4

Bridges

INTRODUCTION

This chapter explains how bridges interconnect LANs. The focus is on learning and spanning tree bridges, those bridges that perform many of their operations automatically. We examine token ring bridges, also known as source routing bridges. The chapter provides examples of how the LLC protocol, configured with the type 2 option, is accommodated in a wide area internet. The chapter also explains the operations of a bridge that connects LANs on a point-to-point link to WANs, known as a half-bridge.

WHY USE BRIDGES?

In Chapter 1, several points were made about why internetworking with routers is valuable to the communications industry. These statements apply to this chapter as well. Bridges are also important because in some networks, such as LANs, they may be a requirement to restrict the number of nodes (workstations, routers, servers, etc.) that are placed on the network media. Consequently, an enterprise may be limited in its growth potential if there is no means to connect the geographically-limited LANs together. The bridge is one tool used to connect these LANs.

Second, LANs (for example, Ethernet) are limited in the distance that the media can be strung through a building or a campus. This geographical restriction can be overcome by placing a bridge between the geographically-challenged LAN segments.

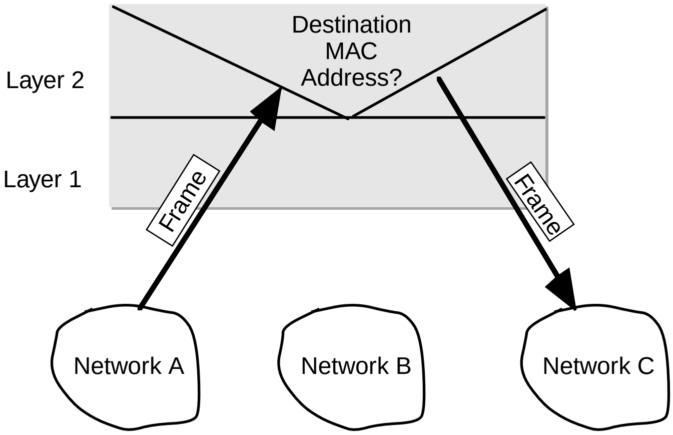
Third, as we mentioned in Chapter 1, the ability to use internetworking units, such as bridges, allows the network manager to contain the amount of traffic that is sent across the expensive network media.

Now that I have said all these wonderful things about bridges, it must also be stated that in many internetworking situations, the router is used in place of a bridge, because it has more capabilities than a bridge.

THE MAC BRIDGE

Bridges are designed to interconnect LANs. Therefore, they use a destination MAC address (see Appendix B, Figure B–2) in determining how to relay the traffic between LANs. A bridge "pushes" the conventional network layer responsibilities of route discovery and forwarding operations into the data link layer. In effect, a bridge has no conventional network layer.

Figure 4–1 shows a multiport bridge, which accepts a frame coming in on a port from network A. The frame is examined by the MAC relay



Where:
MAC Media access control (a LAN address)

Figure 4–1 Bridge Operations

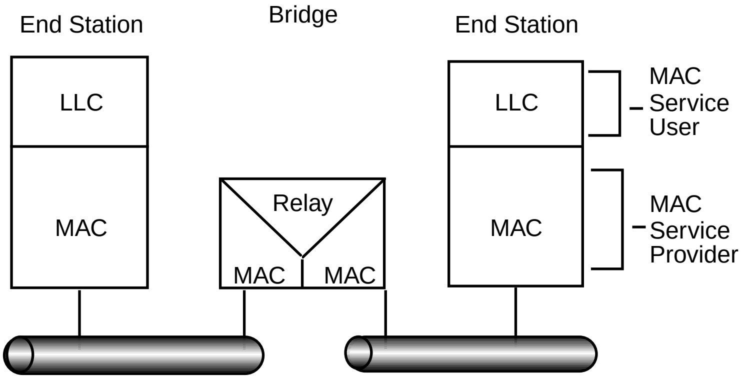


Figure 4–2 The MAC Relay Entity

entity and a decision is made to relay the traffic on an output port to network C.

There is no provision for data integrity in bridges (such as the acknowledgment of traffic, and the possible retransmission of erred traffic). As a consequence, frames can be discarded if the bridge becomes congested. On the other hand, bridges are fast, and they are very easy to implement. Indeed, most bridges are self-configuring. This feature relieves network managers of many onerous tasks, such as the ongoing management of a number of naming and network reconfiguration parameters.

THE OTHER BRIDGE LAYERS

The IEEE internetworking entity is positioned at the MAC layer. As shown in Figure 4–2, the relay entity is designated as a bridge. In this example, the MAC service user is LLC and the MAC service provider is (a) MAC and (b) the MAC relay entity.

Traffic transported across a MAC bridge need only access the MAC layer. Except for certain network management functions, the operation does not require the invocation of any protocol above MAC.

TYPES OF BRIDGES

Several different types of bridges are available for internetworking LANs. They are introduced in this section, and summarized in Table 4–1.

Table 4–1 Types of Bridges

Transparent basic bridge

Places incoming frame onto all outgoing ports except original incoming port

Source routing bridge

Relies on routing information in frame to relay the frame to an outgoing port

Transparent learning bridge

Stores the origin of a frame (from which port) and later uses this information to relay frames to that port

Transparent spanning bridge

Uses a subset of the LAN topology for a loop-free operation

The Transparent Basic Bridge

The simplest type of bridge is called the transparent basic bridge. This bridge receives traffic coming in on each port and stores the traffic until it can be transmitted on the outgoing ports. It will not forward the traffic from the port from which it was received. The bridge does not make any conversion of the traffic. It merely extends LANs beyond what could be achieved with simple repeaters.

Source Routing Bridge

The source routing bridge is so named because the route through the LAN internet is determined by the originator (the source) of the traffic. As shown in Figure 4–3, the routing information field (RIF), contained in the LAN frame header, contains information on the route that the traffic takes through the LAN internet.

At a minimum, routing information must identify the intermediate nodes that are required to receive and send the frame. Therefore, source routing requires that the user traffic follow a path that is determined by the routing information field.

The architecture for source routing is similar to the architecture for all bridges in that both use a MAC relay entity at the LAN node. Interfaces are also provided through primitives to the MAC relay entity and to LLC. However, the frames of the source routing protocol are different from those of other bridge frames because the source routing information must be contained within the frame.

Figure 4–4 shows the functional architecture for source routing bridges. Two primitives are invoked between the MAC entities and LLC.

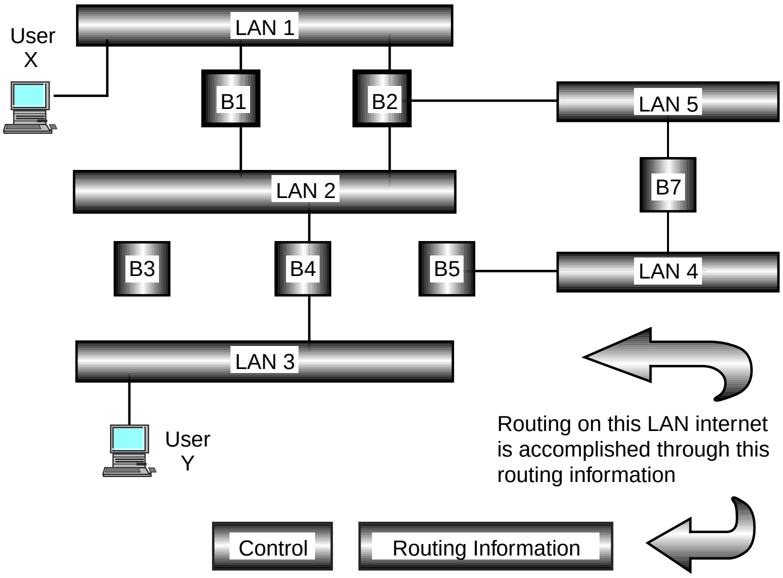

User X
LAN 1
B1
B2
LAN 5
LAN 2
B7
B3
B4
B5
LAN 4
LAN 3
User Y
Routing on this LAN internet is accomplished through this routing information
Control
Routing Information

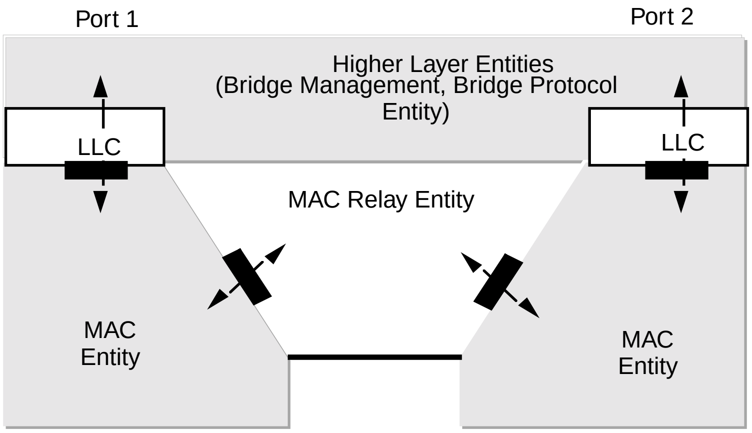
Figure 4–3 Source Routing Concept

The first primitive is the M_UNITDATA.request, and the second primitive is the M_UNITDATA.indication.

The parameters in these primitive calls must contain the information to create the frame (frame control), and the MAC addresses, and of course the routing information that is used to forward the traffic through the LAN internet. A frame check sequence value is included if frame check sequence operations are to be performed. The primitives also contain a data parameter, a user priority parameter, and a service class parameter. These latter two parameters are used only with token rings and are not found in the primitives calls for other LANs, such as Ethernet or token bus.

The Transparent Learning Bridge

The transparent learning bridge, depicted in Figure 4–5, finds the location of user stations by examining the source and destination addresses in the frame when the frame is received at the bridge. The destination address is stored if it is not in a routing table and the frame is sent to all LANs except the LAN from which it came. In turn, the source address is stored with the direction (incoming port) from which it came.



M_UNITDATA request

(frame_control, destination_address, source_address, routing_information, frame_check_sequence, data, user_priority, service_class, suppress_fcs)

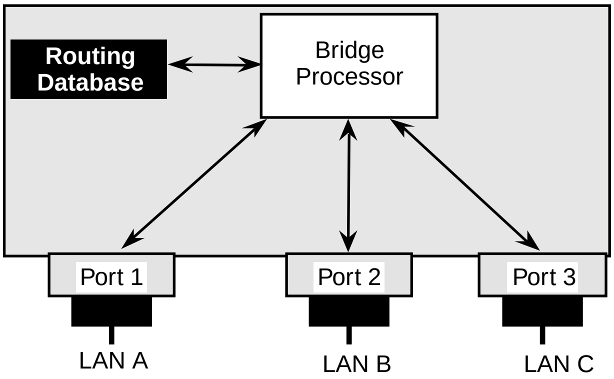
M_UNITDATA indication

(frame_control, destination_address, source_address, routing_information, data, frame_check_sequence, user_priority, service_class)

Figure 4–4 Source Routing Layers and Primitives

Consequently, if another frame is received in which this source address is now a destination address, it is forwarded across this port. The only restriction to the use of a transparent learning bridge is that the physical topology cannot allow loops.

The learning bridge operates with a bridge processor, which is responsible for routing traffic across its ports. The processor accesses a routing database which contains the destination ports of associated MAC addresses. When a frame arrives at an incoming port on the bridge, the bridge examines its database to determine the output port on which the frame will be relayed. If the destination address is not in the directory, the bridge processor will broadcast the frame onto all ports except the port from which the frame arrived. As mentioned earlier, the bridge processor also stores information about the source address in the frame. This information is stored in the database and contains the source port



- Processor examines both source and destination addresses in frames
- Looks for destination address in routing database; if found, routes according to the database; if not found, broadcasts frame to all ports except the originating port
- Also looks for source address in the routing database; stores the direction from which it came—on which port it arrived

Figure 4–5 The Transparent Learning Bridge

from which the frame arrived. This information aids the processor in determining where to route a later frame that contains (in its destination field) an address that was received earlier as a source address.

Figure 4–6 shows how a bridge processes an incoming frame in relation to its destination address (DA) and its source address (SA). The bridge is processing a frame coming in from port 1 with a DA of A and SA of B. Upon accessing its routing database, it finds that it does not have the DA of A in its database. Therefore, it broadcasts this frame out to all ports except the port from which this frame came (port 1). After it has forwarded the frame, it determines if it knows about the SA. If the SA is stored in its routing database, it will update this entry in the database by refreshing a timer which means that this address is still “timely and valid.” In this example, it does not know about the SA of B. Therefore it stores in its database that B is an active station on the LAN and that, from the viewpoint of this bridge, B can be found on port 1.

In Figure 4–7, a frame arrives at the bridge on port 3 containing destination address B and source address C. The first task of the bridge is to route the frame. Therefore, it consults its routing database and determines that B can be reached through its port 1. This determination is

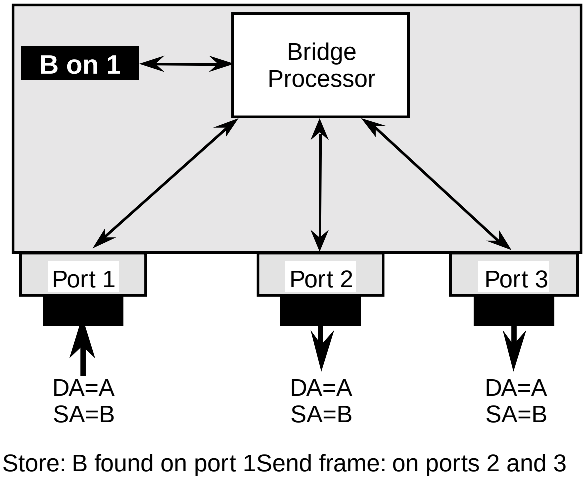


Figure 4–6 Learning, Forwarding and Filtering Operations

made from a previous operation in which a frame arrived on port 1 with B's address in the source address field. Since the bridge understands that address B is on port 1, it does not forward this frame to port 2. The bridge also stores in its routing database that the source address C can be reached on port 3. Additionally, it does not forward the frame to port 3 because this would send the frame backward. This latter statement is

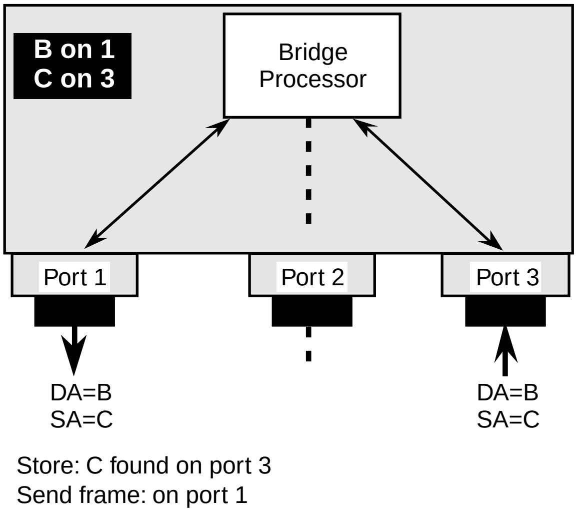


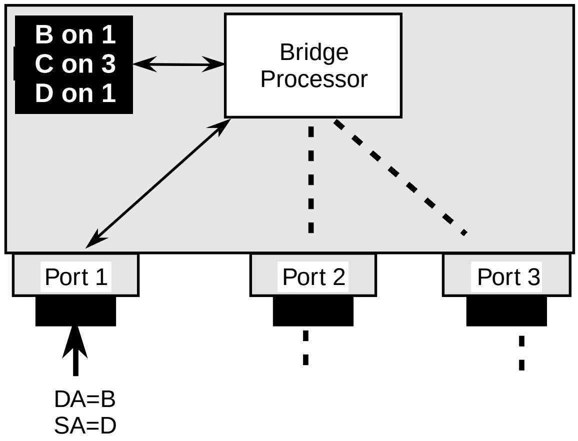
Figure 4–7 Bridge Learns About C, Forwards to Port 1

important because the learning bridge is based on trust. That is to say, the bridge assumes that the frame received on an incoming port has been properly delivered by the downstream bridges and LANs.

In some situations, a bridge will not forward the frame to any port. Figure 4–8 shows one example of why complete filtering is possible. A frame has arrived at the bridge on port 1. Its contents contain a DA of B and a SA of D. Once again, the bridge consults its routing database which reveals that DA B can be found on port 1. Since the frame arrived on port 1, it will not forward this frame to ports 2 and 3 nor will it send it “backward” to port 1. In addition, once it has taken care of the relaying operations, it makes certain that the SA is checked against its routing database. In this instance, the SA is D; it is not known in the database at this time, and therefore an entry to the database is added and a time is attached to the entry.

A learning bridge permits the use of multicasting and broadcasting. In Figure 4–9, a frame arrives from port 1 with a DA set to ALL (all 1s in the address field). The source address is D. The bridge processor does not update its table because D is already known as coming from port D, and the relaying process is straightforward. It need only relay the traffic to all other outgoing ports. In this example, the traffic is sent to ports 2 and 3.

Figure 4–10 provides examples of how a bridge forwards and filters frames. A frame transmitted on the LAN from station A to station B is



Store: D found on port 1
Send frame: No

Figure 4–8 Bridge Learns About D, but Filters

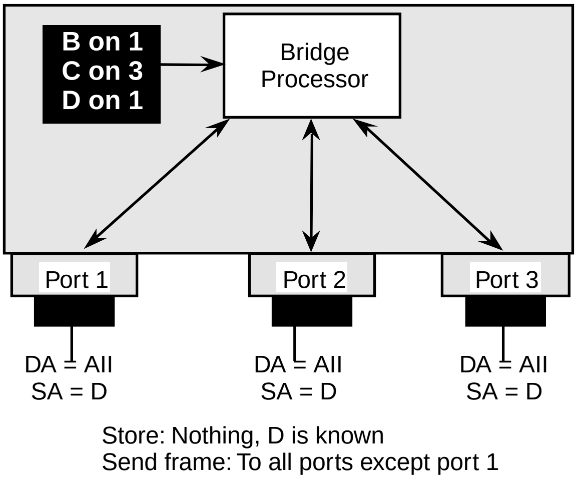


Figure 4–9 Multicasting—Filtering on Incoming Port Only

not forwarded by bridge 1. The bridge assumes the traffic was successfully transferred on the broadcast network between A and B. Traffic destined from station A to station C must be forwarded by bridge 1 in order to reach station C. However, this frame is discarded (filtered) by bridge 2. Both bridges 1 and 2 must forward traffic destined from station A to station D.

Figure 4–11 shows a flowchart used by a learning bridge to (a) determine the destination port for a frame and (b) update the routing database. Upon receiving a frame from a port (in this example, port A), the bridge examines the routing database to determine if the destination

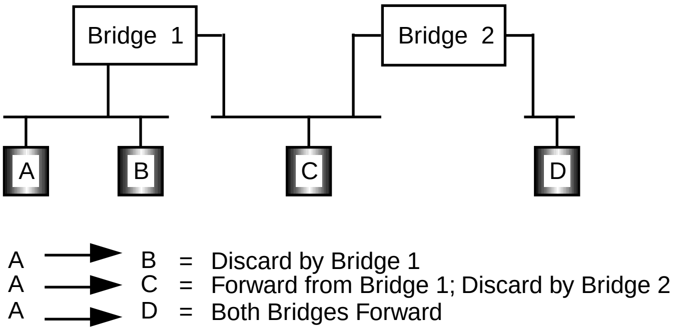

Bridge 1
Bridge 2
A
B
C
D
A → B = Discard by Bridge 1
A → C = Forward from Bridge 1; Discard by Bridge 2
A → D = Both Bridges Forward

Figure 4–10 Discarding Frames at the Bridges

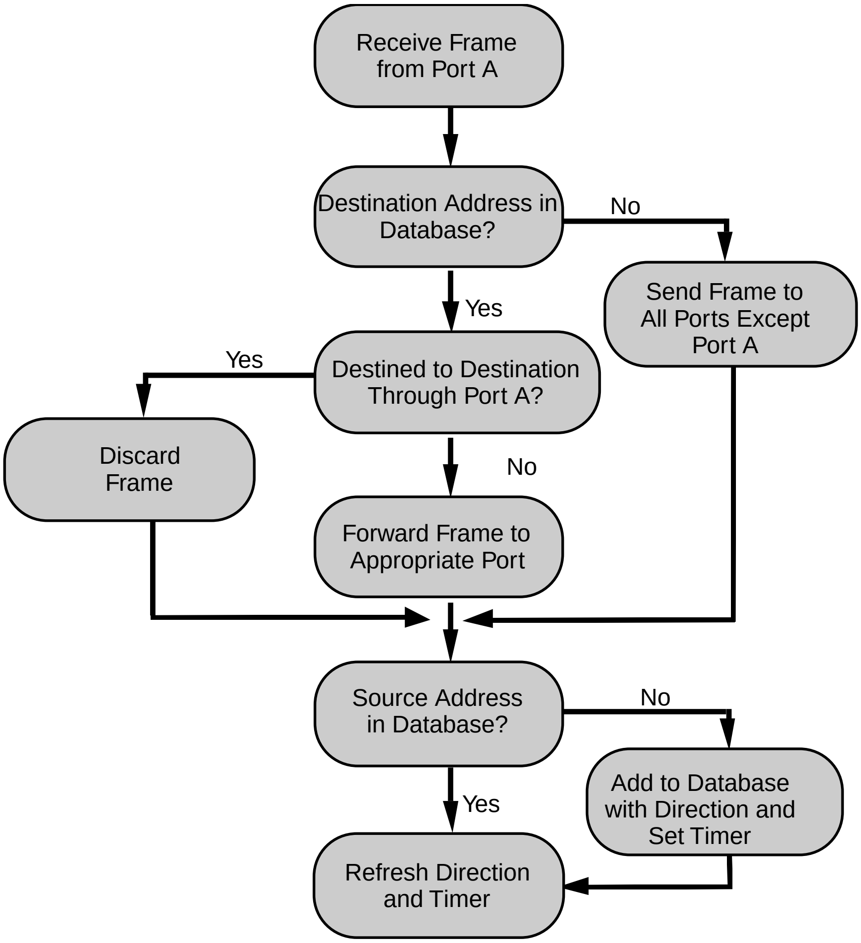

Receive Frame from Port A
Destination Address in Database?
No
Send Frame to All Ports Except Port A
Yes
Yes
Destined to Destination Through Port A?
Discard Frame
No
Forward Frame to Appropriate Port
Source Address in Database?
No
Add to Database with Direction and Set Timer
Yes
Refresh Direction and Timer

Figure 4–11 Learning Bridge Logic

MAC address exists. If not, the frame is broadcast to all ports except the source port (port A). If the address exists in the database, it is forwarded to the appropriate port. Otherwise, the frame is discarded.

The next step is to determine if the MAC source address that was in the frame exists in the routing database. If it does not exist, the address is added to the database with an entry revealing that it came from port A. A timer is set on this entry in order to keep the routing database up-to-date. If the database becomes full, older entries are cashed out. If the source address already exists in the database, the direction is checked, perhaps refreshed, and the timer is reset.

The Transparent Spanning Tree Bridge

The last type of bridge is called a spanning tree (or transparent spanning) bridge. Unlike the previous examples in this explanation, the spanning tree bridge uses a subnet of the full topology to create a loop-free operation.

Figure 4–12 shows the functional logic of the IEEE 802.1 bridge. The received frame is examined by the relay entity in the following manner. The destination MAC address contained in the frame is matched against a routing database (known in some IEEE documents as the filtering database). In addition, information is stored relative to the bridge ports. This information is called port state information and reveals if a port can be used for this destination address. A port could be in a blocked state to fulfill the requirements of spanning tree operations. If the filtering database reveals an outgoing port for the frame and the port is in a forwarding state, the frame is routed across the port.

The 802.1 standard requires that the bridges' ports operate in other conditions as well. For example, a port state might be "disabled" for

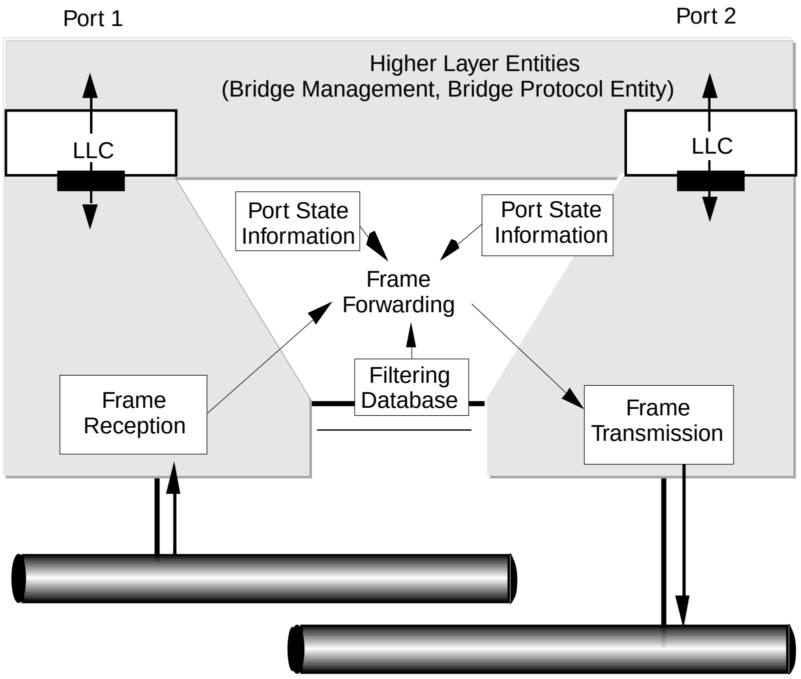


Figure 4–12 Spanning Tree Relay Operations

reasons of maintenance or because of malfunctions. Ports may also be temporarily unavailable if filtering databases are being changed in the bridge because of a result of changes noted during route discovery operations on the network.

The Configuration Message

Figure 4–13 shows the format for the configuration message, also called a bridge protocol data unit (BPDU). The protocol identifier is set to 0. Also, the version identifier is 0. The message type for the configuration message is 0.

The flags field contain a topology change notification flag. It is used to inform nonroot bridges that they should age-out station entries in cache. This field also contains a topology change notification bit. It is used to inform the bridges that they do not have to inform a parent bridge that a topology change has occurred. The parent bridge will perform this task.

The root identifier contains the ID of the root, plus a 2-octet field that can be used to establish a priority for the selection of the root bridge

	Octets
Protocol ID	2
Version	1
BPDU type	1
Flags	1
Root identifer	8
Path cost to root	4
Bridge identifier	8
Port identifier	2
Message age	2
Max age	2
Hello time	2
Forward delay	2

Figure 4–13 802.1 Bridge Message or Protocol Data Unit (BPDU)

and the designated bridge. The root path cost field represents the total cost from the transmitting bridge to the bridge that is listed in the root identifier field.

The bridge and port identifiers are the priority and ID of the bridge (and the reported port) that is sending the configuration message. The message age field is a time, in 1/256th of a second, since the root bridge sent its configuration message from which this message is derived. The max age field, also in 1/256th of a second, contains the time when the configuration message is no longer valid and should be deleted. The hello time field, also in 1/256th of a second, defines the time between the sending of configuration messages by the root bridge. The forward delay field, also in 1/256th of a second, is the time lapse in which a port should stay in an intermediate state (learning, listening) before moving from a blocking state to a forwarding state.

POTENTIAL LOOPING AND BLOCKING PROBLEMS

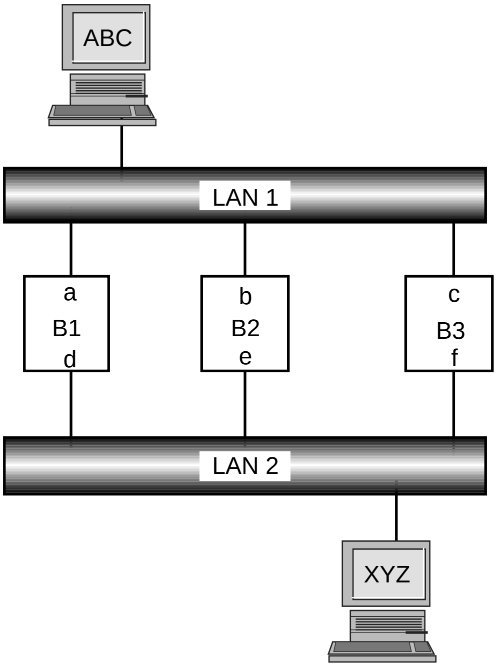
Many LANs are internetworked with many multiport bridges, where the bridges permit a looped, nontree topology. In such a configuration, it is possible for packets to loop around through the network over and over again. Depending on how the networks and bridges are set up, it also possible for packets to be blocked by a bridge and not allowed to transit to a proper destination.

The next two sections provide examples of looping and blocking problems. I have made up these examples for the purpose of showing these potential problems; in real implementations, the bridges do not permit these operations to occur (unless the bridges have been incorrectly configured).

Looping

As illustrated in Figure 4–14, bridges B1, B2, and B3 have two ports each for access to LAN 1 and LAN 2. This topology presents potential problems in that the three bridges could possibly forward the same copy of a frame, and continue sending the frame onto both LANs indefinitely [PERL92].[1] For example, assume a frame is sent by station ABC onto LAN 1, destined for station XYZ on LAN 2. The three bridges receive the

[1] [PERL92] Perlman, Radia, *Interconnections: Bridges and Routers,* Addison-Wesley, 1992.



Event	Result
1: ABC sends packet onto LAN 1, received on ports a, b, c at bridges	Bridges note ABC is on LAN 1 and, Queue packet on ports d, e, f for LAN 2
2: Bridge 3 sends packet onto LAN 2	Bridges 1 and 2 note ABC is on LAN 2 and, queue packet on ports a and b for LAN 1
3. Bridge 1 sends packet onto LAN 2	Bridge 2: ABC still on LAN 2 Bridge 3: ABC has moved to LAN 2 Queue packet on ports b and C for LAN 1
4. Bridge 1 sends packet onto LAN 1	Bridge 2: ABC moved to LAN 1 Bridge 3: ABC moved to LAN 1 Queue packet on ports e and f for LAN 2

Figure 4–14 Looping Problems [PERL92]

frame, and note the direction of the frame. B1 notes that ABC can be found on its port a, LAN 1. B2 notes that ABC can be found its port b, LAN 1. B3 notes that ABC can be found on its port c, LAN 1. The three bridges send the frame to LAN 2 across their ports d, e, and f respectively. These operations are represented by event 1 in Figure 4–14.

Three copies of the frame are now introduced onto LAN 2. For this example, let us assume that B3 sends this frame first. When this frame is processed at B1 and B2 (in event 2), they will note that ABC resides on

LAN 2, and they queue this frame back to LAN 1 on their a and b ports, respectively. Thus, a loop has started. If you follow events 3 and 4 in the table accompanying Figure 4–14, it is revealed that not only do the frames loop between the networks, they multiply: each successful frame transmittal results in yet another copy of the frame being created.

The solution to this potential problem is to prevent the bridges from forwarding the frame onto LAN 1 and to prevent the frame from being sent back to LAN 2. These preventive measures form the basis for spanning tree logic. In essence, a spanning tree protocol logically blocks certain ports such that one and only one route exists between any source and any destination.

Blocking

Another potential problem that spanning tree algorithms solve is also illustrated in Figure 4–14, with operations at users ABC and XYZ, and B1 and B2. First, we must assume that the looping problem in the previous discussion has been solved.

User ABC sends traffic onto LAN 1 that is destined for user XYZ. The bridges note the origin of this traffic: that is, user ABC can be found on LAN 1. Next, the bridges receive each other's traffic on LAN 2. Since the source address in the frame is user ABC, the bridges assume that user ABC has relocated and is now on LAN 2. Next, assume at a later time that user XYZ sends a frame to user ABC. The bridges do not forward this frame after examining the destination address of ABC, since they assume XYZ's transmittal of this frame onto LAN 2 has reached user ABC successfully.

Clearly, these two examples of traffic flow management are not acceptable, and remedial measures are taken to prevent these operations.

THE SPANNING TREE OPERATIONS

Before a spanning tree bridge can operate, it must first prune its topology to a nonlooping tree. In so doing, it follows several well-ordered procedures. See Figure 4–15. The first task is to determine an anchor point from which to calculate a cost through the network. This process is used to identify one bridge among all the bridges in the routing domain to be a "root." This root selection is arbitrary based on the comparison of the ID of the root, an assumed cost to the root (which is a 0 from all bridges initially because they think themselves as the root), the desig-

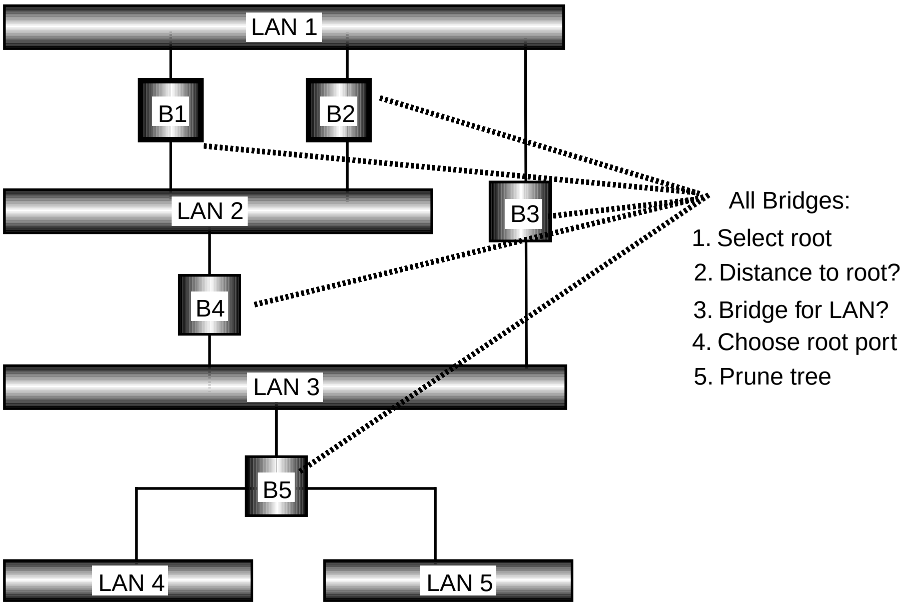


Figure 4–15 Exchange Configuration Messages to Prune the Tree

nated root ID, and the port ID on the root. This number concatenated from left to right is examined by each bridge when it receives messages from other bridges to determine who becomes the root bridge. Once again, this process is arbitrary, and it is not important who becomes the root as long as there is a reference point from which to calculate costs.

Next, configuration messages are exchanged between the bridges with distance values in these messages. The purpose of these exchanges is to allow the bridges to calculate the distance from themselves to the root. During this operation, each LAN will select a designated bridge on that LAN (if multiple bridges exist) to act as the bridge to the route. By examining the costs in the configuration messages, it can be determined which bridge is “closest” to the root. Upon this decision being made, this designated bridge will be assigned the job of sending messages from this LAN toward the root.

The next process involves choosing the best port from the particular bridge to the root. This process is known as “choosing the root port.” Finally, after all these activities, the bridges perform the spanning tree algorithm and essentially prune out paths that could create loops by simply keeping paths open on root ports and any ports that have been designated for that bridge as the ports with the lowest cost to the root.

The configuration messages transmitted by the LAN station are used to inform other stations about the transmitting nodes knowledge of the “reachability” to these other nodes. Figure 4–16 shows the format for the configuration packet. The originator of the packet must place its MAC address in the source address field of the frame and a multicast address value in the destination address field. The SAP values are coded in accordance with specific network implementations. The information content of the frame consists of an assumed root identifier (root ID), the sending bridge ID, the identification of the port from which the message was sent (port ID), and the known cost to the perceived root.

The initial values of the root ID and the perceived cost to the root are “tentative” values in an initial configuration. As subsequent configuration messages are exchanged, these values may change.

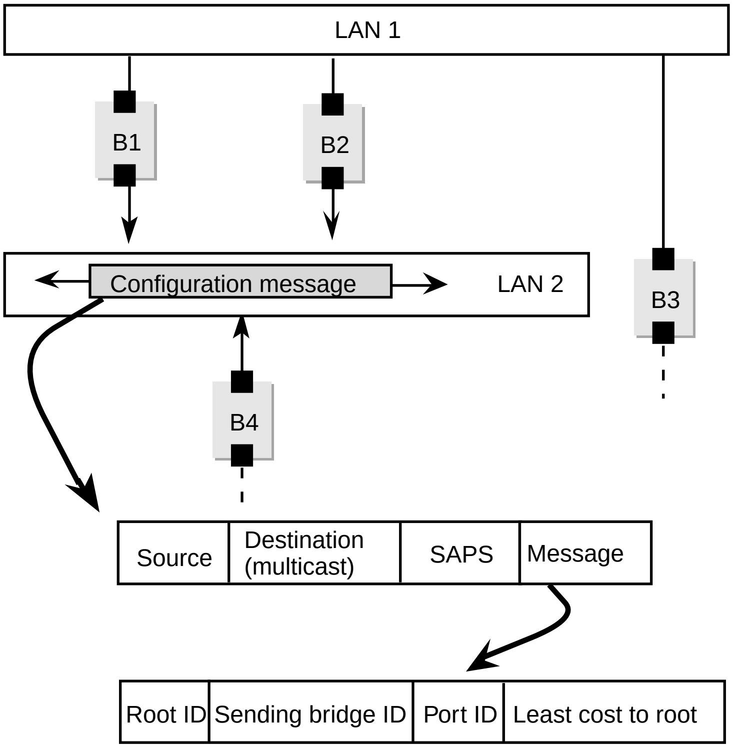


Figure 4–16 Configuration Messages

Each node that participates in the spanning tree operation stores the configuration messages sent to it. It uses these messages to determine the “best route” to various nodes in the network. The best route can be defined with any type of link state metric deemed appropriate by the network manager. Whatever this metric may be, it is conveyed in the configuration packet in the cost field, also known as the “path cost to root” field.

The idea behind the exchange of configuration messages is to select a root bridge for the network, calculate a shortest path to the root bridge, select a designated bridge for each network, and choose a root port from each node to the root bridge.

The “best configuration packet” is performed by comparing configuration messages received at each port to the messages that would be transmitted on that port.

The best route is one in which (a) the root ID is lower, then (b) the cost is numerically lower, then (c) the bridge ID is numerically lower, and then (d) the port ID is lower. In other words, the node looks first at the root ID, and if those values are equal it then looks at the cost field, and if those are equal it looks at the bridge ID, and so on down to the port ID. If this technique seems arbitrary to the reader, you are on target, for it is arbitrary—the idea is to find first an anchor point from which to measure (thus the need for finding a root bridge) and then to calculate the costs in relation to the anchor point. See Figure 4–17.

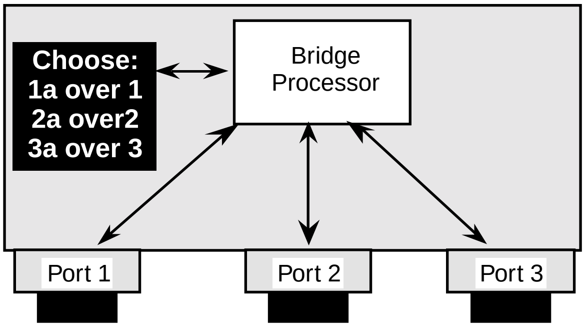
The Spanning Tree Logic

The spanning tree calculation is performed (a) when the timer for a port reaches a maximum age or (b) if a received configuration message (CM) reveals that this message contains a better path than the stored configuration message.

The timer operation is illustrated in Figure 4–18(a). When the incremented timer is equal to the maximum age (MAXage), the configuration message is discarded and the bridge recalculates the root, root path cost, and root port.

The use of a configuration message is illustrated in Figure 4–18(b). When the bridge receives a configuration message on port n, it compares this message with the stored message. Two situations will lead to a recalculation: when the received CM is better than the stored CM, or the received CM has an age field smaller than the stored CM.

Figure 4–19 provides an example of how the bridge processor determines costs and roots on its ports. The bottom part of the figure shows



"Best" Configuration Messages:

Message	Root ID	Send Bridge ID	Cost
1	31	32	4
1a	30	29	3
2	31	32	4
2a	31	29	3
3	31	32	4
3a	31	32	3

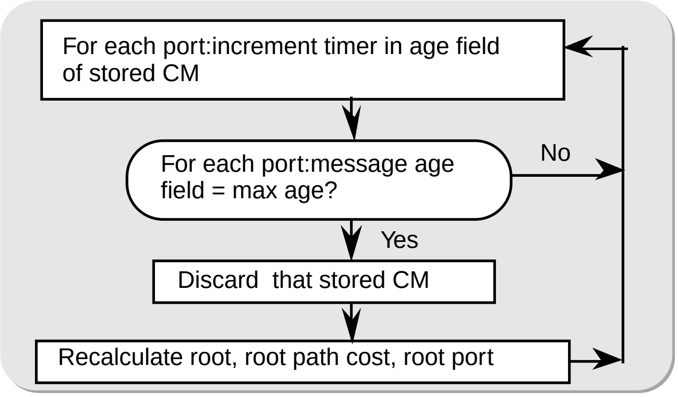
Note: Port ID can also be used as part of selection process

Figure 4–17 Saving "Best" Configuration Messages

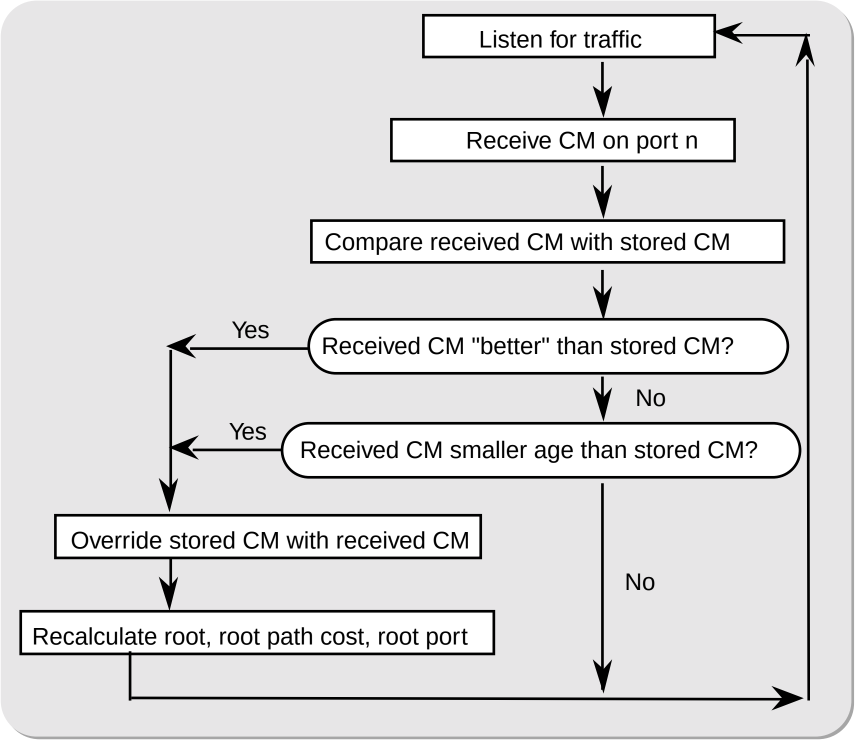
the configuration messages that have been received on ports 1, 2, and 3. The CM on port 1 contains route ID = 10, which is smaller than the route IDs of the CMs on ports 2 and 3. Therefore, the best route is route ID = 10, and the route port is port 1.

As a result to this analysis, the bridge processor will transmit CMs with route ID = 10, sending bridge = 11, and a cost = 6. The value of 6 is used since it is one greater than the cost to the route of 5.

Since the bridge processor has ID = 11, this value is smaller than the route IDs found on ports 2 and 3, consequently it is the designated bridge on these ports and it will transmit its CMs on ports 2 and 3.

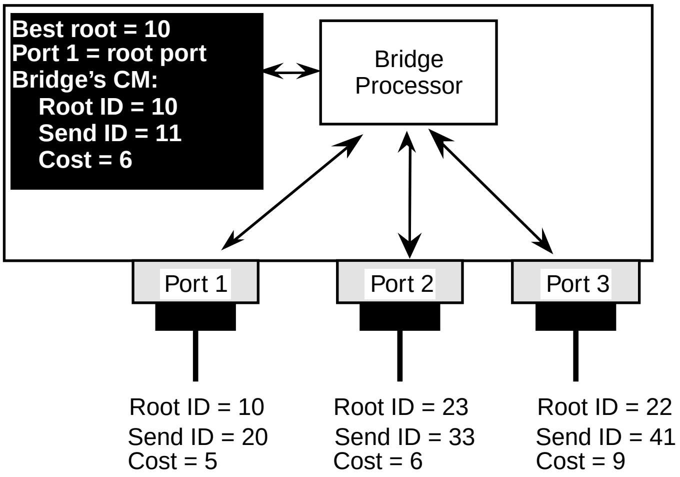

For each port:increment timer in age field of stored CM
For each port:message age field = max age?
No
Yes
Discard that stored CM
Recalculate root, root path cost, root port

(a) Max Age


Listen for traffic
Receive CM on port n
Compare received CM with stored CM
Yes
Received CM "better" than stored CM?
No
Yes
Received CM smaller age than stored CM?
Override stored CM with received CM
No
Recalculate root, root path cost, root port

(b) Configuration Message Receipt

Figure 4–18 Spanning Tree Logic



Bridge CM is "better" than CM on ports 2 and 3
Bridge is "designated bridge" on these ports
Therefore, it transmits its CMs on ports 2 and 3

Figure 4–19 Determining the Root ID, Cost to Root, and Designated Bridge to Ports

The Pruned Topology

After all the exchanges of configuration messages and the selection of the root and the designated bridge for each LAN, each bridge computes the spanning tree. Figure 4–20 shows the effect of one such operation. You will notice that several of the ports have been placed in a blocking state (signified with the dashed lines). Data cannot be sent on these ports. Other ports have been placed in a forwarding state, which permits their use for user data traffic. It is also evident that the LAN internet has full connectivity (all LANs and bridges are reachable), yet no loops exist in the topology.

Table 4–2 provides a comparison of spanning tree and source routing operations. Generally speaking, most people in the industry favor the spanning tree concept over that of source routing. This table summarizes the reasons why transparent spanning tree operations are the preferred choice.

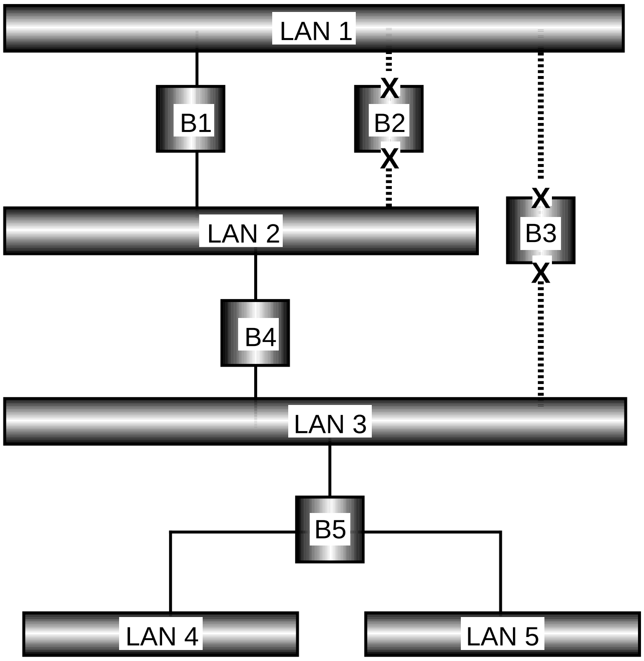


Figure 4–20 Pruned LAN Topology

Table 4–2 Spanning Tree and Source Routing

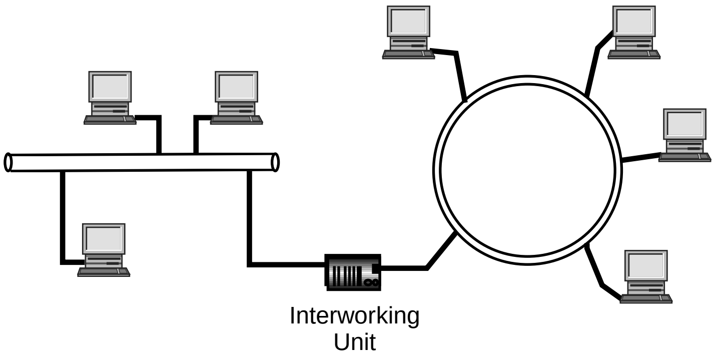
Feature	Spanning Tree	Source Routing
Routing	Usually, not optimal	Very efficient
Headers	Small	Small to very large
Configuration	Easy	Somewhat easy, if one is careful
Path discovery	Low overhead	Low to high overhead
End node responsibility	Very little	Considerable
Explicit route header	No	Yes
Frame size management	Restricted	No restriction
Performance	Fair to good	Good under transient conditions

INTERNETWORKING DIFFERENT LANS

Internetworking the same types of networks is a relatively simple operation. However, internetworking heterogeneous networks requires the IWU to assume additional and significant functions. Figure 4–21 shows the internetworking of an 802.3 LAN with an 802.5 LAN and lists some of the major differences that must be resolved by the IWU.

Internetworking different networks is not a trivial exercise. But the task can be accomplished if it is understood that an end user may not be able to achieve the full benefits of one or both of the internetworked protocols.

In most internetworking situations, the end user is given the capabilities of the network that exhibits the lower quality of service. This



802.3 to 802.5 →

- Change bit ordering
- Buffer for Speed Differences
- Frame Reformat
- Change Checksum Order
- PDU Size Matching

← 802.5 to 802.3

- Change bit ordering
- Buffer for speed differences
- Frame reformat
- Change checksum order
- "Map" status bits
- PDU size matching
- Absorb frame

Figure 4–21 Internetworking CSMA/CD and Token Ring

approach is certainly reasonable. After all, how can one expect a low function network to spontaneously raise its level of service to that of a higher quality of service network?

In any event, Figure 4–21 shows some of the major tasks that must be accomplished for internetworking 802.3 and 802.5 LANs.

Address Mapping

One problem that must be solved is the resolution between MAC addresses. This statement seems contradictory in that one would think the use of MAC addresses between two networks would obviate any type of address resolution/mapping. Unfortunately, such is not the case. While the IEEE committees have done a laudatory job setting up efficient standards for LANs, their reluctance or inability to define the exact syntax of MAC addresses for each IEEE LAN type has complicated the internetworking of the different IEEE networks. The principal problem lies in the manner in which the bits are constructed within the address field. Certain networks place the binary low-order bits in the field first, and other networks place the high-order bits in the field first. These approaches are know as the “little endian” and “big endian” syntaxes. As examples, Ethernet, 802.3, and 802.4 transmit “little-endian,” with least-significant bits first, and 802.5 and FDDI transmit “big-endian,” with most-significant bits first.

Transit Bridging

One technique to support internetworking heterogeneous LANs is known as either simple encapsulation or transit bridging. See Figure 4–22. With this approach, the router is responsible for interpreting the address of a type a network and relating that to the address of a type b network. As part of this support function, the router encapsulates the traffic of the type a network into the information field of the type b network and transports this traffic across the “transit” type b network. At the receiving router between the type a and type b networks, this router decapsulates the traffic and passes the traffic onto the type a network by placing the traffic in the type a frame.

Table 4–3 summarizes the major operations that occur in the bridging of traffic between Ethernets and token rings when translation bridging is employed. In essence, when moving traffic from a token ring to an Ethernet, the bridge must strip the routing information field (RIF), reformat the frame to the Ethernet format, and throw away the bits used by the token ring that are not used with Ethernet.

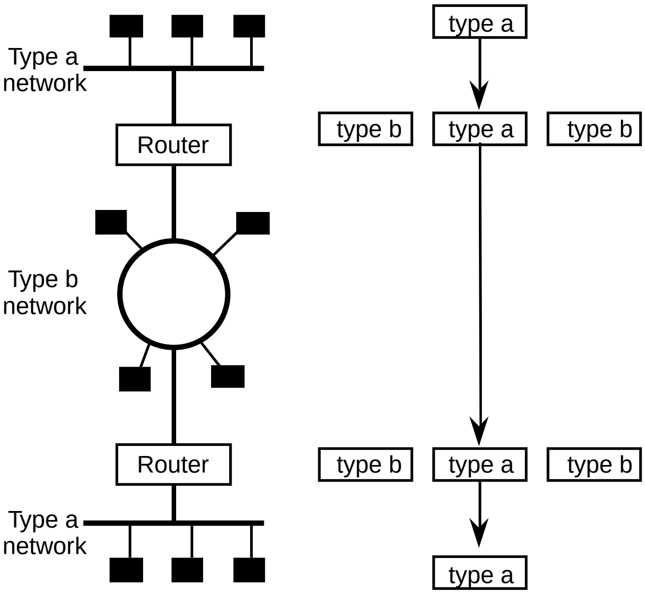


Figure 4–22 Encapsulation/Transit Bridging

Table 4–3 Techniques for Bridging Traffic Between Ethernets and Token Rings: Translation Bridging

- Token ring to Ethernet
    - Bridge caches RIF for use in sending data to source
    - Strips RIF, and reformats frame to Ethernet format
    - Throws away priority bits, token bit, monitor bit, and reservation bits
- Ethernet to token ring
    - Bridge attaches RIF (if available), else frame flooded and response to flood used for RIF
    - Inserts priority bits, token bit, monitor bit and reservation bits
    - Considerations:
        1. Loop avoidance information is not passed
        2. Does not disturb a source-route bridged network
        3. No spanning tree computations, forcing “all rings” explorer packets
        4. What about addresses in other fields (ARP, XNS, RARP)?
        5. How to handle E (error), A (address seen), and C (frame copied) bits?
            - Do nothing
            - Bridge sets C bit, but not A bit

Conversely, when relaying traffic from Ethernet to the token ring network, the bridge must attach an RIF (if this information is available) and build the token ring frame with the priority bits, the token bit, the monitor bit, and the reservation bits.

Each vendor handles translation bridging in their own fashion. Therefore, the network manager should examine how the bridge provides certain features, which are summarized at the bottom part of Table 4–3.

Source Route Transparent Bridging (SRT)

IBM has developed a technique to allow the bridging of traffic between Ethernets and token rings. IBM calls this technique source route transparent bridging (SRT). See Figure 4–23. With this approach, a bridge can support source and nonsource routing as long as the end nodes communicate with each other with the same type of operation. Therefore, an Ethernet transparent routing structure can interwork with a token ring source routing structure.

Obviously, IBM's technique must change frame formats and eliminate RIFs and other fields when traffic is relayed from a token ring to an Ethernet, and the reverse process must be accommodated when relaying the traffic from an Ethernet to a token ring.

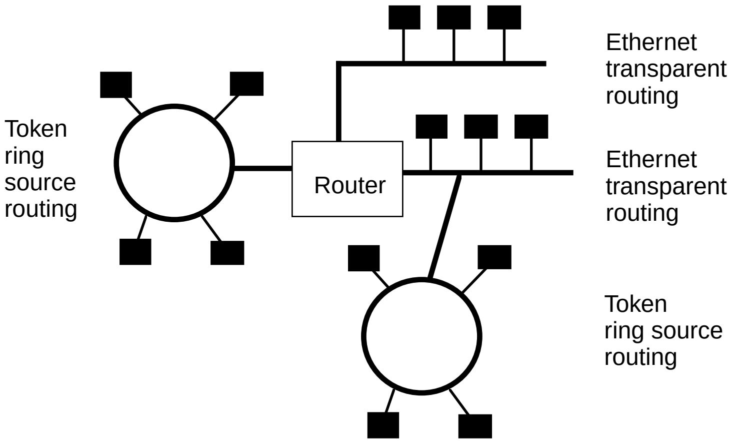

Ethernet transparent routing
Token ring source routing
Router
Ethernet transparent routing
Token ring source routing

Figure 4–23 Techniques for Bridging Traffic Between Ethernets and Token Rings: Source Route Transparent Bridging (SRT)

REMOTE BRIDGES

In many situations, it is not possible for LANs to interwork directly with bridges between LANs. Since many enterprises are widely distributed, the LANs must often be connected with wide area communications links. See Figure 4–24. These links connect LAN bridges as a point-to-point topology. Such a connection is called remote bridging. Some vendors, such as AppleTalk, refer to the bridges as half bridges in the sense that two bridges and the link are considered to be a single bridge.

Spanning tree operations can be applied to remote bridges. The point-to-point link is considered to be part of the spanning tree, and the bridges are obligated to forward traffic on that link to the other bridge.

That is the good news. The bad news is that the IEEE in its initial discussions on spanning tree bridges, did not define fully remote bridge operations. Therefore, vendors have taken it upon themselves to define procedures for two remote bridges to communicate with each other and determine if traffic is to be forwarded through the point-to-point link.

Another issue that should be considered is the fact that if a LAN is connected through bridges into WAN topologies, with rare exceptions, these WANs will not provide the broadcasting capability. Therefore, it may be necessary for disbursed LANs to have their bridges fully meshed in order for the bridges to communicate with each other. This fully meshed network, while expensive, allows each designated LAN bridge to communicate with the other dedicated LAN bridges.

As of this writing, the 802 committee is addressing the issue of adapting standardized procedures for remote bridge operations. Decisions being contemplated include:

- How one bridge on the point-to-point link decides or does not decide how to forward traffic

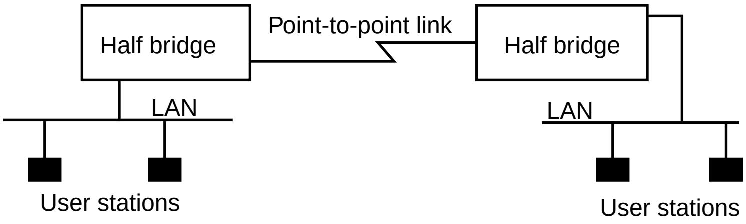


Figure 4–24 Remote Bridges

- How the traffic is represented from the standpoint of its syntax on the point-to-point link
- How bridges can communicate with other bridges using not only the point-to-point link method but a wide area switched network as well.

DATA LINK SWITCHING

As shown in Figure 4–25, routers are designed to support a wide variety of communications protocols: X.25, SDLC, frame relay, TCP/IP, DECnet, IPX, AppleTalk, XNS. It also transports SNA, APPN, and NETBIOS traffic, and functions as a multiport bridge between and among token rings and Ethernets.

Many routers also provide SDLC to LLC 2 conversion, and a technique called Data Link Switching (DLS), which is used to minimize overhead by allowing the use of SNA, APPN, and NETBIOS over the same physical link, and to transport these protocols between LANs over WANs.

SNA and NETBIOS were designed for connection-oriented operations, at least at the communications layers. They do not contain sufficient information to permit the dynamic routing and rerouting found in connectionless network protocols, such as IP, CLNP, IPX, etc.

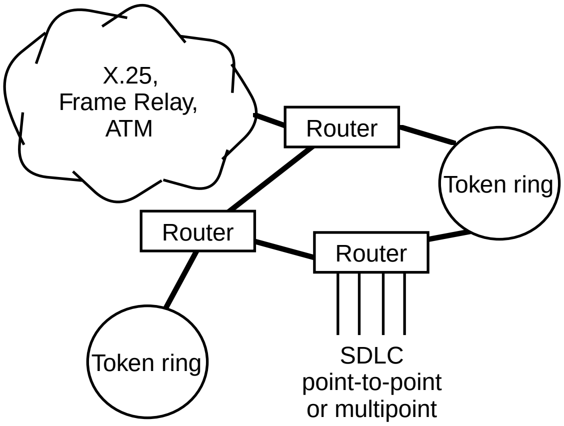


Figure 4–25 Typical Router Internetworking Topology

DLS has been developed to allow the transport of SNA and NETBIOS traffic across an internet. DLS provides the following functions: First, SNA and NETBIOS traffic is transported over a multiprotocol backbone by encapsulating this traffic into the IP data field. Reliable delivery of SNA traffic is assured, and dynamic rerouting of the traffic is provided, if necessary. LLC ACK spoofing is performed on each LAN segment, and broadcast traffic control through a WAN is also provided. DLS also supports LAN and WAN congestion and flow control operations.

DLS Configuration

Figure 4–26 shows a general configuration for DLS. The routers use spoofing (LLC termination) to minimize the impact of LLC 2 T1 timer timeouts. Spoofing also keeps the LLC2 ACKs local. DLS also terminates the IBM token ring routing information field (RIF) at the edge router, which permits the number of hops across a transport internet to be greater than the 7-hop limit that is in the RIF in some implementations. In effect, 7 hops are permitted at the local side of the WAN, and another 7 are permitted on the other side of the LAN.

The concept of a DLS circuit is also shown in this figure. It is a concatenation of the two LLC 2 sessions between the IBM devices and their respective routers, and the TCP session between the routers. This latter part of the circuit is a TPC socket between (only) the routers. This session was established when the router network was initialized.

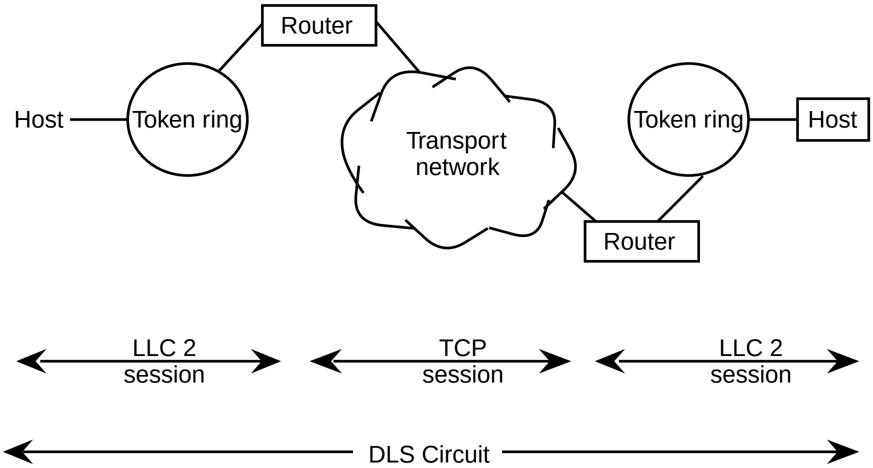


Figure 4–26 A DLS Circuit

For SDLC links, polling and poll response occurs locally, not over the WAN. Broadcast of search frames is controlled by the routers once the location of a target system is discovered. Finally, the switches can apply back pressure to the end systems to provide flow and congestion control.

The DLS Specification: RFC 1795

RFC 1795 is the recognized standard for DLS [WELL95].[2] It is a detailed specification of 91 pages, so this part of the chapter provides an overview of this RFC.

The RFC defines the operations of the switch-to-switch protocol (SSP) that is used between data link switches (DLSw); that is, the routers. It defines switching at the SNA data link layer and encapsulation in TCP/IP for transport over the Internet. It also documents the frame formats and protocols for multiplexing data between the data link switches.

The DLSw in RFC 1795 can support SNA [Physical Unit (PU) 2, PU 2.1 and PU 4] systems and optionally NetBIOS systems attached to IEEE 802.2 LLC-based LANs, as well as SNA [PU 2 (primary or secondary) and PU 2.1] systems attached to IBM SDLC links. For the latter case, the SDLC attached systems are provided with a LAN appearance within the DLSw: each SDLC protocol unit is presented to SSP as a unique MAC/SAP address pair. For the token ring LAN, the DLSw appears as a source-routing bridge.

Since the DLSw is acting as a bridge, it must support the exchange of token ring traffic, notably LLC data units. Copies of the link protocol data units (LPDU) are sent between the switches in SSP messages. Retries of the LPDU are absorbed by switch that receives it. The switch that transmits the LPDU received in an SSP message to a local data link control (LLC control) will perform retries in a manner appropriate for the local DLC. In summary, DLS handles the following token ring MAC and LLC bridging operations across the WAN internet:

- Timeouts
- Acknowledgments and retries
- Flow and congestion control
- Broadcast control of search packets
- Source route bridging hop count limits

[2] [WELL95] Wells, L, RFC 1795. “Data Link Switching: Link-to-Link Protocol,” April, 1995.

Example of DLS Operations

DLS specifies several messages for the operations between the switches. The principle messages perform the following functions, and Figure 4–27 shows the flow of these messages.

The CANUREACH, ICANREACH, and REACH_ACK message types all carry the data link ID, consisting of the MAC and LLC SAP values associated with the two end stations. The MAC and LLC identifiers are used in a token ring network to uniquely identify traffic from a host, so DLS must support the exchange of these parameters.

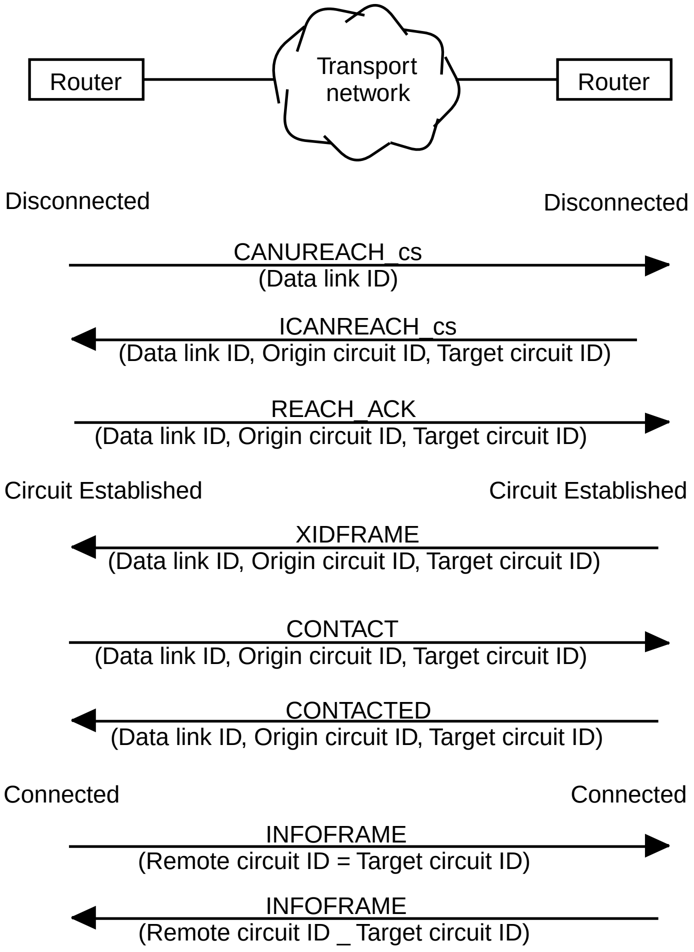

Router
Transport network
Router
Disconnected
Disconnected
CANUREACH_cs
(Data link ID)
ICANREACH_cs
(Data link ID, Origin circuit ID, Target circuit ID)
REACH_ACK
(Data link ID, Origin circuit ID, Target circuit ID)
Circuit Established
Circuit Established
XIDFRAME
(Data link ID, Origin circuit ID, Target circuit ID)
CONTACT
(Data link ID, Origin circuit ID, Target circuit ID)
CONTACTED
(Data link ID, Origin circuit ID, Target circuit ID)
Connected
Connected
INFOFRAME
(Remote circuit ID = Target circuit ID)
INFOFRAME
(Remote circuit ID _ Target circuit ID)

Figure 4–27 Example of a DLS Message Flow to Initialize the DLS Circuit

The CANUREACH and ICANREACH messages are coded as CANUREACH_ex, ICANREACH_ex (explorer messages) and CANUREACH_cs, ICANREACH_cs (circuit start messages). The CANUREACH_ex is used to find a remote MAC and LLC SAP address without establishing an SSP circuit. Upon receipt of a CANUREACH_cs message, the target DLSw starts a data link for each port, thereby obtaining a data link correlator. The purpose of the data link correlator is to provide an additional identifier for the messages and the links involved.

If the target station can be reached, an ICANREACH_cs message is returned to the originating DLSw containing a target circuit ID parameter. Upon receipt of this information, the originating DLSw starts a data link and returns the origin circuit ID to the target DLSw with the REACH_ACK message.

During the exchange of the XIDFRAME, CONTACT, and CONTACTED messages, the pair of Circuit ID parameters is included in the message exchanges. The INFOFRAME messages are then exchanged with a header that contains only the Circuit ID associated with the remote DLSw. The Remote Data Link Correlator and the Remote DLC Port ID are set equal to the Data Link Correlator and the DLC Port ID that are associated with the origin or target Data Link Switch, depending upon the direction of the packet.

How a Router Handles DLS

This part of our DLS analysis shows more examples of how the router implements DLS. The examples here are specific to IBM routers [TEAG92],[3] [KUBE92],[4] but other routers do about the same thing if they comply with RFC 1795.

As depicted in Figure 4–28, a new circuit is established by sending conventional explorer frames from a host to another host. The frame is broadcast or multicast to stations within an internet subnetwork. Each router relays the frame on to its outgoing ports. These frames reach the final destination, where they are analyzed for the “best” route.

[3] [TEAG92]. “Data Link Switching on 6611,” March 31, 1992, E. Teagarden, L. Bobbitt, G. Cox, J. Massara, Complex System Support, Dept. B19, Building 651, Research Triangle Park, NC.

[4] [KUBE92]. IBM 6611 Performance Presentation Script, October 1992, CB Kube, IBM Washington Systems Center, Dept. JLK, Building 183, Gaithersburg, MD.

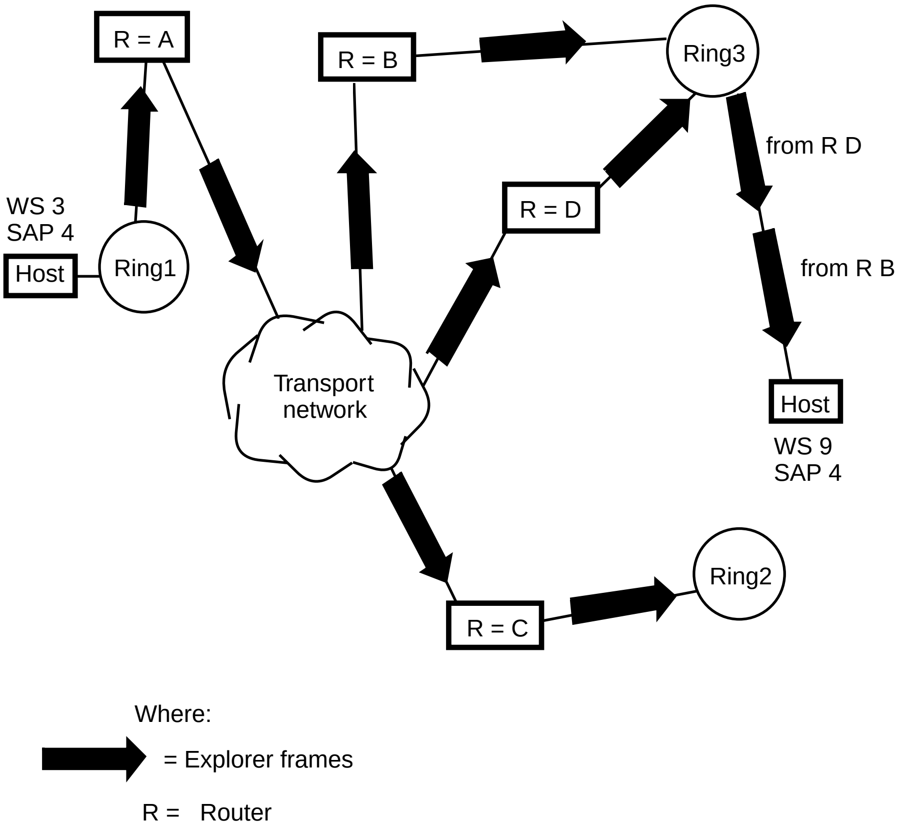


Figure 4–28 DLS Circuit Establishment

In Figure 4–28, the work station with a MAC address of 3 (work station is a host), and a SAP of 4 sends the explorer frame into the internet. The frame is received by router A, and forwarded to routers B, C, and D.

The explorer frame is intended for the workstation identified with a MAC address of 9 and a SAP of 4. This station receives the frame twice, one frame from router B and another frame from router D. Both of these frames contain the routes (in the RIF) that have been traversed from station 3 to station 9. The explorer frame is also sent to ring 2, but the workstation is not to be found there.

Figure 4–29 shows a simplified view of the explorer frames sent by routers B and D and received at work station 9. The S(3,4) identifies the MAC address (3) and SAP (4) of the sender. The D(9,4) identifies the MAC address (9) and SAP (4) of the intended receiver. The routing

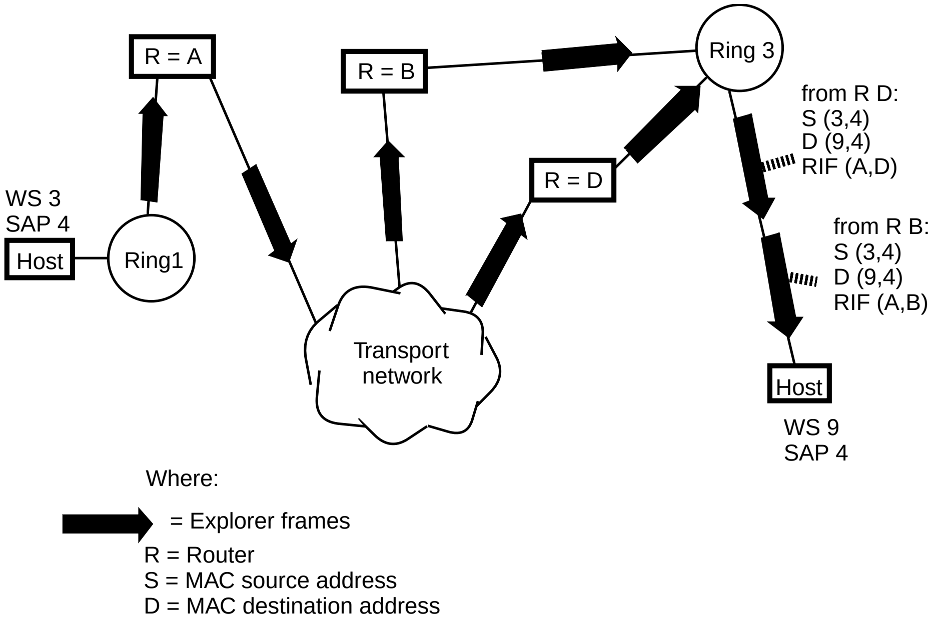


Figure 4–29 The Explorer Frames Received at Host 9, SAP 4

information field (RIF) contains a record of the route that each frame has followed in its traversal through the internet. One explorer frame records the route through router B and another records the route through router D. Although the explorer frames are sent to ring 2, this ring does not interface with station 9, and is deleted from further examples.

Station 9 does not know that these frames have been sent through a wide area transport network. It views the frames as coming from one hop beyond routers B and D. This is known as a “phantom ring segment.” The router, using DLS, uses source route bridging on its LAN ports for encapsulating the frames into the router. Then, the router encapsulates the SNA or NETBIOS traffic into TCP/IP for transport across the internet.

Station 9 must respond to the explorer frame by sending responses back to the originator. See Figure 4–30. A response is sent to router B and router D. These routers store information about station 3; it can be reached (“preferred”) through router A. This operation obviates querying each of the routers in the internet.

In Figure 4–31, router A receives the two explorer frames and determines which one is best. In this illustration, it is assumed it receives the frame from router B first, and makes this router the preferred router to reach station 9. Router D is also noted as being capable of reaching station 9.

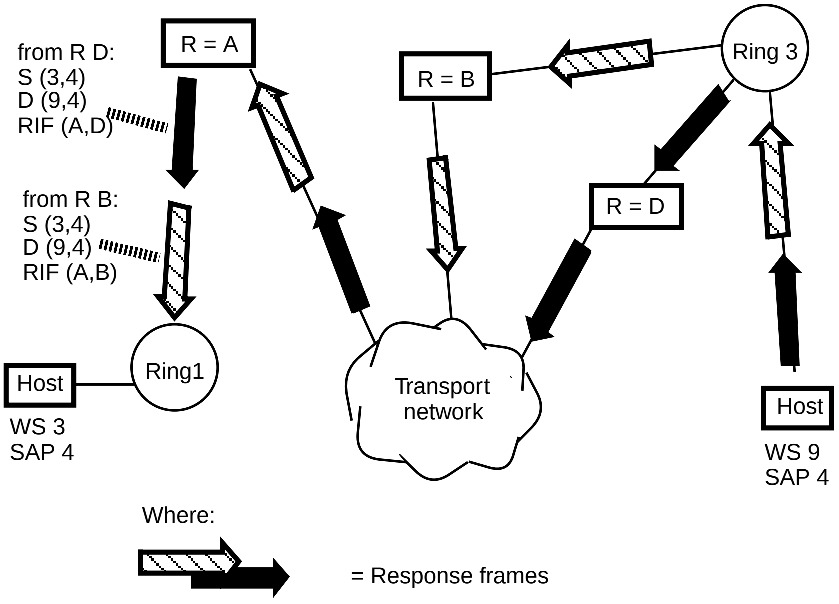
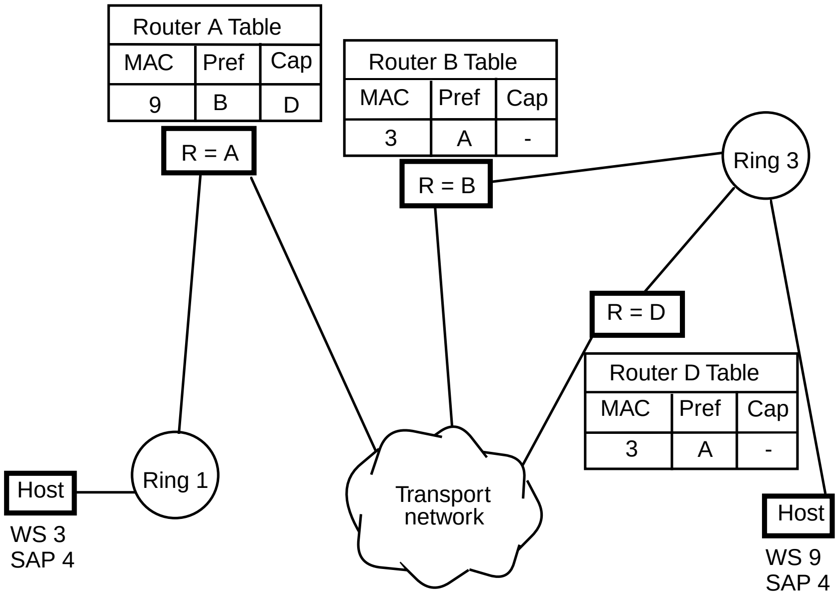


Figure 4–30 Establishment of DLS Circuit



Router A Table		
MAC	Pref	Cap
9	B	D

Router B Table		
MAC	Pref	Cap
3	A	-

Router D Table		
MAC	Pref	Cap
3	A	-

Figure 4–31 Effect of Circuit Establishment on Routing Tables

A few more thoughts are pertinent to this discussion. If another station attached to ring 1 were to send an explorer frame destined for station 9 into this internet, router B intercepts this frame, because it knows a preferred route to station 9. Consequently, it sends back a response to the sending station.

After a circuit is established, traffic is managed by the router on an individual circuit basis. Flow control is provided locally (spoofing) with the conventional receive ready (RR) and receive not ready (RNR) LLC frames on each circuit.

The routers keep records on the relationship of the LLC part of the circuit to the TCP part of the circuit. Therefore, congestion problems experienced at the routers and/or within the internet can be mapped back to the LLC part of the circuit.

In effect, the end stations can be controlled, their timers satisfied, and SNA sessions will not pull themselves down (because of non-response to transmissions).

SUMMARY

Bridges are important in data communications networks because a bridge is an efficient and cost-effective tool used to connect LANs.

LANs are limited in the distance that the media can be strung through a building or a campus of buildings. This geographical restriction can be overcome by placing a bridge between the LAN segments.

The ability to use internetworking units, such as bridges, allows the network manager to contain the amount of traffic that is sent across the expensive network media.

Data link switching is used in the token ring environment to handle topologies that use SDLC and LLC type 2. TCP serves the function of providing traffic acknowledgments between the LAN routers across the internet. Explorer frames are tunneled through the internet with the DLS protocol.

FOLLOW-UP READING

I have cited the Perlman text earlier, and I recommend it to you for excellent descriptions of bridging and other routing operations. Of course, there is no substitute for the actual standards, and the IEEE specifications have been cited earlier in the book. For readers wishing to delve into detail about bridges and how to configure bridges, consult your vendor's user manuals. If you do not have access to these manuals, I recommend a book from the Cisco IOS Reference Library titled: *Cisco IOS Bridging and IBM Network Solutions*, by Cisco Press (available from Cisco or Macmillian Technical Publishing).