CHAPTER 23

Chapter Goals

- Understand transparent bridge processes of learning, filtering, forwarding, and flooding.
- Explain the purpose of the spanning-tree algorithm.
- Describe the bridge and port modes in a spanning-tree network.

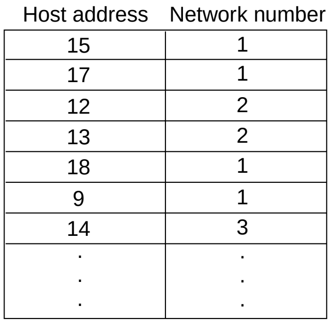
Transparent Bridging

Transparent bridges were first developed at Digital Equipment Corporation (Digital) in the early 1980s. Digital submitted its work to the Institute of Electrical and Electronic Engineers (IEEE), which incorporated the work into the IEEE 802.1 standard. Transparent bridges are very popular in Ethernet/IEEE 802.3 networks. This chapter provides an overview of transparent bridging's handling of traffic and protocol components.

Transparent Bridging Operation

Transparent bridges are so named because their presence and operation are transparent to network hosts. When transparent bridges are powered on, they learn the workstation locations by analyzing the source address of incoming frames from all attached networks. For example, if a bridge sees a frame arrive on port 1 from Host A, the bridge concludes that Host A can be reached through the segment connected to port 1. Through this process, transparent bridges build a table (the learning process), such as the one in Figure 23-1.

Figure 23-1 Transparent Bridges Build a Table That Determines a Host's Accessibility



Host address	Network number
15	1
17	1
12	2
13	2
18	1
9	1
14	3
. . .	. . .

The bridge uses its table as the basis for traffic forwarding. When a frame is received on one of the bridge's 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 also are flooded in this way.

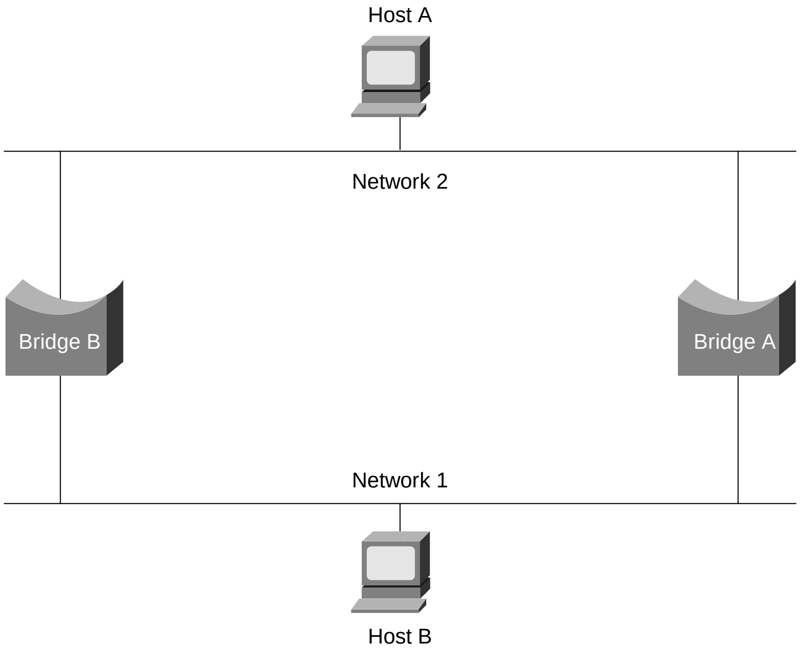
Transparent bridges successfully isolate intrasegment traffic, thereby reducing the traffic seen on each individual segment. This is called *filtering* and occurs when the source and destination MAC addresses reside on the same bridge interface. Filtering usually improves network response times, as seen 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 multiple paths of bridges and local-area networks (LANs) exist between any two LANs in the internetwork. Figure 23-2 illustrates such a bridging loop.

Suppose that Host A sends a frame to Host B. Both bridges receive the frame and correctly learn that Host A is on segment 2. Each bridge then forwards the frame onto segment 2. Unfortunately, not only will Host B receive two copies of the frame (once from bridge 1 and once from bridge 2), but each bridge now believes that Host A resides on the same segment as Host B. When Host B replies to Host A's frame, both bridges will receive and subsequently filter the replies because the bridge table will indicate that the destination (Host A) is on the same network segment as the frame's source.

Figure 23-2 Bridging Loops Can Result in Inaccurate Forwarding and Learning in Transparent Bridging Environments



In addition to basic connectivity problems, the proliferation of broadcast messages in networks with loops represents a potentially serious network problem. Referring again to Figure 23-2, assume that Host A's initial frame is a broadcast. Both bridges 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 23-2, can be useful as well as potentially harmful. A loop implies the existence of multiple paths through the internetwork, and a network with multiple paths from source to destination can increase overall network fault tolerance through improved topological flexibility.

Spanning-Tree Algorithm

The *spanning-tree algorithm (STA)* was developed by Digital Equipment Corporation, a key Ethernet vendor, to preserve the benefits of loops while eliminating their problems. Digital's algorithm subsequently was 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 compatible.

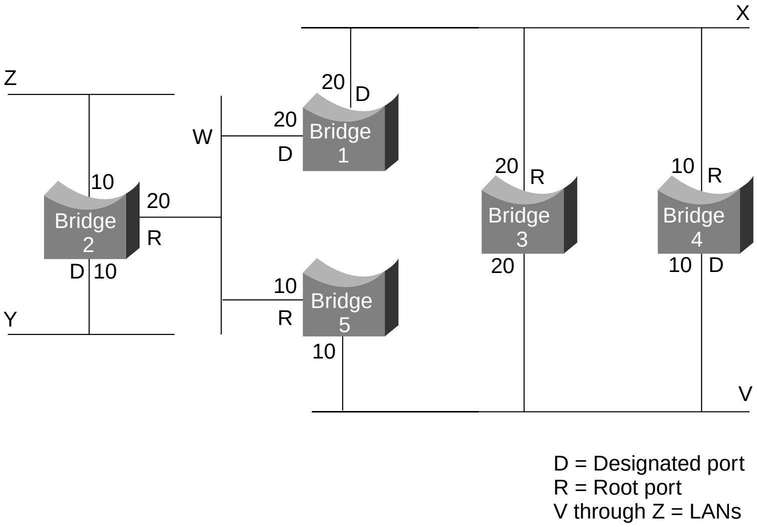
The STA designates a loop-free subset of the network's topology by placing those bridge ports that, if active, would create loops into a standby (blocking) condition. Blocking bridge ports can be activated in the event of a primary link failure, providing a new path through the internetwork.

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, a spanning tree of edges maintains the connectivity of the graph but contains no loops.

Figure 23-3 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 an administratively assigned priority. Each port in every bridge also is assigned a unique identifier (within that bridge), which is typically its own MAC address. Finally, each bridge port is associated with a path cost, which represents the cost of transmitting a frame onto a LAN through that port. In Figure 23-3, path costs are noted on the lines emanating from each bridge. Path costs are usually defaulted but can be assigned manually by network administrators.

Figure 23-3 STA-Based Bridges Use Designated and Root Ports to Eliminate Loops



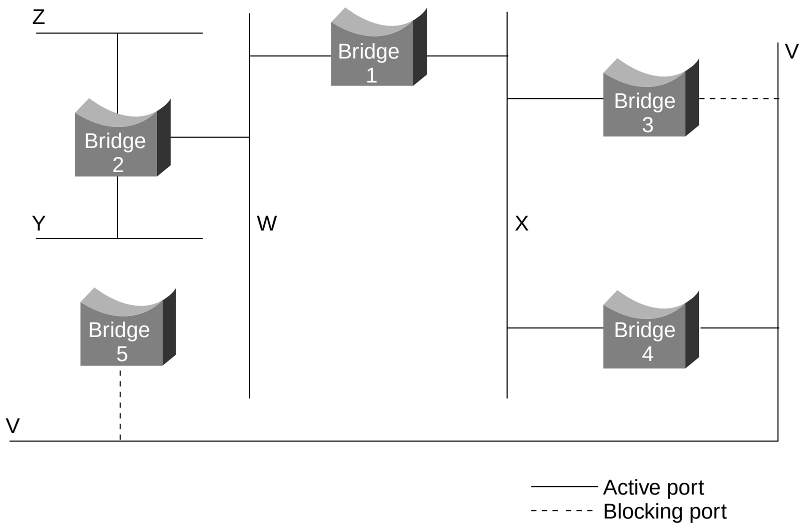
The first activity in spanning-tree computation is the selection of the *root bridge*, which is the bridge with the lowest-value bridge identifier. In Figure 23-3, the root bridge is Bridge 1. Next, the *root port* on all other bridges is determined. A bridge's root port is the port through which the root bridge can be reached with the least aggregate path cost, a value that 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. In Figure 23-3, for example, 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 23-4 shows the results of applying the STA to the network shown in Figure 23-3. Figure 23-4 shows the tree topology more clearly. It also shows that the STA has placed both Bridge 3 and Bridge 5's ports to LAN V in standby mode.

Figure 23-4 A Loop-Free Tree Topology and an STA-Based Transparent-Bridge Network



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, as well as 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 will detect the lack of configuration messages and will initiate a spanning-tree recalculation.

All transparent-bridge topology decisions are made locally by each bridge. Bridges exchange configuration messages with neighboring bridges, and no central authority exists to determine 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. This forces bridges to relearn the location of hosts because a host may originally have been accessed from port 1, although after the topology change it may be reached through port 2.

Figure 23-5 illustrates the IEEE 802.1d configuration-message format.

Figure 23-5 Twelve Fields Comprise the Transparent-Bridge Configuration Message

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

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

- **Protocol Identifier**—Contains the value zero.
- **Version**—Contains the value zero.
- **Message Type**—Contains the value zero.
- **Flag**—Contains 1 byte, of which only 2 bits are used. The topology-change (TC) least significant bit signals a topology change. The topology-change acknowledgment (TCA) most significant 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 multiple attached bridges to be detected and handled.
- **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 might be ready to change their state, and loops can result.

Topology-change messages consist of only 4 bytes. These include a Protocol-Identifier field, which contains the value zero; a *Version* field, which contains the value zero; and a Message-Type field, which contains the value 128.

Review Questions

Q—*What three frame types does a transparent bridge flood?*

A—Transparent bridges flood unknown unicast frames (where the bridge has no entry in its table for the destination MAC address), broadcast frames, and mulitcast frames.

Q—*How does a bridge learn the relative location of a workstation?*

A—A bridge learns about the direction to send frames to reach a station by building a bridge table. The bridge builds the table by observing the source MAC address of each frame that it receives and associating that address with the received port.

Q—*What two bridge PDUs does a transparent bridge generate, and what are they used for?*

A—Transparent bridges create either a configuration PDU or a topology-change PDU. Configuration PDUs help bridges learn about the network topology so that loops may be eliminated. Topology-change PDUs enable bridges to relearn the network topology whenever a significant change occurs when a segment may no longer have connectivity or when a new loop is created.

Q—*What is the difference between forwarding and flooding?*

A—Bridges forward frames out a *single* interface whenever the bridge knows that the destination is on a different port than the source. On the other hand, bridges flood whenever the bridge does not know where the destination is located.

Q—*After bridges determine the spanning-tree topology, they will take on various roles and configure ports into various modes. Specifically, the roles are root and designated bridges, and the modes are designated ports and root ports. If there are 10 bridges and 11 segments, how many of each are there in the broadcast domain?*

A—There is one and only one root bridge in a broadcast domain, and all other bridges are designated bridges. Therefore, there is one root bridge and nine designated bridges. There must be one designated port for each segment, so there are ten. Each bridge, except the root, must have one and only one root port. Therefore there are nine root ports.

For More Information

Clark, Kennedy, and Kevin Hamilton. *CCIE Professional Development: Cisco LAN Switching*. Indianapolis: Cisco Press, 1999.

Perlman, Radia. *Interconnections*, Second Edition: *Bridges, Routers, Switches, and Internetworking Protocols*. Boston: Addison Wesley, 1999.