Designing SRB Internetworks

This chapter discusses source-route bridging (SRB) and remote source-route bridging (RSRB). SRB is evaluated within two contexts: Systems Network Architecture (SNA) and NetBIOS.

When IBM developed SRB technology in the mid-eighties, it was viewed as a local technology that would interconnect a few rings and terminate at a remote 3745. The challenge for any SRB internetwork occurs when the scale exceeds what was originally intended by IBM. This technology encounters problems when non-IBM protocols are required to coexist with native Token Ring traffic. Source-route bridges were intended to be the primary internetworking tool for creating a corporate-wide Token Ring internetwork. These bridges were never meant to scale to the level that many customers require. This chapter addresses the challenges of this environment and aims to help network designers successfully implement SRB within a large, multiprotocol topology. This chapter is grouped into the following topics:

- SRB technology and implementation overview

- Internet Protocol (IP) routing protocol selection and implementation

- SRB network design recommendations and guidelines

Note   For information concerning IBM serial line connections, refer to Appendix B, "IBM Serial Link Implementation Notes."

SRB Technology Overview and Implementation Issues

The following discussions address SRB-related technology, features provided to support SRB requirements, and implementation issues that can affect large-scale, router-based SRB networks. Specific topics include the following:

- Typical SRB Environments

- Multiport Bridging

- Explorer Packets and Propagation

- NetBIOS Broadcast Handling

- LAN Framing

- WAN Framing

- WAN Parallelism

- WAN Frame Sizes

- SNA Host Configuration Considerations for SRB

> **Note** If you have eight or fewer routers operating as SRBs, you can skip this chapter. You probably do not need to tune your network.

Typical SRB Environments

SRB is used in three types of user environments:

- *Many end stations to few end stations (hierarchical)*—In a hierarchical SNA network, end users from multiple access sites need connectivity to a host site through a limited number of front-end processors (FEPs).

- *Many end stations to several end stations (distributed)*—Many users need to access a limited number of servers or a limited number of devices, such as an AS/400.

- *Any-to-any (flat)*—End users at one site need to access end stations at another site.

The following discussions evaluate SRB environment design issues in relation to these user environments.

Multiport Bridging

The fundamental design of an SRB, as initially created by IBM, was a two-port, ring-to-bridge-to-ring combination. IBM also created a half-bridge configuration that consisted of a ring-to-wide-area-network (WAN) combination followed by a second WAN-to-ring half-bridge combination.

To support more than two rings, multiport routers adopt an implementation that allows SRBs to include multiple rings on a single internetworking node. This is accomplished via the *virtual ring* capability. A virtual ring is a conceptual entity that connects two or more physical rings together, locally or remotely.

Figure 4-1 illustrates the concept of multiport bridges and a virtual ring.

Figure 4-1   Multiport bridge using virtual ring concept to permit multiple ring interconnection.
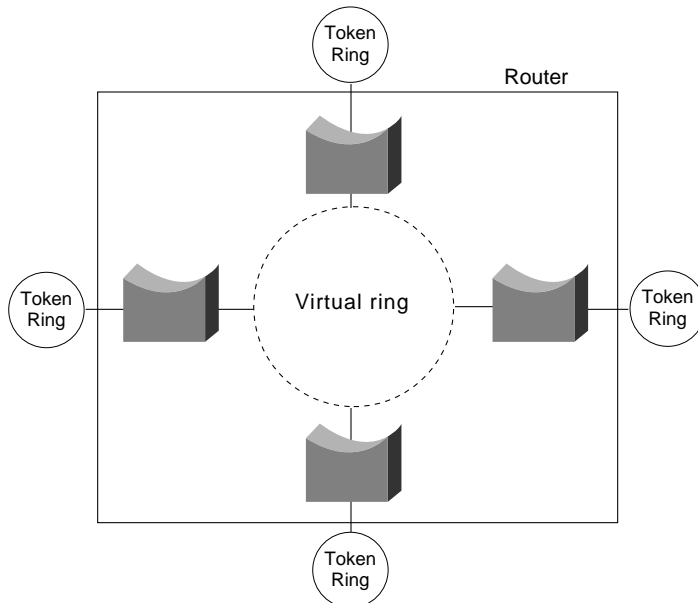
The concept of virtual rings can be expanded across router boundaries. A large virtual ring can connect several access points to a central router with an FEP. Figure 4-2 illustrates this expansion.

**Figure 4-2   Virtual rings expanded across an IP cloud.**
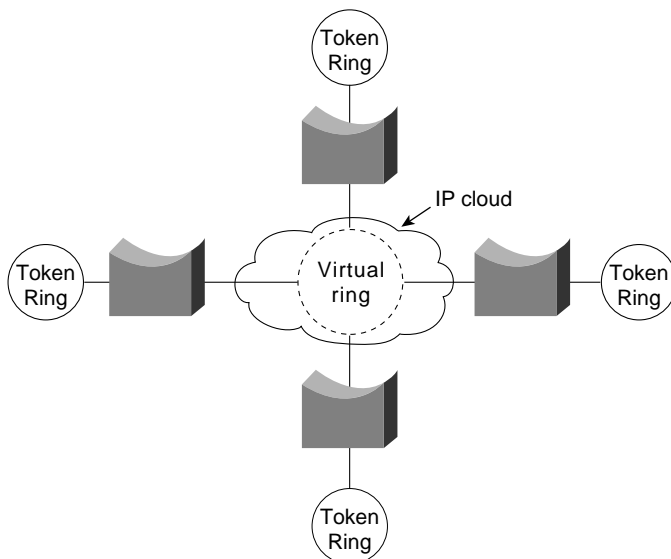
Routers support simple bridging, multiport bridging, and connections to both local and remote virtual rings. A virtual ring configuration is required to communicate with remote rings. The half-bridge configuration is not supported. The IBM half bridge does not use the concept of virtual rings; two IBM half bridges use two rings. The virtual ring advantage is in a topology that features many SRBs. In such an arrangement, only a single unit is required at a central site.

Remote virtual rings have a property not found in physical ring topologies: The logical connectivity is determined by the network administrator. Two options are available: partially meshed topologies (sometimes called *redundant star topologies*) or fully meshed topologies. In a partially meshed topology, a single central location (such as an FEP Token Ring) is connected to all access locations. Each access location is logically connected to the central FEP rings and is not connected to any other ring. Partially meshed topologies using virtual rings do not permit *direct* communication between remote rings. However, communication is allowed from the central ring to the remote rings, which also allows communication among remote rings through the central ring.

In a fully meshed virtual ring topology, any ring can communicate with any other ring. Figure 4-3 and Figure 4-4 illustrate partially meshed and fully meshed topologies. In the partially meshed topology depicted in Figure 4-3, all rings are logically bridged to Token Ring 10. The access rings are not bridged together. In the fully meshed topology illustrated in Figure 4-4, all rings are bridged to all other rings.

**Figure 4-3   Typical hierarchical topology.**
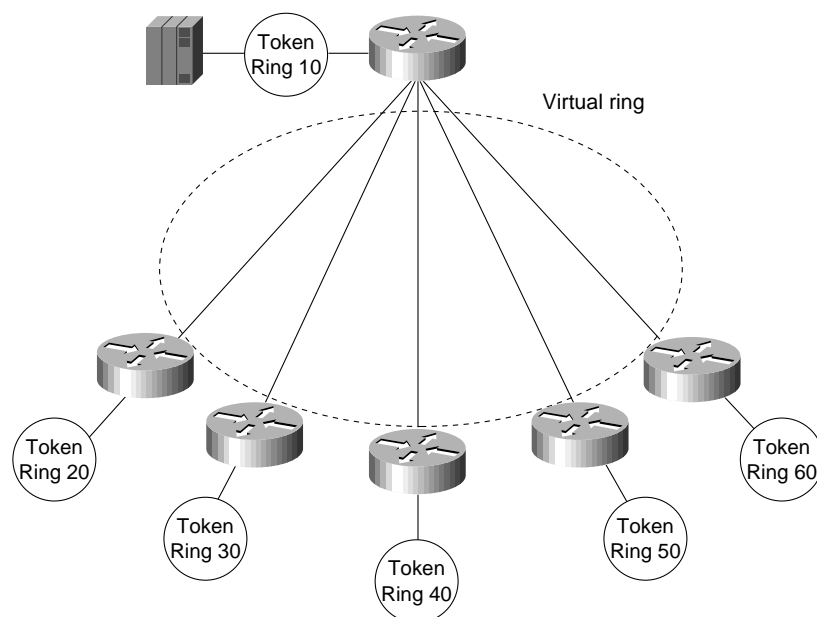
In the topology illustrated in Figure 4-3, each of the access routers is a peer to the FEP router. They are not peers to one another. Thus, SRB is enabled between all rings and Token Ring 10 and is not enabled between token rings 20, 30, 40, 50 and 60.

Assuming this is only a hierarchical SNA environment, users connected to these rings do not have SRB connectivity. Broadcasts are not forwarded across the lower layer rings (token rings 20 through 60); broadcasts are sent only from Token Ring 10 to or from the other rings.

In the topology illustrated in Figure 4-4, each router is a peer to each other router. All rings are logically bridged to all other rings. The actual physical topology is less important than the logical topology. In Figure 4-4, the same logical topology can exist even if there are no physical connections between the access routers.

Figure 4-4 Typical fully meshed (flat) topology.
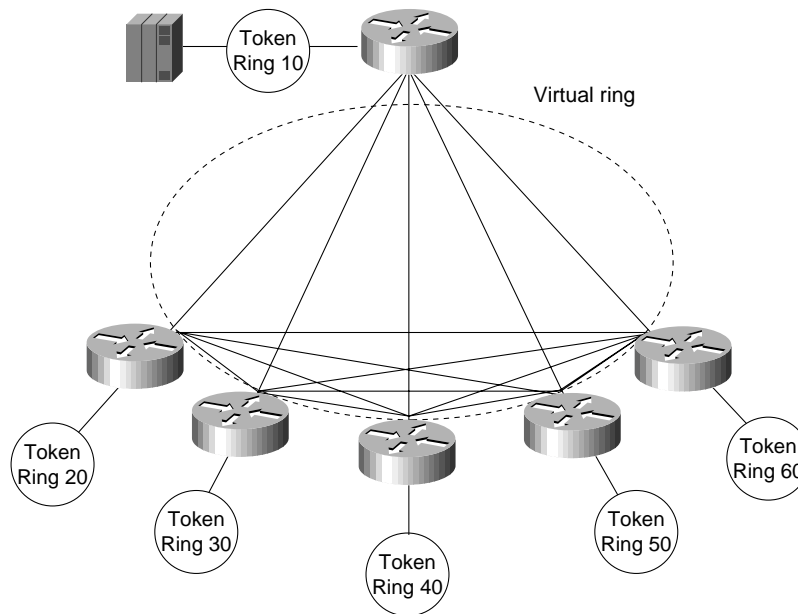
Explorer Packets and Propagation

Once you build a network of ring and bridge combinations, you must have a method for the end stations to find other end stations in the network.

An IBM bridge uses a system of *explorer packet marking* to propagate routing information through an SRB internetwork. The explorer packet is produced by the source end station and is marked (updated) by each bridge that it traverses. The marked field is called the Routing Information Field (RIF). Two important transactions occur in the explorer packet handling exchange: the transmission of the explorer packet and the reply by the end station to the explorer packets that it receives.

In this environment, the source end stations must know the Token Ring Media Access Control (MAC) address of the destination end stations. Once the MAC address is understood, the source end station produces an explorer packet.

The source-route bridge updates the explorer packet to include its bridge-ring combination in the explorer packet's RIF in the MAC frame. By accumulating this information, the explorer packet gathers a hop-by-hop description of a path through the SRB network. In addition, the bridge forwards the explorer to each destination ring it encounters, therefore creating a complete topological map for each end station trying to find its way through the network.

Explorer Packet Types

There are three types of explorer packets: *local explorer packets*, *spanning explorer packets*, and *all-routes explorer packets*. Note that all-routes explorer packets are also known as *all-rings explorer packet*s, and spanning explorer packets are also known as *single-route* and *limited-route explorer packets*. Single router explorers are explorers that pass through a predetermined path constructed by a spanning tree algorithm in the bridge. A station should receive only one single router explorer from the network.

A local explorer packet is generated by some end systems (either NetBIOS or SNA) to find a host connected to the local ring. After this event has occurred without finding a local host, the end station produces either a spanning explorer or an all-routes explorer packet. This behavior depends on the type of end station. SNA end stations generally produce an all-routes explorer packet. NetBIOS end stations produce a spanning explorer packet.

Note As of Cisco IOS Software Release 10.2, auto spanning tree (AST) for SRB is supported. The implementation of AST in Cisco IOS Software Release10.2 is based on the IEEE 802.1 standard and is fully compatible with IBM PC bridging. New global and interface configuration commands are required to configure a router for AST. Once configured, AST can be enabled and disabled through LAN Network Manager (LNM). The following discussion of spanning tree explorer packets applies to the manual spanning tree functionality available in software releases prior to Cisco IOS Software Release 10.2.

To pass a spanning explorer packet on a router, the configuration for the router's Token Ring interface must have the **source-bridge spanning** interface configuration command for the specific ring. If this interface command is not included, spanning explorer packets are discarded.

In contrast, an all-routes explorer packet can find any valid SRB ring. No specific router configuration other than specification of SRB is required to pass all-routes explorer packets.

Explorer packet processing works as illustrated in Figure 4-5. If End station X sends an all-routes explorer packet, Bridge B1 and Bridge B2 both forward the explorer packet. End station Y receives two all-routes explorer packets in this configuration. End station Y responds to each of the all-routes explorer packets by sending a directed, nonbroadcast packet. In the example illustrated in Figure 4-5, four packets are generated:

- Two all-routes explorer packets inbound (to End station Y)

- Two nonbroadcast packets outbound (from End station Y)

Figure 4-5   Explorer packet processing (all-routes broadcast).
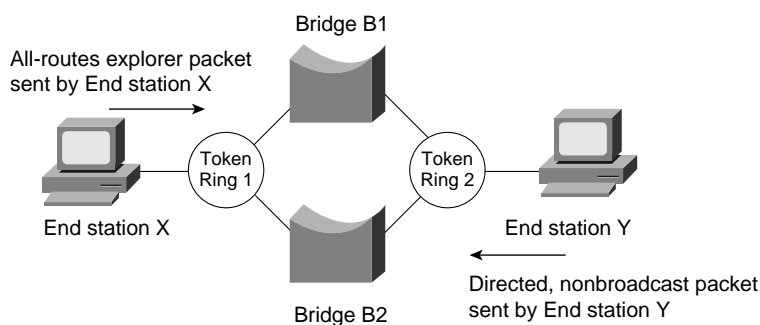

Figure 4-6 illustrates an end station sending a spanning explorer packet. Bridge B1 and Bridge B2 make their respective forwarding decisions based on whether or not spanning is enabled. Assume Bridge B1 has spanning enabled and Bridge B2 does not have spanning enabled. Bridge B1 forwards the spanning explorer packet, and Bridge B2 does not. End station Y receives one spanning explorer packet and returns an all-routes explorer packet for each single route received. As before, Bridge B1 and Bridge B2 forward the all-routes explorer packet. In this example, the following packets are generated:

- One spanning explorer packet inbound (to End station Y)

- Two all-routes explorer packets outbound (to End station X)

Figure 4-6   Explorer packet processing (spanning explorer broadcast).
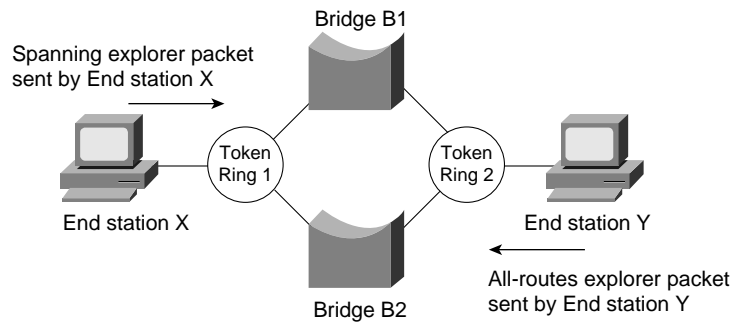
If spanning were enabled on Bridge B2, it would also forward the spanning explorer packet. The following packets would be generated:

- Two spanning explorer packets inbound (to End station Y)

- Four all-routes explorer packets outbound (to End station X)

Note   In general, there should be only a single path through the network for spanning explorer packets. If redundancy is required, a trade-off should be made between automatic redundancy and tolerance for additional explorer packet traffic. When redundancy is required, AST should be used.

Redundancy can be achieved in many instances within the router-based cloud as a result of encapsulation in either TCP or IP, the latter called *Fast Sequenced Transport* (FST). To contrast redundancy provided by a pure SRB environment and an internetwork that combines routing capabilities with SRBs, consider the networks illustrated in Figure 4-7, Figure 4-8, and Figure 4-9. Figure 4-7 illustrates a pure bridged network. Figure 4-8 and Figure 4-9 illustrate an SRB network running over routers.

Figure 4-7　　　**Redundancy in a pure SRB network.**
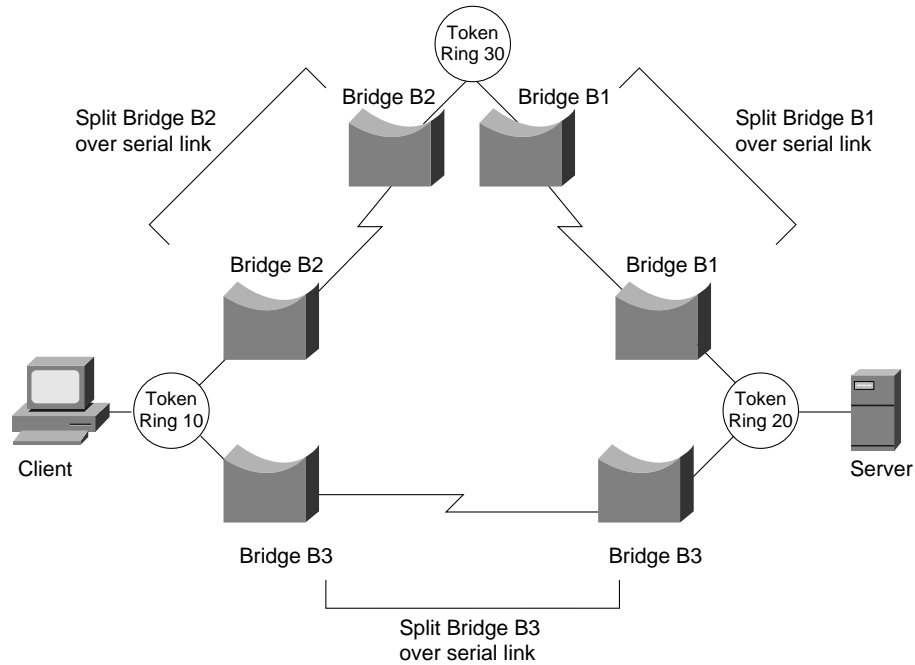
In Figure 4-7, there are two SRB paths between Token Ring 10 and Token Ring 20:

- Token Ring 10 to split Bridge B3 to Token Ring 20
- Token Ring 10 to split Bridge B2 to Token Ring 30 to split Bridge B1 to Token Ring 20

If spanning is enabled on both paths, the traffic resulting from a spanning explorer broadcast from the server is as follows:

- Two spanning explorer packets inbound (to the server)
- Four all-routes explorer packets outbound (to the client)

In router-based networks, the same type of redundancy is achieved in a different, more efficient manner, as illustrated in Figure 4-8 and Figure 4-9.

Figure 4-8　　　**Redundancy in a router-based SRB network (physical router connectivity).**
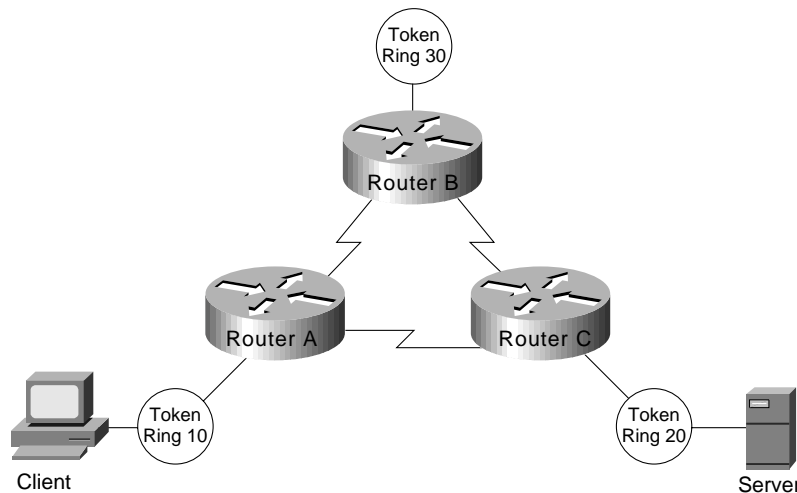
Figure 4-9   Redundancy in a router-based SRB network (logical SRB connectivity).
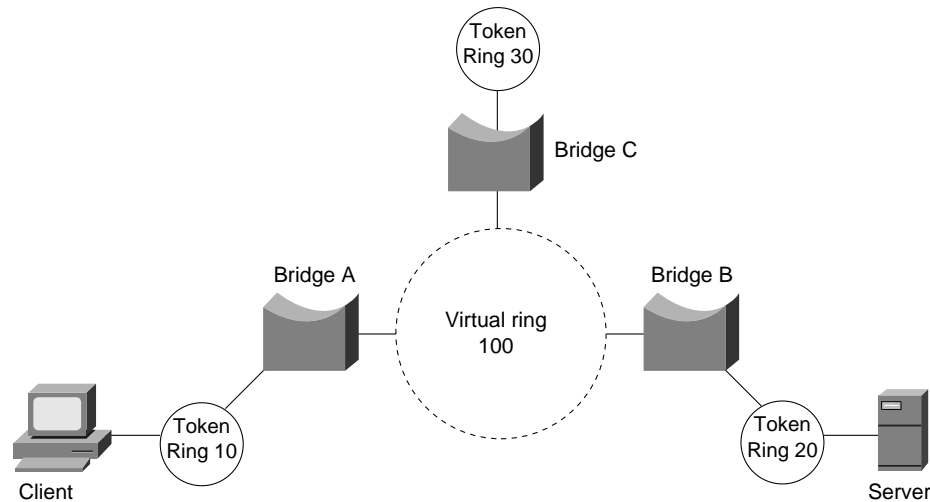
In Figure 4-9, there is only one SRB path between Token Ring 10 and Token Ring 20. The path is Token Ring 10 to Bridge A to Virtual ring 100 to Bridge B to Token Ring 20. When the client sends a spanning explorer packet, the following occurs:

- One spanning explorer packet goes inbound (to the server).

- Two all-routes broadcasts go outbound—one to the client on Token Ring 10 and one to Token Ring 30.

These broadcast rules are valid even when spanning is enabled on all the routers. In this example, spanning does not affect the traffic. The redundancy is a result of router-to-router traffic handling.

Each explorer packet is modified and copied at each destination ring when a multiring bridge is connected to more than two rings or to a virtual ring with multiple remote destinations. The virtual ring in these cases operates indistinguishably from a physical ring. The RIFs are modified exactly as if the virtual ring were a physical ring. All source-route bridges are designed to forward packets, so frame copying can be a limiting factor in both large-scale bridges and topologies with many token rings. In these topologies, your most important job as a network designer is to prevent excessive forwarding of explorer packets, which can disable an entire network.

Most source-route bridges do not propagate an explorer packet onto a ring from which it has just arrived. As a result, explorer packets are not copied from a virtual ring back to the same virtual ring, even in the presence of valid remote peers.

Figure 4-10 illustrates a situation in which incoming explorer packets arriving on virtual ring 1A are transmitted to Bridge 1 but are not copied back to Virtual ring 1A, even in the presence of multiple remote peer statements pointing to Virtual ring 1A. This is desirable behavior. Bridge 2 does not forward frames that originated from Bridge 1 because the frame has been on Virtual ring 1A.

Figure 4-10   Virtual ring and explorer packet behavior.
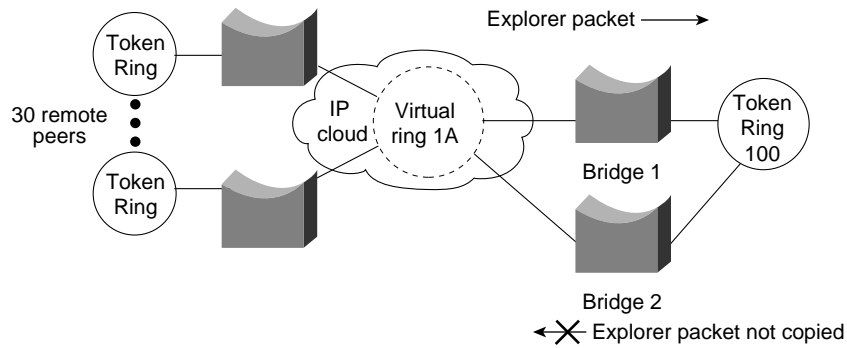
In contrast, Figure 4-11 illustrates a topology that can result in a storm of explorer packets. In this topology, two virtual rings are separated by physical Token Ring 2.

Figure 4-11   Virtual ring topology resulting in explorer packet storms.
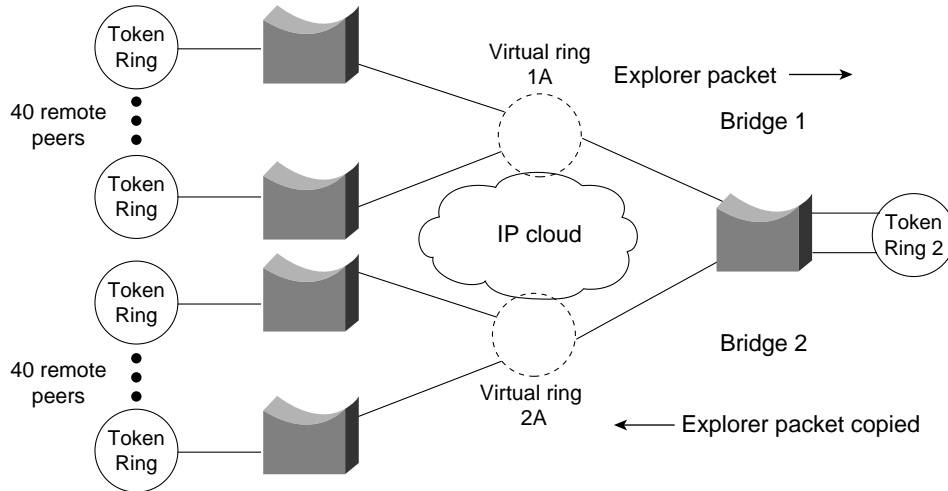
An incoming explorer packet arriving on Virtual ring 1A is propagated to physical Token Ring 2 through Bridge 1. This explorer packet is then propagated into Bridge 2 and copied 40 times for each remote peer statement. Because the SRB protocol does not scale effectively, it results in this kind of explorer packet explosion that causes performance problems in Token Ring environments. The bridge must modify and copy the explorer packet in the CPU, causing inefficient use of the CPU and system bus for copying and modifying each explorer packet bound for a new destination.

You can reduce the number of forwarded explorer packets by enabling the explorer packet processing queue. The queue is used to divide traffic into data frames and explorer packets, as illustrated in Figure 4-12.

Figure 4-12   Queuing process resulting in the division of frames between real data and explorer packets.

Token Ring

Output queue

Explorer packets

Real data packets
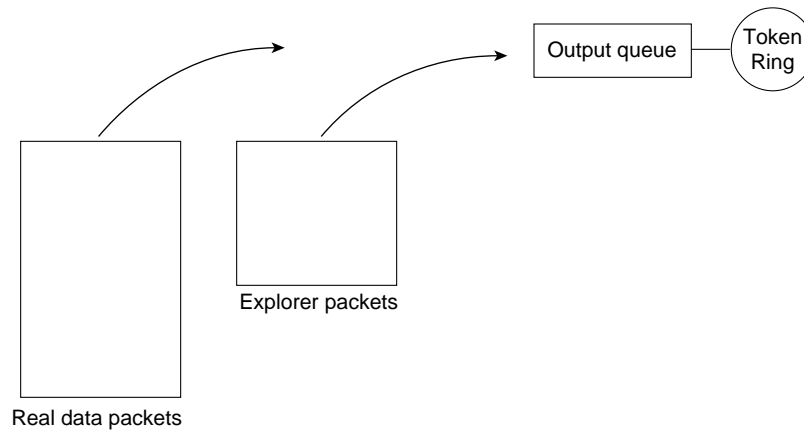
Reduce the number of forwarded explorer packets and improve overall efficiency by allowing the CPU to spend more cycles transmitting frames for routing and bridging and less time copying, modifying, and forwarding explorer packets. To enable the explorer packet processing queue, use the following global configuration command (available with Software Release 9.1.8(5) and subsequent releases):

source-bridge explorerq-depth *number*

The value of *number* specifies the queue depth. The default value of *number* is 30 queue entries. The disadvantage of enabling the explorer packet processing queue is the potential for suboptimal paths. For most SRB networks that are plagued by excessive explorer packet, this potential is an acceptable trade-off.

Limiting the copying of explorer packets is an important factor in designing SRB networks. Poorly designed SRB networks can collapse under high explorer packet copying loads and the resulting volume of explorer packet traffic. Although good internetwork design, such as a single unified virtual ring, can eliminate large-scale explorer packet copying, this solution does not scale infinitely. For very large internetworks, contact your technical support representative for more information about specific limitations. Also, refer to the section "SRB Network Design" later in this chapter for more information about how different topologies scale.

Proxy Explorer

Another way of limiting explorer packet traffic is to use the *proxy explorer* feature. The function of the *proxy explorer* feature is to create an explorer packet reply cache, the entries of which are reused when subsequent explorer packets need to find the same host. The proxy explorer feature allows the SRB network designer to minimize exploding explorer packet traffic throughout the network. Routers cache the explorer packet reply and reuse it for subsequent explorer packets that are searching for the same MAC address.

Proxy explorer functionality is very useful in traditional SNA configurations because most explorer packets are destined for a single FEP on a single ring. However, if the host to be reached is an FEP on two rings (with a single locally administered address duplicated on both rings), this feature will select a single path without the possibility of redundant paths from a single router. Different routers can use different paths.

If your configuration does not involve duplicate FEPs with the same locally administered address, you can use the proxy explorer function in any SNA environment. Use the following interface configuration command:

source-bridge proxy-explorer

NetBIOS Broadcast Handling

NetBIOS stations issue broadcasts for several reasons: to verify at startup that a station name is unique to the network, to find the route to a particular server, and to provide a heartbeat function to maintain connectivity between servers and requesters. These broadcasts are addressed either to a specific name, or to the NetBIOS functional address (such as C000 0000 0080). Station requests, such as a NAME QUERY frame, are sent as a spanning explorer broadcast to a unique NetBIOS name, and the corresponding response is returned as a broadcast of all-routes explorer packets.

NetBIOS is a broadcast-intensive protocol that can quickly consume lower bandwidth bridge paths. To address this problem, the router provides four different methods of preventing single and all-routes broadcast traffic from consuming your network:

- NetBIOS Name Caching
- NetBIOS Datagram Broadcast Handling
- NetBIOS Broadcast Throttling
- NetBIOS Broadcast Damping

NetBIOS Name Caching

NetBIOS name caching allows the router to maintain a cache of NetBIOS names that it uses to avoid the high overhead of transmitting many of the broadcasts used between client and server PCs in an SRB environment.

Name caching allows the router to detect when any host sends a series of duplicate query frames and to limit the host to one frame per configurable time period. The name cache includes a cache of mappings between NetBIOS server and client names and their MAC addresses. The name cache allows the router to send broadcast requests from clients to find servers and from servers in response to their clients directly to their destinations. It does this rather than sending the broadcast across the entire bridged network.

In most cases, the NetBIOS name cache is best used in situations in which large amounts of broadcast traffic creates bottlenecks on the WAN media. However, the traffic savings of NetBIOS name caching is probably not worth the router processor overhead when two local-area network (LAN) segments are interconnected.

As NetBIOS broadcasts traverse the router, the router caches the NetBIOS name in the NAME-QUERY and NAME-RECOGNIZED broadcast frames along with the station MAC address, RIF, and the physical port from which the broadcast was received. Because the router has the NetBIOS name as well as the route to the station, it can respond locally to broadcasts and eliminate the overhead of propagating broadcast frames throughout the network.

NetBIOS name caching can be enabled on each interface by using the following interface configuration commands:

source-bridge proxy-explorer

netbios enable-name-cache

The **source-bridge proxy-explorer** command is a prerequisite for NetBIOS name caching. To limit proxy-explorer to NetBIOS only, use the following configuration command:

source-bridge proxy-netbios-only

NetBIOS Name Caching Operation

Figure 4-13 illustrates the NetBIOS name-caching process. Workstation A issues a NAME-QUERY frame looking for Server C. The single-route broadcast is propagated to all rings in the network and Server C responds with a NAME-RECOGNIZED response as a broadcast of all-routes explorer packets. The all-routes broadcast propagates throughout the network, and generates two duplicate NAME-RECOGNIZED responses to Workstation A, each with different routes reflected in the MAC header. Workstation A and Server C are now cached in routers 1, 2, and 3.

Figure 4-13   NetBIOS name-caching process.
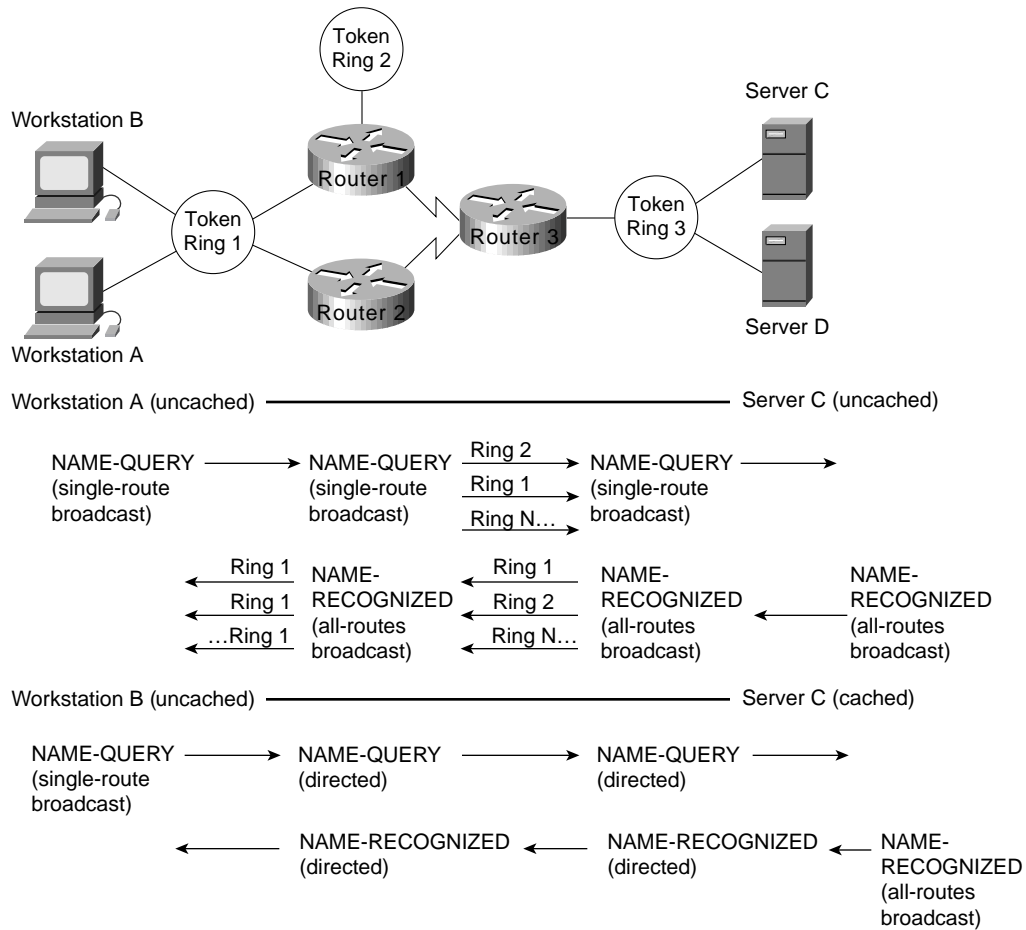
Workstation B now broadcasts a NAME-QUERY frame also looking for Server C. The broadcast is received by Router 1, which finds Server C in its cache. To verify that Server C and the cached route are still available, the router converts the broadcast frame to a directed frame using the cached RIF information, forwards the NAME-QUERY frame, and starts the RIF validate-age timer. When Server C receives the NAME-QUERY frame, it responds with a NAME-RECOGNIZED (all-routes) broadcast. If the router receives Server C's response before the validate-age timer expires, it keeps the RIF information; if not, the router deletes the RIF information from the cache.

Router 3 copies the NAME-RECOGNIZED broadcast and checks its cache for Workstation B. If an entry exists, the all-routes broadcast is converted to a directed frame and is forwarded to Workstation B. This example demonstrates that once a station name is broadcast into the network and its name is cached, no further broadcasts traverse the network. Without name caching, the

broadcast activity in a network with 100 fully meshed ring segments can become a serious issue. NetBIOS name caching significantly reduces the bandwidth consumed by nonproductive broadcast traffic.

Each NetBIOS name cache entry is aged out of the table if activity does not occur within a configurable period of time. Aging ensures the information in the cache is current and that the cache is kept to a minimum size to maintain optimal performance.

The following global configuration command controls the name-caching age timer:

> **netbios name-cache timeout** *minutes*

The default is 15 minutes.

NetBIOS Datagram Broadcast Handling

The router also checks the NetBIOS name cache when it receives NetBIOS datagram broadcasts (addressed to unique names), which allows the router to handle NetBIOS datagram broadcasts locally in a way that is similar to NAME-QUERY and NAME-RECOGNIZED broadcast handling. The difference is that datagram broadcasts are generally one-way flows with no corresponding reply. If datagram broadcasts represent a small percentage of overall broadcast traffic, you can disable datagram handling and avoid expending additional router overhead for relatively minor effect. This decision can be made only with an understanding of your broadcast traffic patterns.

NetBIOS Broadcast Throttling

NetBIOS applications broadcast by issuing multiple successive copies of the broadcast frame into the network. For example, IBM's OS/2 LAN Requester sends six successive copies of a NAME-QUERY frame, with a pause of a half second between each repeated transmission. Some applications allow you to tune this behavior, but tuning NetBIOS broadcasts is difficult to maintain if the number of NetBIOS workstations in your network is high.

As illustrated in Figure 4-14, when NetBIOS name caching is enabled, the router forwards the first of these six broadcasts, and drops the duplicate five broadcasts. The duplicate broadcasts (which originated from the same station), continue to be dropped until the dead timer expires. Two global configuration commands control relevant timers:
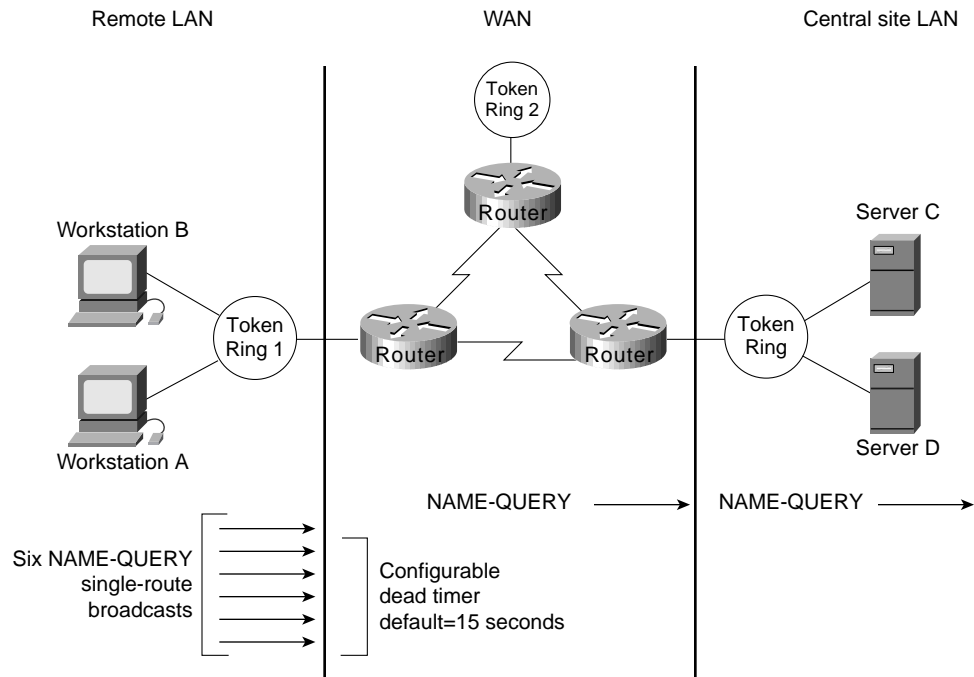
> **netbios name-cache query-timeout** *seconds*

The default is 6 seconds.

> **netbios name-cache recognized-timeout** *seconds*

The default is 1 second.

Figure 4-14   Throttling NetBIOS broadcasts.

NetBIOS Broadcast Damping

The router remembers the physical port from which a NetBIOS station's route was cached. As a result, the router can remember where a cached station resides relative to the router. If the router receives a broadcast frame that is addressed to a cached NetBIOS name and if the router knows that the route to that station exists off of the same interface, the router does not need to forward the broadcast to find the target station. Instead, the router drops the broadcast and prevents unnecessary broadcast traffic from traversing the network.

As illustrated in Figure 4-15, a NetBIOS broadcast addressed to Server D is received by Router 1 on interface T0. Router 1 finds a cached entry for Server D, which indicates that the route to Server D is via interface T0. Because the broadcast was received on T0 and because the route to Server D is via T0, the broadcast is prevented from continuing on the network, and the requester finds Server D via the local SRB topology.

Figure 4-15   NetBIOS broadcast damping.
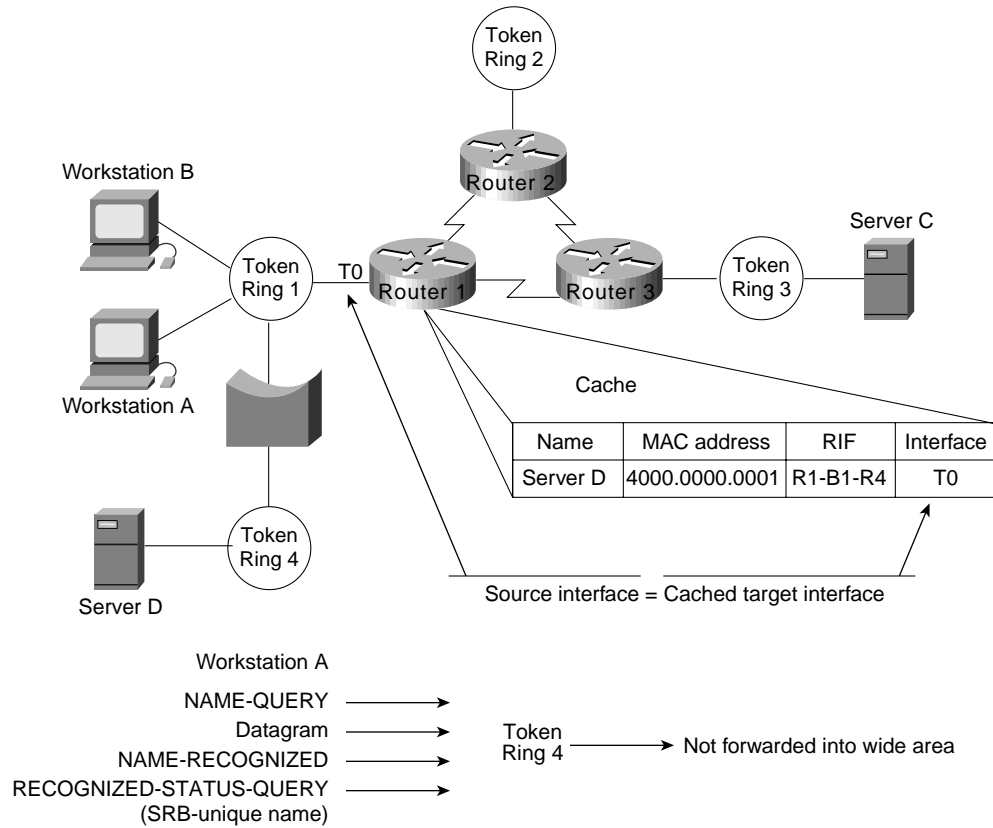
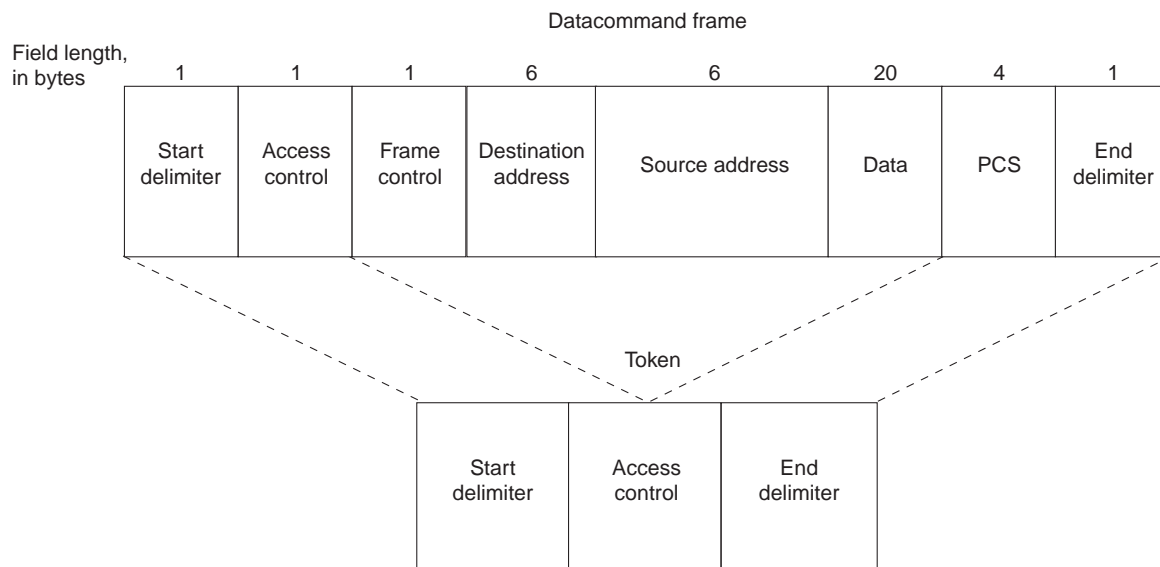## LAN Framing

Framing for SRB networks is straightforward. Using a basic IEEE 802.5 frame with Logical Link Control type 2 (LLC2) 802.2 framing, a RIF field follows the source MAC field in the IEEE 802.5 frame. The presence of a RIF field is indicated by setting the Routing Information Identifier (RII), which is the high-order bit of the source MAC field, as shown in Figure 4-16.

Figure 4-16   IEEE 802.5/Token Ring frame formats.

Datacommand frame

Field length,
in bytes

1	1	1	6	6	20	4	1
Start delimiter	Access control	Frame control	Destination address	Source address	Data	PCS	End delimiter

Token

Start delimiter	Access control	End delimiter

An SRB-configured router evaluates incoming frames based on IEEE 802.5 values, which is mostly a Subnetwork Access Protocol (SNAP) evaluation. Once the router determines whether the packet is to be routed, it evaluates whether to use SRB based on the value of the RII bit. If the bit is set and the router is not configured to route a specific protocol, the router sends the frame using SRB. Figure 4-17 illustrates this process.

Figure 4-17   Process for identifying routable versus SRB packets.
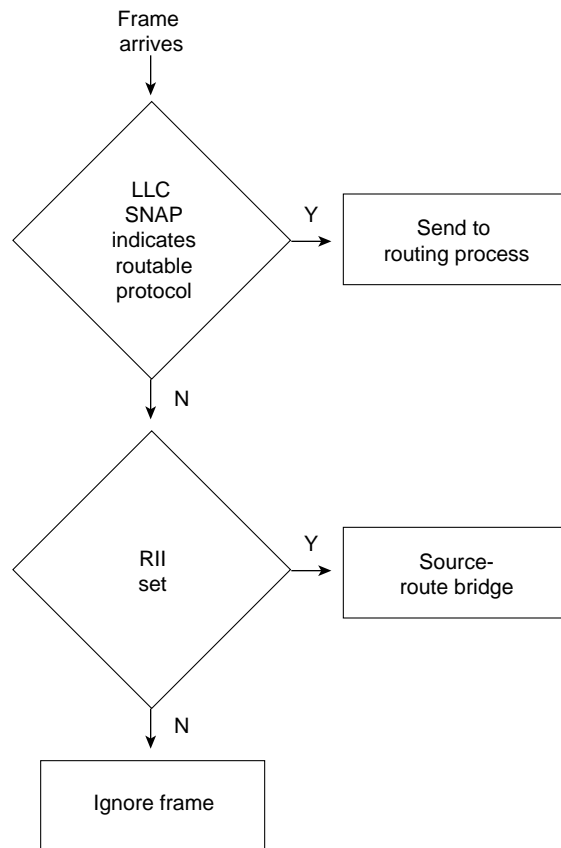
IBM's original Token Ring bridging designs use a Routing Information Field (RIF) in the Token Ring frame header. This information stored the path that the frame took to get across multiple Token Ring segments (allowing the response to be sent along the same path). The fields of the RIF are as follows:

- The *routing control* filed, which consists of the following subfields:

    — The *type* subfield in the RIF indicates whether the frame should be routed to a single node, a group of nodes that make up a spanning tree of the internetwork, or all nodes. The first type is called a *specifically routed* frame; the second type is called a *spanning-tree explorer;* and the third type is called an *all paths explorer.* The spanning-tree explorer can be used as a transit mechanism for multicast frames. It can also be used as a replacement for all-paths explorer in outbound route queries. In this case, the destination responds with an all-paths explorer.

    — The *length* subfield indicates the total length (in bytes) of the RIF.

    — The *D* bit indicates the direction of the frame (forward or reverse).

    — The *largest* field indicates the largest frame that can be handled along this route.

- The *route descriptor* field, of which there can be more than one. Each router descriptor field carries a ring number-bridge number pair that specifies a portion of a route. Routes, then, are simply alternating sequences of LAN and bridge numbers that start and end with LAN numbers.

Routers provide a feature called *multiring,* which provides several benefits when SRB networks are mixed with multiprotocol routers. The first benefit is realized when connecting a multiprotocol router to an existing pure SRB network to support routable protocols (such as Novell's IPX). In this case, the multiring feature allows you to connect IPX and SRB networks seamlessly by routing IPX

even in the presence of SRB framing. IPX stations can be linked via SRB networks or locally connected to a Token Ring with SRB framing. A router will route to the IPX station by first searching for the station and then framing each Token Ring frame with RII and a RIF.

The second benefit of multiring is that all outgoing packets for a specific routable protocol are framed in an SRB frame. The router creates a valid SRB frame by transmitting an explorer packet to create a valid RIF entry for the SRB frame of a routable network packet.

The third benefit of multiring is that it allows a smooth transition from a previously framed SRB network to a routed network. For example, a locally connected Token Ring can either use an IPX frame or an SRB frame depending on what is currently in use. To leverage existing IPX servers with SRB drivers, configure the multiring for that specific Token Ring. A typical **multiring** interface configuration example might be as follows:

```
interface tokenring 0
source-bridge 10 1 100
multiring ipx spanning
```

WAN Framing

Routers recognize two forms of SRB. The first is *local* SRB, which is characterized by either the standard single ring-to-bridge-to-ring combination, or a flexible form using a multiple ring-to-bridge-to-virtual ring arrangement. The second form of SRB involves WAN connections and is called *remote* SRB (RSRB).

The framing that occurs to support WAN activities is twofold. First, the SRB frame is encapsulated in one of three ways: Transmission Control Protocol/Internet Protocol (TCP/IP) encapsulation, Fast Sequence Transport (FST) encapsulation, or direct High-Level Data Link Control (HDLC) encapsulation. Next, the frame is placed in the WAN frame for the appropriate WAN media, such as HDLC, Frame Relay, or Switched Multimegabit Data Service (SMDS).

If you select direct encapsulation for a WAN serial link, you avoid the overhead of encapsulating into either IP or TCP. The datagram is framed directly into HDLC. Direct encapsulation for WAN frames works only for HDLC. Over a multiaccess media, such as Ethernet or Fiber Distributed Data Interface (FDDI), direct encapsulation can be used to transmit data from one router to another.

Selection of encapsulation is critical to the performance of the underlying network and affects the degree to which the topology can scale to very large networks of token rings. Each encapsulation form is addressed in the following sections.

TCP/IP Encapsulation

TCP/IP encapsulation is the most common encapsulation format. Figure 4-18 illustrates a TCP/IP-encapsulated SRB frame. The chief benefit of TCP/IP encapsulation is a robust set of capabilities that ensures reliable transport.

Figure 4-18   SRB frame encapsulated in TCP/IP with HDLC header.
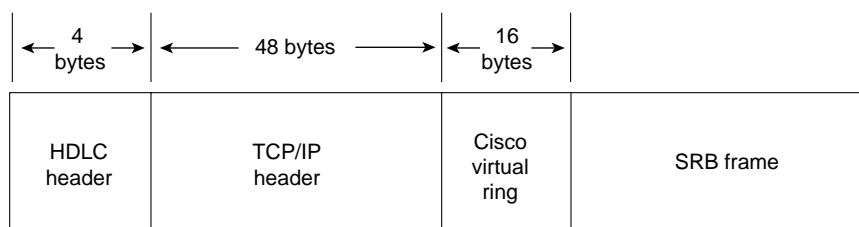
Because many tasks are involved in TCP/IP encapsulation, such as packet reordering, running timers for retransmission, and sending acknowledgments, TCP/IP encapsulation is costly in terms of CPU overhead. For both LANs and WANs, TCP/IP encapsulation incurs additional CPU overhead because all framing occurs in the CPU and the resulting IP frame is then process switched, which incurs additional overhead. (Process switching and its associated costs are discussed in "Process Switching" later in this chapter.)

Because of the high overhead associated with TCP/IP encapsulation, there is a significant upper boundary to maximum traffic forwarding. Performance is not the only constraint for using TCP/IP; fewer connections to other SRB rings can be supported using TCP/IP than any other encapsulation because of the processor overhead required to maintain the TCP structure. In general, you should limit the maximum number of remote peers connected to a single Cisco CSC/4 or RP card using TCP/IP encapsulation. Issues that can affect the acceptable number of remote peers include link speed, traffic load, number of supported protocols, routing platform implemented, and the level of other non-SRB activity occurring in the router.

## Fast Sequenced Transport (FST) Encapsulation

Fast Sequenced Transport (FST) encapsulation is an alternative to TCP/IP encapsulation. FST encapsulation creates an IP frame with a sequence number; this frame is transmitted to an IP destination. At arrival, FST encapsulation strips the IP frame. If the sequence number of the arriving frame is greater than the sequence number of the last frame that arrived, FST encapsulation places the frame on the destination ring. If the sequence number of the arriving frame is less than the last frame transmitted by FST encapsulation, the frame is discarded, and the router relies on the transport mechanism of LLC2 to request the discarded or out-of-order frames to be retransmitted.

FST encapsulation is configured on a per-remote-ring basis. A typical example of using the **fst** keyword with the **source-bridge remote-peer** global configuration command follows:

```
source-bridge remote-peer 10 fst 131.108.3.2
```

The benefit of FST encapsulation is sustained end-to-end performance across multiple hops. FST is fast because the IP encapsulation happens on the interface card (AGS+, Cisco 7000, MGS, and CGS) or the system memory (IGS, Cisco 2000, Cisco 2500, Cisco 3000, and Cisco 4000) while the processor is in interrupt mode. For WAN transmissions, once the framing occurs, you can select an IP switching mechanism, either process switching or fast switching, depending on the result you want. Figure 4-19 illustrates the frame format of FST encapsulation.

Figure 4-19   SRB frame encapsulated in FST with HDLC header.
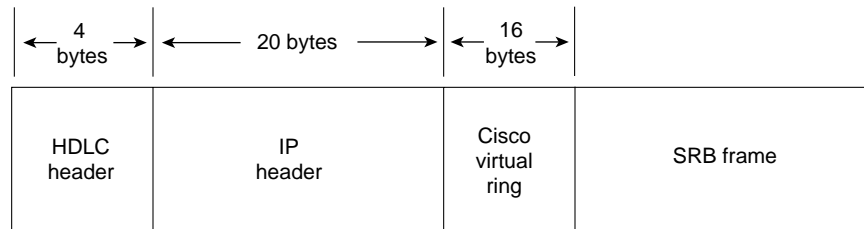

There is a cost to implementing FST encapsulation. Because the packet discard feature associated with FST encapsulation does not guarantee delivery, FST encapsulation cannot be used in conjunction with the router's local acknowledgment feature.

Direct HDLC Encapsulation

Direct HDLC encapsulation is the fastest SRB encapsulation, but has the most restrictions. Direct HDLC encapsulation allows the network designer to configure two token rings separated by a single Ethernet, FDDI ring, Token Ring, or serial link.

For multiaccess media such as Ethernet or FDDI, you must know the destination MAC address of the neighbor. For HDLC on a WAN link, you need only to know the serial interface over which you intend to transmit traffic. As with FST, direct HDLC encapsulation occurs at processor interrupt level and is very fast. Figure 4-20 illustrates the format.

Figure 4-20   SRB frame encapsulated in direct HDLC.
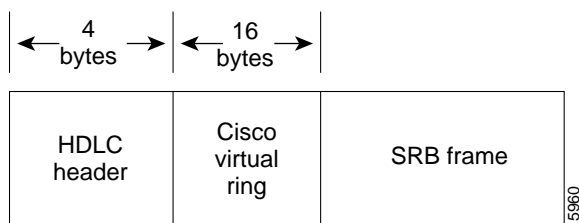
The following is an example of a global configuration command that configures direct HDLC encapsulation on a serial interface:

```
source-bridge remote-peer 10 interface Serial0
```

The following is an example of a global configuration command that configures direct HDLC encapsulation on an FDDI interface:

```
source-bridge remote-peer 10 interface Fddi0 00c0.3456.768a
```

When connected to parallel WAN links, direct HDLC encapsulation can operate over only one of the links. Contact your technical support representative for specific information regarding likely performance characteristics, given your specific network configuration and type of encapsulation.

WAN Parallelism

Parallelism implies that multiple paths exist between two points that are parallel to each other. These paths might be of equal or unequal cost. Parallel links present a number of potential problems to network designers. Parallelism is not specifically a WAN issue. However, because WAN links are expensive, parallelism becomes an important design factor. For that reason, this chapter explores some of the considerations for implementing parallel links.

Problems with parallel links in an SRB environment result from the tandem objectives of minimizing session loss when links fail and maximizing traffic across a WAN infrastructure. Pure SRB networks maximize the WAN infrastructure but cause session losses at each link failure. IP-routed SRB networks minimize session loss but leave the challenge of maximizing WAN links to network designers. The goal of this section is to explore the issues that affect your efforts to balance these objectives.

Setting up parallel links between either two routers (see Figure 4-21) or several routers (see Figure 4-22Figure 4-21) can pose challenges in an SRB environment. First, consider environments running NetBIOS and SNA over SRB environments. When an SNA or NetBIOS frame is delivered out of sequence, the end station might declare a protocol violation and terminate the session. Session loss is probably the worst possible outcome for a user or a network administrator.

Figure 4-21   **Parallel paths between two WAN routers.**
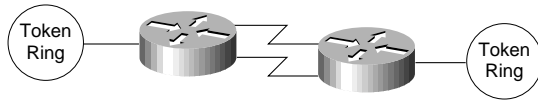
Figure 4-22   **Parallel WAN connections among several routers.**
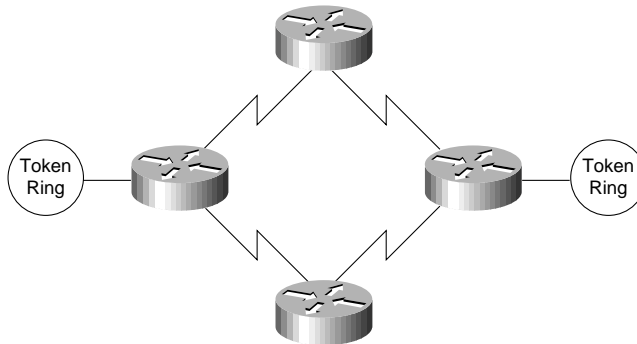
Delivering frames in sequence is the key objective of any SRB delivery mechanism. When you create parallel WAN links, you expect parallel delivery. In an SRB universe, this might not be achievable because timing differences on WAN links alone can cause packet resequencing. If the router uses parallel links and starts one frame header ahead of a second frame header, the frames might not arrive with the same sequencing. The second frame might arrive before the first frame because of WAN link delays. This is particularly true of packet-switched WANs.

When selecting or applying an encapsulation strategy with parallel WAN links, other factors influence the encapsulation that is used. These factors include the WAN switching technology and the IP routing protocols. A discussion of these factors follows. Some choices are predetermined. For example, direct HDLC encapsulation voids all parallel connections across a single virtual ring. In a multiprotocol environment, you can place SRB traffic on a single parallel link, whereas other protocols are load balanced on parallel links. As an alternative, you can configure the second link exclusively for multiprotocol (non-SRB) traffic.

WAN technologies can use two primary switching types: *process switching* and *fast switching*. Process switching provides full route evaluation and per-packet load balancing across parallel WAN links. Fast switching associates an IP host destination to a single interface to avoid out of order frames. The fact that the destination of a remote peer is a single IP destination can impact SRB decisions.

Process-switching and fast-switching techniques provide different features and performance characteristics. Each technique must be applied in situations that can optimize its respective capabilities. These switching strategies are addressed in detail in the following sections, "Process Switching" and "Fast Switching." Later in this chapter, "IP Routing Protocols with Parallel Links" addresses routing and switching in the context of SRB framing options.

Process Switching

Process switching is the most expensive switching operation that the CPU can perform. Process switching involves transmitting entire frames to the router CPU. Frames are then repackaged for delivery to or from a WAN interface, and the router makes a route selection for each packet. TCP/IP framing must be process switched because switching must occur when the rings are encapsulating or unencapsulating data, which occurs at processor level. FST framing can be process switched or fast switched.

Process switching begins when a frame arrives on a Token Ring interface and causes an interrupt to be transmitted to the CPU. The CPU then determines that the frame must be process switched and schedules the switch in noninterrupt mode. The frame is then transferred to the CPU and placed on an input queue, whose depth is viewable with the **show interfaces** EXEC command. Once the entire frame is transferred across the system bus, the frame is reworked for appropriate TCP headers and header compression.

Next, the IP route for the destination is examined. If multiple paths exist, the frame pointer is updated to use the next path for the next frame that arrives. After a route is selected, the frame is transmitted across the system bus to the output interface queue of the specific interface card from which the frame will exit. The queue entry is placed on the specific exit interface and the SCI card dequeues and transmits the frame down the WAN link.

Fast Switching

Fast switching maximizes the volume of traffic that the router can handle by streamlining the router's queuing mechanisms. Fast switching deals with incoming frames in *processor interrupt mode* and minimizes the number of decisions that must be applied.

Fast switching precaches routes. Once an IP destination is process switched, its route is cached and associated with a specific interface. When an IP destination is precached, it is tied to a specific path. For either FST or TCP/IP encapsulations, a single IP destination carries all of the SRB traffic to an FEP destination. If fast switching is used with multiple IP paths, a single path exists for each ring destination. You must use process switching to load-balance traffic across multiple paths.

Two of the SRB framing techniques are capable of being fast switched: direct HDLC encapsulation and FST encapsulation. Direct HDLC encapsulation is by definition fast switched; it cannot be process switched. FST can be fast switched or process switched.

Two IBM SRB WAN options do not allow fast switching of a frame: TCP header compression and priority or custom queuing. If either of these features is invoked, the frame cannot be fast switched. The reason for these caveats is that certain frame components are modified when using fast switching in AGS+, MGS, CGS, or Cisco 7000 interface memory and not in the CPU memory. If the frame needs to be extensively modified, it must be done in CPU system buffers and not in buffers associated with individual interface cards.

In addition, fast switching uses only interface buffers that are not generally reported using monitoring EXEC commands, such as **show interfaces**. The buffers reported in the **show interfaces** EXEC command are CPU buffers for input and output that are only used during process switching. Fast switching uses preconfigured interface buffers. You can view the allocation of buffers using the **show controllers** EXEC command.

For direct HDLC encapsulation, SRB frames are directly linked to an output serial port (such as interface serial 0). When an SRB frame enters the 2R or CTR card, an interrupt is transmitted to the CPU. The CPU verifies that the frame is an SRB frame and that the buffer on either the 2R card or the ciscoBus controller is modified to create an HDLC header. The new frame is transmitted two bytes at a time through the CPU from either the 2R card or the ciscoBus controller across the system bus to the SCI card or SIP card.

A similar process occurs for FST encapsulation; however, SRB frames are directly linked to a destination IP address. When an SRB frame enters the 2R or CTR card, an interrupt is transmitted to the CPU. The CPU verifies that the frame is an SRB frame and the buffer on either the 2R card or the ciscoBus controller is modified to create an IP datagram with appropriate WAN framing for the destination. The new frame is transmitted two bytes at a time through the CPU from either the 2R card or the ciscoBus controller across the system bus to the SCI card or SIP card.

Use the EXEC command **show ip route cache** to determine whether an IP destination is being fast switched. This command lists IP destinations as well as the relevant MAC frame and destination interface that the specific IP address uses. If the entry is fast switched, the destination IP address is present. If the destination IP address is not present, the router is using process switching to reach the destination. By default, HDLC WAN links are fast switched for IP. If the router is configured for direct HDLC encapsulation, the only status indication is the output for the **show source-bridge** EXEC command. The bridge indicates it is using a direct serial interface and not an IP address.

IP Routing Protocols with Parallel Links

IP routing protocols play a part in the parallel SRB WAN decisions because they can create wider parallelism than two routers with parallel links. Given the parallel links shown in Figure 4-23, load balancing makes three critical assumptions: equal-cost paths, routing protocol support for equal-cost load balancing, and process switching for a single IP destination.

Figure 4-23   Parallel WAN paths.
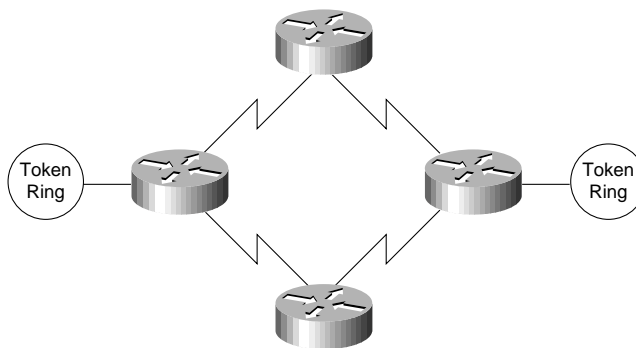
Process switching is discussed extensively in the section "Process Switching" earlier in this chapter. Issues relating to equal-cost path IP routing and unequal-cost path routing are discussed in the sections that follow, "IP Routing over Equal-Cost Paths" and "IP Routing over Unequal-Cost Paths Using Variance."

IP Routing over Equal-Cost Paths

An *equal-cost path* is *metrically* equivalent to another parallel path between two points. In RIP, equivalence is parallel WAN paths of equal hops. In Interior Gateway Routing Protocol (IGRP) and Open Shortest Path First (OSPF), metric equivalence translates into WAN paths of equal bandwidth, where the bandwidth is declared by the network administrator. IGRP also adds the concept of delay to determine metrically equivalent links. To create parallel links for equal-cost paths and to actively use these paths, the router must use process switching because all frames sent from one ring to another have the same IP destination.

The following list outlines the capability of supported IP routing technologies to create equal-cost paths:

- *Static routes*—For Cisco software releases prior to Software Release 9.1, static routes cannot be created in parallel; only a single path can be selected. As of Software Release 9.1, static routes can be created in parallel.

- *IGRP and Enhanced Interior Gateway Routing Protocol (Enhanced IGRP)*—Can use up to four equal-cost paths in parallel. Ensure that the bandwidth command is correctly configured on all links.

- *OSPF*—If paths are of equal declared metrics, OSPF can use up to four equal-cost paths in parallel.

- *RIP*—RIP can use four equal-cost paths in parallel. Remember that this will not take into account anything but hops, so even unequal bandwidth links will be evaluated as having equivalent cost.

IGRP, Enhanced IGRP, and OSPF can route traffic across equal-cost paths and split SRB traffic across equal-cost links if the router is process switching. RIP will route across equal-cost paths and it will assume that all WAN links are the same speed regardless of reality. Static routes allow parallel paths and are a tool for the advanced network designer.

A router's capability to use parallel paths is determined in part by the encapsulation method. If TCP/IP encapsulation is used, parallel paths are used. If FST encapsulation is used under normal operational conditions, all traffic must use only one of the parallel links. This is because all the RSRB traffic sent to another FST peer goes to a single IP destination address. When using fast switching, the router might alternate some traffic across parallel links based on destination address. However, because all traffic to a peer router uses only one destination IP address, all RSRB traffic flows across one link.

IP Routing over Unequal-Cost Paths Using Variance

The only routing protocols that can handle intentional unequal-cost path balancing are IGRP and Enhanced IGRP. Using a feature called *variance*, the router can load balance over unequal-cost paths. Figure 4-24 illustrates one such configuration from A to B. In this figure, load balancing the link from C to B is assumed to be faster than the link from A to B.

Figure 4-24   Unequal-cost load balancing with IGRP.
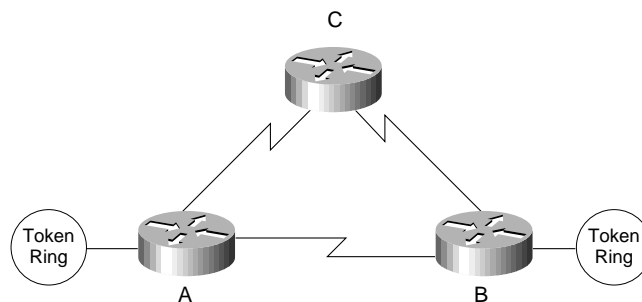
Variance has two rules that apply in this or any unequal-cost load balancing situation:

- *Rule 1*—Parallelism must exist in the topology.

- *Rule 2*—Packets must make *forward progress* on any parallel link toward an intended destination. In other words, a router will not forward traffic to another router that has the same (or greater) relative distance metric to a destination. This rule prevents loops. The rule of forward progress is straightforward. If the next-hop router is closer to the destination (than some other router) a path through it will be used as a valid alternative path.

If these rules are met and the network administrator adds variance to the IGRP configuration, the router will load balance over parallel paths for a single IP destination when it is process switching. Figure 4-25 illustrates a case in which variance might be used.

Figure 4-25   **Environmental illustrating variance applications.**
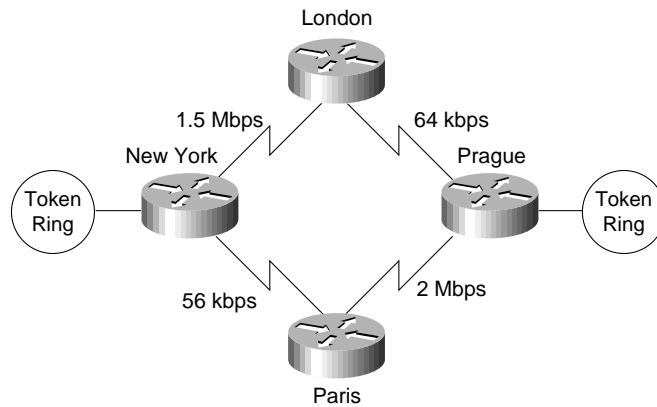
Consider a set of routers connected via WAN links in a circle, where each WAN link is the same speed, as illustrated in Figure 4-26. Assume that a data center is at location A and that all the link speeds are the same. Consider parallelism from B to A. A parallel link exists from A to B and A to C to D to E to B; however, routing protocols are not intuitive. This topology satisfies the first rule because parallelism clearly exists; however, this topology fails the forward progress rule.

The way to evaluate the forward progress rule is to examine the obvious short path separately from the long variant path, subtracting the first hop. Is C to D to E to B a better path than A to B? The answer is *no*; variance will have no effect in this topology for the problem as described.

Figure 4-26   **Unequal-cost path and variance implementation example.**
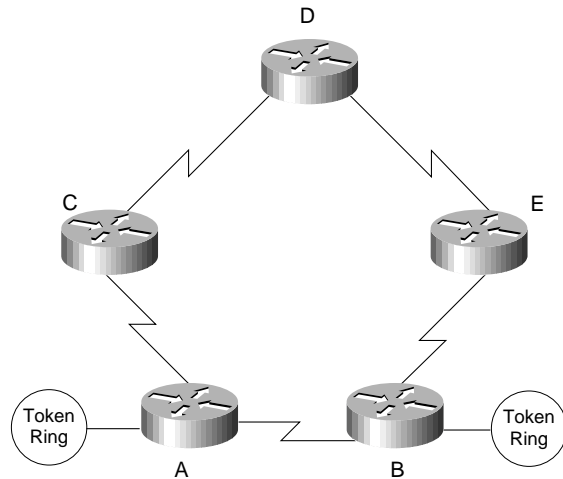
Now evaluate the problem from the perspective of A to E. Using the forward progress rule, compare A to B to E with C to D to E. In this topology, these paths are equal and they fail the forward progress rule. If these paths are to pass data in parallel, router A must have two paths: one to C and one to B. If C had variance configured, it would have two paths: one to A and one to D. This leaves the possibility of C routing to A and A routing to C in a loop. Thus, the variance rule is that the metric of the next-hop router must be less than the metric through the shortest path. In a five-router topology with equal-cost WAN links, parallelism cannot be achieved.

By default, variance is not configured. If it is, it must be configured as an integer multiple of the allowable metric variance. Consider the following use of the **variance** router configuration command:

```
router igrp 1343
variance 2
```

Using this particular instance of the **variance** command results in a load-balanced topology with a 2:1 ratio of bandwidth. For all practical topologies, this should be an upper maximum because you should not load balance an overly high variance of WAN links such as E1 and 64 Kbps.

Use variance carefully. Because IP fast switching links an IP destination to an interface or next hop, a single IP destination can be stuck on a 64-Kbps link while most other IP destinations are wired to a fast link such as an E1. This situation will cause users to call their network administrators to determine why transmission is slow one day when it was fast the day before. If SRB is fast switched, all users of a destination ring can be linked to the slower path using variance.

Variance has another major benefit: If a link fails for any reason, the router immediately switches all traffic to the parallel link without any convergence overhead. The router can do this because the parallel link is a known valid path and the router does not need to wait for the routing protocols to converge.

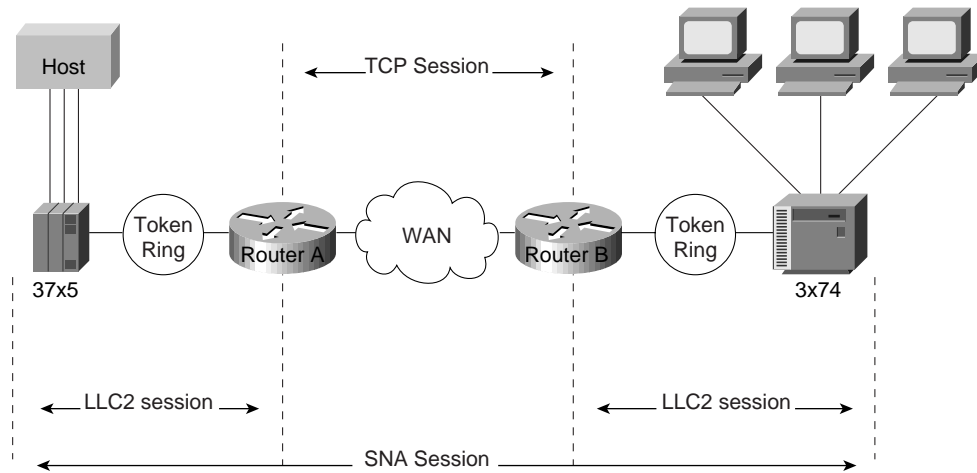## Local Acknowledgment Recommendations

If you configure your remote source-route bridge to use IP encapsulation over a TCP connection, you also can enable LLC2 Local Acknowledgment. The following explains how this feature can be useful.

LLC2, or Logical Link Control-Type 2, an ISO standard data link level protocol used in Token Ring networks, was designed to ensure reliable transmission of data across LAN media with minimal or predictable time delays. With the advent of remote source route bridging (RSRB) and wide area network (WAN) backbones, LANs are now separated by wide, geographic distances spanning countries and continents. As a result, these LANs have time delays that are longer than LLC2 allows for bidirectional communication between hosts. The Local Acknowledgment capability in router/bridges supporting remote source-route bridging addresses the problem of unpredictable time delays, multiple retransmissions, and many user sessions.

In a typical LLC2 session, when host A sends a frame to host B, the sending host A expects host B to respond positively or negatively in a certain amount of predefined time commonly called the T1 time. If host A does not receive an acknowledgment of the frame it sent to host B within the T1 time, it will retry a few number of times (normally 8 to 10). If there is still no response from host B, host A will drop the session.

Figure 4-27 shows how Local Acknowledgment operates in an SRB environment. With Local Acknowledgment for LLC2 enabled in both routers, Router A acknowledges frames received from the 37x5. The 37x5 still thinks that the acknowledgments it receives are from the 3x74. The 3x74 thinks that the acknowledgments it receives are from the 37x5. Router B looks like the 3x74 to 37x5. Because the frames no longer have to travel the WAN backbone networks to be acknowledged, but instead are locally acknowledged by routers, the end machines do not time out, resulting in no loss of sessions.

Figure 4-27   How Local Acknowledgment operates in an SRB environment.

The following recommendations apply to the implementation of local acknowledgment. Use local acknowledgment under the following conditions:

- When the WAN implementation must accommodate long network delays

- When the internetwork includes slow links, heavily used links, or poor quality links

- When the internetwork requires that sessions remain active during router convergence

- When WAN traffic must be minimized

- When the amount of LLC traffic on backbone needs to be reduced (when more than 50 percent of packets are LLC packets)

- When WAN costs must be reduced

- When network integrity must be improved, assuming TCP/IP encapsulation is used

- When unreliable WAN links exist that are causing frequent session loss

- When end station timer or retry modifications are difficult or costly

- When bandwidth constraints require the elimination of acknowledgment traffic

Parallel Link Recommendations

The following recommendations apply to parallel WAN link configuration:

- Do not combine multiple CTR cards with multiple WAN links; create a separate router primarily with WAN links. For example, do not create an 8-T1/E1 process-switched WAN solution on top of a 75-kilopackets-per-second (Kpps) Token Ring engine. You will run out of CPU bandwidth.

- Use FST encapsulation whenever possible.

- Use TCP/IP encapsulation when local acknowledgment or prioritization is required.

- Maximize fast switching.

When link speeds are primarily 64 Kbps and slower and local acknowledgment or prioritization is a requirement, use TCP/IP encapsulation with IGRP variance in meshed topologies when the topology can take advantage of these features.

When link speeds are primarily greater than 64 Kbps and local acknowledgment is a requirement, follow this recommendation:

- Use TCP/IP encapsulation only on those links that have a history of session loss (local acknowledgment).

- Use FST encapsulation on the remaining links.

SNA Host Configuration Considerations for SRB

When designing SRB-based internets that feature routers and IBM SNA entities, you must carefully consider the configuration of SNA nodes, as well as routing nodes. Appendix C, "SNA Host Configuration for SRB Networks," provides examples of SNA host configurations that focus on three specific SNA devices:

- Front-end processors

- VTAM-switched major nodes

- 3174 cluster controllers

IP Routing Protocol Selection for SRB Networks

When designing large SRB networks, the goal is to optimize the underlying IP network so it can carry SNA and NetBIOS traffic more efficiently. To do this, select your IP routing protocol carefully. You should consider the following parameters when selecting your routing protocol:

- Time to converge

- Maintainability of the internetwork routing environment

If you select a protocol using only one criterion, you might build a network that cannot be expanded and that might eventually break.

Three interior gateway routing protocols work best in an SRB environment: IGRP, Enhanced IGRP, and OSPF. In general, IGRP, Enhanced IGRP, and OSPF are the only options for building an IP SNA network. You can also consider RIP. However, because RIP does not provide any consistent WAN bandwidth sensitivity, it is not a good choice for redundancy or meshed topologies.

The following discussion focuses on network topology and convergence considerations.

Convergence Considerations

Convergence is the time it takes a router to start using a new route when an active link fails in a network where alternative routes are available.

Rapid convergence is critical for SNA environments, particularly when local acknowledgment is not used. Consider a 3174 failure recovery scenario—an SNA device can lose its session with the host in 13 seconds. The result is session loss and route rediscovery for all affected units. If the 3174 had not just sent data to the host, the session would not be lost for somewhere between 13 and 42 seconds, depending on the value of the T1 Timer Inactivity parameter when the link failed. If local acknowledgment is used, the SNA session does not fail while the routers are converging.

Convergence becomes an issue when installing large meshed networks with multiple valid alternative paths. Distance vector protocols such as RIP or IGRP cannot determine the source of a learned route. A route could be learned by Router A from a neighbor that had originally learned the route from Router A. If Router A and its neighbor both use this route, they create a *routing loop*. Routing loops imply *broadcast storms* and, as a result, are widely viewed as undesirable events.

Enhanced IGRP provides superior convergence properties and operating efficiencies. It uses a convergence algorithm that eliminates routing loops throughout a route computation. More importantly, convergence time with Enhanced IGRP is reduced to a level below the threshold for session loss.

OSPF was also designed to minimize convergence time. It is good at convergence, but has side effects that will be discussed in the next section. For routers, total convergence time has two primary components:

- Link failure detection time
- IP routing protocol convergence time

Link failure detection time is the minimum, maximum, or average time it takes the router to detect that no frames are crossing the link. IP routing protocol convergence time is the time it takes the routing protocol to detect a failure and switch to alternative links.

Link Failure Effects on Convergence

Links fail in a hierarchical order of occurrence. Serial links are the most unreliable. FDDI networks, Token Ring networks, and Ethernet networks are about equal in reliability and rarely fail.

The following sections describe the significance of media-failure detection mechanisms with respect to recovery from media failure and the effects of different media failures on convergence in an IBM internetwork.

Keepalives and Convergence

Routers institute *keepalives* to verify the stability of a link. A router transmits a packet every 10 seconds by default, and when three keepalives sequentially fail to cross the link, the router declares the link to be down. To recover, the router retransmits the packet every few seconds.

For IBM IP networks, keepalives should be active only on serial and Ethernet links. Ethernet link keepalives are acceptable because the failure rate is low, but serial links (especially those faster than 64 Kbps) should be set to three seconds. Use the **keepalive** interface configuration command to adjust the keepalive timer for a specific interface. For example:

```
interface serial 0
keepalive 3
```
This configuration reduces the maximum failure detection for the serial interface from 30 seconds to nine seconds. (The interface is declared down after three consecutive update intervals pass with no keepalives detected.) Media-related keepalive specifics are provided in the sections that follow.

Enhanced IGRP uses small hello packets to verify link stability. Hello packets are transmitted by default every five seconds. When three hello packets fail to cross the link, the router immediately converges. Hello packets originate from the network layer and are protocol dependent. Use the **ip hello-interval eigrp** interface configuration command to configure a different hello packet interval for IP. For example:

```
ip hello-interval eigrp 109 3
```
This example configures a hello packet interval of three seconds for the IP protocol on Enhanced IGRP autonomous system number 109.

Serial Link Failure Effects

Serial links are inherently unreliable because they usually extend over long distances and because they are subject to a variety of failures.

In general, if a router detects loss of the carrier signal, it immediately disables the link. Unfortunately, carrier loss is not a guaranteed way of detecting a failed link, so the router must also use keepalives or hello packets to determine whether an interface is connected to an operational medium.

When the carrier signal is lost, the router detects the failure immediately. For any other serial failure, given the default keepalive timer of 10 seconds and the rule that three keepalives must be missed before the router declares that the interface is down, failure detection takes at least 21 seconds and could take as long as 30 seconds, with an average detection time of 25.5 seconds. When the keepalive timer is three seconds, the failure is detected within seven to nine seconds.

Token Ring Failure Effects

Token Ring media, whether twisted-pair or IBM media attachment units (MAUs), rarely encounter failures. When media failures occur, the Token Ring protocol fails, causing the ring to transition, beacon, and reinitialize.

Token Ring has built-in reliability that allows the interface to determine whether the ring is up or down: The returning token indicates an active ring. Keepalives, which provide a fail-safe mechanism in case the Token Ring protocol itself fails, are also available but can be disabled in most networks to prevent unnecessary network traffic. Any keepalive failure usually indicates that the Token Ring interface is under tremendous load or may have already failed. The failure detection time for Token Ring is immediate.

FDDI Failure Effects

Like Token Ring, FDDI rings are reliable media. Users who turn their devices off, which causes the FDDI ring to "wrap," are the most common cause of failure in dual-attached FDDI networks. Keepalives are available, but are not particularly useful. Enabling keepalives with FDDI can cause problems in high-load environments because the keepalives add to the traffic. Because the router disables the interface when it is experiencing intolerably heavy traffic loads, detection of a keepalive loss is usually a false error indication. The failure detection time for FDDI rings is immediate.

Ethernet Failure Effects

Ethernet media is generally reliable but lacks a failure-detection protocol. Therefore, keepalives play a critical role in determining the availability of the media. The keepalive must fail three times before the router disables the interface. There is no indication of the location or source of the failure, whether it is from router to MAU or across the physical media.

Given the default keepalive timer of 10 seconds and the rule that three keepalives must be missed before the router declares that the interface is down, failure detection takes at least 21 seconds and could take as long as 30 seconds, with an average detection time of 25.5 seconds.

Routing Protocol Convergence

When analyzing routing convergence, it is assumed that a link has failed or router keepalives have not been detected. The router waits for a link failure detection period to expire. After this waiting period passes, the router incurs a *routing protocol convergence time*. The following discussions address convergence for IGRP, Enhanced IGRP, and OSPF.

IGRP Convergence

IGRP convergence is controlled by a single factor: whether *holddown* is enabled (the default). This discussion focuses on determining when it is appropriate to disable holddown.

Because a router learns about routes from its neighbors, a distance vector routing protocol never actually understands the topologies to which it is connected; instead, it approximates the topologies. When enabled, holddown, which is a property of distance vector routing protocols, specifies that alternative paths are not used until the paths in question are determined to be actual alternative routes. When a failure occurs and alternative paths exists, the router holds down any routing protocol changes until the holddown timer expires to determine that the network is now completely known.

IGRP allows you to configure the protocol so that it will *not* hold down a link. The danger of administratively disabling holddown is that the routers might loop packets to each other for networks that are unreachable, which would cause the receipt of high volumes of errant traffic that could dominate low-bandwidth links. Any errant datagram would loop up to 254 times before the "counting to infinity" process causes the datagram to be dropped. The length of time associated with "counting to infinity" can be modified using the **metric maximum-hops** *hops* router configuration command. The default *hops* value is 100; the maximum is 255.

Generally, meshed WAN bandwidth that consists of fractional T1/E1 or greater can converge faster than parallel WAN bandwidth that is 64 Kbps. Network topologies with high WAN bandwidth can support disabling holddown, so you can safely disable holddown on all routers in any network with a high WAN bandwidth.

If convergence time is worth trading off against potential bandwidth for sites with lower-speed links, you can disable holddown on these sites. However, if a loop occurs when a link is lost, the network performance for end systems connected to affected sites might be poor until "counting to infinity" ends. If you require faster convergence and can live with congestion for a brief period, you can disable holddown in any case. To disable holddown, enter the following router configuration commands for all routers in the network:
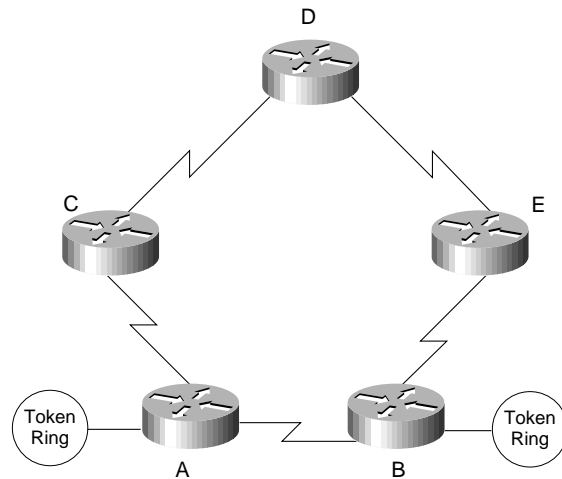
> **router igrp** *autonomous-system*
>
> **network** *network-number*
>
> **no metric holddown**

Including the **no metric holddown** router configuration command changes the convergence of IP to 50 percent of neighbor update time (on average) assuming a neighbor is using this other valid route. Consider a topology as illustrated in Figure 4-28.

Figure 4-28   Convergence topology.

Assume that all links illustrated in Figure 4-28 are of equal speed and that the link from A to B fails. If C is using A to get to B, the IGRP Flash update tells C that its route to B is probably down. When D sends the next IGRP update, C uses D to get to B. A knows its route to B is down, and waits for two updates from C (on average) to get a new route to B. Most topologies converge with a single neighbor update.

If variance is active and there are two separate paths to a destination network, the network converges immediately to the remaining path when the router receives a Flash update.

Also note that the default values for IGRP timers are appropriate for general IP networks, not for IBM IP networks. It is necessary to change a few timer defaults for an IBM IP environment. The basic neighbor update timer is set to 90 seconds. For IBM IP networks, use 20 seconds, which results in an average IBM IP convergence for IGRP of 10 seconds with a Flash update. To make this change, modify the IGRP configuration of each router. The router configuration commands are as follows:

router igrp *autonomous-system*

network *network-number*

timers basic *update invalid holddown flush* [*sleeptime*]

Consider the following configuration for the **timers basic** router configuration command:

```
timers basic 20 60 60 120
```
These values optimize IGRP convergence in an IBM IP environment. If holddown is enabled, the worst-case convergence is three update periods of 20 seconds each, for a total of 60 seconds. Although these values optimize convergence, the worst-case convergence time can break IBM sessions. Try using local acknowledgment to keep sessions up while IGRP converges.

Enhanced IGRP Convergence

Enhanced IGRP is an advanced version of IGRP. The same distance vector technology found in IGRP is used in Enhanced IGRP, and the underlying distance information remains unchanged. Enhanced IGRP implements a new convergence algorithm that permits loop-free operation throughout a route computation, which improves Enhanced IGRP convergence properties and operating efficiency. Enhanced IGRP allows all routers involved in a topology change to synchronize at the same time. Routers that are not affected by topology changes are not involved in the recomputation. The result is very fast convergence time.

OSPF Convergence

OSPF uses two mechanisms for detecting failure. The first mechanism consists of interface status changes, such as carrier loss on a serial link or keepalive loss. The second mechanism is failure of OSPF to transmit and receive a hello packet within a timing window called a *dead timer*. Once the dead timer expires, the router assumes the link is dead. Once a router running OSPF assumes a link is dead, it produces an area-wide broadcast that causes all nodes to recompute their topology maps.

When OSPF receives an active multicast with link down information, the convergence time is less than one second. Suboptimal OSPF convergence occurs when a link is down but the router receives no forward indication. In this failure situation, the router must wait for the dead timer to expire. By default, the OSPF dead timer is set to 40 seconds. In most IP networks, you can set the dead timer equal to at least three OSPF hello packets.

In the IBM IP environment, the default values of the OSPF timers are too high for the session layer convergence that SNA and NetBIOS require; therefore, you should change the dead timer to 18 seconds and the hello timer to six seconds for each interface in your network. For example:

```
interface tokenring 0
ip ospf dead-interval 18
ip ospf hello-interval 6
```

Convergence Summary

If you followed all the recommendations in this section, the behavior of the two routing protocols is as follows:

- IGRP provides instant convergence with carrier loss and active variant paths. Assuming that *serial keepalive* = 3 and *update* = 20, convergence delays are as follows:

  — Seven- to nine-second convergence (eight-second convergence on average) with serial keepalive loss and active variant paths. This assumes that three keepalives have expired, there was no holddown, and no routing update was required.

  — One- to 20-second convergence (10.5-second convergence on average) with carrier loss, Flash update, and no holddown. This assumes failure detection is immediate; therefore time for routing update is the only factor.

  — Ten- to 29-second convergence with keepalive loss, Flash update, and no holddown. This assumes a nine-second keepalive loss, the Flash update was immediate, and one to 20 seconds for the routing update.

  — Sixty-nine-second convergence worst-case scenario (nine-second keepalive loss, three updates of 20 seconds each).

- Enhanced IGRP provides instant convergence with carrier loss and presence of a feasible successor. Convergence delays are as follows:

  — Eleven- to 15-second convergence by default for Hello packet loss in all cases.

- OSPF provides instant convergence with carrier loss and active broadcasts. Convergence delays are as follows:

  — Nine-second convergence with serial keepalive loss and active broadcasts.

  — Eighteen-second convergence, worst-case scenario.

Note Assuming that OSPF is configured with realistic timer settings, the total convergence time is the sum of the time it takes the interface to change its state from up to down, combined with the time it takes the routing protocol to converge.

Routing Protocol Design and Maintenance Issues

You must consider two key design and maintenance factors when creating networks based on IGRP, Enhanced IGRP, or OSPF for primarily SNA traffic:

- Routing Protocol Network Design

- Routing Protocol Scalability

Routing Protocol Network Design

Some routing protocols do not require an additional topological structure to build a successful internetwork. Other routing protocols require a separate topological structure outside of the existing addressing structure that must be maintained and well understood. IGRP, Enhanced IGRP, and OSPF show how different routing protocols handle network design issues.

IGRP Routing Protocol Network Design

IGRP has no implicit network design requirements. IGRP networks can scale as nonhierarchical topologies to thousands of networks and hundreds of routers.

However, implementing hundreds of IGRP routers in the same autonomous system results in the transmission of an extremely large routing update every 90 seconds (by default). The impact of routing update transmission is dampened by a feature of IGRP called *route summarization*, which summarizes unconnected network numbers into a single routing table entry.

For example, if 1000 subnets of TCP/IP are distributed evenly across 10 IP networks, a single router with route summarization would see the 100 subnets of its locally connected network and nine summary routes to all other networks. Route summarization reduces the routing table of large networks, but can result in suboptimal routing at the border points.

### Enhanced IGRP Routing Protocol Network Design

Enhanced IGRP, like IGRP, has no implicit network design requirements. Unlike IGRP, Enhanced IGRP does not make large routing table updates in a single large autonomous system, which makes the use of Enhanced IGRP even more scalable. Only routers that are directly involved in a topology change are involved in route recomputation, which saves processor overhead and results in minimal bandwidth utilization for routing updates.

Enhanced IGRP uses an automatic redistribution mechanism so IGRP routes are imported into Enhanced IGRP and vice versa, for compatibility and seamless interoperation with IGRP routers. The compatibility feature allows you to take advantage of the benefits of both protocols while migrating from IGRP to Enhanced IGRP and allows Enhanced IGRP to be enabled in strategic locations carefully without disrupting IGRP performance. By default, IGRP routes take precedence over Enhanced IGRP routes, but a configuration command that does not require routing processes to restart can change the default.

OSPF Routing Protocol Network Design

OSPF has a network structure that must be maintained separately from the addressing structure of IP. The concept in OSPF is that a single backbone of routers will communicate with several leaf areas. Consider the general environment illustrated in Figure 4-29.

Figure 4-29   OSPF backbone communicating with several leaf areas.
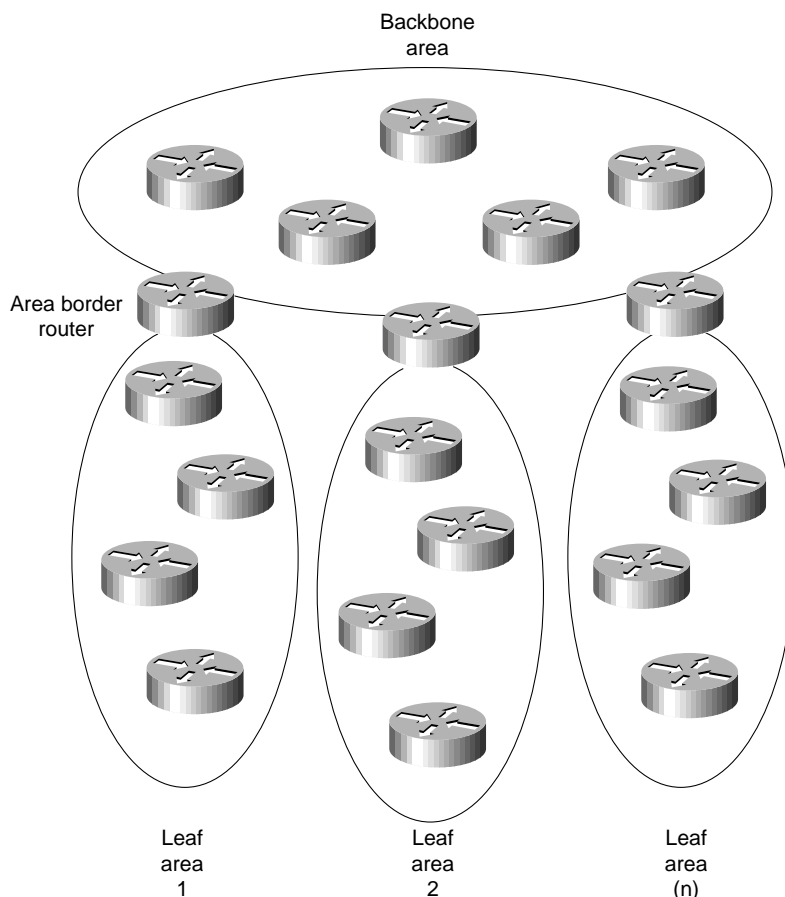
Communication between areas occurs through the backbone only. There is no interarea communication except through the backbone, which is not an overwhelming constraint for most SNA networks because they are already hierarchical in nature. NetBIOS networks, however, are not hierarchical in nature, which poses a potential design challenge.

A hierarchical structure limits the extent of link-state broadcasts that indicate link failures. In each router, OSPF builds a full area topological database that describes each router and each link. When any link changes state, each router within an area recomputes the entire database and builds a new routing table. Traffic associated with this recomputation process occurs across all links in the area. With a typical large installation (for example, 400 routers), you might expect several link updates per second. However, link updates can occur more often, flooding the network and forcing the routers to use all active cycles maintaining routing tables instead of forwarding traffic.

To avoid these problems, create a *structure* of leaf areas and a unique backbone. To create this structure, take the square root of the number of routers and subtract one for the backbone. For example, 100 routers would optimally be allocated with 10 routers in the backbone and nine areas each with 10 routers. Each area must touch the backbone, so the selection of the backbone routers is critical.

Modifying an existing topology to add an additional 10 routers to a geographically remote location poses a greater challenge. You must decide whether to create an unbalanced area that connects the remote location to the backbone, or to rebalance the topology by adding an OSPF backbone router at the remote location.

After you create the topology, you must impose IP addressing on it. If you do not assign a separate network to each leaf area, the boundaries between the leaf areas and the backbone are meaningless and link status changes will propagate throughout the entire network. Each backbone router that bounds an area (called an *area border router*) must summarize the routes imported to and from the backbone. Route summarization does not occur by default, so for most IP networks you must include a common set of commands at each area border router. The following is a typical configuration for area border routers:

```
router ospf 10
network 192.30.0.0 0.0.0.255 area 0
network 131.108.0.0 0.0.255.255 area 0.0.0.1
area 0.0.0.1 range 131.108.0.0 255.255.0.0
```

In this example, the importation of routes into the backbone of network 131.108.0.0 is limited. Unfortunately, it specifies only a single point of entry for network 131.108.0.0. If several area border routers are connected to leaf area 1 using network 131.108.0.0, the router uses the nearest area border router with connectivity to 131.108.0.0.

The techniques used for addressing an OSPF using multiple areas are discussed in the "Addressing and Route Summarization" section in Chapter 3, "Designing Large-Scale IP Internetworks."

Routing Protocol Scalability

Only one significant design challenge exists for large scalable IBM networks using IGRP as the routing protocol: low-speed links individually connected as leaf networks in which IGRP transmits large routing tables. To prevent potential problems, configure the router to transmit IGRP information in a single direction—toward the backbone. The leaf router uses default routing to find the backbone and all other valid routes. The leaf router will transmit IGRP information about its routes to the backbone. The backbone router does not transmit any IGRP information to the leaf. The following examples illustrate configurations for leaf and backbone routers:

```
! Leaf router configuration
router igrp 109
network 131.108.0.0
ip route 0.0.0.0 Serial0
ip route 131.108.0.0 Serial0

! Backbone router configuration
router igrp 109
network 131.108.0.0
passive-interface Serial0
```

Figure 4-30 illustrates what happens when the preceding leaf and backbone router configurations are used. This configuration does not send routing information to the lower-speed leaf routers, while the backbone retains all valid routes in the network.

Figure 4-30   Effects of using the passive-interface router configuration command.
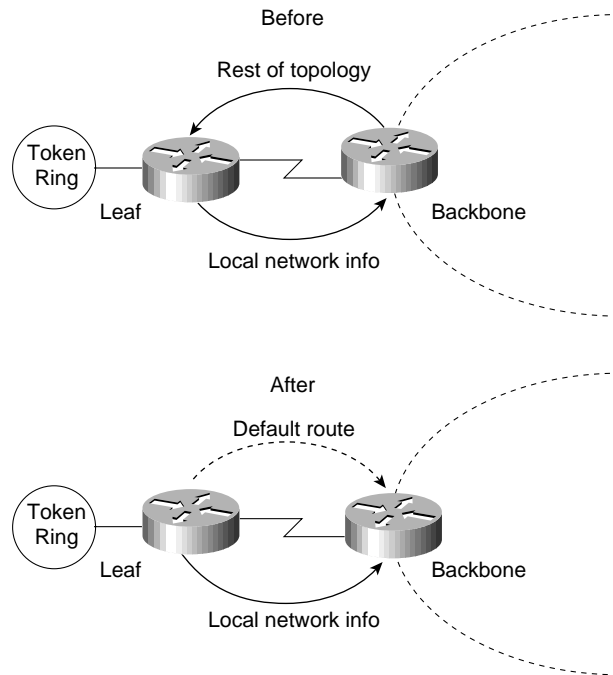
Note   When designing large branch networks based on OSPF routing, OSPF has a natural limit. When link instability raises broadcast traffic and route recomputation to an unacceptable level, the network is at its limit. Always contact your router technical support representative when designing large OSPF-based internetworks.

SRB Network Design

The key to building predictable and scalable SRB networks is to follow the network design guidelines in this chapter. Ultimately, there is a limit to the maximum diameter of a single meshed virtual ring, so before you begin designing a network, consider four critical questions. Answering these questions helps you assess the available options.

- *How many routers are required?* This question assesses the capability to build a simple SRB network. If you are implementing a large internetwork, contact your technical support representative for specific information about virtual ring limitations.

- *Are there any T1/T3, E1/E3, fractional T1/T3, or fractional E1/E3 links?* This question assesses SRB WAN traffic that may reduce a meshed topology to a smaller radius. If you are using T1/T3 or E1/E3 technology, you can take advantage of their increased bandwidth capabilities by increasing traffic loads to and from the rings, which allows you to reduce the number of routers.

- *Is the design for an SNA network, a NetBIOS network, or both?* This question helps you determine whether a partially meshed topology can be used when an FEP-connected ring is a peer of each Token Ring in the network. The remote token rings are allowed to be peers only of the FEP rings, not of one another. This topology is called a partially meshed network because certain points can connect only to certain points. Partially meshed SRB networks are much more scalable than fully meshed networks, in which all rings can reach all rings. Fully meshed topologies are often required in NetBIOS environments.

- *Is the network a multiprotocol environment?* This question implicitly raises the topic of *prioritization*. When dealing with a multiprotocol internetwork, you must consider your options for prioritizing traffic to ensure acceptable response time for interactive traffic, while maintaining adequate internetworking resources to handle other types of traffic, such as file transfers.

In general, it is best to design a router network in a hierarchical fashion; there are typically three logical service layers: the backbone (or core) service layer, the distribution service layer, and the access service layer. Figure 4-31 illustrates these basic service layers.

When designing a router network for SRB, consideration should be given to the design of virtual rings. Two key issues affect the design of virtual rings:

- The type of SRB connectivity required (hierarchical, distributed, or flat)

- The corporate organizational structure

Figure 4-31   Backbone, distribution, and access service layers in an SRB environment.
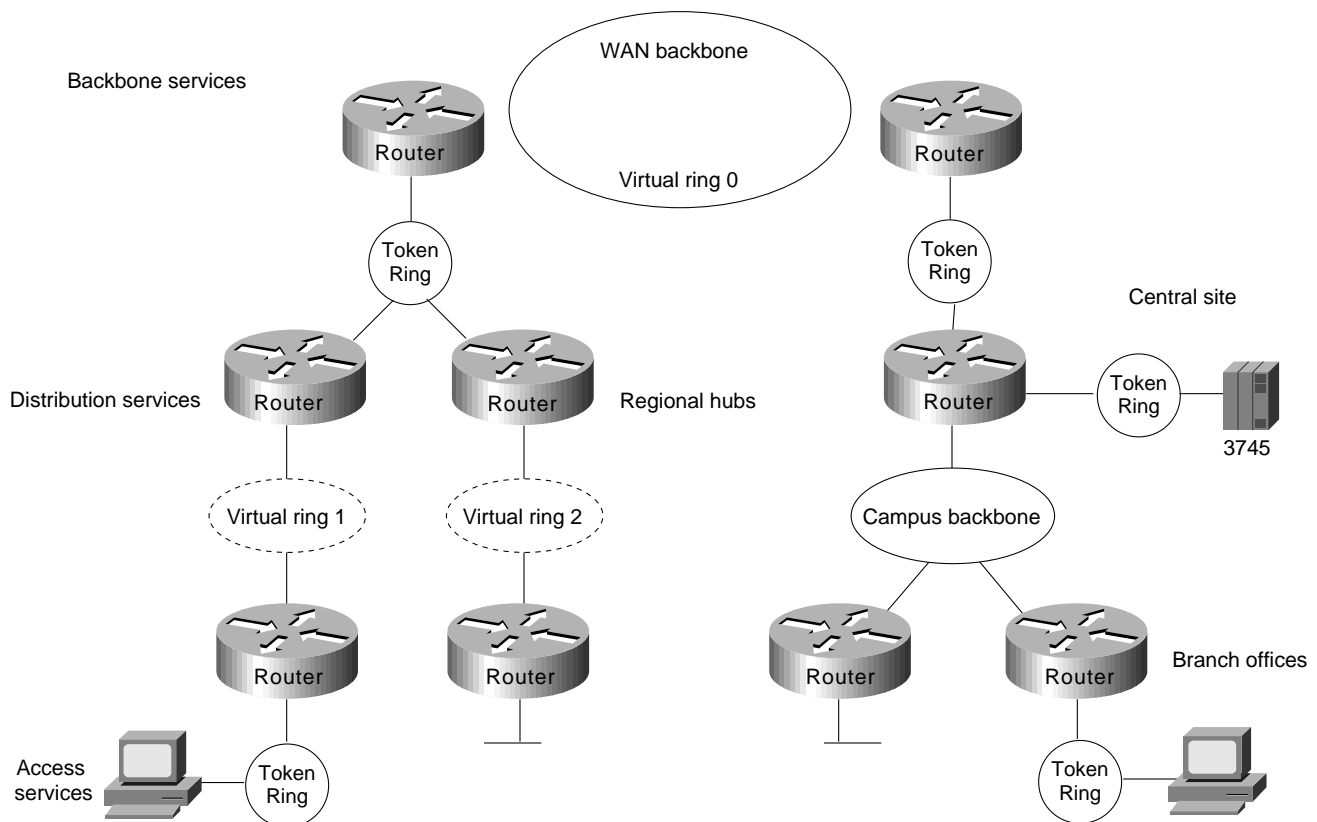

The remainder of this section focuses on network design approaches that help create scalable networks. The following topics are discussed:

- Hierarchical Design for SNA Environments

- Hierarchical Design for NetBIOS Environments

- Queuing and Prioritization Schemes

Hierarchical Design for SNA Environments

In SNA-only networks, all processing is hierarchical, where a single FEP or a few FEPs (one primary and one secondary) are the target of all remote rings. The SRB topology is focused from all remote rings to a single or a few redundant rings. A topology featuring a single FEP is illustrated in Figure 4-32.

Figure 4-32   Hierarchical topology featuring a single FEP.
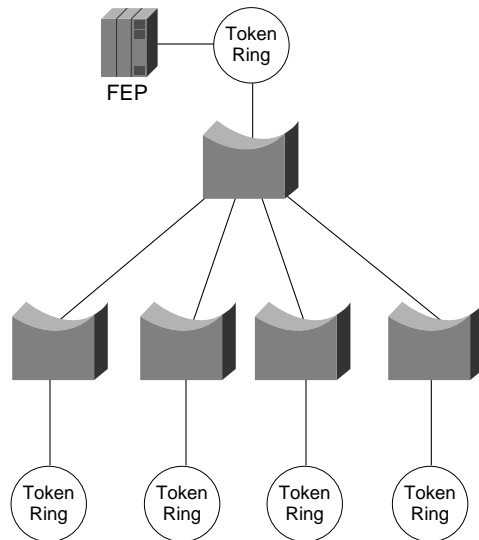
A topology featuring duplicate FEPs on duplicate rings is illustrated in Figure 4-33.

Figure 4-33   Duplicate FEPs on duplicate rings.
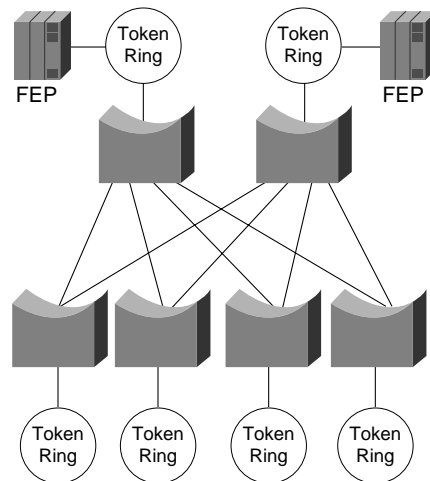
The topology in Figure 4-33 is a partially meshed topology because the remote nodes cannot reach each other; they can reach only the core of the network where the FEPs are located.

When you are designing a partially meshed topology, several options are available. SNA traffic can be generalized as having few explorer packets and having the requirement to connect many remote sites. The suggested topology for a partially meshed topology depends on whether the link speed to

the core is greater than 64 Kbps. Contact your technical support representative for specific limitations and capabilities regarding the maximum number of peers for the various encapsulation implementations and your specific network attributes.

To scale a partially meshed network to diameters of greater than 15 to 100 remote rings, you can take two approaches: Build a hierarchical structure of virtual rings, or build a scalable partially meshed structure using a single virtual ring.

Proceed with caution to avoid uncontrolled growth in virtual rings, especially in parallel, because parallel virtual rings replicate explorer packets, which results in unnecessary explorer packet traffic.

Scalable Partially Meshed Rings

With a partially meshed ring topology, the objective is to leverage the advantage of a network that does not require *any-to-any* connectivity. You can use a single virtual ring to connect a series of routers at the FEP sites. For each additional 15 to 100 remote peers, you must add a router to the central site. Figure 4-34 illustrates this kind of environment. Contact your technical support representative for more information about specific limitations and recommendations that might apply to your network specifications.

Figure 4-34   Virtual ring environment interconnecting multiple remote peers.
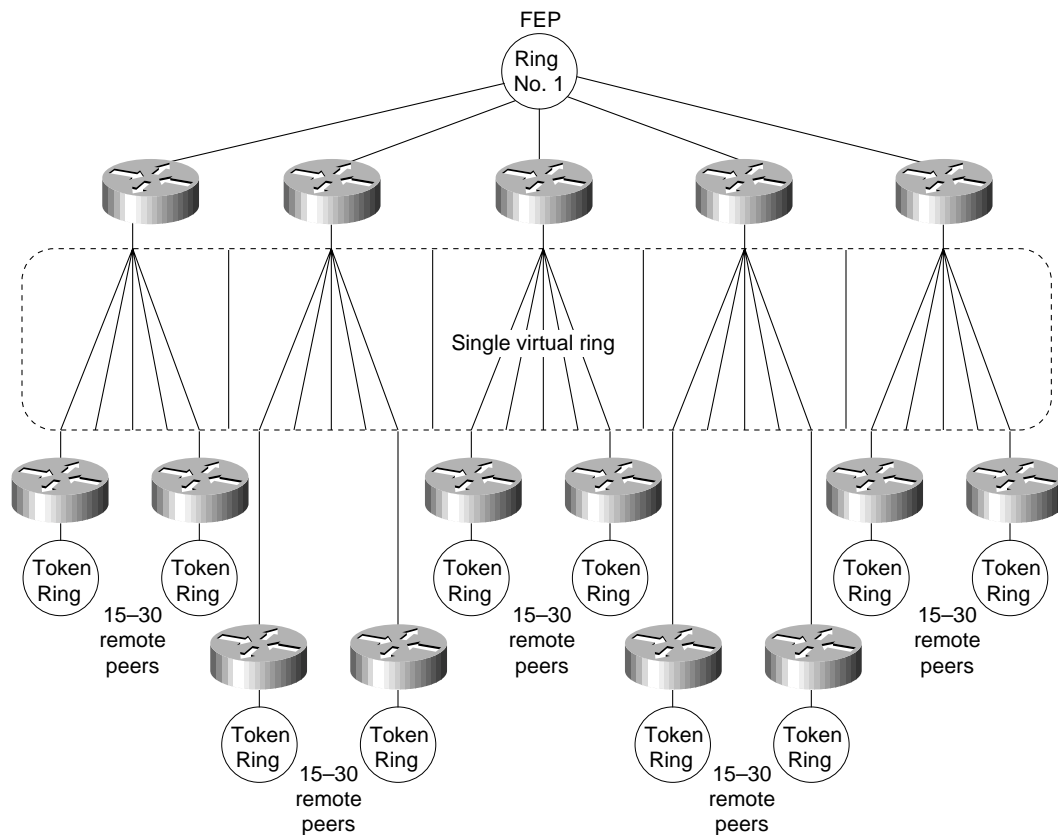

The network illustrated in Figure 4-34 will work for local acknowledgment because all traffic exists on a single virtual ring. A potential problem with this topology is that the number of routers is related to the number of virtual rings, not to the LAN or WAN connectivity at the FEP site. A site with two WAN links and a Token Ring can require several routers if it is the main FEP site.

Hierarchical Virtual Rings

Using *hierarchical* virtual rings, you can use physical token rings and virtual rings to create a hierarchy of virtual rings. Figure 4-35 illustrates such a hierarchy.

Figure 4-35   **Hierarchical virtual ring topology.**
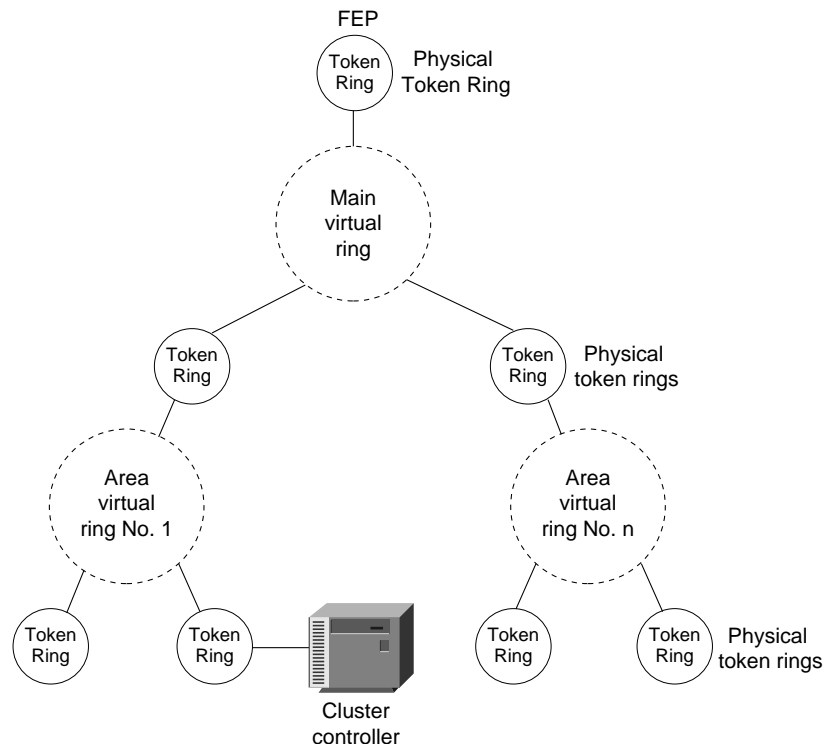
Because the destination of all explorer packets is to the core virtual ring, you can use filters to eliminate explorer packet traffic crossing between local-access virtual rings at the point where rings meet. The filters would also filter out FEP traffic. As an alternative, you can use the same virtual ring number for each virtual ring to filter out FEP traffic that might otherwise traverse the local-access virtual rings.

This design is limited in that the hop count might limit Token Ring SRB connectivity. Because the connectivity from access point to FEP uses four hops, additional local bridges at the access points or at the central site might not be reachable from the entire network.

Combined Designs

Networks can be built from hierarchical designs and from scalable partially meshed designs as long as you prevent explorer packet traffic from reexploring access points. To fulfill this requirement, write access lists to prevent explorer packet traffic from entering a ring if that traffic did not originate from the ring that has an FEP.

Hierarchical Design for NetBIOS Environments

The challenge of NetBIOS is that applications might require unrestricted ring-to-ring connectivity. The SRB protocol was not designed to scale, and network designers often demand that routers help scale beyond the original design of the protocol.

Limitations on the maximum number of peers mandates that your only topological option for a NetBIOS SRB network is the hierarchical environment illustrated in Figure 4-35. This design poses certain challenges because of increased explorer packet traffic. It is imperative that you create a single-stream, spanning connection through the network to minimize explorer packets. To succeed, a hierarchical NetBIOS network needs three elements:

- Proxy explorer

- Aggressive explorer packet caching

- NetBIOS name caching

These features allow switching of valid NetBIOS traffic even under the worst conditions of high explorer packet load. You might not be able to use the partially meshed network design if you have to maintain unrestricted ring-to-ring connectivity. Contact your technical support representative to determine any specific limitations for your network topology and design implementation. Refer to the "Explorer Packets and Propagation" and "NetBIOS Broadcast Handling" sections earlier in this chapter for additional details concerning these topics.

Queuing and Prioritization Schemes

The following information focuses on current prioritization mechanisms. Prioritization tools discussed include:

- Priority Queuing

- Custom Queuing

- SAP Prioritization

- Enhanced LU Address Prioritization

- SAP Filters on WAN Links

Note   The queuing and prioritization schemes described in this section rely on process switching. If the router is configured for fast switching or for autonomous switching, the configuration of a queuing or prioritization scheme will increase processor utilization. However, increased processor utilization is usually not a problem when the router is sending traffic over low-speed WAN links.

Priority Queuing

Priority queuing (introduced in Software Release 9.1) allows packets to be prioritized. When priority queuing is enabled on an interface, the router maintains up to four output queues for that interface. During congestion, the packets are placed in one of the four queues according to their priority. The router services all packets on the highest priority queue before moving on to the next highest priority queue. In other words, the queuing delay of a packet on a lower priority queue is nondeterministic: An RSRB session set to normal priority might time out if, for example, IPX packet traffic is heavy and is configured for the highest priority queue.

This scheme introduces a problem in that packets configured for lower priority queues might not be serviced in a timely manner, or at all, depending on the bandwidth used by packets sent from the higher priority queues. Priority queuing does not provide bandwidth allocation.

Priority queuing can be used when there is sufficient bandwidth to accommodate all packets destined for a particular interface, but where packets from certain protocols such as file transfers cause other protocols such as Telnet sessions to suffer from poor response.

If there is insufficient bandwidth on an output interface to pass data from various sources, priority queuing cannot solve the limited bandwidth condition. If there is not enough bandwidth to pass all of the data destined for an interface, protocols assigned to the lower priority queues suffer packet loss.

Priority queuing introduces processor overhead that might be acceptable for slow interfaces, but might be unacceptable for higher speed interfaces such as Ethernet, Token Ring, or FDDI. If you are currently fast switching packets, be aware that priority queuing requires that these packets be process switched, which would negatively impact performance.

Use the **priority-list** global configuration command to define priority lists and the **priority-group** interface command to assign a list to an interface. Priority queuing can be configured instead of, but not in addition to, custom queuing.

Note   Priority queuing does not operate over X.25.

Custom Queuing

Custom queuing (introduced in Software Release 9.21 and enhanced in Cisco IOS Software Release 11.0) allows you to allocate a percentage of bandwidth to a particular kind of traffic when the available bandwidth is unable to accommodate the aggregate traffic queued.

When custom queuing is enabled on an interface, the router maintains 16 output queues (numbered from 0 to 15) for that interface that can be used to modify queuing behavior. The router cycles through queue numbers 1 to 15 in a sequential fashion, delivering packets in the current queue before moving on to the next. Associated with each output queue is a configurable byte count, which specifies how many bytes of data should be delivered from the current queue by the router before the router moves on to the next queue.When a particular queue is being processed, packets are sent until the number of bytes sent exceeds the queue byte count or the queue is empty.

Queue number 0 is a system queue; its queue is emptied before any of the queues numbered 1 to 15 are processed. The router queues high priority packets to this queue, such as interface keepalive packets. Routing protocol packets are not automatically placed in the system queue.

The custom queuing implementation should not impact the performance of existing packet queuing code. The queuing algorithm implementation is time-critical because it affects packet delivery time when custom queuing is in use.

When custom queuing (or priority queuing) is enabled, it should take much longer for the router to switch packets because each packet has to be classified by the processor card.

Use the **queue-list** global configuration command to define custom queue lists and the **custom-queue-list** interface configuration command to assign a custom queue list to an interface. Custom queuing can be configured instead of, but not in addition to, priority queuing.

Figure 4-36 describes the syntax of the **priority-list** and **queue-list** commands.

Note   Custom queuing does not operate over X.25.

Figure 4-36   Priority and custom queuing command syntax.

| Command | List | Protocol | Queue Priority   Custom | Optional arguments |

```
Command        List        Protocol        Queue                    Optional arguments
                                        Priority    Custom

                                apollo
                                appletalk
                    1           bridge
                    2           chaos                        1
                    3           decnet                       2
                    4                           high         3       list       access-list*
priority-list       5   protocol ip          medium         .       lt         byte-count
   or               6           ipx†         normal          .      gt         byte-count
queue-list          7           pup            low          .       tcp        port-number
                    8           rsrb                         14      udp        port-number
                    9           stun                         15      bridge list access-list
                    10          vines                        16
                                xns
```

† In releases prior to Cisco IOS 10.0, the protocol argument is "novell".

*Applies only to AppleTalk, bridging, IP, IPX, VINES, and XNS.

SAP Prioritization

The purpose of the SAP prioritization feature is to allow you to specify the priority (precedence and bandwidth) of a protocol over another protocol across the RSRB/SDLLC WAN. The prioritization is based on the destination service access point (DSAP) address and source service access point (SSAP) address.

SAP prioritization can be built based on priority queuing or on custom queuing. The actual SAP classification code can be developed regardless of the underlying prioritization mechanism. The priority queuing mechanism addresses only the *precedence* criteria. The custom queuing mechanism provides *precedence* and guarantees *bandwidth*. This section describes SAP prioritization using priority queuing.

To provide a fine granularity in the prioritization of packets, the **sap priority-list** global configuration command (available in Software Release 9.1[9]) allows you to specify any combination of DSAP, SSAP, destination MAC (DMAC) address, and source MAC (SMAC) address.

For example, if you want to prioritize all SNA traffic (SAP 04) over NetBIOS traffic (SAP F0), only the DSAP or SSAP must be configured. In contrast, if you want to give precedence to traffic on a particular LLC2 session, you must specify four parameters: DSAP address, SSAP address, DMAC address, and SMAC address. Use the **sap-priority** *list* interface configuration command (available in Software Release 9.1[9]) to tie the priority list to a particular input interface.

You must also specify the **priority** option in the **source-bridge remote-peer** global configuration command to enable SAP prioritization. In addition, you must configure the **priority-list** global configuration command for the appropriate interfaces and use the **priority-group** interface configuration command on the output interface.

Enhanced LU Address Prioritization

The enhanced logical unit (LU) address-prioritization feature allows you to specify the physical unit (PU) on which an LU resides. This is important because multiple PUs on a Token Ring or on a multidropped SDLC line might have LUs with the same LU address. For example, Figure 4-37 illustrates a situation in which LU02 on 3174-2 is a 3287 printer, and LU02 on 3174-1 is a 3278 terminal. It is undesirable to assign the same priority to the printer and the terminal.

Figure 4-37   LU prioritization for RSRB.
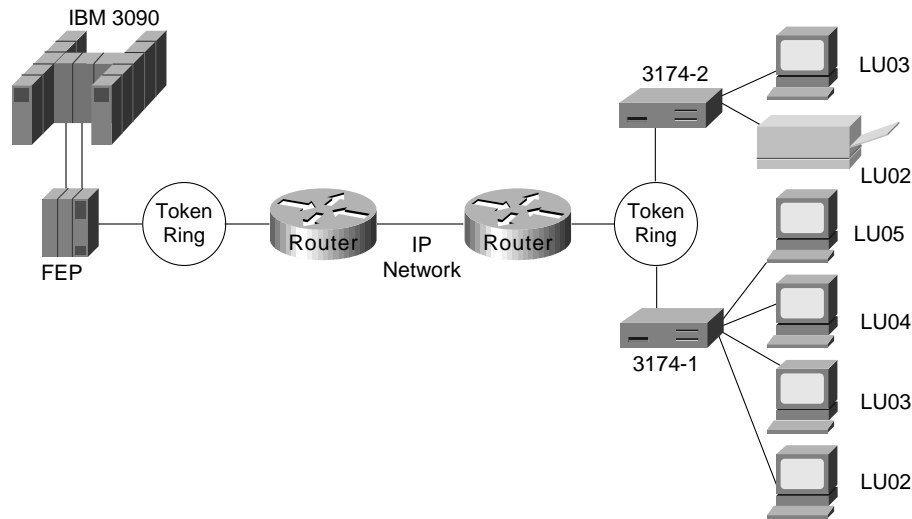
As of Software Release 9.1(9), the LU address prioritization for both RSRB and serial tunneling (STUN) allow you to prioritize the following addresses in addition to the LU address:

- In the RSRB case, you can specify the MAC and SAP address, which together uniquely identify a PU.

- In the STUN case, you can specify the SDLC address to identify a PU on a multidropped SDLC line.

SAP Filters on WAN Links

SAP filters, which are currently available for serial, Token Ring, and Ethernet interfaces, can be used to prevent local NetBIOS and other broadcasts from traversing the RSRB/SDLLC WAN. To implement SAP filter logic on the RSRB/SDLLC WAN interface, it is desirable to place the code at the RSRB level independent from the encapsulation type on the interface. The same filter code should work for direct HDLC encapsulation, TCP/IP encapsulation, and FST encapsulation. In addition to filtering by SAP address, SAP filters can also be used to filter packets by NetBIOS name.

The commands, which are available in Software Release 9.1.(9), are the same as those used for SRB on the Token Ring interface:

- The **access-list** *list* global configuration command builds the access list.

- The **rsrb remote-peer** *ring-group* interface configuration command filters by Local Service Access Point (LSAP) address or by NetBIOS station name on the RSRB WAN interface.

- The **netbios access-list** *host* global configuration command builds a NetBIOS access filter by host name.

SRB Design Checklist

Before implementing a source-route bridging (SRB) network, be sure to familiarize yourself with the technical reference material in the *Router Products Configuration Guide* and the *Router Products Command Reference* publications that deals with SRB internetworking.

Next, read the "Multiport Bridging" through "WAN Framing" sections earlier in this chapter. Depending on your implementation, you should review the "IP Routing Protocol Selection for SRB Networks" and "SRB Network Design" sections earlier in this chapter. If you require more than eight routers, continue as follows:

Step 1   Evaluate the following requirements:

- Determine which protocols are to be used. Relevant options are hierarchical Systems Network Architecture (SNA) and NetBIOS. If you are running hierarchical SNA, determine the link speeds to the core front end processor (FEP) sites.

- Determine whether parallel paths exist in the network either between individual routers or in the general network. If they do, refer to the "WAN Parallelism" section earlier in this chapter.

- Determine whether the network requires greater than 2-kilobyte frames to be sent across WAN links. If so, refer to the "WAN Frame Sizes" section earlier in this chapter.

Step 2   If the access ring and the FEP-connected sites exceed 15 token rings, you must address the following issues:

- Determine whether local acknowledgment is a requirement. Refer to the "Local Acknowledgment Recommendations" section earlier in this chapter.

- Select an encapsulation method. Refer to the "WAN Framing" section.

- Design a network topology incorporating the rules outlined in the "SRB Network Design" section.

- Select a routing protocol described in the "WAN Parallelism" and "IP Routing Protocol Selection for SRB Networks" sections.

Step 3   If performance is important for your internetwork, review the "IP Routing Protocol Selection for SRB Networks" section.

Step 4   Prepare each router's configuration for the following:

- SRB (Refer to the "Explorer Packets and Propagation" and "WAN Framing" sections.)

- IP route tuning (Refer to the "IP Routing Protocol Selection for SRB Networks" section.)

Step 5   Turn on proxy explorer as needed. Refer to the "Explorer Packets and Propagation" section.

Step 6   If the network requires NetBIOS, proceed as follows:

- Turn on NetBIOS name caching.

- Limit the explorer packet processing queue to 20 entries. Refer to the "Explorer Packets and Propagation" section.

Step 7   If you expect to exceed 250 Token Rings, contact your technical support representative for additional information.

Summary

This chapter discussed source-route bridging (SRB) and remote source-route bridging (RSRB). It addressed the challenges of this environment and helped network designers successfully implement SRB within a large, multiprotocol topology, including covering the following areas:

- SRB technology and implementation overview

- Internet Protocol (IP) routing protocol selection and implementation

- SRB network design recommendations and guidelines