

Troubleshooting Transparent Bridging Environments

Document ID: 10543

This information from the Internetwork Troubleshooting Guide was first posted on CCO here. As a service to our customers, selected chapters have been updated with the most current and accurate information. The complete update to the Internetwork Troubleshooting Guide will soon be available in print and online.

Objectives

Transparent Bridging Technology Basics

- Bridging Loops
- The Spanning Tree Algorithm
- Frame Format
- Message Fields
- Different IOS Bridging Techniques

Troubleshooting Transparent Bridging

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

Before You Call Cisco Systems TAC Team

Additional Sources

Related Information

Objectives

Transparent bridges were first developed at Digital Equipment Corporation (DEC) 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 the spanning tree protocol. An in-depth description of the spanning tree protocol is included.
- Cisco devices that implement transparent bridges used to be split into two categories: routers that run Cisco IOS® software and the Catalyst range of switches that run specific software. This is no longer the case. Several Catalyst products are now based on the IOS. This chapter introduces the different bridging techniques that are available on IOS devices. For Catalyst software-specific configuration and troubleshooting, refer to the LAN Switching chapter.
- Finally, we introduce some troubleshooting procedures that are classified by the symptoms of potential problems that typically occur in transparent bridging networks.

Transparent Bridging Technology Basics

Transparent bridges derive their name from the fact that their presence and operation are transparent to network hosts. When transparent bridges are powered on, they learn the topology of the network by analysis of the source address of inbound frames from all attached networks. If, for example, 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 an internal bridging table such as the one in Table

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 bridging table as the basis for traffic-forwarding. When a frame is received on one of the bridge interfaces, the bridge looks up the destination address of the frame in its internal table. If the table is mapped between the destination address and any of the ports of the bridge (aside from the one on which the frame was received), the frame is forwarded to the specified port. If no map is found, the frame is flooded to all outbound ports. Broadcasts and multicasts are also flooded in this way.

Transparent bridges successfully isolate intra-segment traffic and reduce the traffic seen on each individual segment. This usually improves network response times. The extent to which traffic is reduced and response times are improved depends on the volume of intersegment traffic (relative to total traffic) as well as on 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.

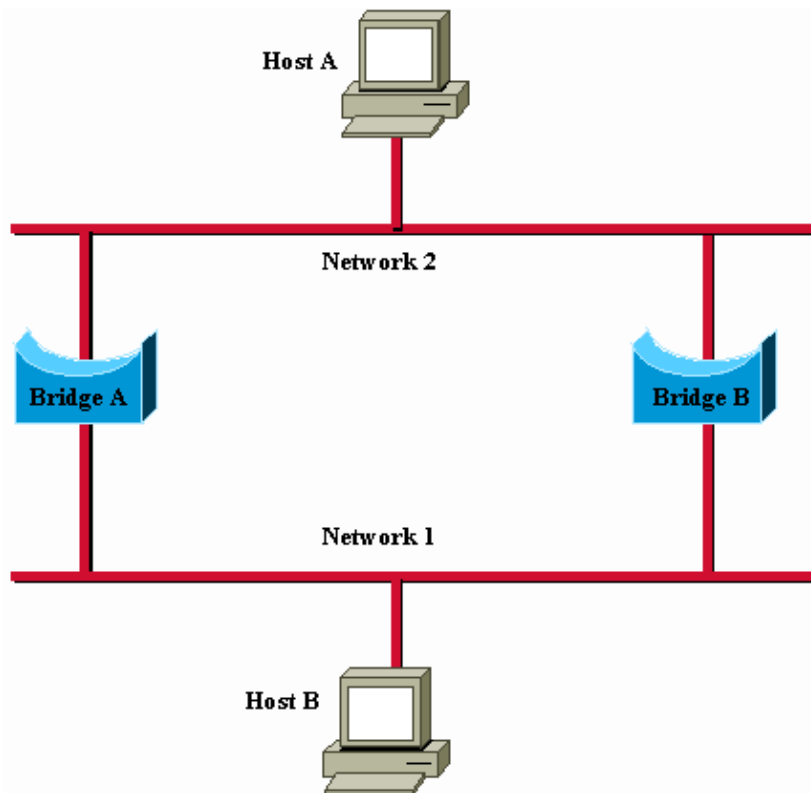


Figure 20–1: Inaccurate Forwarding and Learning in Transparent Bridging Environments

Suppose 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 the frame of Host A, 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 will then change their internal tables to indicate that Host A is on Network 1. If this is the case, when Host B replies to the frame of Host A, 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 source of the frame.

In addition to basic connectivity problems, such as the one described, the proliferation of broadcast messages on networks with loops represents a potentially serious network problem. In reference to Figure 20–1, assume that the initial frame of Host A is a broadcast. Both bridges forward the frames endlessly, use all available network bandwidth, and block the transmission of other packets on both segments.

A topology with loops such as that shown in Figure 20–1 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 has what is called improved topological flexibility which increases overall network fault tolerance.

The Spanning Tree Algorithm

The spanning tree algorithm (STA) was developed by DEC, a key Ethernet vendor, to preserve the benefits of loops yet eliminate their problems. The DEC algorithm was subsequently revised by the IEEE 802 committee and published in the IEEE 802.1d specification. The DEC algorithm and the IEEE 802.1d algorithm are not the same, nor are they compatible.

The STA designates a loop-free subset of the topology of the network by the placement of those bridge ports, so, if active, it can create loops into a standby (blocking) condition. Bridge port blocking can be activated in the event of primary link failure, which provides a new path through the internetwork.

The STA uses a conclusion from graph theory as a basis for the construction of a loop-free subset of the topology of the network. Graph theory states: "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 Media Access Control (MAC) addresses of the bridge plus a priority indication. 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 the transmittal of a frame onto a LAN through that port. In Figure 20–2, path costs are noted on the lines that emanate from each bridge. Path costs are usually default values, but they can be assigned manually by network administrators.

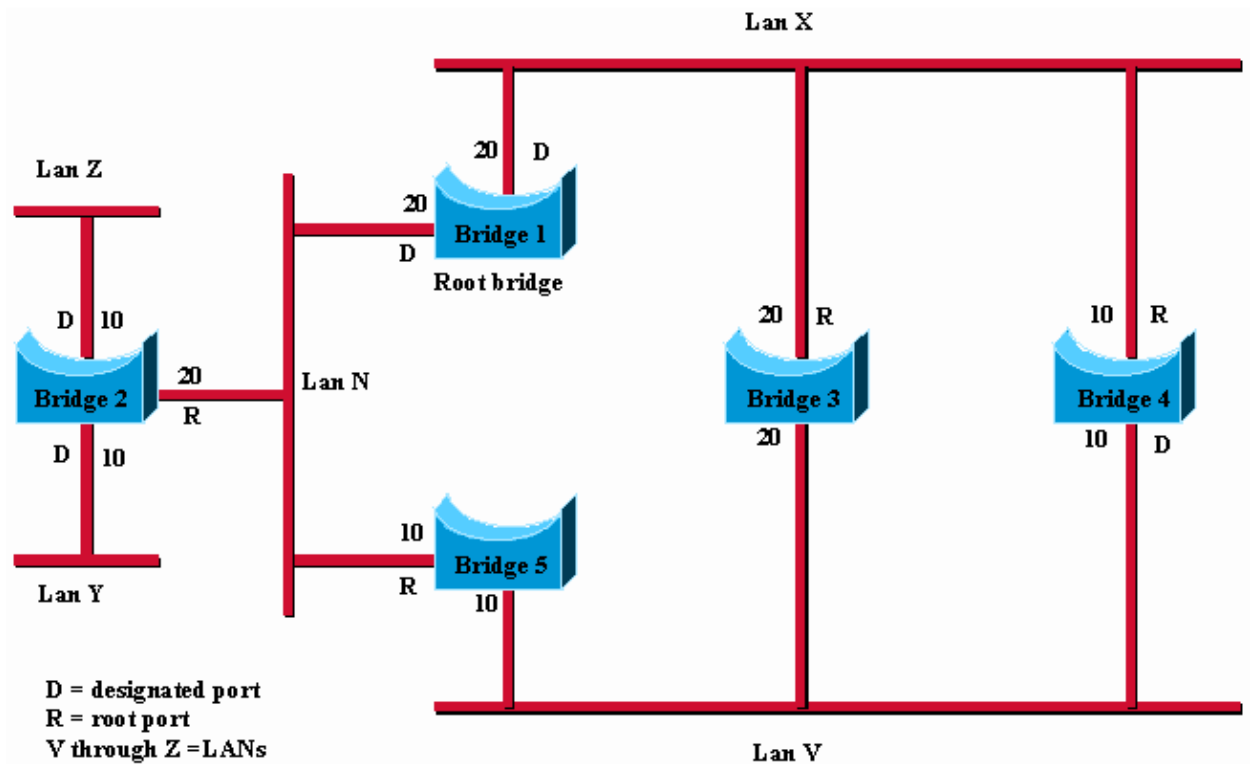


Figure 20–2: Transparent Bridge Network (Before STA)

The first activity in a spanning tree computation is the selection of the root bridge, which is the bridge with the lowest bridge identifier value. In Figure 20–2, the root bridge is Bridge 1. Next, the root port on all other bridges is determined. A root port of a bridge 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 designated bridge of a LAN is the only bridge allowed to forward frames to and from the LAN for which it is the designated bridge. A designated port of a LAN 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 20–2, 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. The LAN V port of Bridge 4 is selected over the LAN V port of Bridge 5.

With this process, all but one of the bridges directly connected to each LAN are eliminated, which removes all two-LAN loops. The STA also eliminates loops that involve more than two LANs, yet still preserve connectivity. Figure 20–3 shows the results from the application of the STA to the network shown in Figure 20–2. Figure 20–2 shows the tree topology more clearly. A comparison of this figure to Figure 20–3 shows that the STA has placed the ports to LAN V in both Bridge 3 and Bridge 5 in standby mode.

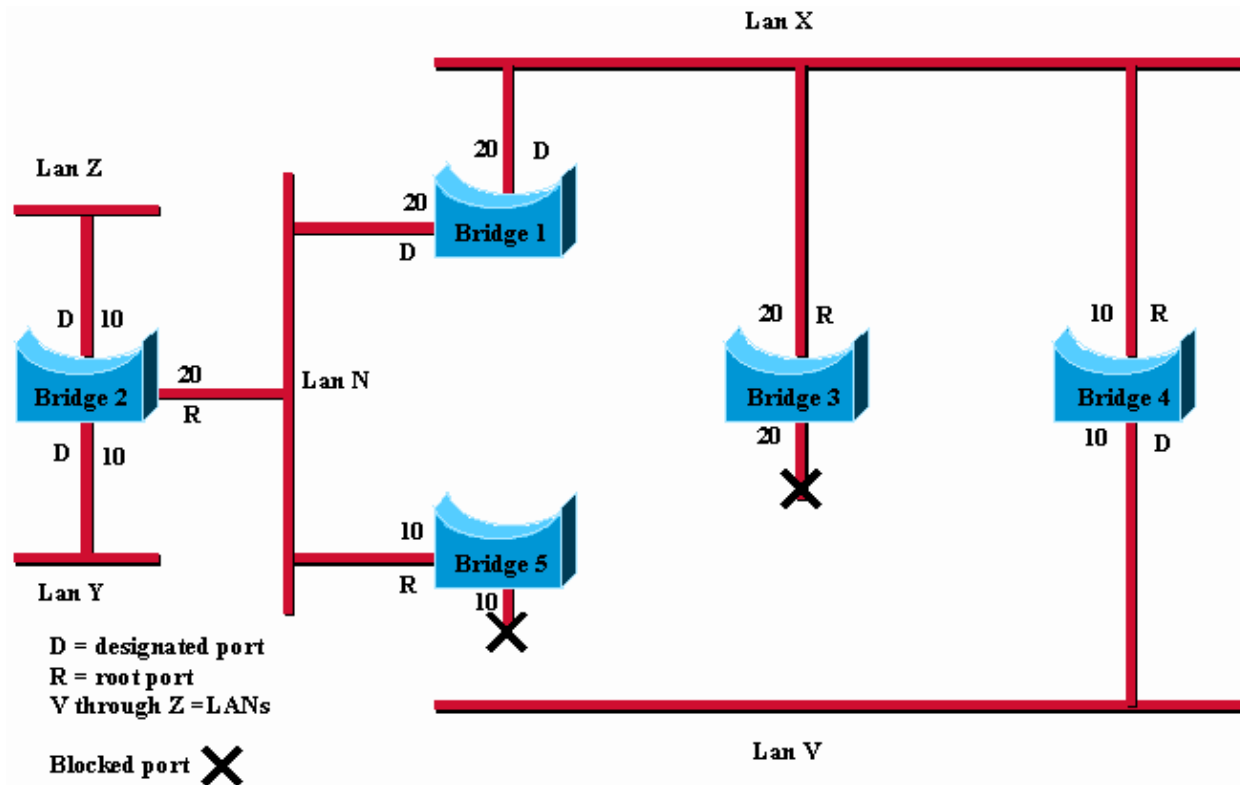


Figure 20–3: Transparent Bridge Network (After STA)

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 that identify 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 one to four seconds). If a bridge fails (which causes a topology change), nearby 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 nearby 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 must be rerun.

Table 20–2 shows the IEEE 802.1d configuration message format.

Table 20–2: Transparent Bridge Configuration

Protocol identifier	Version	Message type	Flags	Root id	Root path cost	Bridge id	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 Fields

Transparent bridge configuration messages consist of 35 bytes. These are the message fields:

- Protocol identifier: Contains the value 0.
- Version: Contains the value 0.
- Message type: Contains the value 0.
- Flag: A one–byte field, of which only the first two 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 and lists its 2–byte priority followed by its six–byte ID.
- Root path cost: Contains the cost of the path from the bridge that sends the configuration message to the root bridge.
- Bridge ID: Identifies the priority and ID of the bridge that sends the message.
- Port ID: Identifies the port from which the configuration message was sent. This field allows loops created by multiple attached bridges to be detected and dealt with.
- Message age: Specifies the elapsed time since the root sent the configuration message on which the current configuration message is based.
- Maximum age: Indicates when the current configuration message must be deleted.
- Hello time: Provides the time period between root bridge configuration messages.
- Forward delay: Provides the amount of time bridges must wait before a transition to a new state after a topology change. If a bridge transitions too soon, not all network links can be ready to change their state, and loops can result.

The topology–change message format is similar to that of the transparent bridge configuration message, except it consists only of the first four bytes. These are the message fields:

- Protocol identifier: Contains the value 0.
- Version: Contains the value 0.
- Message type: Contains the value 128.

Different IOS Bridging Techniques

Cisco routers have three different ways to implement bridging: Default Behavior, Concurrent Routing and Bridging (CRB), and Integrated Routing and Bridging (IRB).

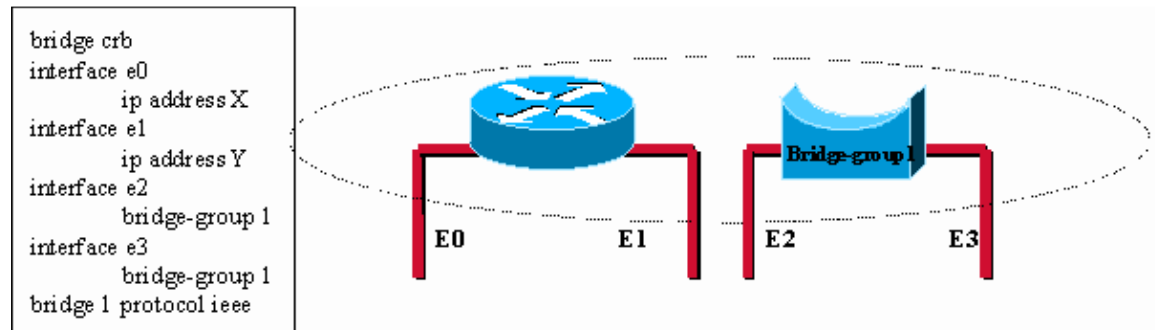
Default Behavior

Before IRB and CRB features were available, you were only able to bridge or route a protocol on a platform basis. That is, if the **ip route** command was used, for example, IP routing was done on all interfaces. In this situation, IP could not be bridged on any of the interfaces of the router.

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.

This example illustrates that, for a given protocol, a single router can logically act as separate, independent devices: one router and one or more bridges:



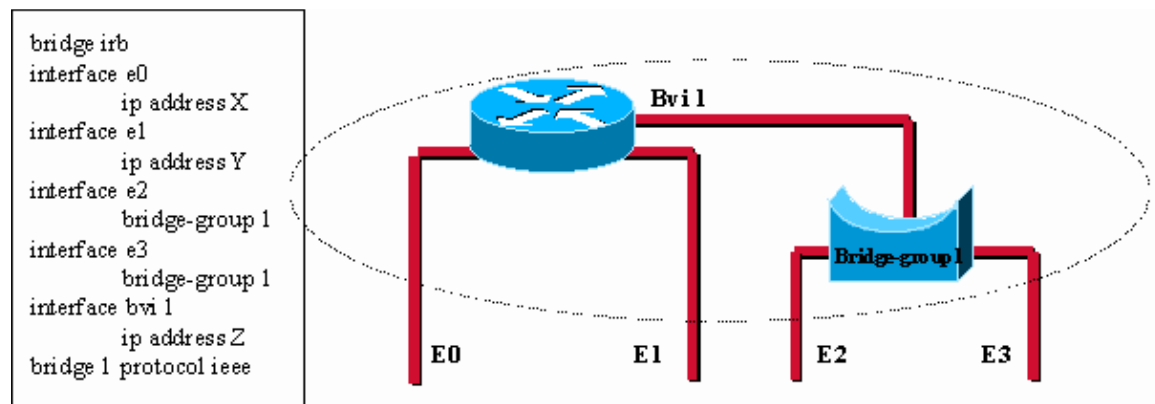
In this configuration, for the IP protocol, the Cisco device is acting like a router for interface e0 and e1 and is acting like a bridge for interface e2 and e3. Note that there is no communication possible between the two functions (a host connected on e0 would never be able to reach a host connected on e2 through the router with this configuration).

Figure 20-4: Concurrent Routing and Bridging (CRB)

Integrated Routing and Bridging (IRB)

IRB provides the ability to route between a bridge-group and a routed interface with a concept called Bridge-Group Virtual Interface (BVI). Because bridging occurs at the data link layer and routing at 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 with its own IP network address.

The concept of BVI was created to enable these interfaces to exchange packets for a given protocol. Conceptually, as shown in this example, the Cisco router looks like a router connected to one or more bridge-groups:

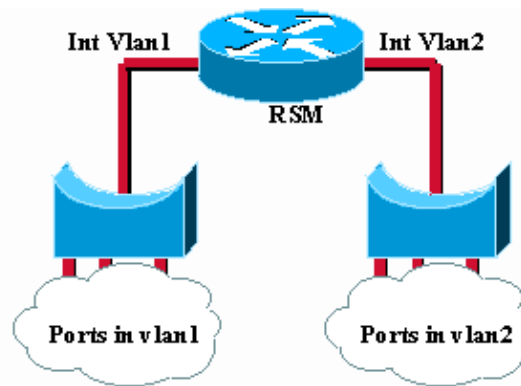


The bridge group virtual interface brings routing to bridge-group 1. One can assign an IP address to the whole bridge-group and routed communication is now possible between a host connected to E0 and a host connected to E2 for instance.

Figure 20–5: Integrated Routing and Bridging (IRB)

The BVI is a virtual interface within the router that acts like a normal routed interface. The BVI represents the correspondent bridge–group to routed interfaces within the router. The interface number of the BVI is the number of the bridge–group represented by this virtual interface. The number is the link between this BVI and the bridge–group.

This example illustrates how the BVI principle applies to the Route Switch Module (RSM) in a Catalyst switch:



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

Figure 20–6: Route Switch Module (RSM) in a Catalyst Switch.

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."

In order to efficiently troubleshoot your bridged network, you must have a basic knowledge of its design, especially when a spanning tree is involved.

These must be available:

- Topology map of the bridged network
- Location of the root bridge
- Location of the redundant link (and blocked ports)

When you troubleshoot connectivity issues, reduce the problem to a minimum number of hosts, ideally only a client and a server.

These 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: The client cannot connect to hosts across a transparently bridged network.

Table 20–3 outlines the problems that can cause this symptom and suggests solutions.

Table 20–3: Transparent Bridging: No Connectivity

Possible Causes	Suggested Actions
Hardware or media problem	<ol style="list-style-type: none"> 1. Use the show bridge EXEC command to see if there is a connectivity problem. If so, 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. <p>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.</p>
Host is down	<ol style="list-style-type: none"> 1. Use the show bridge EXEC command on bridges to make sure that the bridging table includes the MAC addresses of attached end-nodes. <p>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.</p> <ol style="list-style-type: none"> 2. If any expected end-nodes are missing, check the status of the nodes to verify that they are connected and properly configured.

	<ol style="list-style-type: none"> 3. Reinitialize or reconfigure end-nodes as necessary and reexamine the bridging table with the show bridge command.
<p>Bridging path is broken</p>	<ol style="list-style-type: none"> 1. Identify the path that packets must take between end-nodes. If there is a router on this path, split the troubleshooting into two parts: Node 1–Router and Router–Node 2. 2. Connect to each bridge on the path and check the status of the ports used on the path between end-nodes (as described in the "Hardware or media problem" table entry. 3. Use the show bridge command to make sure that the MAC address of the nodes are learned on the correct ports. If not, there can be instability on your spanning tree topology. See Table 20–2, "Transparent Bridging: Unstable Spanning Tree." 4. Check the state of the ports with the show span command. If the ports that can transmit traffic between the end-nodes are not in the forwarding state, the topology of your tree can have changed unexpectedly. See Table 20–4, "Transparent Bridging Unstable Spanning Tree."
<p>Misconfigured bridging filters</p>	<ol style="list-style-type: none"> 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 is restored. 3. If connectivity is not restored, the filter is not the problem. If connectivity is restored after filters are removed, one or more bad filters are the cause of the connectivity problem. 4. If either multiple filters exist or filters that use 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 the TYPE filters, which can be used simultaneously to block different protocols. For example, LSAP (F0F0) can be used to block NetBIOS, and

	<p>TYPE (6004) can be used to block local area transport.</p> <p>5. Modify any filters or access lists that block traffic. Continue to test filters until all filters are enabled and connections still work.</p>
Input and output queues full	<p>Excessive multicast or broadcast traffic can cause input and output queues to overflow, which results in dropped packets.</p> <ol style="list-style-type: none"> 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 above 80% of the current size of the input queue, the size of the input queue needs to be tuned to accommodate the 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 can still overflow the queue. 2. Reduce broadcast and multicast traffic on attached networks with the use of bridging filters, or segment the network with more internetwork devices. 3. If the connection is a serial link, increase bandwidth, apply priority queues, increase the hold queue size, or modify the system buffer size. For more information, refer to Chapter 15, "Troubleshooting Serial Line Problems."

[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 20–4 outlines the problems that can cause this symptom and suggests solutions.

Table 20–4: Transparent Bridging: Unstable Spanning Tree

Possible Causes	Suggested Actions
Link flapping	<ol style="list-style-type: none"> 1. Use show span command to see if the number of topology changes steadily

	<p>increases.</p> <ol style="list-style-type: none"> If so, check the link between your bridges with the show interface command. If this command does not reveal a link flapping between two bridges, use the debug spantree event privileged EXEC command on your bridges. <p>This logs all changes related to the spanning tree. In a stable topology, there cannot be any. The only links to track are the ones that connect the bridge devices together. A transition on a link to an end-station should have no impact on the network.</p> <p>Note: Because the debug output is assigned a high priority in the CPU process, to use the debug spantree event command can render the system unusable. For this reason, use debug commands only to troubleshoot specific problems or when in sessions to troubleshoot problems with Cisco technical support staff. Moreover, it is best to use debug commands within periods of low network traffic and fewer users. If you debug within these periods, it decreases the likelihood that increased debug command overhead processes will affect system use.</p>
<p>Root bridge continues to change/ multiple bridges claim to be the root</p>	<ol style="list-style-type: none"> Check the consistency of the root bridge information all over the bridged network with the show span commands on the different bridges. If there are several bridges that claim to be the root, make sure that you run the same spanning tree protocol on every bridge (see the Spanning tree algorithm mismatch" table entry in Table 20-6). Use the bridge <group> priority <number> command on the root bridge to force the desired bridge to become the root. The lower the priority, the more likely it is for the bridge to become the root. Check the diameter of your network. With a standard spanning tree set up, there must never be more than seven bridge hops between two hosts.
<p>Hellos not exchanged</p>	<ol style="list-style-type: none"> Check to see if bridges communicate with one another. Use a network analyzer or the debug spantree tree

privileged EXEC command to see if spanning tree hello frames are exchanged.

Note: Because the debug output is assigned a high priority in the CPU process, to use the **debug spantree event** command can render the system unusable. For this reason, use **debug** commands only to troubleshoot specific problems or when in sessions to troubleshoot problems with Cisco technical support staff. Moreover, it is best to use **debug** commands within periods of low network traffic and fewer users. If you debug within these periods, it decreases the likelihood that increased **debug** command overhead processes will affect system use.

2. If hellos are not exchanged, check the physical connections and software configuration on the bridges.

Transparent Bridging: Sessions Terminate Unexpectedly

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

Table 20–5 outlines the problems that can cause this symptom and suggests solutions.

Table 20–5: Transparent Bridging: Sessions Terminate Unexpectedly

Possible Causes	Suggested Actions
Excessive retransmissions	<ol style="list-style-type: none"> 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 how to configure your hosts, refer to the vendor documentation. For information on how to troubleshoot serial lines, refer to Chapter 15, "Troubleshooting Serial Line Problems." <p>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 switch). Troubleshoot the LAN media as</p>

	<p>appropriate. For more information, refer to the chapter about how to troubleshoot media that covers the media type used in your network.</p> <p>3. Use a network analyzer to determine whether the number of retransmissions subsides.</p>
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, "Troubleshooting Serial Line Problems."

Transparent Bridging: Looping and Broadcast Storms Occur

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

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

Table 20–6 outlines the problems that can cause this symptom and suggests solutions.

Bridging loops are the worst case scenario in a bridged network since it will potentially impact every user. In case of emergency, the best way to recover connectivity quickly is to manually disable all the interfaces that provide redundant path in the network. Unfortunately, the cause of the bridging loop will be very difficult to identify afterward if you do so. If possible, try Table 20–6 actions beforehand.

Table 20–6: Transparent Bridging: Looping and Broadcast Storms Occur

Possible Causes	Suggested Actions
No spanning tree implemented	<ol style="list-style-type: none"> 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	<ol style="list-style-type: none"> 1. Use the show span EXEC command on each bridge to

	<p>determine which spanning tree algorithm is used.</p> <ol style="list-style-type: none"> 2. Make sure that all bridges run the same spanning tree algorithm (either DEC or IEEE)[1]. It can be necessary to use both the DEC and IEEE spanning tree algorithms in the network for some very specific configurations (generally, those that involve IRB). If the mismatch in the spanning tree protocol is not intended, reconfigure the bridges as appropriate so that all the bridges use the same spanning tree algorithm. <p>Note: The DEC and IEEE spanning tree algorithms are incompatible.</p>
<p>Multiple bridging domains incorrectly configured</p>	<ol style="list-style-type: none"> 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 other domains.
<p>Link error (unidirectional link), duplex-mismatch, high level of error on a port.</p>	<p>Loops occur when a port that should block moves to the forwarding state. A port needs to receive BPDUs from a nearby bridge in order to stay in the blocking state. Any error that leads to lost BPDUs can then be the cause of a bridging loop.</p> <ol style="list-style-type: none"> 1. Identify blocking ports from your network diagram.

2. Check the status of the ports that should block in your bridged network with the **show interface** and **show bridge EXEC** commands.
3. If you find a possibly blocked port that is currently forwarding or is about to forward (that is, in the learn or listen state) you have found the real source of the problem. Check to see if this port receives BPDUs. If not, there is probably an issue on the link connected to this port. Then check link errors, duplex setting, and so on).

If the port still receive BPDUs, go to the bridge that you expect to be designated for this LAN. Then 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 You Call Cisco Systems TAC Team

When your network is stable, collect as much information as you can about its topology.

At a minimum collect this data:

- Physical topology of the network
- Expected location of the root bridge (and backup root bridge)
- Location of blocked ports

Additional Sources

Books:

- Interconnections, Bridges and Routers, Radia Perlman, Addison–Wesley
- Cisco Lan Switching, K.Clark, K.Hamilton, Cisco Press

Related Information

- [Transparent Bridging Documentation](#)
 - [Technical Support & Documentation – Cisco Systems](#)
-

