BaseTX
```

After connecting the cables between the two switches, the status is as follows:

```
1999 Dec 14 20:32:44 %PAGP-5-PORTTOSTP:Port 2/1 joined bridge port 2/1
1999 Dec 14 20:32:44 %PAGP-5-PORTTOSTP:Port 2/2 joined bridge port 2/2
1999 Dec 14 20:32:44 %PAGP-5-PORTTOSTP:Port 2/3 joined bridge port 2/3
1999 Dec 14 20:32:44 %PAGP-5-PORTTOSTP:Port 2/4 joined bridge port 2/4

Switch-A show port
Port  Name               Status     Vlan       Level  Duplex Speed Type
----- ------------------ ---------- ---------- ------ ------ ----- ------------
 2/1                     connected  1          normal a-full a-100 10/100BaseTX
 2/2                     connected  1          normal a-full a-100 10/100BaseTX
 2/3                     connected  1          normal a-full a-100 10/100BaseTX
 2/4                     connected  1          normal a-full a-100 10/100BaseTX
Switch-B  show port
Port  Name               Status     Vlan       Level  Duplex Speed Type
----- ------------------ ---------- ---------- ------ ------ ----- ------------
 2/1                     connected  1          normal a-full a-100 10/100BaseTX
 2/2                     connected  1          normal a-full a-100 10/100BaseTX
 2/3                     connected  1          normal a-full a-100 10/100BaseTX
 2/4                     connected  1          normal a-full a-100 10/100BaseTX
```

Because the switch configurations were cleared before starting this test, the ports are in their default conditions. They are all in vlan1, and their speed and duplex are set to auto. After connecting the cables, they negotiate to a speed of 100 Mbps and full-duplex mode. The status is connected, so we can **ping** the other switch.

```
Switch-A  ping 172.16.84.17
172.16.84.17 is alive
```

In your network, you may want to set the speeds manually to 100 Mbps and full-duplex mode instead of relying on autonegotiation because you will probably want your ports to always run at the fastest speed. For a discussion of autonegotiation, see the section "Troubleshooting Ethernet 10-/100-Mb Half-/Full-Duplex Autonegotiation."

Step 4 Verify that the ports to be grouped have the same settings. This is an important point and will be covered in more detail in the troubleshooting section. If the command to set up EtherChannel doesn't work, it is usually because the ports involved in the channel have differing configurations. This includes the ports on the other side of the link as well as the local ports. In our case, because the switch configurations were cleared before starting this test, the ports are in their default conditions. They are all in VLAN1, their speed and duplex are set to auto, and all spanning-tree parameters for each port are set the same. We saw from the previous output that after the cables are connected, the ports negotiate to a speed of 100 Mbps and full-duplex mode. Because Spanning-Tree Protocol runs for each VLAN, it is easier to just configure the channel and respond to error messages than to try to check every spanning-tree field for consistency for each port and VLAN in the channel.

Step 5 Identify valid port groups. On the Catalyst 5000, only certain ports can be put together into a channel. These restrictive dependencies do not apply to all platforms. The ports in a channel on a Catalyst 5000 must be contiguous. Notice from the **show port capabilities** command that for port 2/1, the possible combinations are these:

```
Switch-A  show port capabilities
Model                    WS-X5225R
Port                     2/1
...
Channel                  2/1-2,2/1-4
```

Notice that this port can be a part of a group of two (2/1-2) or part of a group of four (2/1-4). An Ethernet Bundling Controller (EBC) on the module causes these configuration limitations. Let's look at another port.

```
Switch-A  show port capabilities 2/3
Model                     WS-X5225R
Port                      2/3
...
Channel                   2/3-4,2/1-4
```

This port can be grouped into a group of two ports (2/3-4) or into a group of four (2/1-4).

Note Depending on the hardware, there might be additional restrictions. On certain modules (WS-X5201 and WS-X5203), you cannot form an EtherChannel with the last two ports in a "port group" unless the first two ports in the group already form an EtherChannel. A port group is a group of ports that are allowed to form an EtherChannel (2/1-4 is a port group in the previous example). For example, if you are creating separate EtherChannels with only *two* ports in a channel, you cannot assign ports 2/3-4 to a channel until you have first configured ports 2/1-2 to a channel, for the modules that have this restriction. Likewise, before configuring ports 2/6-7, you must configure ports 2/5-6. This restriction does not occur on the modules used for this document (WS-X5225R, WS-X5234).

Because we are configuring a group of four ports (2/1-4), this is within the approved grouping. We would not be able to assign a group of four to ports 2/3-6. This is a group of contiguous ports, but they do not start on the approved boundary, as shown by the **show port capabilities** command (valid groups would be ports 1-4, 5-8, 9-12, 13-16, 17-20, and 21-24).

Step 6 Create the channel. To create the channel, use the command **set port channel <mod/port on** for each switch. We recommend turning off the ports on one side of the channel using the **set port disable** command before turning on EtherChannel manually. This will avoid possible problems with Spanning-Tree Protocol during the configuration process. Spanning-Tree Protocol could shut down some ports (with a port status of errdisable) if one side is configured as a channel before the other side can be configured as a channel. Because of this possibility, it is much easier to create EtherChannels using PAgP, which we will cover in the section " Using PAgP to Configure EtherChannel " coming up later in this chapter. To avoid this situation when configuring EtherChannel manually, we will disable the ports on Switch A, configure the channel on Switch A, configure the channel on Switch B, and *then* re-enable the ports on Switch A.

First verify that channeling is off.

```
Switch-A (enable) show port channel
No ports channelling
Switch-B (enable) show port channel
No ports channelling
```

Now disable the ports on Switch A until both switches have been configured for EtherChannel so that Spanning-Tree Protocol will not generate errors and shut down the ports.

```
Switch-A (enable) set port disable 2/1-4
Ports 2/1-4 disabled.
[output from SwitchA upon disabling ports]
1999 Dec 15 00:06:40 %PAGP-5-PORTFROMSTP:Port 2/1 left bridg1
1999 Dec 15 00:06:40 %PAGP-5-PORTFROMSTP:Port 2/2 left bridge port 2/2
1999 Dec 15 00:06:40 %PAGP-5-PORTFROMSTP:Port 2/3 left bridge port 2/3
1999 Dec 15 00:06:40 %PAGP-5-PORTFROMSTP:Port 2/4 left bridge port 2/4
```

Turn the channel mode to on for Switch A.

```
Switch-A (enable) set port channel 2/1-4 on
Port(s) 2/1-4 channel mode set to on.
```

Check the status of the channel. Notice that the channel mode has been set to on, but the status of the ports is disabled (because we disabled them earlier). The channel is not operational at this point, but it will become operational when the ports are enabled.

```
Switch-A (enable) show port channel
Port  Status     Channel    Channel     Neighbor                  Neighbor
                 mode       status      device                    port
----- ---------- ---------- ----------- ------------------------- ----------
 2/1  disabled   on     channel
 2/2  disabled   on     channel
 2/3  disabled   on     channel
 2/4  disabled   on     channel
----- ---------- ---------- ----------- ------------------------- ----------
```

Because Switch A ports were (temporarily) disabled, Switch B ports no longer have a connection. The following message is displayed on Switch B's console when Switch A ports were disabled.

```
Switch-B (enable)
2000 Jan 13 22:30:03 %PAGP-5-PORTFROMSTP:Port 2/1 left bridge port 2/1
2000 Jan 13 22:30:04 %PAGP-5-PORTFROMSTP:Port 2/2 left bridge port 2/2
2000 Jan 13 22:30:04 %PAGP-5-PORTFROMSTP:Port 2/3 left bridge port 2/3
2000 Jan 13 22:30:04 %PAGP-5-PORTFROMSTP:Port 2/4 left bridge port 2/4
```

Turn on the channel for Switch B.

```
Switch-B (enable) set port channel 2/1-4 on
Port(s) 2/1-4 channel mode set to on.
```

Verify that channel mode is on for Switch B.

```
Switch-B (enable) show port channel
Port  Status     Channel    Channel     Neighbor                  Neighbor
                 mode       status      device                    port
----- ---------- ---------- ----------- ------------------------- ----------
 2/1  notconnect on     channel
 2/2  notconnect on     channel
 2/3  notconnect on     channel
 2/4  notconnect on     channel
----- ---------- ---------- ----------- ------------------------- ----------
```

Notice that the channel mode for Switch B is on, but the status of the ports is notconnect. That is because Switch A ports are still disabled.

Finally, the last step is to enable the ports on Switch A.

```
Switch-A (enable) set port enable 2/1-4
Ports 2/1-4 enabled.
1999 Dec 15 00:08:40 %PAGP-5-PORTTOSTP:Port 2/1 joined bridge port 2/1-4
1999 Dec 15 00:08:40 %PAGP-5-PORTTOSTP:Port 2/2 joined bridge port 2/1-4
1999 Dec 15 00:08:40 %PAGP-5-PORTTOSTP:Port 2/3 joined bridge port 2/1-4
1999 Dec 15 00:08:40 %PAGP-5-PORTTOSTP:Port 2/4 joined bridge port 2/1-4
```

Verifying the EtherChannel Configuration

To verify that the channel is set up properly, use the **show port channel** command.

```
Switch-A (enable) show port channel
Port  Status     Channel    Channel     Neighbor                  Neighbor
                 mode       status      device                    port
----- ---------- ---------- ----------- ------------------------- ----------
 2/1  connected  on         channel     WS-C5505    066509957(Sw  2/1
 2/2  connected  on         channel     WS-C5505    066509957(Sw  2/2
 2/3  connected  on         channel     WS-C5505    066509957(Sw  2/3
 2/4  connected  on         channel     WS-C5505    066509957(Sw  2/4
----- ---------- ---------- ----------- ------------------------- ----------

Switch-B (enable) show port channel
Port  Status     Channel    Channel     Neighbor                  Neighbor
                 mode       status      device                    port
----- ---------- ---------- ----------- ------------------------- ----------
 2/1  connected  on         channel     WS-C5505    066507453(Sw  2/1
 2/2  connected  on         channel     WS-C5505    066507453(Sw  2/2
 2/3  connected  on         channel     WS-C5505    066507453(Sw  2/3
 2/4  connected  on         channel     WS-C5505    066507453(Sw  2/4
----- ---------- ---------- ----------- ------------------------- ----------
```

Spanning-Tree Protocol is shown to treat the ports as one logical port in the following command. In the following output, when the port is listed as 2/1-4, this means that Spanning-Tree Protocol is treating ports 2/1, 2/2, 2/3, and 2/4 as one port.

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

Designated Root             00-10-0d-b2-8c-00
Designated Root Priority    32768
Designated Root Cost        8
Designated Root Port        2/1-4
Root Max Age   20 sec    Hello Time 2  sec   Forward Delay 15 sec

Bridge ID MAC ADDR          00-90-92-b0-84-00
Bridge ID Priority          32768
Bridge Max Age 20 sec    Hello Time 2  sec   Forward Delay 15 sec

Port      Vlan  Port-State     Cost   Priority  Fast-Start  Group-Method
--------- ----  -------------  -----  --------  ----------  ------------
 2/1-4   1     forwarding         8        32   disabled       channel
```

EtherChannel can be implemented with different ways of distributing the traffic across the ports in a channel. The EtherChannel specification does not dictate how the traffic should be distributed across the links in a channel. The Catalyst 5000 uses the last bit or the last 2 bits (depending on how many links are in the channel) of the source and destination MAC addresses in the frame to determine which port in the channel to use. You should see similar amounts of traffic on each of the ports in the channel, assuming that traffic is generated by a normal distribution of MAC addresses on one side of the channel. To verify that traffic is going over all the ports in the channel, you can use the show mac command. If your ports were active before configuring EtherChannel, then you may reset the traffic counters to zero by the clear counters command. Then the traffic values will represent how EtherChannel has distributed the traffic.

In our test environment, we did not get a real-world distribution because no workstations, servers, or routers are generating traffic. The only devices generating traffic are the switches themselves. We issued some pings from Switch A to Switch B, and you can tell from the following output that the unicast traffic is using the first port in the channel. The Receive information in this case (Rcv-Unicast) shows how Switch B distributed the traffic across the channel to Switch A. A little lower in the output, the Transmit information

```
(Xmit-Unicast) shows how Switch A distributed the traffic across the channel to Switch B.
We also see here that a small amount of switch-generated multicast traffic (Dynamic ISL,
CDP) goes out all four ports. The broadcast packets are ARP queries (for the default
gateway, which doesn't exist in our lab here). If we had workstations sending packets
through the switch to a destination on the other side of the channel, we would expect to
see traffic going over each of the four links in the channel. You can monitor the packet
distribution in your own network using the show mac command.
Switch-A (enable) clear counters
This command will reset all MAC and port counters reported in CLI and SNMP.
Do you want to continue (y/n) [n]? y
MAC and Port counters cleared.
Switch-A (enable) show mac

Port     Rcv-Unicast          Rcv-Multicast        Rcv-Broadcast
-------- -------------------- -------------------- --------------------
 2/1                        9                  320                  183
 2/2                        0                   51                    0
 2/3                        0                   47                    0
 2/4                        0                   47                    0
(...)

Port     Xmit-Unicast         Xmit-Multicast       Xmit-Broadcast
-------- -------------------- -------------------- --------------------
 2/1                        8                   47                  184
 2/2                        0                   47                    0
 2/3                        0                   47                    0
 2/4                        0                   47                    0
(...)

Port     Rcv-Octet            Xmit-Octet
-------- -------------------- --------------------
 2/1                    35176                17443
 2/2                     5304                 4851
 2/3                     5048                 4851
 2/4                     5048                 4851
(...)

Last-Time-Cleared
--------------------------
Wed Dec 15 1999, 01:05:33
```

Using PAgP to Automatically Configure EtherChannel (Preferred Method)

The Port Aggregation Protocol (PAgP) facilitates the automatic creation of EtherChannel links by exchanging packets between channel-capable ports. The protocol learns the capabilities of port groups dynamically and informs the neighboring ports.

When PAgP identifies correctly paired channel-capable links, it groups the ports into a channel. The channel is then added to the spanning tree as a single bridge port. A given outbound broadcast or multicast packet is transmitted out one port in the channel only, not out every port in the channel. In addition, outbound broadcast and multicast packets transmitted on one port in a channel are blocked from returning on any other port of the channel.

Four user-configurable channel modes exist: on, off, auto, and desirable. PAgP packets are exchanged only between ports in auto and desirable modes. Ports configured in on or off modes do not exchange PAgP packets. The recommended settings for switches that you want to form and EtherChannel is to have both switches set to desirable mode. This gives the most robust behavior in case one side encounters error situations or must be reset. The default mode of the channel is auto.

Both the auto and desirable modes allow ports to negotiate with connected ports to determine whether they can form a channel, based on criteria such as port speed, trunking state, native VLAN, and so on.

Ports can form an EtherChannel when they are in different channel modes, as long as the modes are compatible. For example:

- A port in desirable mode can form an EtherChannel successfully with another port that is in desirable or auto mode.
- A port in auto mode can form an EtherChannel with another port in desirable mode.
- A port in auto mode cannot form an EtherChannel with another port that is also in auto mode because neither port will initiate negotiation.
- A port in on mode can form a channel only with a port in on mode because ports in on mode do not exchange PAgP packets.
- A port in off mode will not form a channel with any port.

When using EtherChannel, if a "SPANTREE-2: Channel misconfig—*x/x-x* will be disabled" or similar syslog message is displayed, it indicates a mismatch of EtherChannel modes on the connected ports. We recommend that you correct the configuration and re-enable the ports by entering the **set port enable** command. Valid EtherChannel configurations include these:

Port Channel Mode	Valid Neighbor Port Channel Mode(s)
Desirable	Desirable or auto
Auto (default)[1]	Desirable or auto
On	On
Off	Off

1. If both the local and neighbor ports are in auto mode, an EtherChannel bundle will not form.

Table 23-4 is a summary of all the possible channeling mode scenarios. Some of these combinations may cause the Spanning-Tree Protocol to put the ports on the channeling side into errdisable state (that is, shut them down).

Table 23-4 Summary of All Possible Channeling Mode Scenarios

Switch A Channel Mode	Switch B Channel Mode	Channel State
On	On	Channel
On	Off	Not Channel (errdisable)
On	Auto	Not Channel (errdisable)
On	Desirable	Not Channel (errdisable)
Off	On	Not Channel (errdisable)
Off	Off	Not Channel
Off	Auto	Not Channel
Off	Desirable	Not Channel
Auto	On	Not Channel (errdisable)
Auto	Off	Not Channel
Auto	Auto	Not Channel

Table 23-4 Summary of All Possible Channeling Mode Scenarios

Auto	Desirable	Channel
Desirable	On	Not Channel (errdisable)
Desirable	Off	Not Channel
Desirable	Auto	Channel
Desirable	Desirable	Channel

We turned off the channel from the previous example using the following command on Switch A and Switch B:

Switch-A (enable) set port channel 2/1-4 auto

Port(s) 2/1-4 channel mode set to auto.

The default channel mode for a port that is capable of channeling is auto. To verify this, enter the following command:

```
Switch-A (enable) show port channel 2/1
Port  Status     Channel    Channel     Neighbor                  Neighbor
                 mode       status      device                    port
----- ---------- ---------- ----------- ------------------------- ----------
 2/1  connected  auto       not channel
```

The previous command also shows that currently the ports are not channeling. Another way to verify the channel state is as follows:

```
Switch-A (enable) show port channel
No ports channelling
Switch-B (enable) show port channel
No ports channelling
```

To make the channel work with PAgP is really very simple. At this point, both switches are set to auto mode, which means that they will channel if a connected port sends a PAgP request to channel. Setting Switch A to desirable causes Switch A to send PAgP packets to the other switch, asking it to channel.

```
Switch-A (enable) set port channel 2/1-4 desirable
Port(s) 2/1-4 channel mode set to desirable.
1999 Dec 15 22:03:18 %PAGP-5-PORTFROMSTP:Port 2/1 left bridg1
1999 Dec 15 22:03:18 %PAGP-5-PORTFROMSTP:Port 2/2 left bridge port 2/2
1999 Dec 15 22:03:18 %PAGP-5-PORTFROMSTP:Port 2/3 left bridge port 2/3
1999 Dec 15 22:03:18 %PAGP-5-PORTFROMSTP:Port 2/4 left bridge port 2/4
1999 Dec 15 22:03:19 %PAGP-5-PORTFROMSTP:Port 2/2 left bridge port 2/2
1999 Dec 15 22:03:19 %PAGP-5-PORTFROMSTP:Port 2/3 left bridge port 2/3
1999 Dec 15 22:03:20 %PAGP-5-PORTFROMSTP:Port 2/4 left bridge port 2/4
1999 Dec 15 22:03:23 %PAGP-5-PORTTOSTP:Port 2/1 joined bridge port 2/1-4
1999 Dec 15 22:03:23 %PAGP-5-PORTTOSTP:Port 2/2 joined bridge port 2/1-4
1999 Dec 15 22:03:23 %PAGP-5-PORTTOSTP:Port 2/3 joined bridge port 2/1-4
1999 Dec 15 22:03:24 %PAGP-5-PORTTOSTP:Port 2/4 joined bridge port 2/1-4
```

To view the channel, do as follows:

```
Switch-A (enable) show port channel
Port  Status     Channel   Channel     Neighbor                  Neighbor
                 mode      status      device                    port
----- ---------- --------- ----------- ------------------------- ----------
 2/1  connected  desirable channel     WS-C5505    066509957(Sw  2/1
 2/2  connected  desirable channel     WS-C5505    066509957(Sw  2/2
 2/3  connected  desirable channel     WS-C5505    066509957(Sw  2/3
 2/4  connected  desirable channel     WS-C5505    066509957(Sw  2/4
----- ---------- --------- ----------- ------------------------- ----------
```

Because Switch B was in auto mode, it responded to the PAgP packets and created a channel with Switch A.

```
Switch-B (enable)
2000 Jan 14 20:26:41 %PAGP-5-PORTFROMSTP:Port 2/1 left bridg1
2000 Jan 14 20:26:41 %PAGP-5-PORTFROMSTP:Port 2/2 left bridge port 2/2
2000 Jan 14 20:26:41 %PAGP-5-PORTFROMSTP:Port 2/3 left bridge port 2/3
2000 Jan 14 20:26:41 %PAGP-5-PORTFROMSTP:Port 2/4 left bridge port 2/4
2000 Jan 14 20:26:45 %PAGP-5-PORTFROMSTP:Port 2/2 left bridge port 2/2
2000 Jan 14 20:26:45 %PAGP-5-PORTFROMSTP:Port 2/3 left bridge port 2/3
2000 Jan 14 20:26:45 %PAGP-5-PORTFROMSTP:Port 2/4 left bridge port 2/4
2000 Jan 14 20:26:47 %PAGP-5-PORTTOSTP:Port 2/1 joined bridge port 2/1-4
2000 Jan 14 20:26:47 %PAGP-5-PORTTOSTP:Port 2/2 joined bridge port 2/1-4
2000 Jan 14 20:26:47 %PAGP-5-PORTTOSTP:Port 2/3 joined bridge port 2/1-4
2000 Jan 14 20:26:48 %PAGP-5-PORTTOSTP:Port 2/4 joined bridge port 2/1-4

Switch-B (enable) show port channel
Port  Status     Channel   Channel     Neighbor                  Neighbor
                 mode      status      device                    port
----- ---------- --------- ----------- ------------------------- ----------
 2/1  connected  auto      channel     WS-C5505    066507453(Sw  2/1
 2/2  connected  auto      channel     WS-C5505    066507453(Sw  2/2
 2/3  connected  auto      channel     WS-C5505    066507453(Sw  2/3
 2/4  connected  auto      channel     WS-C5505    066507453(Sw  2/4
----- ---------- --------- ----------- ------------------------- ----------
```

Note It is recommended to set both sides of the channel to desirable so that both sides will try to initiate the channel in case one side drops out. Setting the EtherChannel ports on Switch B to desirable mode, even though the channel is currently active and in auto mode, poses no problem. The command is as follows:

```
Switch-B (enable) set port channel 2/1-4 desirable
Port(s) 2/1-4 channel mode set to desirable.

Switch-B (enable) show port channel
Port  Status     Channel   Channel     Neighbor                  Neighbor
                 mode      status      device                    port
----- ---------- --------- ----------- ------------------------- ----------
 2/1  connected  desirable channel     WS-C5505    066507453(Sw  2/1
 2/2  connected  desirable channel     WS-C5505    066507453(Sw  2/2
 2/3  connected  desirable channel     WS-C5505    066507453(Sw  2/3
 2/4  connected  desirable channel     WS-C5505    066507453(Sw  2/4
----- ---------- --------- ----------- ------------------------- ----------
```

Now if Switch A drops out for some reason, or if new hardware replaces Switch A, then Switch B will try to re-establish the channel. If the new equipment cannot channel, then Switch B will treat its ports 2/1-4 as normal nonchanneling ports. This is one of the benefits of using the desirable mode. If the

channel was configured by using the PAgP on mode and one side of the connection has an error of some kind or a reset, it could cause an errdisable state (shutdown) on the other side. With PAgP set in desirable mode on each side. The channel will stabilize and renegotiate the EtherChannel connection.

Trunking and EtherChannel

EtherChannel is independent of trunking. You can turn trunking on, or you can leave trunking off. You also can turn on trunking for all the ports before creating the channel, or you can turn it on after creating the channel (as we will do here). As far as EtherChannel is concerned, it does not matter; trunking and EtherChannel are completely separate features. What does matter is that all the ports involved are in the same mode: Either they are all trunking before you configure the channel, or they are all not trunking before you configure the channel. All the ports must be in the same trunking state before creating the channel.

After a channel is formed, whatever is changed on one port is also changed for the other ports in the channel. The modules used in this test bed can do ISL or 802.1q trunking. By default, the modules are set to auto trunking and negotiate mode, which means that they will trunk if the other side asks them to trunk, and they will negotiate whether to use the ISL or 802.1q method for trunking. If they are not asked to trunk, they will work as normal nontrunking ports.

```
Switch-A (enable) show trunk 2
Port      Mode         Encapsulation  Status        Native vlan
--------  -----------  -------------  ------------  -----------
 2/1      auto         negotiate      not-trunking  1
 2/2      auto         negotiate      not-trunking  1
 2/3      auto         negotiate      not-trunking  1
 2/4      auto         negotiate      not-trunking  1
```

There are a number of different ways to turn on trunking. For this example, we will set Switch A to desirable. Switch A is already set to negotiate. The combination desirable/negotiate will cause Switch A to ask Switch B to trunk and to negotiate the type of trunking to do (ISL or 802.1q). Because Switch B defaults to autonegotiate, Switch B will respond to Switch A's request. The following results occur:

```
Switch-A (enable) set trunk 2/1 desirable
Port(s) 2/1-4 trunk mode set to desirable.
Switch-A (enable)
1999 Dec 18 20:46:25 %DTP-5-TRUNKPORTON:Port 2/1 has become isl trunk
1999 Dec 18 20:46:25 %DTP-5-TRUNKPORTON:Port 2/2 has become isl trunk
1999 Dec 18 20:46:25 %PAGP-5-PORTFROMSTP:Port 2/1 left bridge port 2/1-4
1999 Dec 18 20:46:25 %PAGP-5-PORTFROMSTP:Port 2/2 left bridge port 2/1-4
1999 Dec 18 20:46:25 %DTP-5-TRUNKPORTON:Port 2/3 has become isl trunk
1999 Dec 18 20:46:26 %PAGP-5-PORTFROMSTP:Port 2/3 left bridge port 2/1-4
1999 Dec 18 20:46:26 %DTP-5-TRUNKPORTON:Port 2/4 has become isl trunk
1999 Dec 18 20:46:26 %PAGP-5-PORTFROMSTP:Port 2/4 left bridge port 2/1-4
1999 Dec 18 20:46:28 %PAGP-5-PORTTOSTP:Port 2/1 joined bridge port 2/1-4
1999 Dec 18 20:46:29 %PAGP-5-PORTTOSTP:Port 2/2 joined bridge port 2/1-4
1999 Dec 18 20:46:29 %PAGP-5-PORTTOSTP:Port 2/3 joined bridge port 2/1-4
1999 Dec 18 20:46:29 %PAGP-5-PORTTOSTP:Port 2/4 joined bridge port 2/1-4

Switch-A (enable) show trunk 2
Port      Mode         Encapsulation  Status        Native vlan
--------  -----------  -------------  ------------  -----------
 2/1      desirable    n-isl          trunking      1
 2/2      desirable    n-isl          trunking      1
 2/3      desirable    n-isl          trunking      1
 2/4      desirable    n-isl          trunking      1
```

The trunk mode was set to desirable. The result was that trunking mode was negotiated with the neighbor switch, and both decided on ISL (***n-isl***). The current status now is *trunking*. The following shows what happened on Switch B because of the command issued on Switch A.

```
Switch-B (enable)
2000 Jan 17 19:09:52 %DTP-5-TRUNKPORTON:Port 2/1 has become isl trunk
2000 Jan 17 19:09:52 %DTP-5-TRUNKPORTON:Port 2/2 has become isl trunk
2000 Jan 17 19:09:52 %PAGP-5-PORTFROMSTP:Port 2/1 left bridge port 2/1-4
2000 Jan 17 19:09:52 %DTP-5-TRUNKPORTON:Port 2/3 has become isl trunk
2000 Jan 17 19:09:52 %PAGP-5-PORTFROMSTP:Port 2/2 left bridge port 2/1-4
2000 Jan 17 19:09:53 %DTP-5-TRUNKPORTON:Port 2/4 has become isl trunk
2000 Jan 17 19:09:53 %PAGP-5-PORTFROMSTP:Port 2/3 left bridge port 2/1-4
2000 Jan 17 19:09:53 %PAGP-5-PORTFROMSTP:Port 2/4 left bridge port 2/1-4
2000 Jan 17 19:09:55 %PAGP-5-PORTTOSTP:Port 2/1 joined bridge port 2/1-4
2000 Jan 17 19:09:55 %PAGP-5-PORTTOSTP:Port 2/2 joined bridge port 2/1-4
2000 Jan 17 19:09:55 %PAGP-5-PORTTOSTP:Port 2/3 joined bridge port 2/1-4
2000 Jan 17 19:09:55 %PAGP-5-PORTTOSTP:Port 2/4 joined bridge port 2/1-4

Switch-B (enable) show trunk 2
Port      Mode          Encapsulation  Status        Native vlan
--------  ------------  -------------  ------------  -----------
 2/1      auto          n-isl          trunking      1
 2/2      auto          n-isl          trunking      1
 2/3      auto          n-isl          trunking      1
 2/4    auto      n-isl      trunking    1
```

Notice that all four ports (2/1-4) became trunking even though we specifically change only one port (2/1) to desirable. This is an example of how changing one port in the channel affects all the ports.

Troubleshooting EtherChannel

The challenges for EtherChannel can be divided into two main areas: troubleshooting during the configuration phase, and troubleshooting during the execution phase. Configuration errors usually occur because of mismatched parameters on the ports involved (different speeds, different duplex, different spanning-tree port values, and so on). But you can also generate errors during the configuration by setting the channel on one side to on and waiting too long before configuring the channel on the other side. This causes spanning tree loops, which generate an error and shut down the port.

When an error is encountered while configuring EtherChannel, be sure to check the status of the ports after correcting the EtherChannel error situation. If the port status is errdisable, the ports have been shut down by the software, and they will not come on again until you enter the **set port enable** command.

Note If the port status becomes errdisable, you must specifically enable the ports using the **set port enable** command for the ports to become active. Currently, you can correct all the EtherChannel issues, but the ports will not come up or form a channel until they are enabled again. Future versions of the operating system may periodically check whether errdisable ports should be enabled.

For the following tests, we will turn off trunking and EtherChannel. The following topics will be covered:

- Setting mismatched parameters
- Waiting too long before configuring the other side
- Correcting errdisable state
- Showing what happens when a link breaks and is restored

Setting Mismatched Parameters

This section examines an example of mismatched parameters. We will set port 2/4 in VLAN 2 while the other ports are still in VLAN 1. To create a new VLAN, we must assign a VTP domain for the switch and then create the VLAN.

```
Switch-A (enable) show port channel
No ports channelling

Switch-A (enable) show port
Port  Name                 Status     Vlan       Level  Duplex Speed Type
----- ------------------ ---------- ---------- ------ ------ ----- ------------
 2/1                     connected  1          normal a-full a-100 10/100BaseTX
 2/2                     connected  1          normal a-full a-100 10/100BaseTX
 2/3                     connected  1          normal a-full a-100 10/100BaseTX
 2/4                     connected  1          normal a-full a-100 10/100BaseTX

Switch-A (enable) set vlan 2
Cannot add/modify VLANs on a VTP server without a domain name.

Switch-A (enable) set vtp domain testDomain
VTP domain testDomain modified

Switch-A (enable) set vlan 2 name vlan2
Vlan 2 configuration successful

Switch-A (enable) set vlan 2 2/4
VLAN 2 modified.
VLAN 1 modified.
VLAN  Mod/Ports
----  ------------------------
2     2/4

Switch-A (enable)
1999 Dec 19 00:19:34 %PAGP-5-PORTFROMSTP:Port 2/4 left bridg4

Switch-A (enable) show port
Port  Name                 Status     Vlan       Level  Duplex Speed Type
----- ------------------ ---------- ---------- ------ ------ ----- ------------
 2/1                     connected  1          normal a-full a-100 10/100BaseTX
 2/2                     connected  1          normal a-full a-100 10/100BaseTX
 2/3                     connected  1          normal a-full a-100 10/100BaseTX
 2/4                     connected  2          normal a-full a-100 10/100BaseTX

Switch-A (enable) set port channel 2/1-4 desirable
Port(s) 2/1-4 channel mode set to desirable.

Switch-A (enable)
1999 Dec 19 00:20:19 %PAGP-5-PORTFROMSTP:Port 2/1 left bridge port 2/1
1999 Dec 19 00:20:19 %PAGP-5-PORTFROMSTP:Port 2/2 left bridge port 2/2
1999 Dec 19 00:20:19 %PAGP-5-PORTFROMSTP:Port 2/3 left bridge port 2/3
1999 Dec 19 00:20:20 %PAGP-5-PORTFROMSTP:Port 2/4 left bridge port 2/4
1999 Dec 19 00:20:20 %PAGP-5-PORTFROMSTP:Port 2/2 left bridge port 2/2
1999 Dec 19 00:20:22 %PAGP-5-PORTFROMSTP:Port 2/3 left bridge port 2/3
1999 Dec 19 00:20:22 %PAGP-5-PORTFROMSTP:Port 2/4 left bridge port 2/4
1999 Dec 19 00:20:24 %PAGP-5-PORTTOSTP:Port 2/1 joined bridge port 2/1-2
1999 Dec 19 00:20:25 %PAGP-5-PORTTOSTP:Port 2/2 joined bridge port 2/1-2
1999 Dec 19 00:20:25 %PAGP-5-PORTTOSTP:Port 2/3 joined bridge port 2/3
1999 Dec 19 00:20:25 %PAGP-5-PORTTOSTP:Port 2/4 joined bridge port 2/4

Switch-A (enable) show port channel
Port  Status     Channel    Channel     Neighbor                 Neighbor
                 mode       status      device                   port
```

```
----- ---------- --------- ----------- ------------------------ ----------
 2/1  connected  desirable channel     WS-C5505    066509957(Sw  2/1
 2/2  connected  desirable channel     WS-C5505    066509957(Sw  2/2
----- ---------- --------- ----------- ------------------------ ----------
```

Notice that the channel formed only between ports 2/1 and 2/2. Ports 2/3 and 2/4 were left out because port 2/4 was in a different VLAN. There was no error message; PAgP just did what it could to make the channel work. You need to watch the results when you create the channel to make sure that it did what you wanted it to do.

Now let's set the channel manually to on with port 2/4 in a different VLAN and see what happens. First, we will set the channel mode back to auto to tear down the existing channel, and then we will set the channel manually to on.

```
Switch-A (enable) set port channel 2/1-4 auto
Port(s) 2/1-4 channel mode set to auto.
Switch-A (enable)
1999 Dec 19 00:26:08 %PAGP-5-PORTFROMSTP:Port 2/1 left bridge port 2/1-2
1999 Dec 19 00:26:08 %PAGP-5-PORTFROMSTP:Port 2/2 left bridge port 2/1-2
1999 Dec 19 00:26:08 %PAGP-5-PORTFROMSTP:Port 2/3 left bridge port 2/3
1999 Dec 19 00:26:08 %PAGP-5-PORTFROMSTP:Port 2/4 left bridge port 2/4
1999 Dec 19 00:26:18 %PAGP-5-PORTTOSTP:Port 2/1 joined bridge port 2/1
1999 Dec 19 00:26:19 %PAGP-5-PORTTOSTP:Port 2/2 joined bridge port 2/2
1999 Dec 19 00:26:19 %PAGP-5-PORTTOSTP:Port 2/3 joined bridge port 2/3
1999 Dec 19 00:26:19 %PAGP-5-PORTTOSTP:Port 2/4 joined bridge port 2/4

Switch-A (enable) show port channel
No ports channelling

Switch-A (enable) set port channel 2/1-4 on
Mismatch in vlan number.
Failed to set port(s) 2/1-4 channel mode to on.

Switch-A (enable) show port channel
No ports channelling
```

On Switch B, we can turn on the channel—notice that it indicates that the ports are channeling fine, but we know that Switch A is not configured correctly.

```
Switch-B (enable) show port channel
No ports channelling

Switch-B (enable) show port
Port  Name                 Status     Vlan       Level  Duplex Speed Type
----- ------------------ ---------- ---------- ------ ------ ----- ------------
 2/1                      connected  1          normal a-full a-100 10/100BaseTX
 2/2                      connected  1          normal a-full a-100 10/100BaseTX
 2/3                      connected  1          normal a-full a-100 10/100BaseTX
 2/4                      connected  1          normal a-full a-100 10/100BaseTX

Switch-B (enable) set port channel 2/1-4 on
Port(s) 2/1-4 channel mode set to on.

Switch-B (enable)
2000 Jan 17 22:54:59 %PAGP-5-PORTFROMSTP:Port 2/1 left bridge port 2/1
2000 Jan 17 22:54:59 %PAGP-5-PORTFROMSTP:Port 2/2 left bridge port 2/2
2000 Jan 17 22:54:59 %PAGP-5-PORTFROMSTP:Port 2/3 left bridge port 2/3
2000 Jan 17 22:54:59 %PAGP-5-PORTFROMSTP:Port 2/4 left bridge port 2/4
2000 Jan 17 22:55:00 %PAGP-5-PORTTOSTP:Port 2/1 joined bridge port 2/1-4
2000 Jan 17 22:55:00 %PAGP-5-PORTTOSTP:Port 2/2 joined bridge port 2/1-4
2000 Jan 17 22:55:00 %PAGP-5-PORTTOSTP:Port 2/3 joined bridge port 2/1-4
2000 Jan 17 22:55:00 %PAGP-5-PORTTOSTP:Port 2/4 joined bridge port 2/1-4

Switch-B (enable) show port channel
Port  Status     Channel    Channel     Neighbor                  Neighbor
                 mode       status      device                    port
----- ---------- ---------- ----------- ------------------------- ----------
 2/1  connected  on         channel     WS-C5505    066507453(Sw  2/1
 2/2  connected  on         channel     WS-C5505    066507453(Sw  2/2
 2/3  connected  on         channel     WS-C5505    066507453(Sw  2/3
 2/4  connected  on         channel     WS-C5505    066507453(Sw  2/4
----- ---------- ---------- ----------- ------------------------- ----------
```

This makes it clear that you must check both sides of the channel when manually configuring the channel to make sure that both sides are up, not just one side. The previous output shows that Switch B is set for a channel, but Switch A is not channeling because it has one port that is in the wrong vlan.

Waiting Too Long Before Configuring the Other Side

In our situation, Switch B has EtherChannel turned on, but Switch A does not have EtherChannel turned on because it has a VLAN configuration error (ports 2/1-3 are in VLAN 1, and port 2/4 is in VLAN 2). Here is what happens when one side of a EtherChannel is set to on while the other side is still in auto mode: After a few minutes, Switch B shut down its ports because of a spanning-tree loop detection. This is because Switch B ports 2/1-4 all act like one big port, while Switch A ports 2/1-4 are all totally independent ports. A broadcast sent from Switch B to Switch A on port 2/1 will be sent back to Switch

B on ports 2/2, 2/3, and 2/4 because Switch A treats these ports as independent ports. This is why Switch B thinks there is a spanning-tree loop. Notice that the ports on Switch B are now disabled and have a status of errdisable.

```
Switch-B (enable)
2000 Jan 17 22:55:48 %SPANTREE-2-CHNMISCFG: STP loop - channel 2/1-4 is disabled in vlan
1.
2000 Jan 17 22:55:49 %PAGP-5-PORTFROMSTP:Port 2/1 left bridge port 2/1-4
2000 Jan 17 22:56:01 %PAGP-5-PORTFROMSTP:Port 2/2 left bridge port 2/1-4
2000 Jan 17 22:56:13 %PAGP-5-PORTFROMSTP:Port 2/3 left bridge port 2/1-4
2000 Jan 17 22:56:36 %PAGP-5-PORTFROMSTP:Port 2/4 left bridge port 2/1-4

Switch-B (enable) show port channel
Port  Status     Channel    Channel     Neighbor                  Neighbor
                 mode       status      device                    port
----- ---------- ---------- ----------- ------------------------- ----------
 2/1  errdisable on         channel
 2/2  errdisable on         channel
 2/3  errdisable on         channel
 2/4  errdisable on         channel
----- ---------- ---------- ----------- ------------------------- ----------
Switch-B (enable) show port
Port  Name               Status     Vlan       Level  Duplex Speed Type
----- ------------------ ---------- ---------- ------ ------ ----- ------------
 2/1                     errdisable 1          normal   auto  auto 10/100BaseTX
 2/2                     errdisable 1          normal   auto  auto 10/100BaseTX
 2/3                     errdisable 1          normal   auto  auto 10/100BaseTX
 2/4                     errdisable 1          normal   auto  auto 10/100BaseTX
```

Correcting errdisable State

Sometimes when you try to configure EtherChannel but the ports are not configured the same, it will cause the ports on one side of the channel to be shut down. The link lights will be yellow on the port. You can tell this by the console by typing **show port**. The ports will be listed as errdisable. To recover from this, you should fix the mismatched parameters on the ports involved and then re-enable the ports. Just note that this re-enabling of the ports is a separate step that must be done for the ports to become functional again.

In our example, we know that Switch A had a VLAN mismatch, so we will go to Switch A and put port 2/4 back into VLAN 1. Then we will turn on the channel for ports 2/1-4. Switch A will not show that it's connected until we re-enable Switch B ports. Then when we have fixed Switch A and put it in channeling mode, we will go back to Switch B and re-enable the ports.

```
Switch-A (enable) set vlan 1 2/4
VLAN 1 modified.
VLAN 2 modified.
VLAN  Mod/Ports
---- ----------------------
1     2/1-24

Switch-A (enable) set port channel 2/1-4 on
Port(s) 2/1-4 channel mode set to on.
Switch-A (enable) sh port channel
Port  Status     Channel   Channel     Neighbor                  Neighbor
                 mode      status      device                    port
----- ---------- --------- ----------- ------------------------- ----------
 2/1  notconnect on        channel
 2/2  notconnect on        channel
 2/3  notconnect on        channel
 2/4  notconnect on        channel
----- ---------- --------- ----------- ------------------------- ----------

Switch-B (enable) show port channel
Port  Status     Channel   Channel     Neighbor                  Neighbor
                 mode      status      device                    port
----- ---------- --------- ----------- ------------------------- ----------
 2/1  errdisable on        channel
 2/2  errdisable on        channel
 2/3  errdisable on        channel
 2/4  errdisable on        channel
----- ---------- --------- ----------- ------------------------- ----------

Switch-B (enable) set port enable 2/1-4
Ports 2/1-4 enabled.
Switch-B (enable) 2000 Jan 17 23:15:22 %PAGP-5-PORTTOSTP:Port 2/1 joined bridg4
2000 Jan 17 23:15:22 %PAGP-5-PORTTOSTP:Port 2/2 joined bridge port 2/1-4
2000 Jan 17 23:15:22 %PAGP-5-PORTTOSTP:Port 2/3 joined bridge port 2/1-4
2000 Jan 17 23:15:22 %PAGP-5-PORTTOSTP:Port 2/4 joined bridge port 2/1-4

Switch-B (enable) show port channel
Port  Status     Channel   Channel     Neighbor                  Neighbor
                 mode      status      device                    port
----- ---------- --------- ----------- ------------------------- ----------
 2/1  connected  on        channel
 2/2  connected  on        channel
 2/3  connected  on        channel
 2/4  connected  on        channel
----- ---------- --------- ----------- ------------------------- ----------
```

Showing What Happens When a Link Breaks and Is Restored

When a port in the channel goes down, then any packets that would normally be sent on that port are shifted over to the next port in the channel. You can verify that this is happening by using the **show mac** command. In our test bed, we will have Switch A send **ping** packets to Switch B to see which link the traffic is using. First we will clear the counters and then use **show mac**, send three **ping**s, and use **show mac** again to look at which channel the **ping** responses were received on.

```
Switch-A (enable) clear counters
This command will reset all MAC and port counters reported in CLI and SNMP.
Do you want to continue (y/n) [n]? y
MAC and Port counters cleared.

Switch-A (enable) show port channel
Port  Status     Channel    Channel     Neighbor                  Neighbor
                 mode       status      device                    port
----- ---------- ---------- ----------- ------------------------- ----------
 2/1  connected  on         channel     WS-C5505    066509957(Sw  2/1
 2/2  connected  on         channel     WS-C5505    066509957(Sw  2/2
 2/3  connected  on         channel     WS-C5505    066509957(Sw  2/3
 2/4  connected  on         channel     WS-C5505    066509957(Sw  2/4
----- ---------- ---------- ----------- ------------------------- ----------

Switch-A (enable) show mac

Port     Rcv-Unicast          Rcv-Multicast        Rcv-Broadcast
-------- -------------------- -------------------- --------------------
 2/1                        0                   18                    0
 2/2                        0                    2                    0
 2/3                        0                    2                    0
 2/4                        0                    2                    0
Switch-A (enable) ping 172.16.84.17
172.16.84.17 is alive
Switch-A (enable) ping 172.16.84.17
172.16.84.17 is alive
Switch-A (enable) ping 172.16.84.17
172.16.84.17 is alive
Switch-A (enable) show mac

Port     Rcv-Unicast          Rcv-Multicast        Rcv-Broadcast
-------- -------------------- -------------------- --------------------
 2/1                        3                   24                    0
 2/2                        0                    2                    0
 2/3                        0                    2                    0
 2/4                        0                    2                    0
```

Now at this point, we have received the **ping** responses on port 3/1, so when Switch B console sends a response to Switch A, the EtherChannel uses port 2/1. Now we will shut down port 2/1 on Switch B. From Switch A, we will issue another **ping** and see what channel the response comes back on. (Switch A is sending on the same port to which Switch B is connected. We just show the received packets from Switch B because the transmit packets are farther down in the **show mac** display.)

```
1999 Dec 19 01:30:23 %PAGP-5-PORTFROMSTP:Port 2/1 left bridge port 2/1-4

Switch-A (enable) ping 172.16.84.17
172.16.84.17 is alive
Switch-A (enable) show mac

Port     Rcv-Unicast          Rcv-Multicast        Rcv-Broadcast
-------- -------------------- -------------------- --------------------
 2/1                        3                   37                    0
 2/2                        1                   27                    0
 2/3                        0                    7                    0
 2/4                        0                    7                    0
```

Notice that now that port 2/1 is disabled, EtherChannel automatically uses the next port in the channel, 2/2. Now we re-enable port 2/1 and wait for it to join the bridge group; then we issue two more **ping**s.

```
1999 Dec 19 01:31:33 %PAGP-5-PORTTOSTP:Port 2/1 joined bridge port 2/1-4

Switch-A (enable) ping 172.16.84.17
172.16.84.17 is alive
Switch-A (enable) ping 172.16.84.17
172.16.84.17 is alive
Switch-A (enable) show mac

Port     Rcv-Unicast          Rcv-Multicast        Rcv-Broadcast
-------- -------------------- -------------------- --------------------
 2/1                        5                   50                    0
 2/2                        1                   49                    0
 2/3                        0                   12                    0
 2/4                        0                   12                    0
```

Note that these **ping**s are sent from port 2/1, so when the link comes back up, EtherChannel will again add it to the bundle and use it. All this is done transparently to the user.

Commands Used in This Section

Commands to use for setting the configuration include these:

- **set port channel on**—To turn on the EtherChannel feature
- **set port channel auto**—To reset the ports to their default mode of auto
- **set port channel desirable**—To send PAgP packets to the other side requesting that a channel be created
- **set port enable**—To enable the ports after **set port disable** or after an errdisable state
- **set port disable**—To disable a port while other configuration settings are being made
- **set trunk desirable**—To turn on trunking by causing this port to send a request to the other switch that this be a trunk link and, if the port is set to negotiate (the default setting), to negotiate the type of trunking to use on the link (ISL or 802.1q)

Commands to use for verifying the configuration include these:

- **show version**—To display what version of software the switch is running

- **show module**—To display which modules are installed in the switch
- **show port capabilities**—To determine whether the ports that we want to use have the capability to do EtherChannel
- **show port**—To determine the status of the port (notconnect, connected) and the speed and duplex settings
- **ping**—To test connectivity to the other switch
- **show port channel**—To see the current status of the EtherChannel bundle
- **show port channel** *mod/port*—To give a more detailed view of the channel status of a single port
- **show spantree**—To verify that the Spanning-Tree Protocol looked at the channel as one link
- **show trunk**—To see the trunking status of ports

Commands to use for troubleshooting the configuration include these:

- **show port channel**—To see the current status of the EtherChannel bundle
- **show port**—To determine the status of the port (notconnect, connected) and the speed and duplex settings
- **clear counters**—To reset the switch packet counters to zero. The counters are visible with the **show mac** command
- **show mac**—To view packets received and sent by the switch
- **ping**—To test connectivity to the other switch and generate traffic that shows up with the **show mac** command

Using PortFast and Other Commands to Fix End-Station Startup Connectivity Problems

If you have workstations connected to switches that either are incapable of logging into your network domain (NT or Novell) or are incapable of getting a DHCP address, then you may want to try the suggestions listed in this document before exploring other avenues. The suggestions are relatively easy to implement and are very often the cause of workstation connectivity problems encountered during the workstation's initialization/startup phase.

With more customers deploying switching to the desktop and replacing their shared hubs with switches, we often see problems introduced in client/server environments because of this initial delay. The biggest problem that we see is that Windows 95/98/NT, Novell, VINES, IBM NetworkStation/IBM Thin Clients, and AppleTalk clients cannot connect to their servers. If the software on these devices is not persistent during the startup procedure, *they will give up trying to connect to their server before the switch has even allowed traffic to pass through.*

Note This initial connectivity delay often manifests itself as errors that appear when you first boot up a workstation. The following are several examples of error messages and errors that you might see:

A Microsoft networking client displays "No Domain Controllers Available."

DHCP reports, "No DHCP Servers Available."

A Novell IPX networking workstation does not have the "Novell Login Screen" upon

bootup.

An AppleTalk networking client displays "Access to your AppleTalk network has been interrupted. To re-establish your connection, open and close the AppleTalk control panel." It is also possible that the AppleTalk client's Chooser application either will not display a zone list or will display an incomplete zone list.

The initial connectivity delay is also frequently seen in a switched environment in which a network administrator updates software or drivers. In this case, a vendor may optimize the drivers so that network initialization procedures happen earlier in the client's startup process (before the switch is ready to process the packets).

With the various features that are now included in some switches, it can take close to a minute for a switch to begin servicing a newly connected workstation. This delay would affect the workstation every time that it is turned on or rebooted. The four main features that cause this delay are listed here:

- Spanning-Tree Protocol
- EtherChannel negotiation
- Trunking negotiation
- Link speed/duplex negotiation between the switch and the workstation

These four features are listed in order of causing the most delay (Spanning-Tree Protocol) to causing the least delay (speed/duplex negotiation). A workstation connected to a switch usually does not cause spanning-tree loops, usually does not need EtherChannel, and usually does not need to negotiate a trunking method. (Disabling link speed/detection negotiation can also reduce port delay, if you need to optimize your startup time as much as possible.)

This section shows how to implement startup speed-optimization commands on three Catalyst switch platforms. In the timing sections, we show how the switch port delay is reduced and by how much.

Contents

The following topics will be covered in this section:

- Background
- How to reduce startup delay on the Catalyst 4000/5000/6000 switch
- Timing tests on the Catalyst 5000
- How to reduce startup delay on the Catalyst 2900XL/3500XL switch
- Timing tests on the Catalyst 2900XL
- How to reduce startup delay on the Catalyst 1900/2800 switch
- Timing tests on the Catalyst 1900
- An additional benefit to PortFast

The terms *workstation, end station*, and *server* are all used interchangeably in this section to refer to any device directly connected to a switch by a single NIC card. These terms may also refer to devices with multiple NIC cards in which the NIC card is used only for redundancy—in other words, the workstation or server is not configured to act as a bridge; it just has multiple NIC cards for redundancy.

Note Some server NIC cards support trunking or EtherChannel. In some situations, the server needs to live on several VLANs at the same time (trunking), or the server needs more bandwidth on the link connecting it to the switch (EtherChannel). In these cases, you would *not* turn off PagP, and you would *not* turn off trunking. Also, these devices are rarely turned off or reset. The instructions in this chapter do not apply to these type of devices.

Background

This section covers four features that some switches have that cause initial delays when a device is connected to a switch. Usually a workstation will either not cause the spanning-tree problem (loops), or not need the feature (PAgP, DTP), so the delay is unnecessary.

Spanning Tree

If you have recently started moving from a hub environment to a switch environment, these connectivity problems may show up because a switch works much differently than a hub. A switch provides connectivity at the data link layer, not at the physical layer. The switch must use a bridging algorithm to decide whether packets received on a port need to be transmitted out other ports. The bridging algorithm is susceptible to physical loops in the network topology. Because of this susceptibility to loops, switches run a protocol called the *Spanning-Tree Protocol* that causes loops to be eliminated in the topology. Running this protocol causes all ports that are included in the spanning-tree process to become active much slower than they otherwise would because the protocol detects and blocks loops. A bridged network having physical loops, without spanning tree, will break. So, in spite of the time involved, the Spanning-Tree Protocol is a good thing. The spanning-tree process running on Catalyst switches is an industry-standard specification (IEEE 802.1d).

After a port on the switch has a link and joins the bridge group, it will run Spanning-Tree Protocol on that port. A port running a spanning tree can have one of five states: blocking, listening, learning, forwarding, and disabled. The Spanning-Tree Protocol dictates that the port starts out blocking and then immediately moves through the listening and learning phases. By default, it will spend approximately 15 seconds listening and 15 seconds learning.

During the listening state, the switch is trying to determine where it fits in the spanning-tree topology. It especially wants to know whether this port is part of a physical loop. If it is part of a loop, then this port may be chosen to go into blocking mode. Blocking mode means that the port won't send or receive user data for the sake of eliminating loops. If the port is not part of a loop, it will proceed to the learning state, which involves learning which MAC addresses live off this port. This whole spanning-tree initialization process takes about 30 seconds.

If you are connecting a workstation or a server with a single NIC card to a switch port, this connection *cannot* create a physical loop. These connections are considered leaf nodes. There is no reason to make the workstation wait 30 seconds while the switch checks for loops when the workstation cannot cause a loop. So, Cisco added a feature called PortFast, or Fast-Start, which means that the spanning tree for this port will assume that the port is not part of a loop and will immediately move to the forwarding state, without going through the blocking, listening, or learning states. This can save a lot of time. This command *does not* turn off the spanning tree. It just makes the spanning tree on the selected port skip a few (unnecessary in this circumstance) steps in the beginning.

Note The PortFast feature should *never* be used on switch ports that connect to other *switches, hubs,* or *routers*. These connections may cause physical loops, and it is very important that the spanning-tree process go through the full initialization procedure in these situations. A spanning-tree loop can bring your network down. If PortFast is turned on for a port that *is* part of a physical loop, it can cause a window of time in which packets could possibly be continuously forwarded (and even multiply) in such a way that the network can't recover. In later Catalyst operating system software (5.4(1)), a feature called PortFast BPDU-Guard detects the reception of BPDUs on ports having PortFast enabled. Because this should never happen, BPDU-Guard puts the port into errDisable state.

EtherChannel

Another feature that a switch may have is called EtherChannel (or Fast EtherChannel, or Gigabit EtherChannel). This feature allows multiple links between the same two devices to work as if they were one fast link, with traffic load balanced among the links. A switch can form these bundles automatically with a neighbor using a protocol called Port Aggregation Protocol (PAgP). Switch ports that can run PAgP usually default to a passive mode called auto, which means that they are *willing* to form a bundle if the neighbor device across the link asks them to. Running the protocol in auto mode can cause a port to delay for up to 15 seconds before passing control to the spanning-tree algorithm (PAgP runs on a port before spanning tree does). There is no reason to have PAgP running on a port connected to a workstation. Setting the switch port PAgP mode to off will eliminate this delay.

Trunking

Another switch feature is the capability of a port to form a trunk. A trunk is configured between two devices when they need to carry traffic from multiple VLANs. A VLAN is something that switches create to make a group of workstations appear to be on their own segment or broadcast domain. Trunk ports make these VLANs extend across multiple switches so that a single VLAN can cover an entire campus. They do this by adding tags to the packets, indicating which VLAN the packet belongs to.

Different types of trunking protocols exist. If a port can become a trunk, then it may also have the capability to trunk automatically and, in some cases, even negotiate what type of trunking to use on the port. This capability to negotiate the trunking method with the other device is called Dynamic Trunking Protocol (DTP); the precursor to DTP is a protocol called Dynamic ISL (DISL). If these protocols are running, they can delay a port on the switch becoming active.

Usually a port connected to a workstation belongs to only one VLAN and therefore does not need to trunk. If a port has the capability to negotiate the formation of a trunk, it will usually default to the auto mode. If the port is changed to a trunking mode of off, it will further reduce the delay of a switch port becoming active.

Speed and Duplex Negotiation

Just turning on PortFast and turning off PAgP (if present) is usually enough to solve the problem, but if you need to eliminate every possible second, you could also set the port speed and duplex manually on the switch if it is a multispeed port (10/100). Autonegotiation is a nice feature, but turning it off could save you 2 seconds on a Catalyst 5000 (it does not help much on the 2800 or 2900XL).

There can be complications, though, if you turn off autonegotiation on the switch but leave it active on the workstation. Because the switch will not negotiate with the client, the client might not choose the same duplex setting that the switch is using. See the section "Troubleshooting Ethernet 10-/100-Mb Half-/Full-Duplex Autonegotiation," earlier in this chapter, for additional information on the caveats of autonegotiation.

How to Reduce Startup Delay on the Catalyst 4000/5000/6000 Switch

The following five commands show how to turn on PortFast, turn off PAgP negotiation, turn off trunking negotiation (DISL, DTP), and turn off speed/duplex negotiation. The **set spantree portfast** command can be done on a range of ports at once (**set spantree portfast 2/1-12 enable**). Usually **set port channel** must be turned off using a valid group of channel-capable ports. In the case that follows, module 2 has the capability to channel with ports 2/1-2 or with ports 2/1-4, so either of these groups of ports would have been valid to use.

Note Version 5.2 of Cat OS for Catalyst 4000/5000 has a new command called **set port host,** which is a macro that combines the commands **set spantree portfast, set portchannel off**, **set trunk off** into one command.

Configuration

Use the following commands to reduce startup delay on the Catalyst 4000/5000/6000 switches.

```
Switch-A (enable) set spantree portfast 2/1 enable

Warning: Spantree port fast start should only be enabled on ports connected
to a single host.  Connecting hubs, concentrators, switches, bridges, etc. to
a fast start port can cause temporary spanning tree loops.  Use with caution.

Spantree port 2/1 fast start enabled.
Switch-A (enable) set port channel 2/1-2 off
Port(s) 2/1-2 channel mode set to off.

Switch-A (enable) set trunk 2/1 off
Port(s) 2/1 trunk mode set to off.
```

The changes to the configuration will be automatically saved to NVRAM.

Verification

The version of the switch software used in this document is 4.5(1). For the full output of **show version** and **show module,** refer to the section "Timing Tests with and Without DTP, PAgP, and PortFast on a Catalyst 5000," later in this chapter.

Switch-A (enable) show version

WS-C5505 Software, Version McpSW: 4.5(1) NmpSW: 4.5(1)

The following command shows how to view the current state of a port with regard to spanning tree. Currently the port is in the spanning tree forwarding state (sending and receiving packets), and the Fast-Start column shows that PortFast is currently disabled. In other words, the port will take at least 30 seconds to move to the forwarding state whenever it initializes.

```
Switch-A (enable) show port spantree 2/1

Port      Vlan  Port-State     Cost   Priority  Fast-Start  Group-Method
--------  ----  -------------  -----  --------  ----------  ------------
 2/1      1     forwarding        19        32  disabled
```

Now we will enable PortFast on this switch port. The switch warns us that this command should be used only on ports that are connected to a single host (a workstation, server, and so on) and are never to be used on ports connected to other hubs or switches. The reason that we enable PortFast is so the port will start forwarding immediately. We can do this because a workstation or server will not cause a network loop, so why waste time checking? But another hub or switch could cause a loop, and we want to always go through the normal listening and learning stages when connecting to these types of devices.

```
Switch-A (enable) set spantree portfast 2/1 enable

Warning: Spantree port fast start should only be enabled on ports connected to a single
host.  Connecting hubs, concentrators, switches, bridges, etc. to
a fast start port can cause temporary spanning tree loops.  Use with caution.

Spantree port 2/1 fast start enabled.
```

To verify that PortFast is enabled for this port, use the following command:

```
Switch-A (enable) show port spantree 2/1

Port      Vlan  Port-State     Cost   Priority  Fast-Start  Group-Method
--------  ----  -------------  -----  --------  ----------  ------------
 2/1      1     forwarding        19        32  enabled
```

Another way to view the PortFast settings for one or more ports is to view the spanning-tree information for a specific VLAN. Later, in the section "Timing Tests with and Without DTP, PAgP, and PortFast on a Catalyst 5000," we show how to have the switch report each stage of spanning tree that it moves through in real time. The output that follows also shows the forward delay time (15 seconds), which is how long the spanning tree will be in the listening state and how long it will be in the learning state for each port in the VLAN.

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

Designated Root             00-e0-4f-94-b5-00
Designated Root Priority    8189
Designated Root Cost        19
Designated Root Port        2/24
Root Max Age   20 sec    Hello Time 2  sec   Forward Delay 15 sec

Bridge ID MAC ADDR          00-90-92-b0-84-00
Bridge ID Priority          32768
Bridge Max Age 20 sec    Hello Time 2  sec   Forward Delay 15 sec

Port      Vlan  Port-State     Cost   Priority  Fast-Start  Group-Method
--------- ----  -------------  -----  --------  ----------  ------------

 2/1      1     forwarding        19        32   enabled
...
```

To verify that PAgP is off, use the **show port channel** command, which follows. Be sure to specify the module number (2, in this case) so that the command will show you the channel mode even if there is no channel formed. If we do **show port channel** with no channels formed, it just says "no ports channeling." We want to go further and see the current channel mode.

```
Switch-A (enable) show port channel
No ports channeling

Switch-A (enable) show port channel 2
Port  Status     Channel   Channel     Neighbor                  Neighbor
                 mode      status      device                    port
----- ---------- --------- ----------- ------------------------- ----------
 2/1  notconnect auto      not channel
 2/2  notconnect auto      not channel
...
Switch-A (enable) set port channel 2/1-2 off
Port(s) 2/1-2 channel mode set to off.

Switch-A (enable) show port channel 2
Port  Status     Channel   Channel     Neighbor                  Neighbor
                 mode      status      device                    port
----- ---------- --------- ----------- ------------------------- ----------
 2/1  connected  off       not channel
 2/2  connected  off       not channel
...
```

To verify that trunking negotiation is off, use the **set trunk off** command. We show the default state here. Then we turn trunking to off and show the resulting state. We specify module number 2 so that we can see the current channel mode for the ports in this module.

```
Switch-A (enable) show trunk 2
Port      Mode         Encapsulation  Status        Native vlan
--------  -----------  -------------  ------------  -----------
 2/1      auto         negotiate      not-trunking  1
 2/2      auto         negotiate      not-trunking  1
...

Switch-A (enable) set trunk 2/1-2 off
Port(s) 2/1-2 trunk mode set to off.

Switch-A (enable) show trunk 2
Port      Mode         Encapsulation  Status        Native vlan
--------  -----------  -------------  ------------  -----------
 2/1      off          negotiate      not-trunking  1
 2/2      off          negotiate      not-trunking  1
...
```

We do not show an example here of turning off speed/duplex autonegotiation by manually setting the speed and duplex on the switch; it should not be necessary except in the rarest of cases. We give an example of how to do this in Step 10 of the next section, if you feel that it will be necessary for your situation.

Timing Tests with and Without DTP, PAgP, and PortFast on a Catalyst 5000

The following test shows what happens with switch port initialization timing as the various commands are applied. The default settings of the port are used first to give a benchmark. They have PortFast disabled, PAgP (EtherChannel) mode set to auto (it will channel if asked to channel), and the trunking mode (DTP) set to auto (it will trunk if asked to trunk). The test then proceeds to turn on PortFast and

measure the time, then to turn off PAgP and measure the time, and then to turn off trunking and measure the time. Finally, we turn off autonegotiation and measure the time. All these tests will be done on a Catalyst 5000 with a 10/100 Fast Ethernet card that supports DTP and PAgP.

Note Turning on PortFast is not the same thing as turning off the Spanning-Tree Protocol (as noted earlier in the document). With PortFast on, the Spanning-Tree Protocol is still running on the port; it just skips blocking, listening, and learning, and goes immediately to the forwarding state. Turning off the Spanning-Tree Protocol is not recommended because it affects the entire VLAN and can leave the network vulnerable to physical topology loops, which can cause serious network problems.

Step 1 Show the switch IOS version and configuration (**show version**, **show module**).

```
Switch-A (enable) show version
WS-C5505 Software, Version McpSW: 4.5(1) NmpSW: 4.5(1)
Copyright (c) 1995-1999 by Cisco Systems
NMP S/W compiled on Mar 29 1999, 16:09:01
MCP S/W compiled on Mar 29 1999, 16:06:50

System Bootstrap Version: 3.1.2

Hardware Version: 1.0  Model: WS-C5505  Serial #: 066507453

Mod Port Model      Serial #  Versions
--- ---- ---------- --------- ---------------------------------------
1   0    WS-X5530   006841805 Hw : 1.3
                               Fw : 3.1.2

                               Fw1: 3.1(2)
                               Sw : 4.5(1)
2   24   WS-X5225R  012785227 Hw : 3.2
                               Fw : 4.3(1)
                               Sw : 4.5(1)

       DRAM                    FLASH                   NVRAM
Module Total   Used    Free    Total   Used    Free    Total Used  Free
------ ------- ------- ------- ------- ------- ------- ----- ----- -----
1       32640K  13648K  18992K   8192K   4118K   4074K  512K  119K  393K

Uptime is 28 days, 18 hours, 54 minutes

Switch-A (enable) show module
Mod Module-Name         Ports Module-Type           Model    Serial-Num Status
--- ------------------- ----- --------------------- --------- --------- -------
1                       0     Supervisor III        WS-X5530  006841805 ok
2                       24    10/100BaseTX Ethernet WS-X5225R 012785227 ok

Mod MAC-Address(es)                         Hw     Fw         Sw
--- --------------------------------------- ------ ---------- -----------------
1   00-90-92-b0-84-00 to 00-90-92-b0-87-ff 1.3    3.1.2      4.5(1)
2   00-50-0f-b2-e2-60 to 00-50-0f-b2-e2-77 3.2    4.3(1)     4.5(1)

Mod Sub-Type Sub-Model Sub-Serial Sub-Hw
--- -------- --------- ---------- ------
1   NFFC     WS-F5521  0008728786 1.0
```

Step 2 Set logging for spanning-tree to the most verbose (**set logging level spantree 7**). The following is the default logging level (2) for spanning tree, which means that only critical situations will be reported.

```
Switch-A (enable) show logging

Logging buffer size:          500
        timestamp option:     enabled
Logging history size:         1
Logging console:              enabled
Logging server:               disabled
        server facility:      LOCAL7
        server severity:      warnings(4)

Facility            Default Severity         Current Session Severity
-------------       ----------------------   ------------------------
...
spantree            2                        2
...
0(emergencies)        1(alerts)             2(critical)
3(errors)             4(warnings)           5(notifications)
6(information)        7(debugging)
```

We are going to change the level for the spanning-tree to 7 (**debug**) so we can see the spanning-tree states change on the port. This configuration change lasts only for the terminal session, and then it goes back to normal.

```
Switch-A (enable) set logging level spantree 7
System logging facility <spantree for this session set to severity 7(debugging)

Switch-A (enable) show logging
...

Facility            Default Severity         Current Session Severity
-------------       ----------------------   ------------------------
...
spantree            2                        7
...
```

Step 3 Start with the port on the catalyst shut down.

```
Switch-A (enable) set port disable 2/1
Port 2/1 disabled.
```

Step 4 Now we will check the time and enable the port. We want to see how long it stays in each state.

```
Switch-A (enable) show time
    Fri Feb 25 2000, 12:20:17
    Switch-A (enable) set port enable 2/1
    Port 2/1 enabled.
    Switch-A (enable)
    2000 Feb 25 12:20:39 %PAGP-5-PORTTOSTP:Port 2/1 joined bridge port 2/1
    2000 Feb 25 12:20:39 %SPANTREE-6-PORTBLK: port 2/1 state in vlan 1 changed to blocking.
    2000 Feb 25 12:20:39 %SPANTREE-6-PORTLISTEN: port 2/1 state in vlane 1 changed to
    Listening.
    2000 Feb 25 12:20:53 %SPANTREE-6-PORTLEARN: port 2/1 state in vlan 1 changed to Learning.
    2000 Feb 25 12:21:08 %SPANTREE-6-PORTFWD: port 2/1 state in vlan 1 changed to forwarding.
```

Notice from the previous output that it took about 22 seconds (20:17 to 20:39) for the port to begin the spanning-tree blocking stage. This time was spent negotiating the link and doing DTP and PAgP stuff. When we started blocking, we are in the spanning-tree realm. From blocking, it went immediately to listening (20:39 to 20:39). From listening to learning took approximately 14 seconds (20:39 to 20:53).

From learning to forwarding took 15 seconds (20:53 to 21:08). So, the total time before the port actually became functional for traffic was about 51 seconds (20:17 to 21:08).

Note Technically, the listening and learning stages should both be 15 seconds, which is how the forward delay parameter is set for this VLAN. The learning stage probably is closer to 15 seconds than 14 seconds if we had more accurate measurements. None of the measurements here are perfectly accurate—we are just trying to give a feel for how long things take.

Step 5 We know from the previous output and from the **show spantree** command that the spanning tree is active on this port. Let us look at other things that could slow the port reaching the forwarding state. The **show port capabilities** command shows that this port has the capability to trunk and to create an EtherChannel. The **show trunk** command indicates that this port is in auto mode and that it is set to negotiate the type of trunking to use (ISL or 802.1q, negotiated through Dynamic Trunking Protocol [DTP]).

```
Switch-A (enable) show port capabilities 2/1
Model                     WS-X5225R
Port                      2/1
Type                      10/100BaseTX

Speed                     auto,10,100
Duplex                    half,full
Trunk encap type          802.1Q,ISL
Trunk mode                on,off,desirable,auto,nonegotiate
Channel                   2/1-2,2/1-4
Broadcast suppression     percentage(0-100)
Flow control              receive-(off,on),send-(off,on)
Security                  yes
Membership                static,dynamic
Fast start                yes
Rewrite                   yes
Switch-A (enable) show trunk 2/1
Port      Mode          Encapsulation  Status        Native vlan
--------  ------------  -------------  ------------  -----------
 2/1      auto     negotiate    not-trunking  1
...
```

Step 6 First, we will enable PortFast on the port. Trunking negotiation (DTP) is still in auto mode, and EtherChannel (PAgP) is still in auto mode.

```
Switch-A (enable) set port disable 2/1
Port 2/1 disabled.

Switch-A (enable) set spantree portfast 2/1 enable

Warning: Spantree port fast start should only be enabled on ports connected
to a single host.  Connecting hubs, concentrators, switches, bridges, etc. to
a fast start port can cause temporary spanning-tree loops.  Use with caution.

Spantree port 2/1 fast start enabled.

Switch-A (enable) show time
Fri Feb 25 2000, 13:45:23
Switch-A (enable) set port enable 2/1
Port 2/1 enabled.
Switch-A (enable)
Switch-A (enable)
2000 Feb 25 13:45:43 %PAGP-5-PORTTOSTP:Port 2/1 joined bridgeport 2/1
2000 Feb 25 13:45:44 %SPANTREE-6-PORTFWD: port 2/1 state in vlan 1 change to forwarding.
```

Now we have a total time of 21 seconds! It takes 20 seconds before it joins the bridge group (45:23 to 45:43). But then, because PortFast is enabled, it only takes 1 second until STP starts forwarding (instead of 30 seconds). We saved 29 seconds by enabling PortFast. Let's see if we can reduce the delay further.

Step 7 Now we will turn off PAgP mode. We can see from the **show port channel** command that the PAgP mode is set to auto, which means that it will channel if asked to by a neighbor speaking PAgP. You must turn off channeling for at least a group of two ports; you cannot do it for just an individual port.

```
Switch-A (enable) show port channel 2/1
Port  Status     Channel    Channel     Neighbor                  Neighbor
                 mode       status      device                    port
----- ---------- ---------- ----------- ------------------------- ----------
 2/1  connected  auto       not channel

Switch-A (enable) set port channel 2/1-2 off
Port(s) 2/1-2 channel mode set to off.
```

Step 8 Now let's shut down the port and repeat the test.

```
Switch-A (enable) set port disable 2/1
Port 2/1 disabled.

Switch-A (enable) show time
Fri Feb 25 2000, 13:56:23
Switch-A (enable) set port enable 2/1
Port 2/1 enabled.
Switch-A (enable)
2000 Feb 25 13:56:32 %PAGP-5-PORTTOSTP:Port 2/1 joined bridgeport 2/1
2000 Feb 25 13:56:32 %SPANTREE-6-PORTFWD: port 2/1 state in vlan 1 changed to forwarding.
```

Notice here that now it takes only 9 seconds to reach the forwarding state (56:23 to 56:32) instead of 21 seconds, as in the previous test. Turning PAgP from auto to off in this test saved about 12 seconds.

Step 9 Let's turn trunking to off (instead of auto) and see how that affects the time that it takes for the port to reach forwarding state. We will again turn the port off and then on, and record the time.

```
Switch-A (enable) set trunk 2/1 off
Port(s) 2/1 trunk mode set to off.
Switch-A (enable) set port disable 2/1
Port 2/1 disabled.
```

Start the test with trunking set to off (instead of auto).

```
Switch-A (enable) show time
Fri Feb 25 2000, 14:00:19
Switch-A (enable) set port enable 2/1
Port 2/1 enabled.
Switch-A (enable)
2000 Feb 25 14:00:22 %PAGP-5-PORTTOSTP:Port 2/1 joined bridge port 2/1
2000 Feb 25 14:00:23 %SPANTREE-6-PORTFWD: port 2/1 state in vlan 1 change for forwarding.
```

We saved a few seconds at the beginning because it took only 4 seconds to reach the spanning-tree forwarding state (00:19 to 00:22). We saved about 5 seconds by changing the trunking mode from auto to off.

Step 10 (Optional). If the switch port initialization time was the problem, it should be solved by now. But if you have to shave a few more seconds off the time, you could set the port the speed and duplex manually instead of using autonegotiation.

Setting the speed and duplex manually on our side requires that you set the speed and duplex on the other side as well. This is because setting the port speed and duplex disables autonegotiation on the port, and the connecting device will not see autonegotiation parameters. The connecting device will connect only

at half-duplex mode, and the resulting duplex mismatch results in poor performance and port errors. Remember, if you set speed and duplex on one side, you must set speed and duplex on the connecting device as well to avoid these problems.

To view the port status after setting the speed and duplex, use the **show port** command.

```
Switch-A (enable) set port speed 2/1 100
Port(s) 2/1 speed set to 100 Mbps.
Switch-A (enable) set port duplex 2/1 full
Port(s) 2/1 set to full-duplex.
Switch-A (enable) show port
Port  Name               Status     Vlan       Level  Duplex Speed Type
----- ------------------ ---------- ---------- ------ ------ ----- ------------
 2/1                     connected  1          normal   full   100  10/100BaseTX
...
```

The timing results are as follows:

```
Switch-A (enable) show time
Fri Feb 25 2000, 140528 Eastern
Switch-A (enable) set port enable 2/1
Port 2/1 enabled.
Switch-A (enable)
2000 Feb 25 140529 Eastern -0500 %PAGP-5-PORTTOSTP:Port 2/1 joined bridgeport 2/1
2000 Feb 25 140530 Eastern -0500 %SPANTREE-6-PORTFWD: port 2/1 state in vlan 1 changed
to forwarding.
```

The final result gives a time of 2 seconds (0528 to 0530).

Step 11 We did another visually timed test (watching our watches) by starting a continuous **ping** (**ping -t**) directed to the switch on a PC attached to the switch. We then disconnected the cable from the switch. The **ping**s started failing. Then we reconnected the cable to the switch and checked our watches to see how long it took for the switch to respond to the **ping**s from the PC. It took about 5 or 6 seconds with autonegotiation for speed and duplex turned on, and about 4 seconds with autonegotiation for speed and duplex turned off.

A lot of variables are involved in this test (PC initialization, PC software, switch console port responding to requests, and so on), but we just wanted to get some feel for how long it would take to get a response from the PCs point of view. All the tests were from the switches' internal debug message point of view.

How to Reduce Startup Delay on the Catalyst 2900XL/3500XL Switch

The 2900XL and 3500XL models can be configured from a web browser, by SNMP, or by the command-line interface (CLI). We will use the CLI. The following is an example of viewing the spanning-tree state of a port, turning on PortFast and then verifying that it is on. The 2900XL/3500XL *does* support EtherChannel and trunking, but it *does not* support dynamic EtherChannel creation (PAgP) or dynamic trunk negotiation (DTP) in the version that we tested (11.2[8.2]SA6), so we have no need to turn them off in this test. Also, after turning on PortFast, the elapsed time for the port to come up is already less than 1 second, so there is not much point in trying to change speed/duplex negotiation settings to speed things up. We hope that 1 second will be fast enough! By default, PortFast is off on the switch ports. The commands to turn on PortFast are discussed next.

Configuration

The following commands will help you reduce startup delay on the Catalyst 2900XL and 3500XL platforms:

```
2900XL#conf t
2900XL(config)#interface fastEthernet 0/1
2900XL(config-if)#spanning-tree portfast
2900XL(config-if)#exit
2900XL(config)#exit
2900XL#copy run start
```

This platform is like the router IOS; you must save the configuration (**copy run start**) if you want it to be permanently saved.

Verification

To verify that PortFast is enabled, use this command:

```
2900XL#show spanning-tree interface fastEthernet 0/1
Interface Fa0/1 (port 13) in Spanning tree 1 is FORWARDING
   Port path cost 19, Port priority 128
   Designated root has priority 8192, address 0010.0db1.7800
   Designated bridge has priority 32768, address 0050.8039.ec40
   Designated port is 13, path cost 19
   Timers: message age 0, forward delay 0, hold 0
   BPDU: sent 2105, received 1
   The port is in the portfast mode
```

Or, just look at the switch configuration.

```
2900XL#show running-config
Building configuration...

Current configuration:
!
version 11.2
...
!
interface VLAN1
 ip address 172.16.84.5 255.255.255.0
 no ip route-cache
!
interface FastEthernet0/1
 spanning-tree portfast
!
interface FastEthernet0/2
!
...
```

Timing Tests on the Catalyst 2900XL

The following steps show how to measure the effect of using PortFast on the 2900XL switch:

Step 1 The 11.2(8.2)SA6 version of software was used on the 2900XL for the following tests.

```
Switch#show version
Cisco Internetwork Operating System Software
IOS (tm) C2900XL Software (C2900XL-C3H2S-M), Version 11.2(8.2)SA6, MAINTENANCE INTERIM
SOFTWARE
Copyright (c) 1986-1999 by cisco Systems, Inc.
Compiled Wed 23-Jun-99 16:25 by boba
Image text-base: 0x00003000, data-base: 0x00259AEC

ROM: Bootstrap program is C2900XL boot loader

Switch uptime is 1 week, 4 days, 22 hours, 5 minutes
System restarted by power-on
System image file is "flash:c2900XL-c3h2s-mz-112.8.2-SA6.bin", booted via console

cisco WS-C2924-XL (PowerPC403GA) processor (revision 0x11) with 8192K/1024K bytes of
memory.
Processor board ID 0x0E, with hardware revision 0x01
Last reset from power-on

Processor is running Enterprise Edition Software
Cluster command switch capable
Cluster member switch capable
24 Ethernet/IEEE 802.3 interface(s)

32K bytes of flash-simulated non-volatile configuration memory.
Base ethernet MAC Address: 00:50:80:39:EC:40
Motherboard assembly number: 73-3382-04
Power supply part number: 34-0834-01
Motherboard serial number: FAA02499G7X
Model number: WS-C2924-XL-EN
System serial number: FAA0250U03P
Configuration register is 0xF
```

Step 2 We want the switch to tell us what is happening and when it is happening, so we enter the following commands:

```
2900XL(config)#service timestamps debug uptime
2900XL(config)#service timestamps log uptime
2900XL#debug spantree events
Spanning Tree event debugging is on
2900XL#show debug
General spanning tree:
  Spanning Tree event debugging is on
```

Step 3 Then we shut down the port in question.

```
2900XL#conf t
Enter configuration commands, one per line.  End with CNTL/Z.
2900XL(config)#interface fastEthernet 0/1
2900XL(config-if)#shut
2900XL(config-if)#
00:31:28: ST: sent Topology Change Notice on FastEthernet0/6
00:31:28: ST: FastEthernet0/1 - blocking
00:31:28: %LINK-5-CHANGED: Interface FastEthernet0/1, changed state to
administratively down
00:31:28: %LINEPROTO-5-UPDOWN: Line protocol on Interface FastEthernet0/1, changed
state to down
2900XL(config-if)#exit
2900XL(config)#exit
2900XL#
```

Step 4 At this point, we paste the following commands from the Clipboard into the switch. These commands show the time on the 2900XL and turn the port back on:

```
show clock
conf t
int f0/1
no shut
```

Step 5 By default, PortFast is off. You can confirm it two ways: The first way is that the **show spanning-tree interface** command would not mention PortFast; the second way is to look at the running config, where you would not see the **spanning-tree portfast** command under the interface.

```
2900XL#show spanning-tree interface fastEthernet 0/1
Interface Fa0/1 (port 13) in Spanning tree 1 is FORWARDING
   Port path cost 19, Port priority 128
   Designated root has priority 8192, address 0010.0db1.7800
   Designated bridge has priority 32768, address 0050.8039.ec40
   Designated port is 13, path cost 19
   Timers: message age 0, forward delay 0, hold 0
   BPDU: sent 887, received 1
[Note: there is no message about being in portfast mode is in this spot...]

2900XL#show running-config
Building configuration...
...
!
interface FastEthernet0/1
[Note: there is no spanning-tree portfast command under this interface...]
!
...
```

Step 6 Here is the first timing test with PortFast off:

```
2900XL#show clock
*00:27:27.632 UTC Mon Mar 1 1993
2900XL#conf t
Enter configuration commands, one per line.  End with CNTL/Z.
2900XL(config)#int f0/1
2900XL(config-if)#no shut
2900XL(config-if)#
00:27:27: ST: FastEthernet0/1 - listening
00:27:27: %LINK-3-UPDOWN: Interface FastEthernet0/1, changed state to up
00:27:28: %LINEPROTO-5-UPDOWN: Line protocol on Interface FastEthernet0/1, changed
state to up
00:27:42: ST: FastEthernet0/1 - learning
00:27:57: ST: sent Topology Change Notice on FastEthernet0/6
00:27:57: ST: FastEthernet0/1 - forwarding
```

Total time from shutdown until the port started forwarding was 30 seconds (27:27 to 27:57).

Step 7 To turn on PortFast, do the following:

```
2900XL#conf t
Enter configuration commands, one per line.  End with CNTL/Z.
2900XL(config)#interface fastEthernet 0/1
2900XL(config-if)#spanning-tree portfast
2900XL(config-if)#exit
2900XL(config)#exit
2900XL#
```

To verify that PortFast is enabled, use the **show spanning-tree interface** command. Notice that at the end of the command output, it says that PortFast is enabled.

```
2900XL#show spanning-tree interface fastEthernet 0/1
Interface Fa0/1 (port 13) in Spanning tree 1 is FORWARDING
   Port path cost 19, Port priority 128
   Designated root has priority 8192, address 0010.0db1.7800
   Designated bridge has priority 32768, address 0050.8039.ec40
   Designated port is 13, path cost 19
   Timers: message age 0, forward delay 0, hold 0
   BPDU: sent 1001, received 1
   The port is in the portfast mode
```

You can also see that PortFast is enabled in the configuration output:

```
2900XL#sh ru
Building configuration...
...
interface FastEthernet0/1
 spanning-tree portfast
...
```

Step 8 Now let's do the timing test with PortFast enabled.

```
2900XL#show clock
*00:23:45.139 UTC Mon Mar 1 1993
2900XL#conf t
Enter configuration commands, one per line.  End with CNTL/Z.
2900XL(config)#int f0/1
2900XL(config-if)#no shut
2900XL(config-if)#
00:23:45: ST: FastEthernet0/1 -jump to forwarding from blocking
00:23:45: %LINK-3-UPDOWN: Interface FastEthernet0/1, changed state to up
00:23:45: %LINEPROTO-5-UPDOWN: Line protocol on Interface FastEthernet0/1, changed
state to up
```

In this case, the total time was less than 1 second. If port initialization delay on the switch was the problem, then PortFast should solve it.

Remember, the switch does not currently support trunk negotiation, so we do not need to turn it off. Nor does it support PAgP for trunking, so we do not need to turn it off either. The switch does support autonegotiation of speed and duplex, but because the delay is so small, this would not be a reason to turn it off.

Step 9 We also did the **ping** test (just like Step 11 from the Catalyst 5000 example) from a workstation to the switch. It took about 5 to 6 seconds for the response to come from the switch whether autonegotiation for speed and duplex was on or off.

How to Reduce Startup Delay on the Catalyst 1900/2800 Switch

The 1900/2820 switches call PortFast by another name—they call it "spantree start-forwarding." For the version of software we are running to do our tests (V8.01.05), the switches default to having PortFast *enabled* on the Ethernet (10-Mbps) ports, and PortFast *disabled* on the Fast Ethernet (uplink) ports. So, when you use **show run** to view the configuration, if an Ethernet port says nothing about PortFast, then PortFast is enabled. If it says **no spantree start-forwarding** in the configuration, then PortFast is disabled. On a Fast Ethernet (100-Mbps) port, the opposite is true: For a Fast Ethernet port, PortFast is on only if the port shows **spantree start-forwarding** in the configuration.

Configuration

Here is an example of setting PortFast on a Fast Ethernet port. These examples use Enterprise edition software, version 8. The 1900 automatically saves the configuration after changes have been made. Remember, you would not want PortFast enabled on any port that connects to another switch or hub—only if the port attaches to an end station. The configuration is saved automatically to NVRAM.

```
1900#show version
Cisco Catalyst 1900/2820 Enterprise Edition Software
Version V8.01.05
Copyright (c) Cisco Systems, Inc.  1993-1998
1900 uptime is 0day(s) 01hour(s) 10minute(s) 42second(s)
cisco Catalyst 1900 (486sxl) processor with 2048K/1024K bytes of memory
Hardware board revision is 5
Upgrade Status: No upgrade currently in progress.
Config File Status: No configuration upload/download is in progress
27 Fixed Ethernet/IEEE 802.3 interface(s)
Base Ethernet Address: 00-50-50-E1-A4-80
1900#conf t
Enter configuration commands, one per line.  End with CNTL/Z
1900(config)#interface FastEthernet 0/26
1900(config-if)#spantree start-forwarding
1900(config-if)#exit
1900(config)#exit
1900#
```

Verification

One way to verify that PortFast is on is to look at the configuration. Remember, a Fast Ethernet port must say that it is on. An Ethernet port has PortFast enabled unless the configuration shows that it is off. In the configuration that follows, interface Ethernet 0/1 has PortFast turned off (you can see the command to turn it off), interface Ethernet 0/2 has PortFast on (you see nothing, which means that it is on), and interface Fast Ethernet 0/26 (port A in the menu system) has PortFast on (you can see the command to turn it on).

```
1900#show running-config
Building configuration...
...
!
interface Ethernet 0/1

  no spantree start-forwarding
!
interface Ethernet 0/2

!
...
!
interface FastEthernet 0/26
  spantree start-forwarding
!
```

The easiest way to view the PortFast status is through the menu system. If you select (P) for Port Configuration from the main menu and then select a port, the output will tell you whether port fast mode is enabled. The following output is for port Fast Ethernet 0/26 (which is port "A" on this switch).

```
    Catalyst 1900 - Port A Configuration

    Built-in 100Base-FX
    802.1d STP State:  Blocking     Forward Transitions:  0

---------------------- Settings ----------------------------------------
 [D] Description/name of port
 [S] Status of port                              Suspended-no-linkbeat
 [I] Port priority (spanning tree)               128 (80 hex)
 [C] Path cost (spanning tree)                   10
 [H] Port fast mode (spanning tree)              Enabled
 [E] Enhanced congestion control                 Disabled
 [F] Full duplex / Flow control                  Half duplex

---------------------- Related Menus -----------------------------------
 [A] Port addressing           [V] View port statistics
 [N] Next port                 [G] Goto port
 [P] Previous port             [X] Exit to Main Menu

Enter Selection:
```

Timing Tests on the Catalyst 1900

The timing values are harder to verify on a 1900/2820 because of the lack of debugging tools, so we just started a **ping** from a PC connected to the switch directed to the switch itself. We disconnected and then reconnected the cable, and recorded how long it took for the switch to respond to the **ping** with PortFast on and with PortFast off. For an Ethernet port with PortFast on (the default state), the PC received a response within 5 to 6 seconds. With PortFast off, the PC received a response in 34 to 35 seconds.

An Additional Benefit to PortFast

There is another spanning-tree-related benefit to using PortFast in your network. Every time a link becomes active and moves to the forwarding state in spanning tree, the switch will send a special spanning-tree packet called a Topology Change Notification (TCN). The TCN notification is passed up to the root of the spanning tree, where it is propagated to all the switches in the VLAN. This causes all the switches to age out their table of MAC addresses using the forward delay parameter, which is usually set to 15 seconds. So, every time a workstation joins the bridge group, the MAC addresses on all the switches will be aged out after 15 seconds instead of the normal 300 seconds.

When a workstation becomes active, it does not really change the topology to any significant degree, so as far as all the switches in the VLAN are concerned, it is unnecessary for them to have to go through the fast aging TCN period. If you turn on PortFast, the switch will not send TCN packets when a port becomes active.

Commands to Use for Verifying That the Configuration Is Working

For the 4000/5000/6000 series, use these commands to verify the configuration:

- **show port spantree 2/1**—To see whether "Fast-Start" (PortFast) is enabled or disabled

- **show spantree 1**—To see all ports in VLAN 1 and to determine whether they have "Fast-Start" enabled
- **show port channel**—To see whether you have any active channels
- **show port channel 2**—To see the channel mode (auto, off, and so on) for each port on module 2
- **show trunk 2**—To see the trunk mode (auto, off, and so on) for each port on module 2
- **show port**—To see the status (connected, notconnect, and so on), speed, and duplex mode for all ports on the switch

For the 2900XL/3500XL series, use these commands:

- **show spanning-tree interface FastEthernet 0/1**—To see whether PortFast is enabled on this port (No mention of PortFast means that it is not enabled.)
- **show running-config**—If a port shows the command **spanning-tree portfast,** then PortFast is enabled

For the 1900/2800 series, use this command:

- **show running-config**—To see the current settings (some commands are invisible when they represent the default settings of the switch)

  Use the menu system to view the port status screen.

Commands to Use for Troubleshooting the Configuration

For the 4000/5000/6000 series, use these commands in troubleshooting:

- **show port spantree 2/1**—To see whether "Fast-Start" (PortFast) is enabled or disabled
- **show spantree 1**—To see all ports in VLAN 1 and to determine whether they have "Fast-Start" enabled
- **show port channel**—To see whether you have any active channels
- **show port channel 2**—To see the channel mode (auto, off, and so on) for each port on module 2
- **show trunk 2**—To see the trunk mode (auto, off, and so on) for each port on module 2
- **show port**—To see the status (connected, notconnect, and so on), speed, and duplex mode for all ports on the switch
- **show logging**—To see what type of messages will generate logging output
- **set logging level spantree 7**—To set the switch to log the spanning tree port states in real time on the console
- **set port disable 2/1**—To turn off the port in software (like **shutdown** on the router)
- **set port enable 2/1**—To turn on the port in software (like **no shutdown** on the router)
- **show time**—To show the current time in seconds (used at the beginning of a timing test)
- **show port capabilities**—To see what features are implemented on the port
- **set trunk 2/1 off**—To set the trunking mode to off (to speed port initialization time)
- **set port channel 2/1-2 off**—To set the EtherChannel (PAgP) mode to off (to speed port initialization time)
- **set port speed 2/1 100**—To set the port to 100 Mbps and to turn off autonegotiation
- **set port duplex 2/1 full**—To set the port duplex to full

For the 2900XL/3500XL series, use these commands:

- **service timestamps debug uptime**—To show the time with the debug messages
- **service timestamps log uptime**—To show the time with the logging messages
- **debug spantree events**—So that we can see when the port moves through the spanning tree stages
- **show clock**—To see the current time (for the timing tests)
- **show spanning-tree interface FastEthernet 0/1**—To see whether PortFast is enabled on this port (No mention of PortFast means that it is not enabled.)
- **shut**—To turn off a port from software
- **no shut**—To turn on a port from software

For the 1900/2800 series, use this command:

- **show running-config**—To see the current settings (Some commands are invisible when they represent the default settings of the switch.)

Configuring and Troubleshooting IP Multilayer Switching

This document outlines basic troubleshooting of Multilayer Switching (MLS) for IP. This feature has become a highly desired method of accelerating routing performance through the use of dedicated application-specific integrated circuits (ASICs). Traditional routing is done through a central CPU and software; MLS offloads a significant portion of routing (packet rewrite) to hardware and thus has also been termed *switching*. *MLS* and *Layer 3 switching* are equivalent terms. The NetFlow feature of IOS is distinct and is not covered in this document. MLS also includes support for IPX (IPX MLS) and multicasting (MMLS), but this document will exclusively concentrate on basic MLS IP troubleshooting.

Introduction

As greater demands are placed on networks, the need for greater performance increases. More PCs are being connected to LANs, WANs, and the Internet, and their users require fast access to databases, files/web pages, networked applications, other PCs, and streaming video. To keep connections quick and reliable, networks must be capable of rapidly adjusting to changes and failures and finding the best path, all while remaining as invisible as possible to end users. End users who experience rapid information flow between their PC and server with minimal network slowness are happy ones. Determining the best path is the primary function of routing protocols, and this can be a CPU-intensive process; thus, a significant performance increase is gained by offloading a portion of this function to switching hardware. This is the point of the MLS feature.

There are three major components of MLS: Two of them are the MLS-RP and the MLS-SE. The MLS-RP is the MLS-enabled router, performing the traditional function of routing between subnets/VLANs. The MLS-SE is an MLS-enabled switch, which normally requires a router to route between subnets/VLANs but, with special hardware and software, can handle rewriting of the packet. When a packet transverses a routed interface, nondata portions of the packet are changed (rewritten) as it is carried to its destination, hop by hop.

Confusion can arise here because it seems that a Layer 2 device is taking on a Layer 3 task; actually, the switch is only rewriting Layer 3 information and is “switching” between subnets/VLANs—the router is still responsible for standards-based route calculations and best-path determination. Much of this confusion can be avoided by mentally keeping the routing and switching functions separate, especially when, as is commonly the case, they are contained within the same chassis (as with an internal MLS-RP).

Think of MLS as a much more advanced form of route caching, with the cache kept separate from the router on a switch. Both the MLS-RP and the MLS-SE, along with respective hardware and software minimums, are required for MLS.

The MLS-RP can be internal (installed in a switch chassis) or external (connected via a cable to a trunk port on the switch). Examples of internal MLS-RPs are the Route-Switch Module (RSM) and the Route-Switch Feature Card (RSFC), which are installed in a slot or supervisor of a Catalyst 5xxx family member, respectively; the same applies to the Multilayer Switch Feature Card (MSFC) for the Catalyst 6xxx family. Examples of external MLS-RPs include any member of the Cisco 7500, 7200, 4700, 4500, or 3600 series routers. In general, to support the MLS IP feature, all MLS-RPs require a minimum IOS version in the 11.3WA or 12.0WA trains; consult release documentation for specifics. Also, *MLS must be enabled* for a router to be an MLS-RP**.**

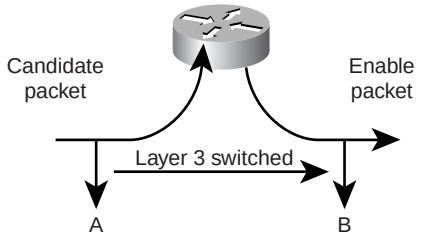
The MLS-SE is a switch with special hardware. For a member of the Catalyst 5xxx family, MLS requires that the supervisor have a NetFlow Feature Card (NFFC) installed; the Supervisor IIG and IIIG have one by default. In addition, a bare minimum of Catalyst OS 4.1.1 software is required. Note that the 4.x train has "gone General Deployment (GD)"—that is, passed rigorous end-user criteria and field-experience targets for stability—so check Cisco's web site for the latest releases. IP MLS is supported and automatically enabled for Catalyst 6xxx hardware and software with the MSFC/PFC (other routers have MLS disabled by default). Note that IPX MLS and MLS for multicasting may have different hardware and software (IOS and Catalyst OS) requirements. More Cisco platforms do/will support the MLS feature. Also, *MLS must be enabled* for a switch to be an MLS-SE**.**

The third major component of MLS is the Multilayer Switching Protocol (MLSP). Because understanding the basics of MLSP gets at the heart of MLS and is essential to performing effective MLS troubleshooting, we will describe MLSP here more in detail. MLSP is utilized by the MLS-RP and the MLS-SE to communicate with one another—tasks include enabling MLS; installing, updating, or deleting flows (cache information); and managing and exporting flow statistics (NetFlow Data Export is covered in other documentation). MLSP also allows the MLS-SE to learn the Media Access Control (MAC, Layer 2) addresses of the MLS-enabled router interfaces, check the flowmask of the MLS-RP (explained later in this chapter), and confirm that the MLS-RP is operational. The MLS-RP sends out multicast "hello" packets every 15 seconds using MLSP; if three of these intervals are missed, then the MLS-SE recognizes that the MLS-RP has failed or that connectivity to it has been lost.

Figure 23-7 illustrates three essentials that must be completed (using MLSP) for a shortcut to be created: the candidate, enabler, and caching steps. The MLS-SE checks for a cached MLS entry; if MLS cache entry and packet information match (a hit), the packet's header is rewritten locally on the switch (a shortcut, or bypassing of the router) instead of being sent on to the router, as would normally happen. Packets that do not match and that are sent on to the MLS-RP are *candidate packets*—that is, there is a possibility of switching them locally.

After passing the candidate packet through the MLS flowmask (explained later in Step 7) and rewriting the information contained in the packet's header (the data portion is not touched), the router sends it toward the next hop along the destination path. The packet is now called an *enabler packet*. If the packet returns to the same MLS-SE from which it left, an MLS shortcut is created and placed into the MLS cache. Rewriting for that packet and all similar packets that follow (called a flow) is now done locally by switch hardware instead of by router software. *The same MLS-SE must see both the candidate and the enabler packets for a particular flow for an MLS shortcut to be created.* (This is why network topology is important to MLS.) Remember, the point of MLS is to allow the communication path between two devices in different VLANs, connected off the same switch, to bypass the router and thus enhance network performance.

Figure 23-7 The Three Essentials That Must Be Completed (Using MLSP) for a Shortcut to Be Created: the Candidate, Enabler, and Caching Steps



By using the flowmask (essentially an access list), the administrator can adjust the degree of similarity of these packets, and thus adjust the scope of the flows: destination address; destination and source addresses; or destination, source, and Layer 4 information. Note that the first packet of a flow always passes through the router; from then on it is locally switched. Each flow is unidirectional—communication between PCs, for example, requires the setup and use of *two* shortcuts. The main purpose of MLSP is to set up, create, and maintain these shortcuts.

These three components (the MLS-RP, the MLS-SE, and the MLSP) free up vital router resources by allowing other network components to take on some of its functions. Depending on the topology and configuration, MLS provides a simple and highly effective method of increasing network performance in the LAN.

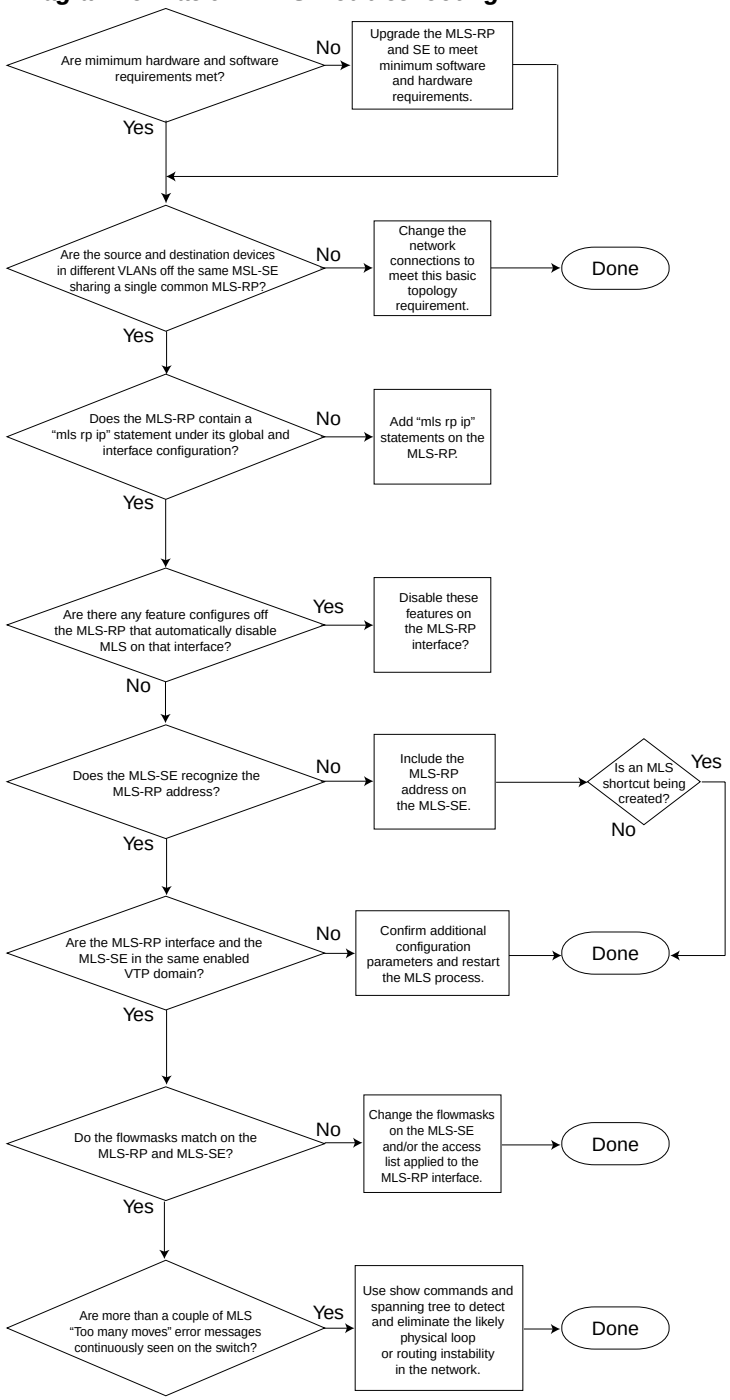
Troubleshooting IP MLS Technology

Figure 23-8 is a flow diagram for basic IP MLS troubleshooting. It is derived from the most common types of MLS-IP cases opened with the Technical Assistance Center (TAC) and faced by our customers and TAC engineers, up to the time that this document was created. MLS is a robust feature, and you should have no problems with it. However, if an issue does arise, the following should help you to resolve the types of IP MLS problems that you might likely face. A few essential assumptions have been made:

- That you are familiar with the basic configuration steps required to enable IP MLS on the router and switches, and that you have completed these steps. See the resources listed at the end of this document for excellent material.
- That IP routing is enabled on the MLS-RP (it is on by default). If the command **no ip routing** appears in the global configuration of a **show run**, it has been turned off, and IP MLS will not function.
- That IP connectivity exists between the MLS-RP and MLS-SE. You can **ping** the IP addresses of the router from the switch and look for exclamation points (called “bangs”) to be displayed in return.
- That the MLS-RP interfaces are in an “up/up” state on the router. Type **show ip interface brief** on the router to confirm this.

Warning **Whenever making configuration changes to a router intended to be permanent, remember to save those changes with a copy running-config starting-config (shortened versions of this command include copy run start and wr mem). Any configuration modifications will be lost if the router reloads or is reset. The RSM, RSFC, and MSFC are routers, not switches. In contrast, changes made at the switch prompt of a Catalyst 5xxx or 6xxx family member are automatically saved.**

Figure 23-8 Flow Diagram for Basic IP MLS Troubleshooting



Step 1 Are minimum hardware and software requirements met?

Upgrade the MLS-RP and SE to meet minimum software and hardware requirements. For the MLS-RP, no additional hardware is required. Although MLS can be configured on nontrunked interfaces, the connection to the MLS-SE is generally through VLAN interfaces (as with an RSM) or support trunking (can be configured to carry multiple VLAN information by configuring ISL or 802.1q).

Also, remember that, as of publication time, only members of the 7500, 7200, 4700, 4500, and 3600 router families support MLS externally. Currently, only these external routers and the routers that fit into the Catalyst 5xxx or 6xxx switch families (such as the RSM and RSFC for the Catalyst 5xxx family, and the MSFC for the Catalyst 6xxx family) can be MLS-RPs. The MSFC requires the Policy Feature Card (PFC) as well, both installed on the Catalyst 6xx Supervisor. IP MLS is now a standard feature in IOS 12.0 and later router software. IOS software lower than IOS 12.0 generally requires a special train; for such IP MLS support, install the latest images in IOS 11.3 that have the letters "WA" in their filenames.

For the MLS-SE, a NetFlow Feature Card (NFFC) is required for a member of the Catalyst 5xxx family; this card is installed in the Supervisor module of the Catalyst switch and is included as standard hardware in newer Catalyst 5xxx series Supervisors (since 1999). The NFFC is not supported on the Supervisors I or II and is an option on early Supervisor IIIs. Also, a minimum of 4.1.1 CatOS is required for IP MLS. In contrast, for the Catalyst 6xxx family, the required hardware comes as standard equipment, and IP MLS has been supported since the first CatOS software release, 5.1.1 (in fact, IP MLS is an essential and default ingredient for its high performance). With new platforms and software being released that support IP MLS, it is important to check documentation and release notes, and to generally install the latest release in the lowest train that meets your feature requirements. Always check the release notes and consult with your local Cisco sales office for new MLS support and feature developments.

Commands to check the installed hardware and software are **show version** on the router, and **show module** on the switch.

Note The Catalyst 6xxx family of switches does *not* support an external MLS-RP at this time. The MLS-RP must be an MSFC.

Step 2 Are the source and destination devices in different VLANs off the same MLS-SE, sharing a single common MLS-RP?

It is a basic topology requirement of MLS that the router have a path to each of the VLANs. Remember that the point of MLS is to create a shortcut between two VLANs so that the "routing" between the two end devices can be performed by the switch, thus freeing the router for other tasks. The switch is not actually routing; it is rewriting the frames so that it appears to the end devices that they are talking through the router. If the two devices are in the same VLAN, then the MLS-SE will switch the frame locally without utilizing MLS, as switches do in such a transparently bridged environment, and no MLS shortcut will be created. It is possible to have multiple switches and routers in the network, and even multiple switches along the flow path, but the path between the two end devices for which an MLS shortcut is desired must include a single MLS-RP in that VLAN for that path.

In other words, the flow from source to destination must cross a VLAN boundary on the same MLS-RP, and a candidate and enabler packet pair must be seen by the same MLS-SE for the MLS shortcut to be created. If these criteria are not met, then the packet will be routed normally without the use of MLS. See the documents suggested at the end of this chapter for diagrams and discussions regarding supported and unsupported network topologies.

Step 3 Does the MLS-RP contain an **mls rp ip** statement under *both* its global and interface configuration?

If one is not present, add **mls rp ip** statements appropriately on the MLS-RP. Except for routers for which IP MLS is automatically enabled (such as the Catalyst 6xxx MSFC), this is a required configuration step. For most MLS-RPs (routers configured for IP MLS), this statement must appear both in the global configuration and under the interface configuration.

Note When configuring the MLS-RP, also remember to place the **mls rp management-interface** command under one of its IP MLS interfaces. This required step tells the MLS-RP out which interface it should send MLSP messages to communicate with the MLS-SE. Again, it is necessary to place this command under one interface only.

Step 4 Are any features configured on the MLS-RP that automatically disable MLS on that interface?

Several configuration options on the router are not compatible with MLS. These include IP accounting, encryption, compression, IP security, network address translation (NAT), and committed access rate (CAR). For further information, see links regarding IP MLS configuration included at the end of this chapter. Packets traversing a router interface configured with any of these features must be routed normally; no MLS shortcut will be created. For MLS to work, you must disable these features on the MLS-RP interface.

Another important feature that affects MLS is access lists, both input and output. Further information on this option is included in the discussion of flowmasks (Step 7).

Step 5 Does the MLS-SE recognize the MLS-RP address?

For MLS to function, the switch must recognize the router as an MLS-RP. Internal MLS-RPs (again, the RSM or RSFC in a Catalyst 5xxx family member, and the MSFC in a Catalyst 6xxx family member) are automatically recognized by the MLS-SE in which they are installed. For external MLS-RPs, you must explicitly inform the switch of the router's address. This address is not actually an IP address, although on external MLS-RPs it is chosen from the list of IP addresses configured on the router's interfaces; it is simply a router ID. In fact, for internal MLS-RPs, the MLS-ID is normally not even an IP address configured on the router. Because internal MLS-RPs are included automatically, it is commonly a loopback address (127.0.0.x). For MLS to function, include on the MLS-SE the MLS-ID found on the MLS-RP.

Use **show mls rp** on the router to find the MLS-ID, and then configure that ID on the switch using **the set mls include <*MLS-ID*>** command. This is a required configuration step when using external MLS-RPs.

Warning **Changing the IP address of MLS-RP interfaces and then reloading the router may cause the MLS process on the router to choose a new MLS-ID. This new MLS-ID may be different from the MLS-ID that was manually included on the MLS-SE, which may cause MLS to cease functioning. This is not a software glitch, just an effect of the switch trying to communicate with a MLS-ID that is no longer valid. Be sure to include this new MLS-ID on the switch to get MLS working once again. You may have to disable/enable IP MLS as well.**

Note When the MLS-SE is not directly connected to the MLS-RP, the address that must be included on the MLS-SE may appear as the loopback address mentioned previously: a switch connected in between the MLS-SE and MLS-RP. You must include the MLS-ID even though the MLS-RP is internal. To the second switch, the MLS-RP appears as an *external* router because the MLS-RP and MLS-SE are not contained in the same chassis.

Step 6 Are the MLS-RP interface and the MLS-SE in the same enabled VTP domain?

MLS requires that MLS components, including the end stations, must be in the same Virtual Trunking Protocol (VTP) domain. VTP is a Layer 2 protocol used for managing VLANs on several Catalyst switches from a central switch; it allows an administrator to create or delete a VLAN on all switches in a domain without having to do so on every switch in that domain. The MLSP, which the MLS-SE and the MLS-RP use to communicate with one another, does not cross a VTP domain boundary. If the network administrator has VTP enabled on the switches (VTP is enabled on Catalyst 5xxx and 6xxx family members by default), use the **show vtp domain** command on the switch to learn in which VTP domain the MLS-SE has been placed. Except for the Catalyst 6xxx MSFC, on which MLS is essentially a plug-and-play feature, add, *in the following steps*, the VTP domain to each of the router's MLS interfaces. This will permit MLSP multicasts to move between the MLS-RP and MLS-SE, and therefore allow MLS to function.

In interface configuration mode of the MLS-RP, enter the following commands:

- **no mls rp ip**—Disable MLS on the affected MLS-RP interface before modifying the VTP domain.
- **mls rp vtp-domain < VTP domain name>**—The VTP domain name on each MLS-enabled interface must match that of the switch.
- **mls rp vlan-id <VLAN #>**—This is required only for non-ISL trunking, external MLS-RP interfaces.
- **mls rp management-interface**—Do this for only one interface on the MLS-RP. This required step tells the MLS-RP out which interface it should send MLSP messages.
- **mls rp ip**—Enable MLS once again on the interface of the MLS-RP.

To change the VTP domain name of the MLS-SE, use the following command at the switch CatOS enable prompt:

set vtp domain name <VTP domain name>

For MLS to work, be sure that VTP is enabled on the switch:

set vtp enable

Step 7 Do the flowmasks agree on the MLS-RP and MLS-SE?

A flowmask is a filter configured by a network administrator that is used by MLS to determine whether a shortcut should be created. Just like an access list, the more detailed the criteria you set up, the deeper into the packet the MLS process must look to verify whether the packet meets those criteria. To adjust the scope of MLS-created shortcuts, the flowmask can be made more or less specific; the flowmask is essentially a "tuning" device.

There are three types of IP MLS modes: destination-ip, destination-source-ip, and full-flow-ip. *Destination-ip* mode, the default, is in use when no access list is applied to the router's MLS-enabled interface. *Source-destination-ip* mode is in use when a standard access list is applied, and *full-flow-ip* is in effect for an extended access list. The MLS mode on the MLS-RP is implicitly determined by the type of access list applied to the interface. By contrast, the MLS mode on the MLS-SE is explicitly configured. By choosing the appropriate mode, you can thus configure MLS so that either only the destination address must match for an MLS shortcut to be created, or both source and destination must match, or even Layer 4 information such as TCP/UDP port numbers must match.

The MLS mode is configurable on both the MLS-RP and the MLS-SE, and in general they must match. However, if either source-destination-ip or full-flow-ip MLS modes are deemed to be required, it is best to configure it on the router by applying the appropriate access list. MLS will always choose the most specific mask, giving the flowmask configured on the MLS-RP precedence over the one found on the MLS-SE. *Be careful* if you change the MLS mode of the switch from the default destination-ip: You

should make sure that it matches the MLS mode on the router for MLS to work. For source-destination-ip and full-flow-ip modes, remember to apply the access list to the appropriate router interface. With no access list applied, even if configured, the MLS mode simply will be destination-ip, the default.

Warning **Whenever the flowmask is changed, whether on the MLS-RP or on the MLS-SE, all cached MLS flows are purged, and the MLS process is restarted. A purge also can occur when applying the command clear ip route-cache on the router. Applying the global router configuration command no ip routing, which turns off IP routing and essentially transforms the router into a transparent bridge, will cause a purge and disable MLS (remember, routing is a prerequisite of MLS). Each of these may temporarily—but seriously—affect router performance in a production network because the router will experience a spike in its load until the new shortcuts are created: After all, it must now handle all the flows that were just previously being processed by the switch.**

Note Especially with a member of the Catalyst 5000 family as the MLS-SE, it is best to avoid the very wide use of flowmasks that are configured with Layer 4 information. By forcing the router to peer so deeply into every packet on the interface, much of the intended benefits of MLS are bypassed. This is much less of an issue when utilizing a Catalyst 6xxx family member as the MLS-SE because the switch ports themselves can recognize Layer 4 information.

Note Until recently, MLS did not support flowmasks configured inbound on an MLS-RP interface, only outbound. Now, by using the **mls rp ip input-acl** command in addition to normal MLS-RP configuration commands on a router interface, an inbound flowmask is supported.

Step 8 Are more than a couple of MLS "Too many moves" error messages continuously seen on the switch?

As the previous note mentions, changing a flowmask, clearing the route cache, or globally turning off IP routing will cause a cache purge. Other circumstances can also cause full or many single entry purges, and cause MLS to complain of "Too many moves." There are several forms of this message, but each contains these three words. Aside from what has already been mentioned, the most common cause of this error occurs when the switch learns multiple identical Ethernet Media Access Control (MAC) address within the same VLAN; Ethernet standards do not allow for identical MAC addresses within the same VLAN. If you see this message infrequently, or just a few times in a row, there is no cause for concern. MLS is a robust feature, and the message may be simply caused by normal network events, such as a PC connection being moved between ports, for example. If you see this message continuously for several minutes, however, it is likely a symptom of a more serious issue.

When such a situation arises, its root cause is commonly the presence of two devices with the same MAC address actually connected to a VLAN, or a physical loop within the VLAN (or multiple VLANs, if bridging across these broadcast domains). Use spanning-tree troubleshooting covered in the section "Troubleshooting Spanning-Tree Protocol and Related Design Considerations" found later in this chapter and the hint that follows to find the loop and eliminate it. Also, any rapid topology changes can cause temporary network (and MLS) instability (flapping router interfaces, a bad NIC,and so on).

Tips Use the **show mls notification** and **show looktable** commands on the switch to point you in the right direction of the duplicate MAC address or physical loop. The first will provide a TA value; the command **show looktable <TA value>** will return a possible MAC address that may be traced to the root of the problem.

Commands or Screen Captures

For descriptions and detailed examples of IP MLS router and switch commands, refer to the excellent documentation listed under the "Additional Sources" section.

Before Calling Cisco Systems' TAC Team

Before calling Cisco Systems's Technical Assistance Center (TAC), make sure you that have read through this chapter and completed the actions suggested for your system's problem.

Additionally, do the following and document the results so that we can better assist you:

- Capture the output of **show module** from all the affected switches.
- Capture the output of **show vtp domain** from all the affected switches.
- Capture the output of **show trunk <mod_num/port_num>** from all the affected ports.
- Capture the output of **show port <mod_num/port_num> capabilities** from all the affected ports.
- Capture the output of **show tech-support** from the MLS-RP.
- Capture the output of **show mls rp** on the MLS-RP and both **show mls** and **sh mls include** on the MLS-SEs.
- The output of additional commands may be necessary, depending on the nature of the issue.

A clear network topology and dial-in or Telnet access also help considerably in effective problem resolution.

Troubleshooting Spanning-Tree Protocol and Related Design Considerations

Note The text in this section comes directly from the Cisco web site www.cisco.com/warp/customer/473/16.html.

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 section 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 following topics:

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

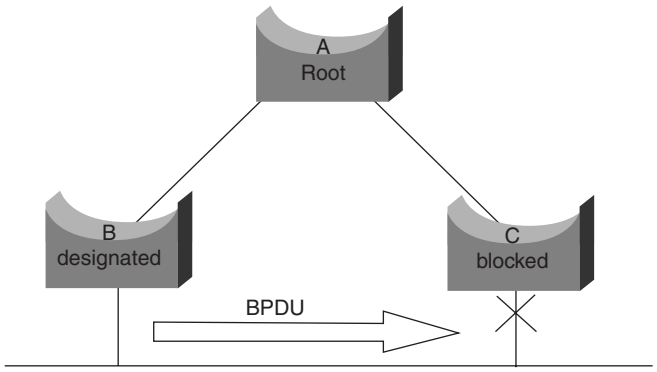
Spanning-Tree Protocol Failure

A failure in the STA generally leads to a bridging loop (not a spanning-tree loop because you don't need STP to have a loop). Most customers calling the TAC for spanning-tree problems suspect a bug, but experience proves that it is seldom the case. Even if the software is at stake, a bridging loop in an 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 first 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 toward the root bridge.

In Figure 23-9, Bridge B has been elected as the designated bridge, and Bridge C is blocking because it is only providing an alternate path to the root. Why is Bridget C blocking, not Bridge B? This is determined practically by the BPDUs that B and C exchange on the LAN. Here, Bridget B had a better BPDU than Bridge C. Bridge B keeps sending BPDUs advertising its superiority over the other bridges on this LAN. If Bridge C fails to receive these BPDUs for a certain period of time (called the max age—20 seconds, by default), it would start a transition to the forwarding mode.

Figure 23-9 Blocked Port on Bridge C Keeps Receiving BPDUs from Bridge B



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

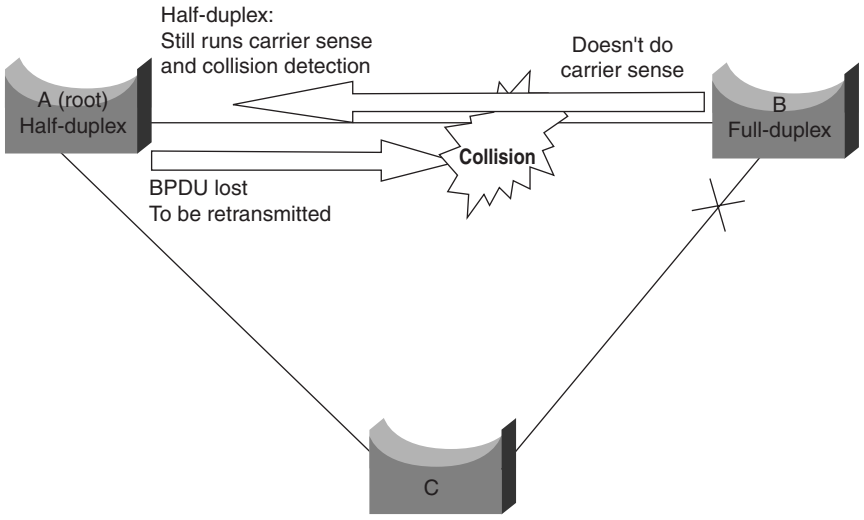
The following subsections list the different situations that can lead the STA to fail. Most of these failures are, in fact, related to a massive loss of BPDUs, causing blocked ports to transition to forwarding mode.

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 hard-coded as full duplex. If you leave the other side in autonegotiation mode, it will end up in half-duplex mode (a port with duplex hard-coded does not negotiate anymore).

The worst-case scenario is when a bridge sending BPDUs is configured for half-duplex operation on a link, whereas its peer is configured for full-duplex mode. In Figure 23-10, the duplex mismatch on the link between bridges A and B can easily lead to a bridging loop. Because Bridge B is configured for full-duplex operation, it does not perform carrier sense when accessing the link. Bridge B will then start sending frames even if Bridge A is already using the link. This is a problem for Bridge A, which detects a collision and runs the back-off 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 Bridge A will be deferred or collisioned and eventually dropped. From an STP point of view, because it does not receive BPDUs from Bridge A anymore, Bridge B has lost its root. This leads Bridge B to unblock its port to Bridge C, hence creating the loop.

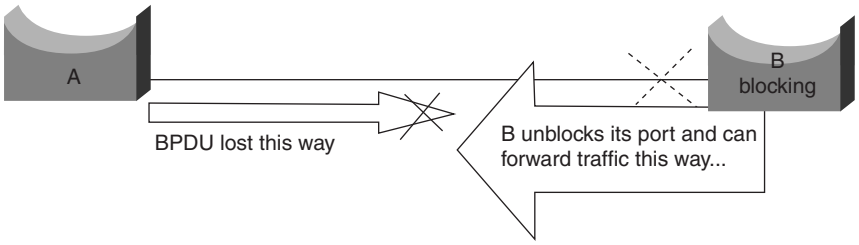
Figure 23-10 Bridging Loop Created by a Duplex Mismatch



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 shown in Figure 23-11 is very straightforward.

Figure 23-11 Bridging Loop in a Unidirectional Link Scenario



Here, let's suppose that the link between bridges A and B is unidirectional and drops traffic from A to B while transmitting traffic from B to A. Suppose that Bridge B should be blocking. We already mentioned that a port can block only if it receives BPDUs from a bridge that has a better priority. In this case, all these BPDUs coming from Bridge 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. This means that rebooting the bridges will have absolutely no effect (whereas it could temporarily help in the previous case).

Cisco introduced the UDLD protocol on high-end switches. This feature is capable of detecting wrong cabling or unidirectional links on Layer 2 and automatically breaks 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 overutilized 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. You will see in the upcoming section "Look for Resource Errors" that there are some guidelines that govern the number of instances of STP that a particular platform can handle.

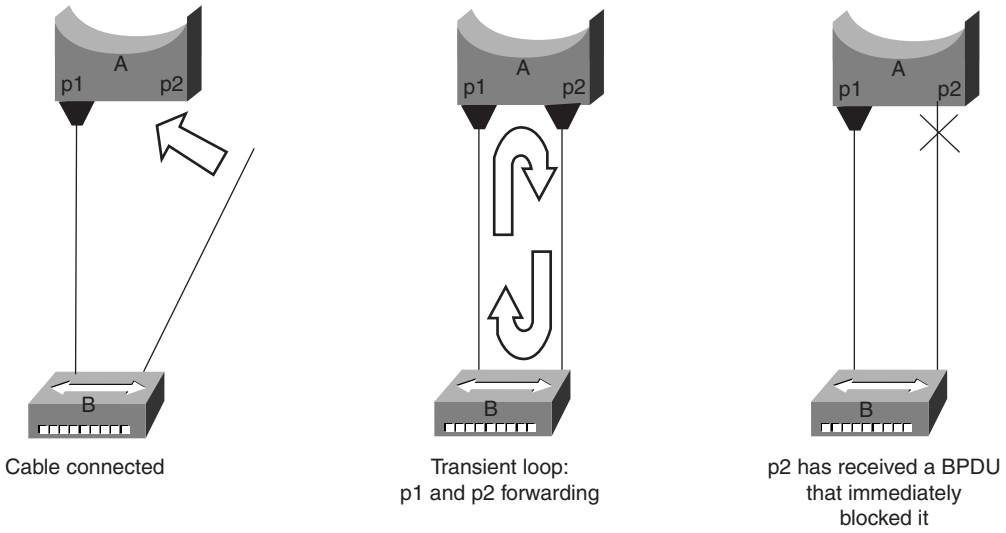
PortFast Configuration Error

PortFast is a feature that you typically will want 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 Figure 23-12, Bridge A is a bridge with port p1 already forwarding and port p2 configured for PortFast. Bridge B is a hub. As soon as the second cable is plugged into Bridge A, p2 goes to forwarding and creates 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 modes. 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 implements a feature called BPDU guard that will even disable the port if it is configured for PortFast and receives a BPDU.

Figure 23-12 Transient Bridging Loop Because of a Wrong PortFast Configuration



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 well known, is related to the diameter of the bridged network. The conservative default values for the STP impose a maximum network diameter of 7. This means that two distinct bridges in the network should not be more than seven bridges away the one to the other. Part of this restriction comes from the Age field BPDUs carry: When a BPDU is propagated from the root bridge toward the leaves of the tree, the Age field is incremented each time that it goes though a bridge. Eventually, when the Age field of a BPDU goes beyond the 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 affect convergence of the spanning tree.

Software Errors

As mentioned in the introduction to this chapter, 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 that 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 an STP issue. This section instead looks like a checklist, recapitulating some of the actions available to you. Most of the indications given here apply to 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 following:

- The topology of the bridged network
- Where the root bridge is located
- Where the blocked ports (and 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 when it is working?
- Most of the troubleshooting steps are simply using **show** commands to try to identify error conditions. Knowledge of the network helps you focus 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 a bridging loop for sure is to capture the traffic on a saturated link and to check that you see similar packets multiple times.

But practically, if all users in a certain bridging domain have connectivity issues 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" upcoming section for additional information.

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 not only gives you the current usage of the switch backplane, but it also specifies the peak usage (and its date). An unusual peak utilization shows you whether there has ever been a bridging loop on this device.

Restore Connectivity Quickly and Be Ready for Another Time

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 to restore connectivity as soon as possible. If you do this, you will not find the real cause of the issue and need to be ready for the next time that it occurs.

Break the Loop Disabling Ports

The easy way out of a bridging loop 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 by 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 break it. Knowing which disabled port stopped the loop, you can be 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, for instance, if the problem is only transient—enable the logging of the 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 because 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** increases 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 this output to a syslog device. Unfortunately, when a bridging loop occurs, you seldom can keep connectivity to a syslog server.

Check Ports

As mentioned before, the critical ports to be investigated first are the blocking ports. The next section gives 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 that you keep receiving BPDUs periodically. Several issues can lead to a port not receiving packets/BPDUs:

- 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 that 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 to checking whether you are receiving BPDUs is to simply enable STP debug with the **debug spantree tree** command.

  For 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 <modele#/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 section "An Additional CatOS Command," later in this chapter, for more information.

Check for Duplex Mismatch

To look for a duplex mismatch, you obviously have to check 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 **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 <modele#/port#>**. The command **show top** automatically evaluates the port utilization over a 30-second period of time and displays the result classified by percentage bandwidth utilization (other options are available). Also, the **show system** command gives an indication on the backplane utilization, even if it does not point 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 <modele#/port#>** gives 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 <modele#/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 <modele#/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 transitions from learning to forwarding. In a stable topology, this counter should always show 1. This counter is reset to 0 if the corresponding port is going down and up. So, if the value is higher than 1, it means that the transition that 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 that the max age expired on this link. Basically, a port expecting BPDUs will wait for the 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 problems 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 percent.

- **CatOS**—Look for the field RsrcErrors (resource error) in the output of **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 that a supervisor engine can handle. Check the release notes of the software that you are running for this.

The following 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)

+ 2*(number of ATM trunks × number of active VLANs on that trunk)

+ number of nontrunking ports.

The sum of all logical ports, as calculated with this formula, 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 to simplify 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 generally, going to a configuration as simple as possible reduces the troubleshooting effort.

Useful Commands

This section lists useful commands for the Catalyst IOS and the Catalyst OS.

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 to Avoid Trouble

We have seen that the spanning tree can fail in some few circumstances and that troubleshooting the related issues can be quite difficult in a live network. This part introduces some guidelines to reduce the risks associated with the spanning tree.

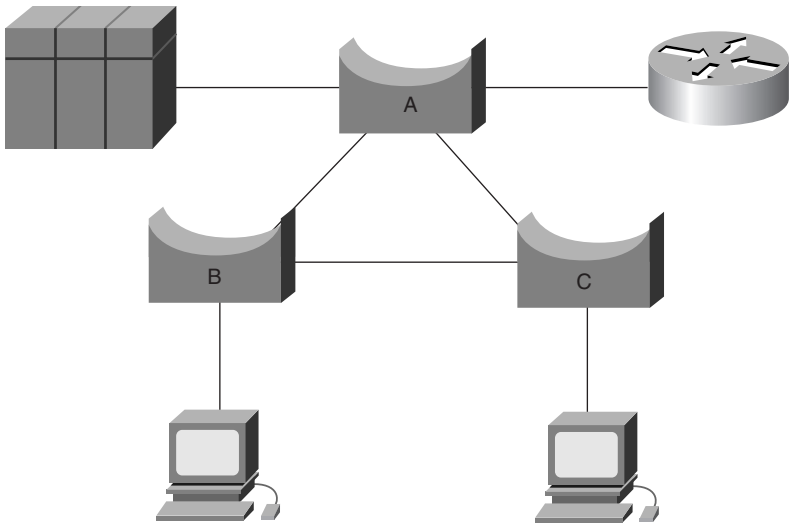
Know Where the Root Is

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, 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 Figure 23-13, you can clearly see that if Bridge B is the root, the link from 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 two hops, and hosts connected to Bridge C in three hops. That makes an average of 2.5 hops.

If Bridge A is the 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 two hops.

Figure 23-13 Root Bridge Location Is Important



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

Important note: For each VLAN, hard-code 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 that 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.

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 you identify the loop and its cause. Knowing where the blocked ports should be also help you to find where the error is coming from (by simple comparison).

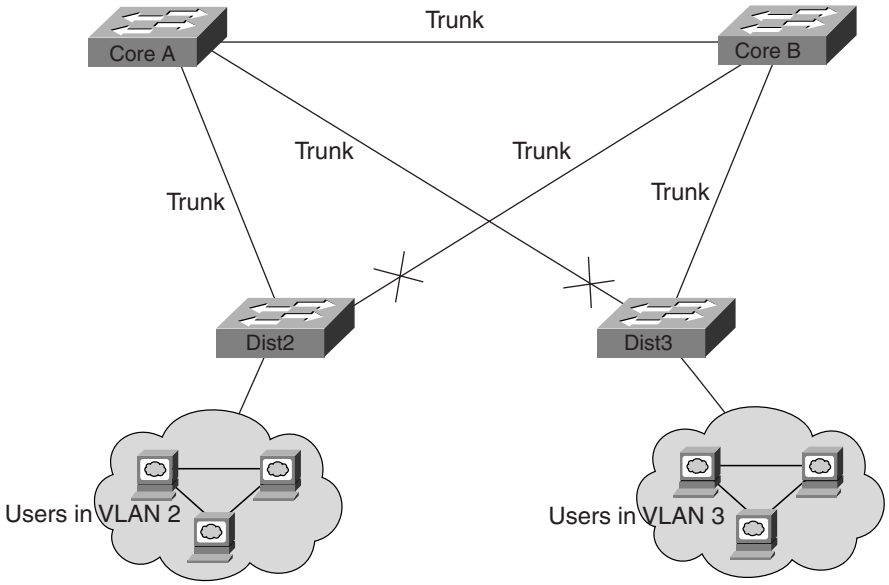
Minimize the Number of Blocked Ports

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 to limit the risk implied by the use of the STP is to reduce the number of blocked ports as much as possible.

Prune VLANs That Are Not Used

You don't need more than two redundant links between two nodes in a bridged network. However, a configuration like that shown in Figure 23-14 frequently appears.

Figure 23-14 Typical Network Design with VLANs Spanning Too Many Links



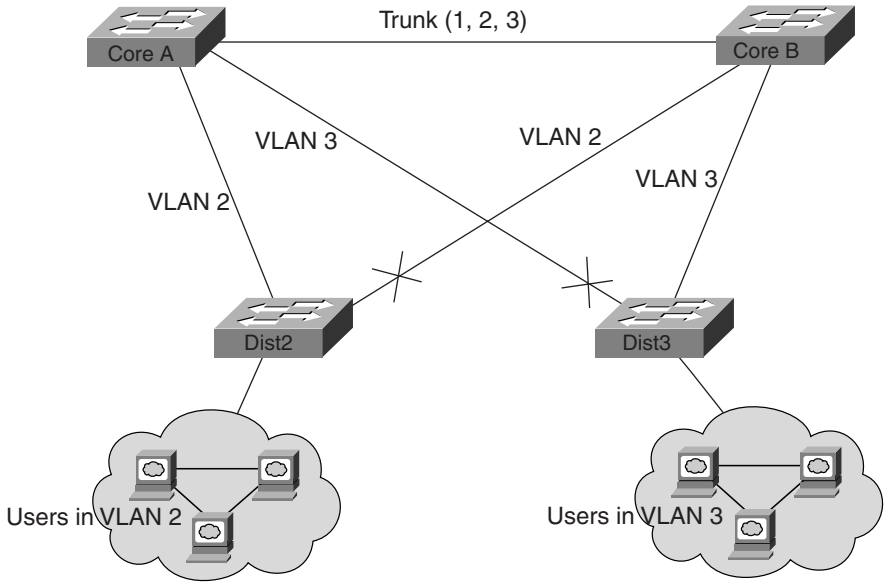
This is a very common design. Distribution switches are dual-attached to two core switches. Users connected on distribution switches are in only a subset of the VLANs available in the network (here, users connected on Dist2 are all in VLAN 2; Dist3 connects only users in VLAN 3). By default, trunks carry all the VLANs defined in the VTP domain. Now only is Dist2 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 increased 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 shown in Figure 23-14. This time, we just use an access VLAN to connect the distribution switches to the core, as shown in Figure 23-15.

Figure 23-15 Pruning VLANs Already Reduces the Number of Blocked Ports and Avoids Unnecessary Flooding



In this design, we have only 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.

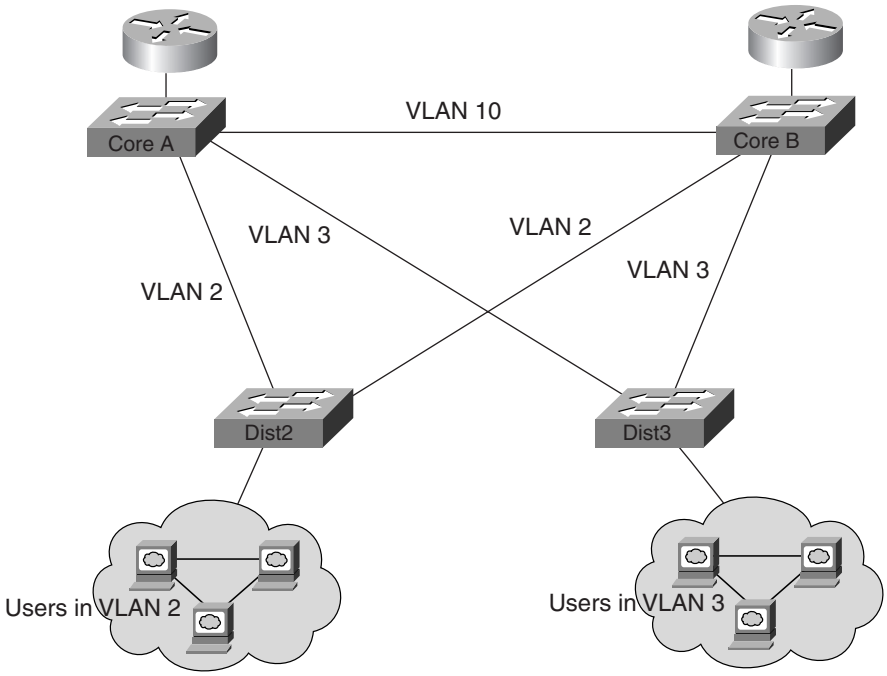
Use Layer 3 Switching

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 now can 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. Figure 23-16 illustrates this, using the same diagram structure.

Figure 23-16 Layer 3 Switching Makes a Design with No Blocking Port Possible



In Figure 23-16, 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 potential for a bridging loop.
- There is no speed penalty because 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 rework 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 melt down in a fraction of second if an operator makes an error on a patch panel, for instance. Generally, 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

Keeping traffic of the administration VLAN and avoiding having a single VLAN spanning the entire network are related points.

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 the CPU and impact its capability to process vital BPDUs. Therefore, it is always a good idea to keep user traffic off the administrative VLAN.

Until recently, in a Cisco implementation, there was no way to remove VLAN 1 from a trunk. This VLAN is generally used as an administrative VLAN, where all switches are accessible in the same IP subnet. Although 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 is. 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 still exists 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 values. (Another option is to use CatOS macros.) Trying to get faster reconvergence from this, for instance, is very dangerous because it has implications on the diameter of the network and the stability of the STP. The only parameters that 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 macro decreases the bridge priority so that it becomes root (or alternate root). You have an additional option that helps you tune 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 such as uplink fast or backbone fast, or a good Layer 3 switching design) and introduces some instability risks in the network. That kind of tuning must 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 this 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 with a blocked port, you have a 50 percent chance of a bridging loop. This is the most dangerous possibility of STP failure because the algorithm is not capable of handling this situation. The latest Catalyst software implements the Uni-Directional Link Detection (UDLD) feature that helps to detect this dangerous condition. This works on point-to-point links between Cisco devices only.

Additional Sources

For further information, including step-by-step configuration materials and full command examples for both the IP MLS-RP and the MLS-SE, you are highly encouraged to view the following (log in to maximize the amount of material that you can view):

- The Technical Assistance Center (TAC) web site, on Cisco Connection Online (CCO), at www.cisco.com/tac
- The MLS Technology Pages, off the CCO TAC web site, at www.cisco.com/cgi-bin/Support/PSP/psp_view.pl?p=Internetworking:Multi-layer_Switching
- For MLS supported and unsupported network topologies, at www.cisco.com/univercd/cc/td/doc/product/lan/cat5000/rel_5_2/layer3/mls.htm#xtocid1101958
- The Layer 3 Switching Software Configuration Guide, at www.cisco.com/univercd/cc/td/doc/product/lan/cat5000/rel_5_2/layer3/index.htm
- The Catalyst 5xxx release notes, at www.cisco.com/univercd/cc/td/doc/product/lan/cat5000/c5krn/index.htm
- The Catalyst 6xxx release notes, at www.cisco.com/univercd/cc/td/doc/product/lan/cat6000/relnotes/index.htm