CHAPTER 20

Troubleshooting Transparent Bridging Environments

Transparent bridges were first developed at Digital Equipment Corporation (Digital) in the early 1980s and are now very popular in Ethernet/IEEE 802.3 networks.

- This chapter first defines a transparent bridge as a learning bridge that implements a Spanning Tree. A deeper description of the Spanning-Tree Protocol is included.
- Cisco devices implementing transparent bridging previously were split in two categories: routers running IOS, and the Catalyst range of switches, running specific software. This is not the case anymore, however, because several Catalyst products are now based on the IOS. This chapter introduces the different bridging techniques available on IOS devices. For Catalyst software specific configuration and troubleshooting, refer to Chapter 23, “Troubleshooting ATM LAN Environments.”
- Finally, this chapter introduces some troubleshooting steps classified by symptoms of potential problems that typically occur in network implementations featuring transparent bridging.

Transparent Bridging Technology Basics

Transparent bridges are so named because their presence and operation are transparent to network hosts. When transparent bridges are powered on, they learn the network’s topology by analyzing the source address of incoming frames from all attached networks. For example, if a bridge sees a frame arrive on line 1 from Host A, the bridge concludes that Host A can be reached through the network connected to line 1. Through this process, transparent bridges build a table such as the one in Table 20-1.

Table 20-1 A Transparent Bridging Table

Host Address	Network Number
0000.0000.0001	1
0000.b07e.ee0e	7
. . .	
0050.50e1.9b80	4
0060.b0d9.2e3d	2
0000.0c8c.7088	1
. . .	

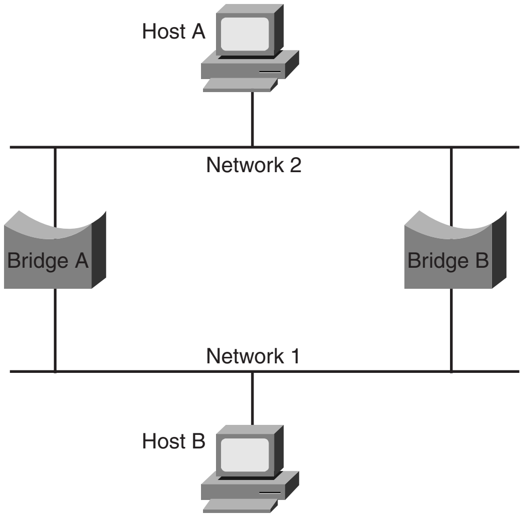
The bridge uses its table as the basis for traffic forwarding. When a frame is received on one of the bridge interfaces, the bridge looks up the frame's destination address in its internal table. If the table contains an association between the destination address and any of the bridge's ports aside from the one on which the frame was received, the frame is forwarded out the indicated port. If no association is found, the frame is flooded to all ports except the inbound port. Broadcasts and multicasts are also flooded in this way.

Transparent bridges successfully isolate intrasegment traffic, thereby reducing the traffic seen on each individual segment. This usually improves network response times as seen by the user. The extent to which traffic is reduced and response times are improved depends on the volume of intersegment traffic relative to the total traffic, as well as the volume of broadcast and multicast traffic.

Bridging Loops

Without a bridge-to-bridge protocol, the transparent bridge algorithm fails when there are multiple paths of bridges and local-area networks (LANs) between any two LANs in the internetwork. Figure 20-1 illustrates such a bridging loop.

Figure 20-1 Inaccurate Forwarding and Learning in Transparent Bridging Environments



Suppose that Host A sends a frame to Host B. Both bridges receive the frame and correctly conclude that Host A is on Network 2. Unfortunately, after Host B receives two copies of Host A's frame, both bridges again receive the frame on their Network 1 interfaces because all hosts receive all messages on broadcast LANs. In some cases, the bridges then change their internal tables to indicate that Host A is on Network 1. If this is the case, when Host B replies to Host A's frame, both bridges receive and subsequently drop the replies because their tables indicate that the destination (Host A) is on the same network segment as the frame's source.

In addition to basic connectivity problems such as the one just described, the proliferation of broadcast messages in networks with loops represents a potentially serious network problem. Referring again to Figure 20-1, assume that Host A's initial frame is a broadcast. Both bridges will forward the frames endlessly, using all available network bandwidth, and blocking the transmission of other packets on both segments.

A topology with loops such as that shown in Figure 20-2 can be useful as well as potentially harmful. A loop implies the existence of multiple paths through the internetwork. A network with multiple paths from source to destination can increase overall network fault tolerance through improved topological flexibility.

The Spanning-Tree Algorithm

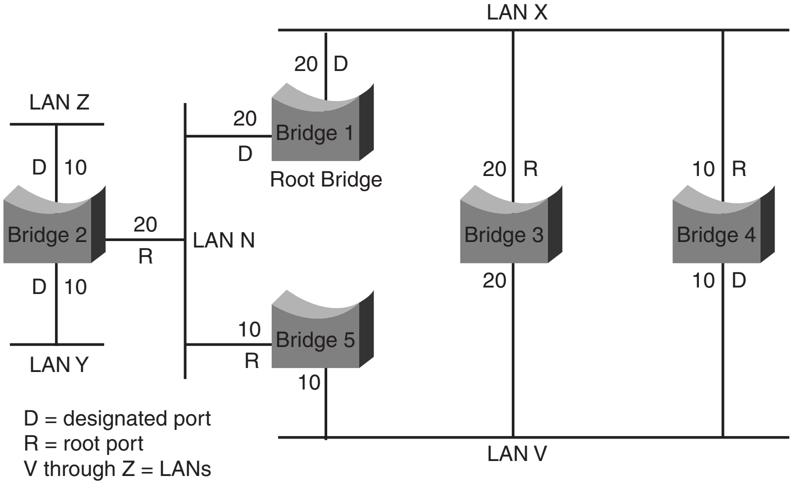
The Spanning-Tree Algorithm (STA) was developed by Digital, a key Ethernet vendor, to preserve the benefits of loops while eliminating their problems. Digital's algorithm was subsequently revised by the IEEE 802 committee and was published in the IEEE 802.1d specification. The Digital algorithm and the IEEE 802.1d algorithm are not the same, nor are they compatible.

The STA designates a loop-free subset of the network's topology by placing those bridge ports that, if active, would create loops into a standby (blocking) condition. Blocking bridge ports can be activated in the event of primary link failure, providing a new path through the internetwork.

The STA uses a conclusion from graph theory as a basis for constructing a loop-free subset of the network's topology. Graph theory states the following: "For any connected graph consisting of nodes and edges connecting pairs of nodes, there is a Spanning Tree of edges that maintains the connectivity of the graph but contains no loops."

Figure 20-2 illustrates how the STA eliminates loops. The STA calls for each bridge to be assigned a unique identifier. Typically, this identifier is one of the bridge's Media Access Control (MAC) addresses, plus a priority. Each port in every bridge is also assigned a unique (within that bridge) identifier (typically, its own MAC address). Finally, each bridge port is associated with a path cost. The path cost represents the cost of transmitting a frame onto a LAN through that port. In Figure 20-3, path costs are noted on the lines emanating from each bridge. Path costs are usually default values, but network administrators can assign them manually.

Figure 20-2 A Transparent Bridge Network Before STA Is Run



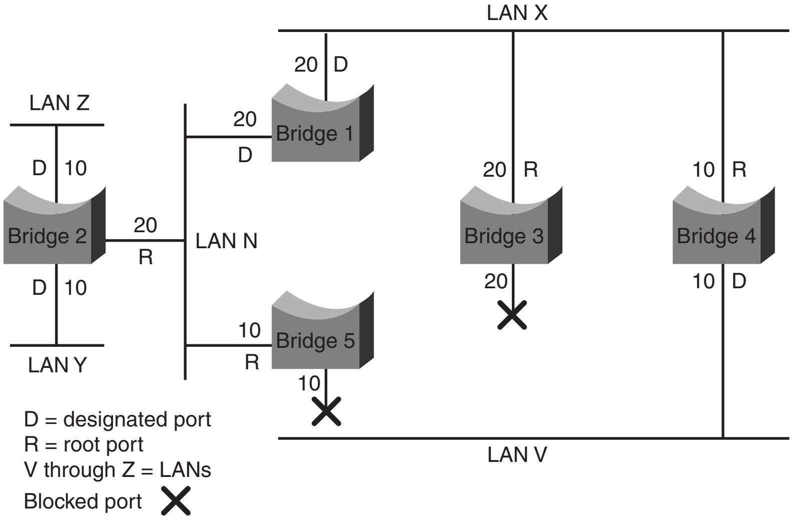
The first activity in Spanning Tree computation is the selection of the root bridge, which is the bridge with the lowest-value bridge identifier. In Figure 20-2, the root bridge is Bridge 1. Next, the root port on all other bridges is determined. A bridge's root port is the port through which the root bridge can be reached with the least aggregate path cost. The value of the least aggregate path cost to the root is called the *root path cost*.

Finally, designated bridges and their designated ports are determined. A designated bridge is the bridge on each LAN that provides the minimum root path cost. A LAN's designated bridge is the only bridge allowed to forward frames to and from the LAN for which it is the designated bridge. A LAN's designated port is the port that connects it to the designated bridge.

In some cases, two or more bridges can have the same root path cost. For example, in Figure 20-3, Bridges 4 and 5 can both reach Bridge 1 (the root bridge) with a path cost of 10. In this case, the bridge identifiers are used again, this time to determine the designated bridges. Bridge 4's LAN V port is selected over Bridge 5's LAN V port.

Using this process, all but one of the bridges directly connected to each LAN are eliminated, thereby removing all two-LAN loops. The STA also eliminates loops involving more than two LANs, while still preserving connectivity. Figure 20-3 shows the results of applying the STA to the network shown in Figure 20-2. Figure 20-3 shows the tree topology more clearly. Comparing this figure to the pre-Spanning Tree figure shows that the STA has placed both Bridge 3 and Bridge 5's ports to LAN V in standby mode.

Figure 20-3 A Transparent Bridge Network After STA Is Run



The Spanning Tree calculation occurs when the bridge is powered up and whenever a topology change is detected. The calculation requires communication between the Spanning Tree bridges, which is accomplished through configuration messages (sometimes called *bridge protocol data units*, or BPDUs). Configuration messages contain information identifying the bridge that is presumed to be the root (root identifier) and the distance from the sending bridge to the root bridge (root path cost). Configuration messages also contain the bridge and port identifier of the sending bridge and the age of information contained in the configuration message.

Bridges exchange configuration messages at regular intervals (typically 1 to 4 seconds). If a bridge fails (causing a topology change), neighboring bridges soon detect the lack of configuration messages and initiate a Spanning Tree recalculation.

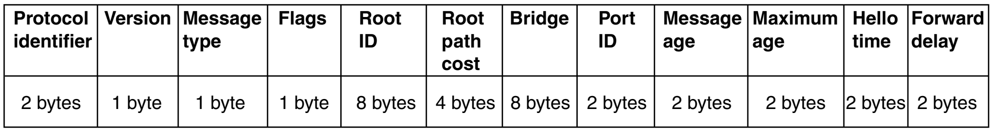
All transparent bridge topology decisions are made locally. Configuration messages are exchanged between neighboring bridges. There is no central authority on network topology or administration.

Frame Format

Transparent bridges exchange configuration messages and topology change messages. Configuration messages are sent between bridges to establish a network topology. Topology change messages are sent after a topology change has been detected to indicate that the STA should be rerun.

The IEEE 802.1d configuration message format is shown in Figure 20-4.

Figure 20-4 The Transparent Bridge Configuration



Protocol identifier	Version	Message type	Flags	Root ID	Root path cost	Bridge	Port ID	Message age	Maximum age	Hello time	Forward delay
2 bytes	1 byte	1 byte	1 byte	8 bytes	4 bytes	8 bytes	2 bytes	2 bytes	2 bytes	2 bytes	2 bytes

Message Format

The fields of the transparent bridge configuration message are as follows:

- **Protocol identifier**—Contains the value 0.
- **Version**—Contains the value 0.
- **Message type**—Contains the value 0.
- **Flag**—A 1-byte field, of which only the first 2 bits are used. The topology change (TC) bit signals a topology change. The topology change acknowledgment (TCA) bit is set to acknowledge receipt of a configuration message with the TC bit set.
- **Root ID**—Identifies the root bridge by listing its 2-byte priority followed by its 6-byte ID.
- **Root path cost**—Contains the cost of the path from the bridge sending the configuration message to the root bridge.
- **Bridge ID**—Identifies the priority and ID of the bridge sending the message.
- **Port ID**—Identifies the port from which the configuration message was sent. This field allows loops created by multiply attached bridges to be detected and dealt with.
- **Message age**—Specifies the amount of time since the root sent the configuration message on which the current configuration message is based.
- **Maximum age**—Indicates when the current configuration message should be deleted.
- **Hello time**—Provides the time period between root bridge configuration messages.
- **Forward delay**—Provides the length of time that bridges should wait before transitioning to a new state after a topology change. If a bridge transitions too soon, not all network links may be ready to change their state, and loops can result.

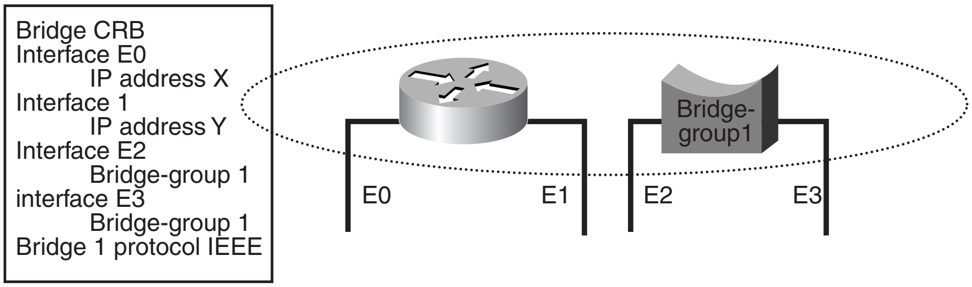
Topological change messages consist of only 4 bytes. They include a Protocol Identifier field, which contains the value 0; a Version field, which contains the value 0; and a Message Type field, which contains the value 128.

Different IOS Bridging Techniques

Cisco routers have three different ways of implementing bridging:

- **Default behavior**—Prior to IRB and the CRB features (see later), you could bridge or route a protocol only on a platform basis. That is, if the **ip route** command was used, for example, then IP routing was done on all interfaces. In this situation, IP could not be bridged on any of the router's interfaces.
- **Concurrent routing and bridging (CRB)**—With CRB, you can determine whether to bridge or route a protocol on an interface basis. That is, you can route a given protocol on some interfaces and bridge the same protocol on bridge group interfaces within the same router. The router can then be both a router and a bridge for a given protocol, but there cannot be any kind of communication between routing-defined interfaces and bridge group interfaces. For a given protocol, the router can be logically considered as different independent devices: one router and one or more bridges, as shown in Figure 20-5.

Figure 20-5 The Router Can Be Logically Considered as Different Independent Devices

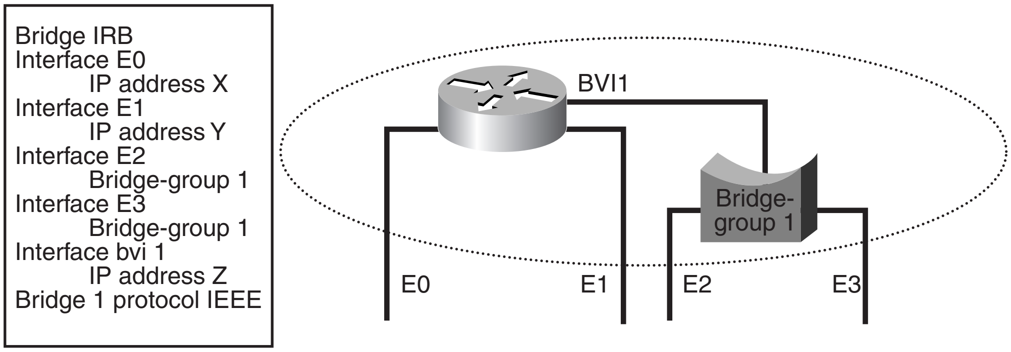


```
Bridge CRB
Interface E0
        IP address X
Interface 1
        IP address Y
Interface E2
        Bridge-group 1
interface E3
        Bridge-group 1
Bridge 1 protocol IEEE
```

In this configuration for the IP protocol, the Cisco device is acting like a router for interfaces E0 and E1 and is acting like a bridge for interfaces E2 and E3. Note that there is no communication possible between the two functions (a host connected on E0 would never be able to reach a host connected on E2 through the router with this configuration).

- **Integrated routing and bridging (IRB)**—IRB provides the capability to route between a bridge group and a routed interface using a concept called Bridge-Group Virtual Interface (BVI). Because bridging is in the data link layer and routing is in the network layer, they have different protocol configuration models. With IP, for example, bridge-group interfaces belong to the same network and have a collective IP network address, while each routed interface represents a distinct network and has its own IP network address. The concept of Bridge-Group Virtual Interface was created to enable these interfaces to exchange packets for a given protocol. Conceptually, the Cisco router looks like a router connected to one or more bridge groups, as shown in Figure 20-6.

Figure 20-6 The Bridge-Group Virtual Interface Brings Routing to Bridge Group 1

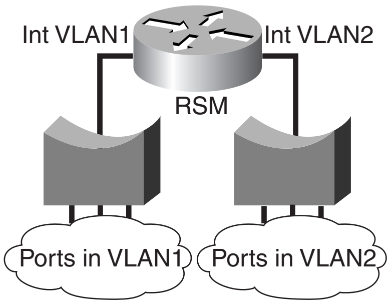


```
Bridge IRB
Interface E0
        IP address X
Interface E1
        IP address Y
Interface E2
        Bridge-group 1
Interface E3
        Bridge-group 1
Interface bvi 1
        IP address Z
Bridge 1 protocol IEEE
```

The Bridge-Group Virtual Interface brings routing to bridge-group1. One can assign an IP address to the whole bridge group and routed communication is now possible between a host connected to E0 and a host connected to E2, for instance.

The BVI is a virtual interface within the router that acts like a normal routed interface that represents the corresponding bridge group to routed interfaces within the router. The interface number of the BVI is the number of the bridge group that this virtual interface represents. The number is the link between this BVI and the bridge group. The sample principle applies to the Route Switch Module in a Catalyst Switch, as shown in Figure 20-7.

Figure 20-7 Route Switch Module in a Catalyst Switch



The IRB concept is also used (but hidden) on the Catalyst Route Switch Module (RSM).
The VLAN interfaces are virtual interfaces connecting different bridge groups (the VLANs).

Troubleshooting Transparent Bridging

This section presents troubleshooting information for connectivity problems in transparent bridging internetworks. It describes specific transparent bridging symptoms, the problems that are likely to cause each symptom, and the solutions to those problems.

Note Problems associated with source-route bridging (SRB), translational bridging, and source-route transparent (SRT) bridging are addressed in Chapter 10, "Troubleshooting IBM."

To do an efficient troubleshooting of your bridged network, you should get a basic knowledge of its design, especially when a Spanning Tree is involved.

Try to have the following available:

- The topology map of the bridged network
- The location of the root bridge
- The location of the redundant link (and blocked ports)

When you are troubleshooting connectivity issues, try to narrow down the problem to a minimum number of hosts (ideally only a client and a server).

The following sections describe the most common network problems in transparent bridged networks:

- Transparent Bridging: No Connectivity
- Transparent Bridging: Unstable Spanning Tree
- Transparent Bridging: Sessions Terminate Unexpectedly
- Transparent Bridging: Looping and Broadcast Storms Occur

Transparent Bridging: No Connectivity

Symptom: Client cannot connect to hosts across a transparently bridged network.

Table 20-2 outlines the problems that might cause this symptom and describes solutions to those problems.

Table 20-2 Transparent Bridging: No Connectivity

Possible Causes	Suggested Actions
Hardware or media problem has occurred	1. Use the **show bridge** exec command to see whether there is a connectivity problem. If there is, the output will not show any MAC[1] addresses in the bridging table. 2. Use the **show interfaces** exec command to determine whether the interface and line protocol are up. 3. If the interface is down, troubleshoot the hardware or the media. Refer to Chapter 3, "Troubleshooting Hardware and Booting Problems." 4. If the line protocol is down, check the physical connection between the interface and the network. Make sure that the connection is secure and that cables are not damaged. If the line protocol is up but input and output packet counters are not incrementing, check the media and host connectivity. Refer to the media troubleshooting chapter that covers the media type used in your network.
Host is down	1. Use the **show bridge** exec command on bridges to make sure that the bridging table includes the MAC addresses of attached end nodes. The bridging table comprises the source and destination MAC addresses of hosts and is populated when packets from a source or destination pass through the bridge. 2. If any expected end nodes are missing, check the status of the nodes to verify that they are connected and properly configured. 3. Reinitialize or reconfigure end nodes as necessary, and re-examine the bridging table using the **show bridge** command.
Bridging path is broken	1. Identify the path that packets should take between end nodes. If there is a router on this path, split the troubleshooting in two parts: node1-router and router-node2. 2. Connect to each bridge on the path, and check the status of the ports that are used on the path between end nodes, just as described in the previous discussion on a hardware or media problem. 3. Using the **show bridge** command, check that the MAC address of the nodes are learned on the correct ports. If not, there may be instability on your Spanning Tree topology. (See Table 20-2.) 4. Check the state of the ports using the **show span** command. If the ports that should transmit traffic between the end nodes are not in the forwarding state, the topology of your tree may have changed unexpectedly. (See Table 20-3.)

Table 20-2 Transparent Bridging: No Connectivity (continued)

Possible Causes	Suggested Actions
Bridging filters are misconfigured	1. Use the **show running-config** privileged exec command to determine whether bridge filters are configured. 2. Disable bridge filters on suspect interfaces, and determine whether connectivity returns. 3. If connectivity does not return, the filter is not the problem. If connectivity is restored after removing filters, one or more bad filters are causing the connectivity problem. 4. If multiple filters or filters using access lists with multiple statements exist, apply each filter individually to identify the problem filter. Check the configuration for input and output LSAP[2] and TYPE filters, which can be used simultaneously to block different protocols. For example, LSAP (F0F0) can be used to block NetBIOS, and TYPE (6004) can be used to block local-area transport. 5. Modify any filters or access lists that are blocking traffic. Continue testing filters until all filters are enabled and connections still work.
Input and output queues are full	Excessive multicast or broadcast traffic can cause input and output queues to overflow, resulting in dropped packets. 1. Use the **show interfaces** command to look for input and output drops. Drops suggest excessive traffic over the media. If the current number of packets on the input queue is consistently at or greater than 80 percent of the current size of the input queue, the size of the input queue may require tuning to accommodate the incoming packet rate. Even if the current number of packets on the input queue never seems to approach the size of the input queue, bursts of packets may still be overflowing the queue. 2. Reduce broadcast and multicast traffic on attached networks by implementing bridging filters, or segment the network using more internetworking devices. 3. If the connection is a serial link, increase bandwidth, apply priority queuing, increase the hold queue size, or modify the system buffer size. For more information, refer to Chapter 15, “Troubleshooting Serial Lines.”

[1] MAC = Media Access Control

[2] LSAP = link services access point

Transparent Bridging: Unstable Spanning Tree

Symptom: Transient loss of connectivity between hosts. Several hosts are affected at the same time.

Table 20-3 outlines the problems that might cause this symptom and describes solutions to those problems.

Table 20-3 Transparent Bridging: Unstable Spanning Tree

Possible Causes	Suggested Actions
Link flapping	1. Use **show span** command to check whether the number of topology changes is steadily increasing. 2. If so, check the link between your bridges, using the **show interface** command. If you cannot find a link flapping between two bridges this way, use the **debug spantree event** privileged exec command on your bridges. This will log all changes related to Spanning Tree; in a stable topology, there should not be any. The only links to track are the ones connecting bridging devices—a transition on a link to an end station should have no impact on the network. **Caution**: Exercise caution when using the **debug spantree event** command. Because debugging output is assigned high priority in the CPU process, it can render the system unusable. For this reason, use **debug** commands only to troubleshoot specific problems or during troubleshooting sessions with Cisco technical support staff. Moreover, it is best to use **debug** commands during periods of lower network traffic and fewer users. Debugging during these periods decreases the likelihood that increased **debug** command processing overhead will affect system use.

Table 20-3 Transparent Bridging: Unstable Spanning Tree (continued)

Possible Causes	Suggested Actions
Root bridge that keeps changing/multiple bridges that claim to be the root	1. Check the consistency of the root bridge information all over the bridged network using the **show span** commands on the different bridges. 2. If several bridges are claiming to be the root, check that you are running the same Spanning-Tree Protocol on every bridge (see the entry "Spanning-Tree Algorithm mismatch," in Table 20-5). 3. Use the **bridge <group> priority <number>** command on root bridge to force the desired bridge to become the root. The lower the priority, the more likely the bridge is to become the root. 4. Check the diameter of your network. With standard Spanning Tree settings, there should never be more that seven bridging hops between two hosts.
Hellos not being exchanged	1. Check whether bridges are communicating with one another. Use a network analyzer or the **debug spantree tree** privileged exec command to see whether Spanning Tree hello frames are being exchanged. **Caution**: Exercise caution when using the **debug spantree tree** command. Because debugging output is assigned high priority in the CPU process, it can render the system unusable. For this reason, use **debug** commands only to troubleshoot specific problems or during troubleshooting sessions with Cisco technical support staff. Moreover, it is best to use **debug** commands during periods of lower network traffic and fewer users. Debugging during these periods decreases the likelihood that increased **debug** command processing overhead will affect system use. 2. If hellos are not being exchanged, check the physical connections and software configuration on bridges.

Transparent Bridging: Sessions Terminate Unexpectedly

Symptom: Connections in a transparently bridged environment are successfully established, but sessions sometimes terminate abruptly.

Table 20-4 outlines the problems that might cause this symptom and describes solutions to those problems.

Table 20-4 Transparent Bridging: Sessions Terminate Unexpectedly

Possible Causes	Suggested Actions
Excessive retransmissions	**1.** Use a network analyzer to look for host retransmissions. **2.** If you see retransmissions on slow serial lines, increase the transmission timers on the host. For information on configuring your hosts, refer to the vendor documentation. For information on troubleshooting serial lines, refer to Chapter 15. If you see retransmissions on high-speed LAN media, check for packets sent and received in order, or dropped by any intermediate device such as a bridge or a switch. Troubleshoot the LAN media as appropriate. For more information, refer to the media troubleshooting chapter that covers the media type used in your network. **3.** Use a network analyzer to determine whether the number of retransmissions subsides.
Excessive delay over serial link	Increase bandwidth, apply priority queuing, increase the hold queue size, or modify the system buffer size. For more information, refer to Chapter 15.

Transparent Bridging: Looping and Broadcast Storms Occur

Symptom: Packet looping and broadcast storms occur in transparent bridging environments. End stations are forced into excessive retransmission, causing sessions to time out or drop.

Note Packet loops are typically caused by network design problems or hardware issues.

Table 20-4 outlines the problems that might cause this symptom and describes solutions to those problems.

Bridging loops are the worst-case scenario in a bridged network because they will potentially impact every user. In case of emergency, the best way of recovering quickly connectivity is to disable manually all the interfaces providing a redundant path in the network. Unfortunately, the cause of the bridging loop will be very difficult to identify afterward, if you do so. Try the actions recommended in Table 20-5 first, if possible.

Table 20-5 Transparent Bridging: Looping and Broadcast Storms Occur

Possible Causes	Suggested Actions
No Spanning Tree implemented	1. Examine a topology map of your internetwork to check for possible loops. 2. Eliminate any loops that exist, or make sure that the appropriate links are in backup mode. 3. If broadcast storms and packet loops persist, use the **show interfaces** exec command to obtain input and output packet count statistics. If these counters increment at an abnormally high rate (with respect to your normal traffic loads), a loop is probably still present in the network. 4. Implement a Spanning-Tree Algorithm to prevent loops.
Spanning-Tree Algorithm mismatch	1. Use the **show span** exec command on each bridge to determine which Spanning-Tree Algorithm is being used. 2. Make sure that all bridges are running the same Spanning-Tree Algorithm (either DEC or IEEE).[1] Use of both DEC and IEEE Spanning-Tree Algorithms may be needed in the network for some very specific configuration (generally involving IRB). If the mismatch in the Spanning-Tree Protocol is not intended, you should reconfigure bridges as appropriate so that all bridges use the same Spanning-Tree Algorithm. **Note:** The DEC and IEEE Spanning-Tree Algorithms are incompatible.

Table 20-5 Transparent Bridging: Looping and Broadcast Storms Occur (continued)

Possible Causes	Suggested Actions
Multiple bridging domains incorrectly configured	1. Use the **show span** exec command on bridges to ensure that all domain group numbers match for given bridging domains. 2. If multiple domain groups are configured for the bridge, ensure that all domain specifications are assigned correctly. Use the **bridge <group> domain <domain-number>** global configuration command to make any necessary changes. 3. Make sure that no loops exist between bridging domains. An interdomain bridging environment does not provide loop prevention based on Spanning Tree. Each domain has its own Spanning Tree, which is independent of the Spanning Tree in another domain.
Link error (unidirectional link), duplex mismatch, high level of error on a port	Loops occur when a port that should block moves to the forwarding state. A port needs to receive BPDUs from a neighbor bridge to stay in a blocking state. Any error that lead to BPDUs being lost can then be the cause of a bridging loop. 1. Identify blocking ports from your network diagram. 2. Check the status of the ports that should be blocking in your bridged network, using the **show interface** and **show bridge** exec commands. 3. If you find a supposedly blocked port that is currently forwarding (or about to forward, in learning or listening state), you have found the real source of the problem. Check where this port is receiving BPDUs. If not, there is probably an issue on the link connected to this port (check then link errors, duplex setting, and so on). If the port is still receiving BPDUs, go to the bridge that you expect to be designated for this LAN. From there, check all the links on the path toward the root. You will find an issue on one of these links (provided that your initial network diagram was correct).

[1] IEEE = Institute of Electrical and Electronic Engineers

Before Calling Cisco Systems' TAC Team

Try to collect as much information as you can on the topology of your network, when stable.

The minimal data to collect is this:

- The physical topology of the network
- The expected location of the root bridge (and the backup root bridge)
- The location of blocked ports

Additional Sources

Books

- Clark, K., and K. Hamilton. *Cisco LAN Switching.* Indianapolis: Cisco Press, 1999.
- Perlman, Radia. *Interconnections, Bridges and Routers*. Boston: Addison-Wesley, 1998.

URLs

Transparent bridging documentation:
www.cisco.com/univercd/cc/td/doc/product/software/ios120/12cgcr/ibm_c/bcprt1/bctb.htm

Technology support pages on CCO: www.cisco.com/tac (Look for the LAN part in the Technology Home Page section.)