Designing DLSw+ Internetworks

This chapter contains the following information:

- Introduction to DLSW+
- Getting Started with DLSW+
- DLSW+ Advanced Features

Introduction to DLSw+

This section describes Data Link Switching Plus (DLSw+) and provides configuration examples to enable you to quickly design and configure simple DLSw+ networks. It reviews the key components of the data-link switching (DLSw+) features and describes the extensions to the standard that are included in DLSw+. This section also describes advanced features, tells when to use them, and includes examples of how to use these features. It provides tuning, hierarchical design, meshed design, debug, and migration guidance. Finally, it recommends how to proceed with designing your network. This section can be used as a reference only (for configuration examples), as a tuning guide, or as a guide to design a complete DLSw+ network.

DLSw+ Defined

DLSw+ is a means of transporting Systems Network Architecture (SNA) and NetBIOS traffic over a campus or wide-area network (WAN). The end systems can attach to the network over Token Ring, Ethernet, Synchronous Data Link Control (SDLC) protocol, Qualified Logical Link Control (QLLC), or Fiber Distributed Data Interface (FDDI). (FDDI is supported on the Cisco 7000 series only and requires Cisco IOS Release11.2 or later.) DLSw+ switches between diverse media and locally terminates the data links, keeping acknowledgments, keepalives, and polling off the WAN. Local termination of data links also eliminates data-link control timeouts that can occur during transient network congestion or when rerouting around failed links. Finally, DLSw+ provides a mechanism for dynamically searching a network for SNA or NetBIOS resources and includes caching algorithms that minimize broadcast traffic.

In this document, DLSw+ routers are referred to as peer routers, peers, or partners. The connection between two DLSw+ routers is referred to as a peer connection. A DLSw+ circuit is comprised of the data-link control connection between the originating end system and the originating router, the connection between the two routers (typically a Transport Control Protocol [TCP] connection), and the data-link control connection between the target router and the target end system. A single peer connection can carry multiple circuits.

DLSw+ supports circuits between SNA physical units (PUs) or between NetBIOS clients and servers. The SNA PU connectivity supported is PU 2.0/2.1-to-PU 4 (attached via any supported data-link controls), PU 1-to-PU 4 (SDLC only), PU 4-to-PU 4 (Token Ring only), and PU 2.1-to-PU 2.1 (any supported data-link control). See Appendix B for details about DLSw+ connectivity.

Note N PU 4-to-PU 4 connectivity supports only a single path between front-end processors (FEPs) because of an idiosyncrasy in how FEPs treat duplicate source-route bridged paths. In addition, remote load is not supported.

DLSw Standard

The DLSw standard was defined at the Advanced Peer-to-Peer Networking (APPN) Implementers Workshop (AIW) in the DLSw-related interest group. The current standard is Version 1, which is documented in RFC 1795. RFC 1795 makes obsolete RFC 1434, which described IBM's original 6611 implementation of DLSw.

The DLSw standard describes the Switch-to-Switch Protocol (SSP) used between routers (called data-link switches) to establish DLSw peer connections, locate resources, forward data, handle flow control, and perform error recovery. RFC 1795 requires that data-link connections are terminated at the peer routers, that is, the data-link connections are locally acknowledged and, in the case of Token Ring, the routing information field (RIF) ends at a virtual ring in the peering router.

By locally terminating data-link control connections, the DLSw standard eliminates the requirement for link-layer acknowledgments and keepalive messages to flow across the WAN. In addition, because link-layer frames are acknowledged locally, link-layer timeouts should not occur. It is the responsibility of the DLSw routers to multiplex the traffic of multiple data-link controls to the appropriate TCP pipe and to transport the data reliably across an IP backbone.

Before any end-system communication can occur over DLSw, the following must take place:

- Establish peer connections
- Exchange capabilities
- Establish circuit

Establish Peer Connections

Before two routers can switch SNA or NetBIOS traffic, they must establish two TCP connections between them. The standard allows one of these TCP connections to be dropped if it is not required. (Cisco routers will drop the extra TCP connection unless they are communicating with another vendor's router that requires two TCP connections.) The standard also allows additional TCP connections to be made to allow for different levels of priority.

Exchange Capabilities

After the TCP connections are established, the routers exchange their capabilities. Capabilities include the DLSw version number, initial pacing windows (receive window size), NetBIOS support, list of supported link service access points (SAPs), and the number of TCP sessions supported. Media Access Control (MAC) address lists and NetBIOS name lists can also be exchanged at this time, and if desired, a DLSw partner can specify that it does not want to receive certain types of search frames. It is possible to configure the MAC addresses and NetBIOS names of all resources that will use DLSw and thereby avoid any broadcasts. After the capabilities exchange, the DLSw partners are ready to establish circuits between SNA or NetBIOS end systems.

Establish Circuit

Circuit establishment between a pair of end systems includes locating the target resource (based on its destination MAC address or NetBIOS name) and setting up data-link control connections between each end system and its data-link switch (local router). SNA and NetBIOS are handled differently. SNA devices on a LAN find other SNA devices by sending an explorer frame (a TEST or an exchange identification [XID] frame) with the MAC address of the target SNA device. When a DLSw router receives an explorer frame, the router sends a canureach frame to each of the DLSw partners. If one of its DLSw partners can reach the specified MAC address, the partner replies with an icanreach frame. The specific sequence includes a canureach ex (explorer) to find the resource and a canureach cs (circuit setup) that triggers the peering routers to establish a circuit.

At this point, the DLSw partners establish a *circuit* that consists of three connections: the two data-link control connections between each router and the locally attached SNA end system, and the TCP connection between the DLSw partners. This circuit is uniquely identified by the source and destination circuit IDs, which are carried in all steady state data frames in lieu of data-link control addresses such as MAC addresses. Each circuit ID is defined by the destination and source MAC addresses, destination and source link service access points (LSAPs), and a data-link control port ID. The circuit concept simplifies management and is important in error processing and cleanup. Once the circuit is established, information frames can flow over the circuit.

NetBIOS circuit establishment is similar, but instead of forwarding a canureach frame that specifies a MAC address, DLSw routers send a name query (NetBIOS NAME-QUERY) frame that specifies a NetBIOS name. Instead of an icanreach frame, there is a name recognized (NetBIOS NAME-RECOGNIZED) frame.

Most DLSw implementations cache information learned as part of the explorer processing so that subsequent searches for the same resource do not result in the sending of additional explorer frames.

Flow Control

The DLSw standard describes adaptive pacing between DLSw routers but does not indicate how to map this to the native data-link control flow control on the edges. The DLSw standard specifies flow control on a per-circuit basis and calls for two independent, unidirectional circuit flow-control mechanisms. Flow control is handled by a windowing mechanism that can dynamically adapt to buffer availability, TCP transmit queue depth, and end-station flow-control mechanisms. Windows can be incremented, decremented, halved, or reset to zero.

The granted units (the number of units that the sender has permission to send) are incremented with a flow-control indication from the receiver (similar to classic SNA session-level pacing). Flow-control indicators can be one of the following types:

- Repeat—Increment granted units by the current window size
- Increment—Increment the window size by one and increment granted units by the new window size
- Decrement—Decrement window size by one and increment granted units by the new window size
- Reset—Decrease window to zero and set granted units to zero to stop all transmission in one direction until an increment flow-control indicator is sent
- Half—Cut the current window size in half and increment granted units by the new window size

Flow-control indicators and flow-control acknowledgments can be piggybacked on information frames or can be sent as independent flow-control messages, but reset indicators are always sent as independent messages.

DLSw+ Features

DLSw+ is Cisco's implementation of DLSw. It goes beyond the standard to include the advanced features of Cisco's current remote source-route bridging (RSRB) and provides additional functionality to increase the overall scalability of DLSw.

DLSw+ includes enhancements in the following areas:

- Scalability—Constructs IBM internetworks in a way that reduces the amount of broadcast traffic and therefore enhances their scalability
- Availability—Dynamically finds alternate paths quickly and optionally load-balances across multiple active peers, ports, and channel gateways
- Transport flexibility—Higher-performance transport options when there is enough bandwidth to handle the traffic load without risk of timeouts, and the option to use lower-overhead solutions when bandwidth is at a premium and nondisruptive rerouting is not required
- Modes of operation—Dynamically detect the capabilities of the peer router and operate according to those capabilities

DLSw+ Improved Scalability

One of the most significant factors that limits the size of LAN internetworks is the amount of explorer traffic that traverses the WAN. There are several optimizations in DLSw+ to reduce the number of explorers.

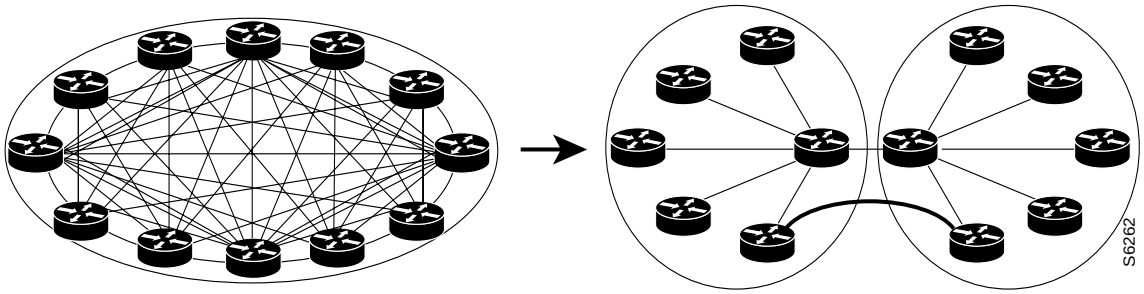
Peer Group Concept

Perhaps the most significant optimization in DLSw+ is a feature known as *peer groups.* Peer groups are designed to address the broadcast replication that occurs in a fully meshed network. When any-to-any communication is required (for example, for NetBIOS or APPN environments), RSRB or standard DLSw implementations require peer connections between every pair of routers.

This setup is not only difficult to configure, it results in branch access routers having to replicate search requests for each peer connection. This wastes bandwidth and router cycles. A better concept is to group routers into clusters and designate a focal router to be responsible for broadcast replication. This capability is included in DLSw+.

With DLSw+, a cluster of routers in a region or a division of a company can be combined into a peer group. Within a peer group, one or more of the routers are designated to be the *border peer*s. Instead of all routers peering to one another, each router within a group peers to the border peer; border peers establish peer connections with each other (see Figure 7-1). When a DLSw+ router receives a TEST frame or NetBIOS NAME-QUERY, it sends a single explorer frame to its border peer. The border peer forwards the explorer on behalf of the peer group member. This setup eliminates duplicate explorers on the access links and minimizes the processing required in access routers.

Figure 7-1 The Peer Group Concept Can Be Used to Simplify and Scale Any-to-Any Networks



Once the correct destination router is found, an end-to-end peer connection (TCP or IP) is established to carry end-system traffic. This connection remains active as long as there is end-system traffic on it, and it is dynamically torn down when not in use, permitting casual, any-to-any communication without the burden of specifying peer connections in advance. It also allows any-to-any routing in large internetworks where persistent TCP connections between every pair of routers would not be possible.

Explorer Firewalls

To further reduce the amount of explorer traffic that enters the WAN, there are a number of filter and firewall techniques to terminate the explorer traffic at the DLSw+ router. A key feature is the explorer firewall.

An explorer firewall permits only a single explorer for a particular destination MAC address to be sent across the WAN. While an explorer is outstanding and awaiting a response from the destination, subsequent explorers for that MAC address are not propagated. Once the explorer response is received at the originating DLSw+, all subsequent explorers receive an immediate local response. This eliminates the start-of-day explorer storm that many networks experience.

DLSw+ Enhanced Availability

One way DLSw+ offers enhanced availability is by maintaining a reachability cache of multiple paths for local and remote destination MAC addresses or NetBIOS names. For remote resources, the path specifies the peer to use to reach this resource. For local resources, the path specifies a port

number. If there are multiple paths to reach a resource, the router will mark one path preferred and all other paths capable. If the preferred path is not available, the next available path is promoted to the new preferred path, and recovery over an alternate path is initiated immediately.

The way that multiple capable paths are handled with DLSw+ can be biased to meet the needs of the network:

- Fault tolerance—Biases circuit establishment over a preferred path, but also rapidly reconnects on an active alternate path if the preferred path is lost
- Load balancing—Distributes circuit establishment over multiple DLSw+ peers in the network or ports on the router

The default for DLSw+ is to use fault-tolerant mode. In this mode, when a DLSw+ peer receives a TEST frame for a remote resource, it checks its cache. If it finds an entry and the entry is fresh (that is, if it is not verified within the last verify interval), the DLSw+ peer responds immediately to the test frame and does not send a canureach frame across the network. If the cache entry is stale, then the originating DLSw+ peer sends a canureach directly to each peer in the cache to validate the cache entries (this is known as a directed verify). If any peer does not respond, it is deleted from the list. This may result in reordering the cache. The SNA-VERIFY-INTERVAL is configurable and is the length of time a router waits before marking the cache entry stale. The SNA-CACHE-TIMEOUT is the interval that cache entries are maintained before they are deleted. It defaults to 16 minutes and is configurable.

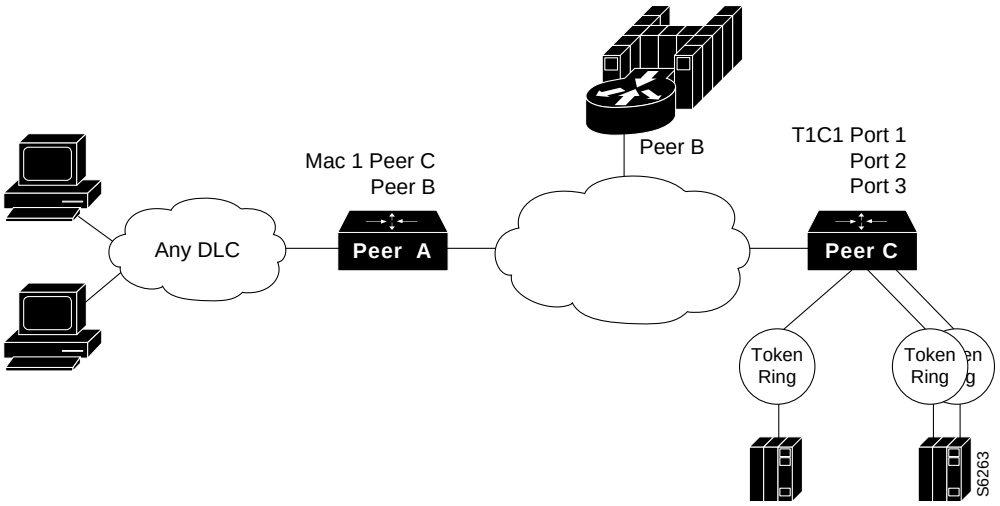
At the destination DLSw+ router, a slightly different procedure is followed using the local cache entries. If the cache entry is fresh, the response is sent immediately. If the cache entry is stale, a single route broadcast test frame is sent over all the ports in the cache. If a positive response is received, an icanreach frame is sent to the originating router. Test frames are sent every 30 seconds (SNA-RETRY-INTERVAL) for a three-minute period (SNA-EXPLORER-TIMEOUT). These timers are configurable.

Alternately, when there are duplicate paths to the destination end system, you can configure load balancing, which causes DLSw+ to alternate new circuit requests in a round-robin fashion through the list of capable peers or ports.

This feature is especially attractive in SNA networks. A very common practice used in the hierarchical SNA environment is assigning the same MAC address to different mainframe channel gateways—for example, FEPs or Cisco routers with Channel Interface Processors (CIPs). If one channel gateway is unavailable, alternate channel gateways are dynamically located without any operator intervention. Duplicate MAC addressing also allows load balancing across multiple active channel gateways or Token Ring adapters.

DLSw+ ensures that duplicate MAC addresses are found, and it caches up to four DLSw peers or interface ports that can be used to find the MAC address. This technique can be used for fault tolerance and load balancing. When using this technique for fault tolerance, it facilitates a timely reconnection after circuit outages. When using this technique for load balancing, it improves overall SNA performance by spreading traffic across multiple active routers, Token Ring or FDDI adapters, or channel gateways, as shown in Figure 7-2. Load balancing not only enhances performance, it also speeds up recovery from the loss of any component in a path through the network because a smaller portion of the network is affected by the loss of any single component.

Figure 7-2 DLSw+ Caching Techniques Provide Load Balancing across Multiple Central Site Routers, Token Rings, and Channel Gateways



In addition to supporting multiple active peers, DLSw+ supports *backup peers*, which are only connected when the primary peer is unreachable.

DLSw+ Transport Flexibility

The transport connection between DLSw+ routers can vary according to the needs of the network and is not tied to TCP/IP as the DLSw standard is. Cisco supports four different transport protocols between DLSw+ routers:

- TCP/IP—Transports SNA and NetBIOS traffic across WANs where local acknowledgment is required to minimize unnecessary traffic and prevent data-link control timeouts and where nondisruptive rerouting around link failures is critical; this transport option is required when DLSw+ is operating in DLSw standard mode.
- FST/IP—Transports SNA and NetBIOS traffic across WANs with an arbitrary topology; this solution allows rerouting around link failures, but recovery may be disruptive depending on the time required to find an alternate path; this option does not support local acknowledgment of frames.
- Direct—Transports SNA and NetBIOS traffic across a point-to-point or Frame Relay connection where the benefits of an arbitrary topology are not important and where nondisruptive rerouting around link failures is not required; this option does not support local acknowledgment of frames.
- DLSw Lite—Transports SNA and NetBIOS traffic across a point-to-point connection (currently only Frame Relay is supported) where local acknowledgment and reliable transport are important, but where nondisruptive rerouting around link failures is not required; DLSw Lite uses RFC 1490 encapsulation of Logical Link Control type 2 (LLC2).

DLSw+ Modes of Operation

Cisco has been shipping IBM internetworking products for several years. There is a substantial installed base of Cisco routers running RSRB today. Therefore, it is essential for DLSw+ and RSRB to coexist in the same network and in the same router. In addition, because DLSw+ is based on the new DLSw standard, it must also interoperate with other vendors' implementations that are based on that DLSw standard.

There are three different modes of operation for DLSw+:

- Dual mode—A Cisco router can communicate with some remote peers using RSRB and with others using DLSw+, providing a smooth migration path from RSRB to DLSw+. In dual mode, RSRB and DLSw+ coexist on the same box; the local peer must be configured for both RSRB and DLSw+; and the remote peers must be configured for either RSRB or DLSw, but not both.
- Standards compliance mode—DLSw+ can detect automatically (via the DLSw capabilities exchange) if the participating router is manufactured by another vendor, therefore operating in DLSw standard mode.
- Enhanced mode—DLSw+ can detect automatically that the participating router is another DLSw+ router, therefore operating in enhanced mode, making all the features of DLSw+ available to the SNA and NetBIOS end systems.

Some of the enhanced DLSw+ features are also available when a Cisco router is operating in standards-compliance mode with another vendor's router. In particular, enhancements that are locally controlled options on a router can be accessed even though the remote router does not have DLSw+. These enhancements include load balancing, local learning (the ability to determine if a destination is on a LAN before sending canureach frames across a WAN), explorer firewalls, and media conversion.

How to Proceed

If you have a simple hierarchical network with a small volume of SNA traffic, read the "Getting Started with DLSw+" section, which describes what configuration commands are required in all DLSw+ implementations and provides configuration examples for SDLC, Token Ring, Ethernet, and QLLC. After reading the "Getting Started" section, you can read about advanced features, customization, and bandwidth management.

If you have a large hierarchical network (hundreds of branch offices), read the "Designing Hierarchical Networks" chapter, which will tell you how to determine the correct number and types of routers to place at the central site and discusses options for peer placement, peer backup, and broadcast reduction.

If you require any-to-any communication between NetBIOS or APPN applications, read the "Designing Meshed Networks" chapter, which describes border peer placement, numbers of peers per group, and how to minimize broadcast replication.

If you are starting with an RSRB network, read the "Migration and Interoperability" chapter.

The "Using Show and Debug Commands" and "Using Maps, SNA View, and Native Service Point" chapters describe network management capabilities available with DLSw and should be read by all DLSw+ users.

Finally, the "Using DLSw+ with Other Features" chapter describes how to use DLSw+ in conjunction with downstream physical unit (DSPU) concentration, LAN Network Manager, APPN, and native client interface architecture (NCIA).

Getting Started with DLSw+

This section describes the basic configuration commands required for a DLSw+ network. It begins with a description of the minimum required configuration and then provides examples for Token Ring, Ethernet, SDLC, and QLLC environments. If you are unfamiliar with router configuration, you should also review the examples in Appendix A. These examples illustrate how to configure not only routers, but also the attaching end systems. They show how to configure canonical addresses, static routes, and loopback addresses.

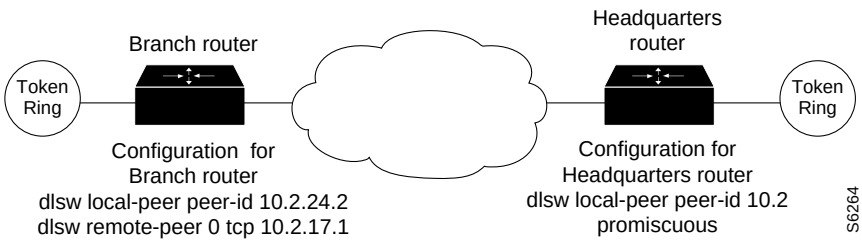
Minimum Required Configuration

Configuring DLSw+ on most networks is not difficult. Every router that supports DLSw+ must have a **dlsw local-peer** command; **dlsw remote-peer** commands are optional, but usually at least one side of a peer connection must configure a remote peer. If a DLSw+ peer configuration omits **dlsw remote-peer** commands, the **dlsw local-peer** command must specify the **promiscuous** keyword. Promiscuous routers will accept peer connection requests from routers that are not preconfigured. This feature allows you to minimize changes to central site routers when branch offices are added or deleted. It also minimizes required coordination of configurations.

If you have used RSRB in the past, you need to know what *not* to configure. With DLSw+, you do not need proxy explorer, NetBIOS name caching, SDLC-to-LLC2 conversion (SDLLC), or source-route translational bridging (SR/TLB). All of these features are built into DLSw+.

In Figure 7-3, the branch router specifies both a **dlsw local-peer** and a **dlsw remote-peer** command. The headquarters router specifies only a **dlsw local-peer** command, but it specifies **promiscuous** on the **dlsw local-peer** command to allow it to dynamically accept connections from branch routers. The peer ID specified on the **dlsw local-peer** command is the router's IP address. It can be a loopback address configured via **interface loopback 0** or the IP address associated with a specific LAN or WAN interface. However, if you use a LAN or WAN IP address, the interface must be up for DLSw to work.

Figure 7-3 Example of dlsw local-peer and dlsw remote-peer Commands



The number following **dlsw remote-peer** is the ring list number. Ring lists are an advanced topic, so for now, specify zero in this space, which indicates that ring lists are not in use. There are other options on the **dlsw local-peer** and **dlsw remote-peer** commands, but they are not required. These options are covered in the "DLSw+ Advanced Features" section.

In addition to specifying local and remote peers, you must map the following local data-link controls to DLSw:

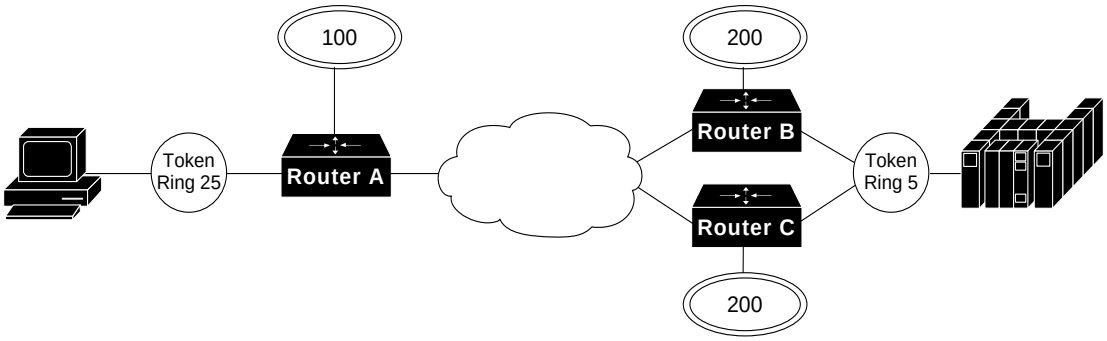
- Token Ring—Define a virtual ring using the source-bridge ring-group command and include a source-bridge command that tells the router to bridge from the external Token Ring to that virtual ring.
- Ethernet—Map a specific Ethernet bridge group to DLSw.
- SDLC—Define the SDLC devices and map the SDLC addresses to DLSw+ virtual MAC addresses.
- QLLC—Define the X.25 devices and map the X.25 addresses to DLSw+ virtual MAC addresses.
- FDDI—Define a virtual ring using the source-bridge ring-group command and include an SRB statement that tells the router to bridge from the external FDDI to that virtual ring; FDDI is supported in Cisco IOS Release 11.2 on the Cisco 7000 series.

The rest of this section provides sample configurations for Token Ring, Ethernet, SDLC, and QLLC.

Token Ring

Figure 7-4 shows a sample DLSw+ configuration for Token Ring. Traffic that originates on Token Ring is source-route bridged from the local ring onto a source-bridge ring group and then picked up by DLSw+. You must include a **source-bridge ring-group** command that specifies a virtual ring number. In addition, you must include a **source-bridge** command that tells the router to bridge from the physical Token Ring to the virtual ring.

Figure 7-4 Simple Token Ring DLSw+ Configuration



DLSw+ supports RIF termination, which means that all remote devices appear to be attached to the virtual ring specified in the **source-bridge** command. In Figure 7-6, from the host end, all the devices attached to Router A would appear to reside on Virtual Ring 200. Conversely, from the remote site, the FEP would appear to reside on Virtual Ring 100. As illustrated in this figure, the virtual rings specified in peer routers do not have to match. If multiple routers are attached to the same physical ring, as shown in Routers B and C, by specifying the same ring group number in each of them, you can prevent explorers from coming in from the WAN and being forwarded back onto the WAN.

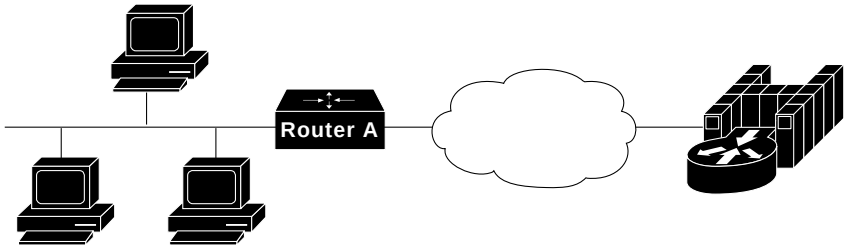
Ethernet

Traffic that originates on Ethernet is picked up from the local Ethernet bridge group and transported across the DLSw network. DLSw always transfers data in noncanonical format. In Figure 7-7, you do not need to configure the left router for translational bridging or worry about what media resides on the other side of the WAN. DLSw will automatically make the correct MAC address conversion depending on the destination media. When DLSw+ receives a MAC address from an Ethernet-attached device, it assumes it is canonical and converts it to noncanonical for transport to the remote peer. At the remote peer, the address is either passed unchanged to Token Ring-attached end systems or converted back to canonical if the destination media is Ethernet. Note that when an SNA resource resides on Ethernet, if you configure a destination SNA address in that device, you

must use canonical format. For example, Ethernet-attached 3174s must specify the MAC address of the FEP in canonical format. If the Token Ring or noncanonical format of the MAC address of the FEP is 4000.3745.0001, the canonical format is 0200.ECA2.0080

In Figure 7-5, the data is transferred directly to a Cisco router with a Channel Interface Processor (CIP), but it could be any DLSw-compliant router, and the upstream SNA end system could reside on any supported media.

Figure 7-5 Simple Ethernet DLSw+ Configuration




```
dlsw local-peer peer-id 10.2.17.1
dlsw remote-peer 0 tcp 10.2.24.2
.
.
dlsw bridge-group 1
interface Ethernet0
no ip address
bridge-group 1
bridge 1 protocol ieee
```

```
source-bridge ring-group 200
dlsw remote-peer 0 tcp 10.2.24.2
promiscuous
.
.
interface channel 0/1
csna 0100 40
csna 0100 41
int chan 0/2
lan tokenring0
source-bridge 1000 1 200
adaptef 0 4000.0000.0401
adapter 0 4000.0000.0403
```

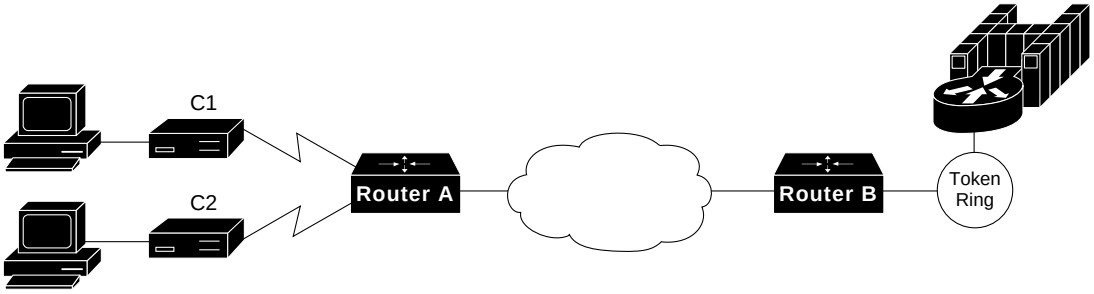
S6266

SDLC

Configuring SDLC devices is a bit more complicated. For SDLC devices, you must know whether the device is a PU 1, PU 2.0, or PU 2.1. For PU 2.0 devices, you must know the IDBLK and IDNUM that was specified in the virtual telecommunications access method (VTAM) for that device because the router plays a greater role in XID processing when SDLC PU 2.0 is involved. You must know if the router is the primary or secondary end of the SDLC line. In addition, if the attachment to the upstream SNA device is over a LAN, you must configure the MAC address of the destination upstream SNA device. In all cases, you must configure a virtual MAC address that will be mapped to an SDLC polling address.

In Figure 7-6, the SDLC-attached devices are each given a common base virtual MAC address of 4000.3174.0000. The router will replace the last two digits of the virtual MAC address with the SDLC address of the device. The device at SDLC address C1 appears to have MAC address 4000.3174.00C1, and the device at SDLC address C2 appears to have MAC address 4000.3174.00C2. In this example, both devices are PU 2.0 devices, so their XID must be configured and it must match what is specified as the IDBLK and IDNUM in VTAM. In addition, the router always assumes the primary role when attaching upstream from PU 2.0 devices.

Figure 7-6 Simple SDLC DLSw+ Configuration



Configuration for Router A
dlsw local-peer peer-id 10.2.17.1
dlsw remote-peer 0 tcp 10.2.24.2
Interface serial 0
encapsulation sdlc
sdlc role primary
sdlc vmac 4000.3174.0000
sdlc address c1
sdlc xid c1 01712345
sdlc partner 4000.3745.0001 c1
sdlc dlsw c1

interface serial1
encapsulation sdlc
sdlc role primary
sdlc vmac 4000.3174.1000
sdlc address c2
sdlc xid c1 01767890
sdlc partner 4000.3745.0001 c2
sdlc dlsw c2

S6267

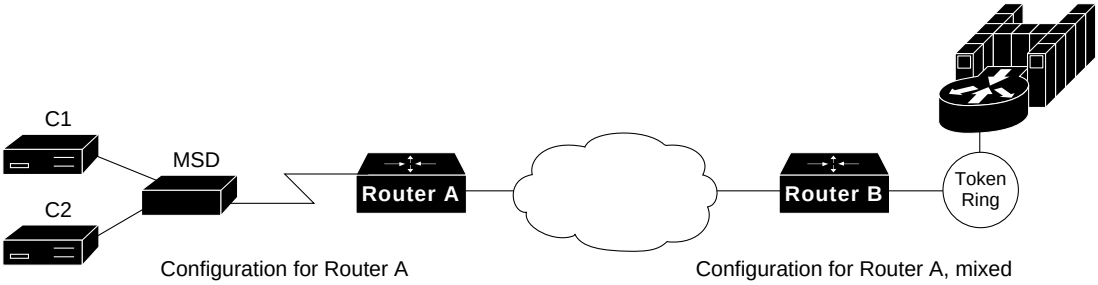
The router can be the secondary end of an SDLC line (for example, when connecting to a FEP over SDLC). In this case, specify **secondary** in the **sdlc role** command, and for PU 2.1 devices, specify **xid-passthru** in the **sdlc address** command.

In Cisco IOS Release 11.0 and later, DLSw+ supports multidrop PU 2.0/2.1. In Figure 7-7, the multidrop PU 2.0 configuration includes an **sdlc xid** command for each PU 2.0 device.

For multidrop lines with a mix of PU 2.1 and 2.0 devices, specify **primary** in the **sdlc role** command. For PU 2.0 devices, you must code the IDBLK and IDNUM in the **sdlc xid** command. For PU 2.1 devices, you can omit the **sdlc xid** command. However, in the **sdlc address** command, you need to specify **xid-poll**.

Alternately, when all devices on a line are PU 2.1, you can specify **sdlc role prim-xid-poll**, in which case you do not need to specify **xid-poll** in each **sdlc address** command.

Figure 7-7 Multidrop SDLC DLSw+ Configuration



Configuration for Router A
Both C1 and C2 are PU 2.0

```
dlsw local-peer peer-id 10.2.17.1
dlsw remote-peer 0 tcp 10.2.24.2
Interface serial 0
mtu 4400
no ip address
encapsulation sdlc
no keepalive
clockrate 19200
sdlc role primary
sdlc vmac 4000.3174.0000
sdlc address  c1
sdlc xid c1 01712345
sdlc partner 4000.3745.0001 c1
sdlc address  c2
sdlc xid c1 01767890
sdlc partner 4000.3745.0001 c2
sdlc dlsw c1 c2
```

Configuration for Router A, mixed
2.0 and 2.1

```
interface serial 0
…
sdlc role primary
sdlc vmac 4000.3174.0000
sdlc address  c1 xid poll
sdlc partner 4000.3745.0001 c1
sdlc address  c2
sdlc xid c1 01767890
sdlc partner 4000.3745.0001 c2
sdlc dlsw c1 c2
```

Configuration for Router A all PL

```
interface serial 0
…
sdlc role prim-xid-poll
sdlc Vamc4000.3174.000
sdlc address c1
sdlc partener4000.3745.000c1
sdlc address c2
sdlc partner4000.3745.000
```

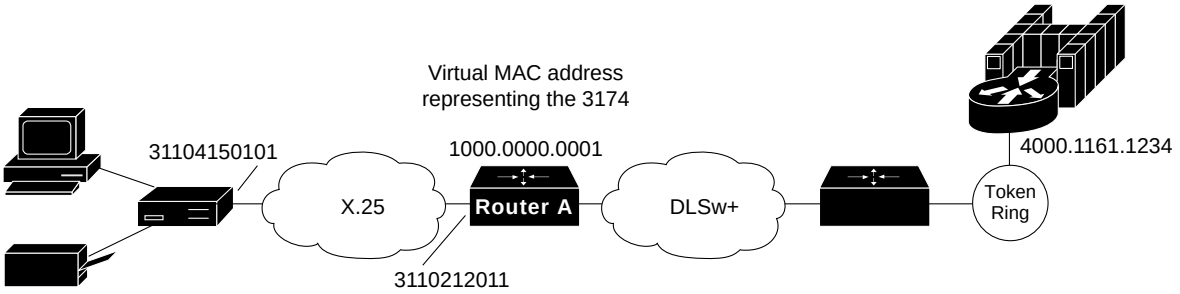
S6268

QLLC

QLLC is the data link used by SNA devices when connecting to X.25 networks. QLLC is a legacy protocol developed by IBM to allow the Network Control Program (NCP) to support remote connections over X.25. The software feature on NCP that supports QLLC is called Network Packet Switching Interface. The QLLC protocol derives its name from using the Q-bit in the X.25 header to identify QLLC protocol primitives. QLLC essentially emulates SDLC over X.25. Thus, DLSw+ performs QLLC conversion in a manner similar to SDLC conversion. Cisco's DLSw+ implementation added support for QLLC in Cisco IOS Release 11.0. Because QLLC is more complicated than Token Ring, Ethernet, or SDLC, three examples are included here.

Figure 7-8 shows DLSw+ being used to allow remote devices to connect to a DLSw+ network over an X.25 public packet switched network. In this example, all QLLC traffic is addressed to destination address 4000.1161.1234, which is the MAC address of the FEP. The remote X.25-attached 3174 is given a virtual MAC address of 1000.0000.0001. This virtual MAC address is mapped to the X.121 address of the 3174 (31104150101) in the X.25 attached router.

Figure 7-8 QLLC DLSw+ Configuration to a Single LAN-Attached Upstream Device



Configuration for Router A

```
dlsw local-peer peer-id 10.2.17.1
dlsw remote-peer 0 tcp 10.2.24.2
Interface serial 0
encapsulation x25
x25 address 3110212011
x25 map qllc 1000.0000.0001 31104150101
qllc dlsw partner 4000.1161.1234
```

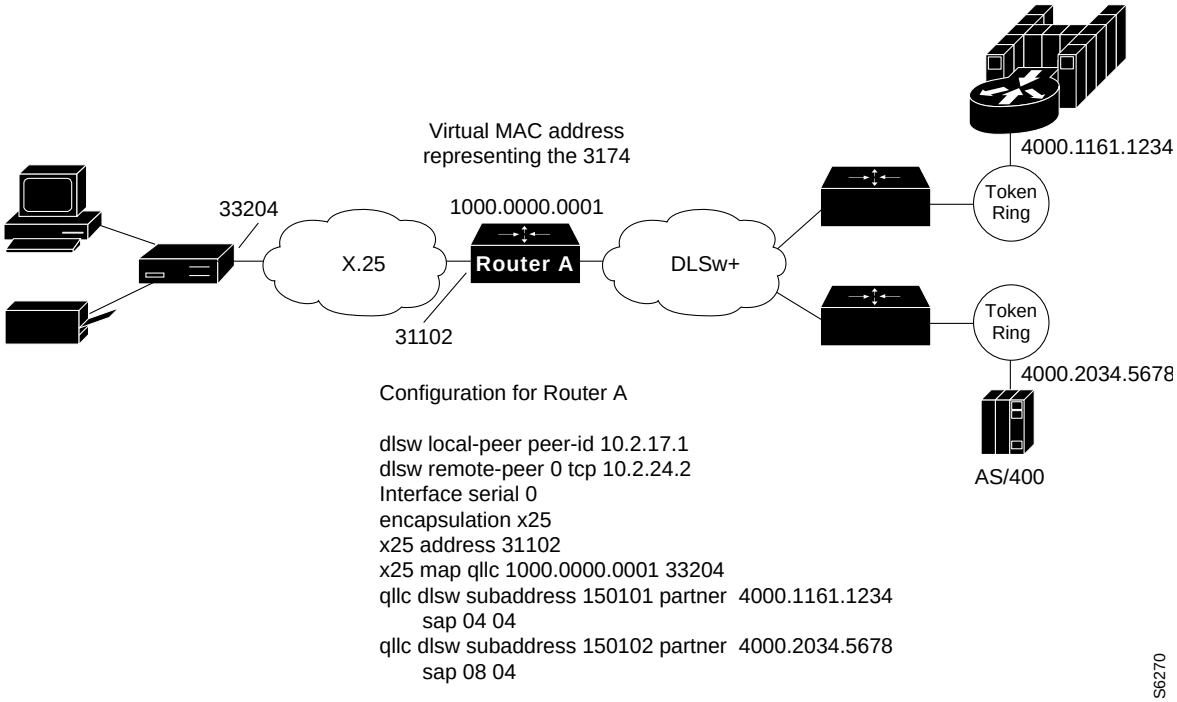
S6269

In Figure 7-9, a single 3174 needs to communicate with both an AS/400 and a FEP. The FEP is associated with subaddress 150101, and the AS/400 is associated with subaddress 150102.

If an X.25 call comes in for 33204150101, the call is mapped to the FEP and forwarded to MAC address 4000.1161.1234. The 3174 appears to the FEP as a Token Ring-attached resource with MAC address 1000.0000.0001. The 3174 uses a source SAP of 04 when communicating with the FEP.

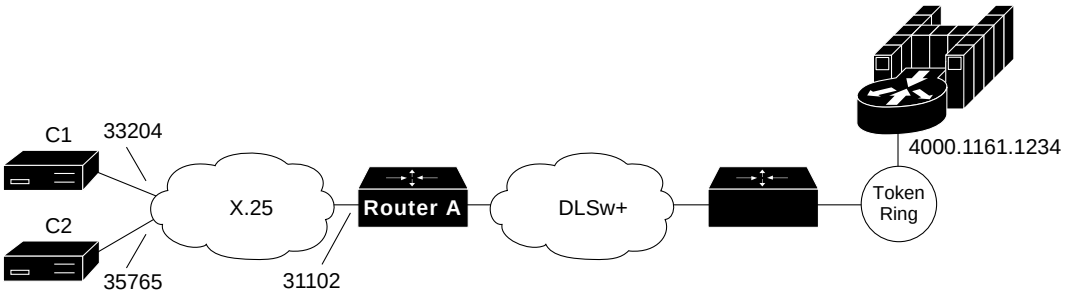
If an X.25 call comes in for 33204150102, the call is mapped to the AS/400 and forwarded to MAC address 4000.2034.5678. The 3174 appears to the AS/400 as a Token Ring-attached resource with MAC address 1000.0000.0001. The 3174 uses a source SAP of 08 when communicating with the AS/400.

Figure 7-9 QLLC DLSw+ Configuration for Support of Multiple Upstream LAN-Attached Devices



In Figure 7-10, two X.25 resources want to communicate over X.25 to the same FEP. In the router attached to the X.25 network, every X.25 connection request for X.121 address 31102150101 is directed to DLSw+. The **qllc dlsw** command creates a pool of two virtual MAC addresses, starting with 1000.0000.0001. The first switched virtual circuit (SVC) established will be mapped to virtual MAC address 1000.0000.0001. The second SVC will be mapped to virtual MAC address 1000.0000.0002.

Figure 7-10 QLLC DLSw+ Configuration for Support of Multiple Downstream X.25-Attached Devices Communicating through an Upstream DLSw+ Network



Configuration for Router A

```
dlsw local-peer peer-id 10.2.17.1
dlsw remote-peer 0 tcp 10.2.24.2
Interface serial 0
encapsulation x25
x25 address 31102
x25 map qllc 33204
x25 map qllc 35765
qllc dlsw subaddress 150101 vmacaddr
      1000.0000.0001 2 partner  4000.1161.1234
```

S6271

DLSW+ Advanced Features

This section describes advanced features of DLSw+, the benefits they provide, and a brief description of when and how to use them. Use this section to determine which options you want to use and to learn how to configure those options to address your requirements.

DLSw+ includes features to enhance availability (load balancing, redundancy, and backup peers), improve performance (encapsulation options), minimize broadcasts (ring lists), and build meshed networks (border peers and peer groups). DLSw+ also provides a feature to maximize central site resources and minimize carrier costs (dynamic peers).

Advanced features are optional and do not apply in all networks. Each feature includes a description of where it should be used.

Background

To understand load balancing, it is useful to understand how DLSw+ peers establish peer connections and find resources. When DLSw+ routers are activated, the first thing they do is establish peer connections with each configured remote peer (unless **passive** is specified, in which case a peer will wait for the remote peer to initiate a peer connection). The routers then exchange their capabilities. Included in the capabilities exchange are any resources configured in **dlsw icanreach** or **dlsw icannotreach** commands. After the capabilities exchange, the DLSw+ peers are idle until an end system sends an explorer frame (explorer frames are SNA TEST or XID frames or NetBIOS NAME-QUERY or ADD NAME-QUERY frames). Explorer frames are forwarded to every active peer and any local ports (other than the port it was received on). It is possible that an end system can be found through multiple remote peers or local ports. The path selected for a given circuit depends on certain advanced configuration options described in this section.

Load Balancing and Redundancy

If you have multiple central site routers supporting DLSw+ for either load balancing or redundancy, this section contains important information . It describes how to balance traffic across multiple central site routers or multiple ports on a single router. Load balancing in this case does not refer to balancing traffic across multiple WAN links or IP paths. That load balancing is done by the underlying IP protocol and is transparent to DLSw+.

If DLSw+ gets multiple positive replies to an explorer, it will cache up to four peers that can be used to reach a remote end system and up to four ports that can be used to reach a local end system. How these cache entries are used depends on whether load balancing is specified on the **dlsw duplicate-path-bias** command. If load balancing is specified, then each new circuit request is established over the next path (remote peer or local port) in the cache in a round-robin fashion.

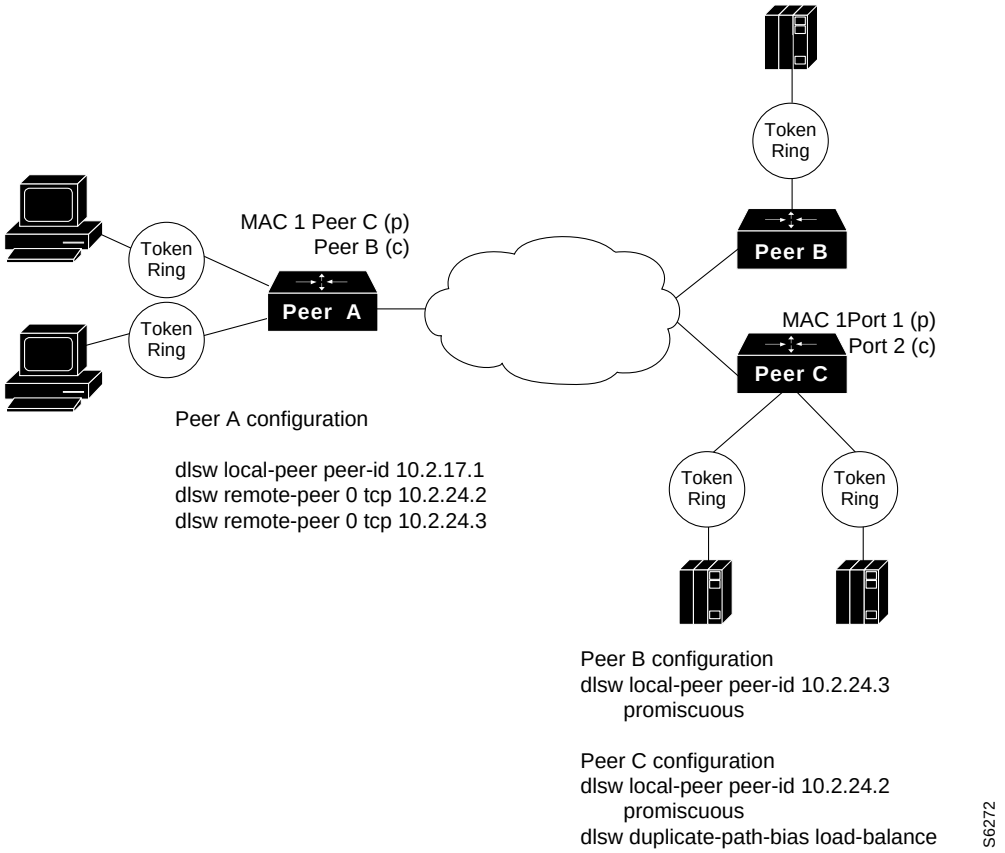
If load balancing is not specified, then the peer selects the first path in the cache and sets up all circuits via that path unless the path is unavailable. The first path in the cache list can be one of the following:

- Peer from which the first positive response was received
- Peer with the least cost
- Port over which the first positive response was received

Cost can be specified on either a **dlsw local-peer** or a **dlsw remote-peer** command. When specified on a **dlsw local-peer** command, it is exchanged with remote DLSw+ peers as part of the capabilities exchange. The following example shows how cost can be used to control which path sessions use.

In Figure 7-11, there are two channel gateways and three Token Ring adapters that can be used to access mainframe applications. All three adapters have been assigned the same MAC address. Assigning duplicate addresses is a common technique for providing load balancing and redundancy in SRB environments. It works because SRB assumes that there are three paths to find the same device and not duplicate LAN addresses. (This technique does not work with transparent bridging.)

Figure 7-11 Possible Configuration and the Resulting Cache Entries Created if All Channel Gateways Illustrated Have the Same MAC Address



In this example, Peer A has **dlsw remote-peer** commands for both Peer B and Peer C. Peer B specifies a cost of 4 in its **dlsw local-peer** command and Peer C specifies a cost of 2. This cost information is exchanged with Peer A during the capabilities exchange.

When the SNA end system (that is, the PU) on the left sends an explorer packet, Peer A forwards the explorer to both Peer B and Peer C. Peer B and Peer C forward the explorer on their local LAN. Peer B will receive a positive reply to the explorer and send a positive response back to Peer A. Peer C will receive two positive replies (one from each port) and will send a positive reply back to Peer A. Peer C records that it has two ports it can use to reach the MAC address of the channel gateway, and Peer A records that it has two peers it can use to reach the MAC address of the channel gateway.

Peer A will forward a positive response to the SNA PU and then establish an end-to-end circuit using Peer C. Peer C is selected because Peer C has a lower cost specified. When the next PU attempts to set up a connection to the same MAC address, it will be set up using Peer C, if available. This is the default method to handle duplicate paths in DLSw+.

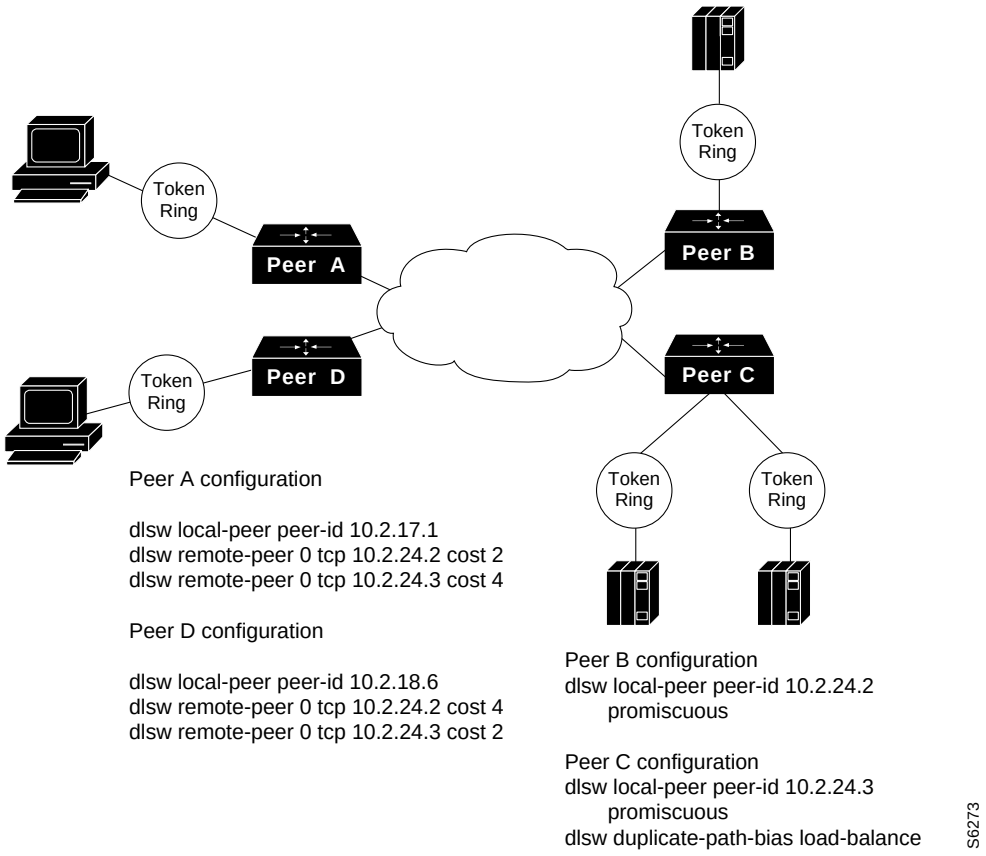
At Peer C, the first circuit will be established using Port 1, but the next circuit will use Port 2. This is because Peer C has specified load balancing in the **dlsw duplicate-path-bias** command. Each new SNA PU will use the next path in the list in a round-robin fashion.

Figure 7-11 shows how to cause all remote connections to prefer one peer over another, but the central site load balances traffic across all the LAN adapters on a given channel gateway. Alternately, load balancing can be specified everywhere to load balance traffic across all central site routers, channel gateways, and LANs. Note that this feature does not require the end systems to be Token

Ring-attached. The remote end systems can connect over SDLC, Ethernet, or QLLC, and this feature will still work. The central site channel gateway must be LAN-attached (preferably Token Ring-attached). Duplicate MAC addresses for channel gateways on Ethernet will only work if: 1) you have a unique bridged Ethernet segment and a unique DLSw+ router for each duplicate MAC address, and 2) you load balance from the remote sites. (Ethernet has no provision to prevent loops, so care must be taken when building redundant networks with Ethernet LANs. Token Ring networks can rely on SRB for loop prevention.)

An alternate way to specify cost is to use the **dlsw remote-peer** command as shown in Figure 7-12. Specifying **cost** in the **dlsw remote-peer** commands allows different divisions or parts of the country to favor different central site gateways. In addition, you must specify **cost** if you want to split SNA traffic across multiple central site routers, but each remote site has only a single SNA PU (all logical unit sessions flow over the same circuit that the PU session flows over). In Figure 7-12, Peer A always favors Peer B and Peer D always favors Peer C.

Figure 7-12 Configuration Where Cost Is Specified in the dlsw remote-peer Command instead of the dlsw local-peer Command



Controlling Peer Selection

A higher-cost peer can be used for a connection even when the lower-cost peer is active, if the higher-cost peer responds to the explorer before the lower-cost peer. If your network configuration allows this possibility, you can prevent it by adjusting a timer.

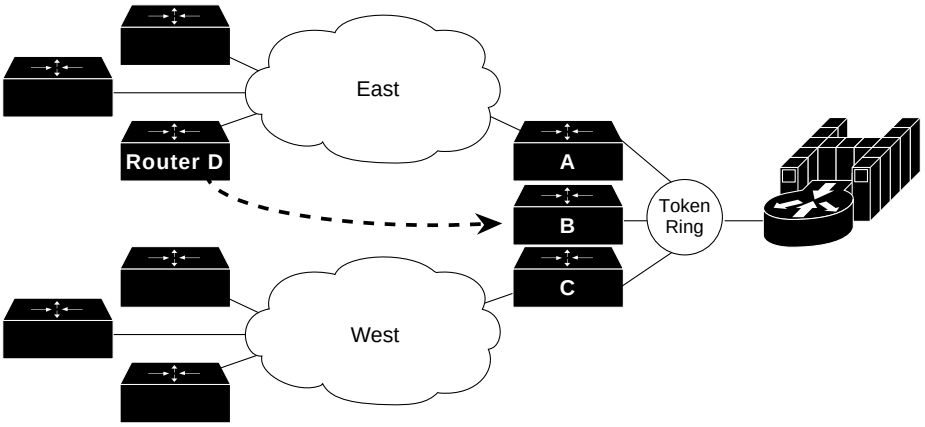
Setting the **dlsw explorer-wait-time** command causes DLSw+ to wait the specified amount of time (for example, one second) before selecting a peer to use for connections.This timer can be set in Cisco IOS Release 11.0 and later. Prior to Cisco IOS Release 11.0, this timer did not exist.

Backup Peers

Having multiple active peers is one way to provide dynamic and immediate recovery from the loss of a central site router. However, in some configurations you may prefer the alternate peer to be active only when required. This may be the case when the backup router resides at a disaster recovery site, or when there are more than 300 to 400 remote sites and a single central site router is providing backup for multiple central site routers.

In this case, use the backup peer capability (first available in Cisco IOS Release 10.3, but enhanced in Release 11.1). Figure 7-13 illustrates how to configure a backup peer. To use backup peers, the encapsulation method used to access the primary peer must be either TCP or Fast-Sequenced Transport (FST).

Figure 7-13 How to Use Backup Peers to Enhance Availability in a Large DLSw+ Network



In this example, there are 400 remote sites. All the routers on the East Coast use Router A as the primary router, and all the routers on the West Coast use Router C as the primary router. In either case, the backup router is Router B. The configuration shown is the configuration in Router D, an East Coast router. (All the East Coast routers will have the same two **dlsw remote-peer** commands.) Both the primary router (Router A) and the backup router (Router B) are configured in **dlsw remote-peer** commands. Router B is configured as a backup only, and the IP address of the router it is backing up is specified.

In the event of a failure in Router A, all SNA sessions are terminated and will reestablish through Router B. When Router A becomes available again, all new sessions are established through Router A, but sessions active on Router B will remain on Router B until the linger timer expires. Omitting the **linger** keyword will cause sessions on Router B to remain active until they terminate

on their own. The **linger** keyword can be used to minimize line costs if the backup peer is accessed over dial lines, but will provide enough time for an operator warning to be sent to all the SNA end users.

Note Prior to Cisco IOS Release 11.1, when the primary peer was activated again, all sessions using the backup peer were terminated immediately and reestablished over the primary router. If that is not the action you want to take, and you are running a level of Cisco IOS software earlier than Release 11.1, consider using duplicate active peers instead (described in the previous section).

Backup Peers Compared to Multiple Active Peers

Backup peers and multiple active peers (with one preferred and others capable) are two ways to ensure that a capable peer can back up the failure of a primary peer. One of the key differences in backup peers is that the peer connections are not active until they are needed. Suppose you have 1000 branch offices, and you want to design a network at minimal cost that will recover dynamically from the failure of any single central site router. Assume four routers at the central site can handle your traffic load. You can install four primary routers at the central site and define 250 branches to peer to each central site router.

To address your availability requirement, one option is multiple concurrently active peer connections. In this case, you would configure each remote router to have two peer connections, one to a preferred router and one to a capable router. The preferred router is the router configured with lower cost. The capable router can be the same router for all remote sites, but in that case, it would have 1000 peer connections. The largest number of peering routers we have seen is 400, and that was in an environment with extremely low traffic. Although 1000 idle peer connections are conceivable, as soon as the capable router takes over for another router, those peer connections could put a strain on the router. The other alternative is to have multiple central site routers as capable routers, but this is not the most cost-effective design.

By using a backup peer statement in each remote branch instead of concurrently peering to two routers, a single backup router at a central site can easily back up any other central site router. There is no work on a backup router until a primary router fails.

Encapsulation Options

DLSw+ offers four different encapsulation options. These options vary in terms of the processing path they use, their WAN overhead, and the media they support. The encapsulation options are TCP, Fast Sequenced Transport (FST), direct, and LLC2.

TCP Encapsulation

TCP is the standard DLSw encapsulation method and is the only encapsulation method supported by RFC 1795. TCP offers the most functionality of the encapsulation options. It provides reliable delivery of frames and local acknowledgment. It is the only option that offers nondisruptive rerouting around link failures. With TCP encapsulation, you can take advantage of dial-on-demand to dynamically dial additional bandwidth if primary links reach a preconfigured amount of congestion. In most environments, it is the recommended encapsulation because its performance is generally more than adequate, it offers the highest availability, and the overhead generally has no negative impact on response time or throughput.

TCP is process switched, so it uses more cycles than FST or direct encapsulation. A Cisco 4700 router running DLSw+ with TCP encapsulation can switch up to 8 Mbps of data, so TCP encapsulation addresses the processing requirements of most SNA environments. Where higher throughput is required, additional routers or alternate encapsulation options can be used.

TCP encapsulation adds the most overhead to each frame (20 bytes for TCP and 20 bytes for IP in addition to the 16-byte DLSw header). TCP header compression or payload compression can be used to reduce the amount of bandwidth required, if necessary. At 56 kbps or higher line speeds, the 40 bytes of overhead adds less than 5.7 ms to the round trip delay, so its impact is negligible.

DLSw+ with TCP encapsulation provides local acknowledgment and local polling and minimizes keepalive traffic across the WAN. It supports any local media and any WAN media. Load balancing across multiple WAN links or IP paths is possible because TCP resequences traffic before forwarding the traffic.

When using TCP encapsulation, you can assign different types of traffic to different TCP ports so that queuing can be granular. LLC2 traffic can be distinguished by SAP (to distinguish NetBIOS and SNA traffic) and SNA devices can be prioritized by LOCADDR or a MAC/SAP pair.

The following is a sample **dlsw remote-peer** command specifying TCP encapsulation:

```
dlsw remote-peer 0 tcp 10.2.24.3
```

FST Encapsulation

FST is a high-performance option used over higher-speed links (256 kb or higher) when high throughput is required. FST uses an IP header with sequencing numbers to ensure that all frames are delivered in sequence (out-of-order frames are discarded and the end system must retransmit them).

FST is fast-switched, not process-switched, so using this encapsulation allows DLSw+ to process more packets per second than TCP encapsulation. FST does not use TCP, so the header is 20 bytes smaller.

FST, however, provides neither reliable delivery of frames nor local acknowledgment. All keepalive frames flow end to end. FST is supported only when the end systems reside on Token Ring. Two FST peers can connect over High-Level Data Link Control (HDLC), Ethernet, Token Ring, FDDI, Asynchronous Transfer Mode (ATM), or Frame Relay. (Some transport media are not available with early maintenance releases. See Appendix B for details.) FST will reroute around link failures, but rerouting may be disruptive. In addition, load balancing across multiple WAN links or IP paths is not recommended with FST because frames may arrive out of order and FST will discard them, causing end systems to retransmit and reducing overall network performance.

Finally, queuing is not as granular with FST because you cannot assign different types of traffic to different TCP ports. This means that when using FST encapsulation, queuing algorithms cannot be distinguished by SAP (so NetBIOS and SNA are treated as LLC2 traffic), and they cannot be distinguished by LOCADDR or MAC address.

The following is a sample **dlsw remote-peer fst** command specifying FST encapsulation:

```
dlsw remote-peer 0 fst 10.2.24.3
```

Direct Encapsulation

Direct encapsulation is a minimal-overhead option for transport across point-to-point lines where rerouting is not required. Direct encapsulation is supported over HDLC lines and Frame Relay. It includes a DLSw 16-byte header and the data-link control header.

Direct encapsulation is fast-switched, not process-switched, so using this encapsulation allows DLSw+ to process more packets per second than TCP encapsulation.

Direct encapsulation provides neither reliable delivery of frames nor local acknowledgment. All keepalive frames flow end-to-end. Direct encapsulation is supported only when the end systems reside on Token Ring. Direct encapsulation does not provide any rerouting.

Finally, queuing is not as granular with direct encapsulation because you cannot assign different types of traffic to different TCP ports. This means that when using direct encapsulation, queuing algorithms cannot be distinguished by SAP (so NetBIOS and SNA are treated as LLC2 traffic), and they cannot be distinguished by SDLC or MAC address.

Direct encapsulation is sometimes considered for very low-speed lines to minimize overhead, but TCP encapsulation with payload compression may offer lower WAN overhead without the limitations of direct encapsulation.

The following is a sample **dlsw remote-peer interface** command specifying direct encapsulation on an HDLC line:

```
dlsw remote-peer 0 interface serial 01
```

The following is a sample **dlsw remote-peer frame relay** command specifying direct encapsulation on a Frame Relay line:

```
dlsw remote-peer 0 frame-relay interface serial 01 33 pass-thru
frame-relay map dlsw 33
```

In this example, data-link connection identifier (DLCI) 33 on serial interface 1 will be used to transport DLSw traffic. Specifying **pass-thru** implies that the traffic is not locally acknowledged. Leaving **pass-thru** off will cause the traffic to be locally acknowledged, which means it is transported in LLC2 to ensure reliable delivery. The next section describes LLC2 encapsulation.

LLC2 Encapsulation (DLSw Lite)

DLSw+ with LLC2 encapsulation is also known as DLSw Lite. It supports many DLSw+ features, including local acknowledgment, media conversion, minimizing keepalive traffic, and reliable delivery of frames, but it uses less overhead (16 bytes of DLSw header and 4 bytes of LLC2). It is currently supported over Frame Relay and assumes a point-to-point configuration over Frame Relay (that is, the peering router at the central site is also the WAN router). DLSw Lite supports Token Ring-, SDLC-, QLLC-, or Ethernet-attached end systems. DLSw Lite is process-switched and processes approximately the same traffic volume as TCP encapsulation.

With DLSw Lite, link failures are disruptive. Availability can be achieved by having multiple active central site peers, which allows for dynamic, but disruptive, recovery from the loss of either a link or a central site peer. Backup peers are not yet supported for DLSw Lite.

Queuing with DLSw Lite is not as granular as with TCP encapsulation, because you cannot assign different types of traffic to different TCP ports. This means that when using DLSw Lite, queuing algorithms cannot distinguish traffic by SAP (so NetBIOS and SNA are treated as LLC2 traffic), and they cannot distinguish traffic by SDLC or MAC address.

The following is a sample **dlsw remote-peer frame-relay** command specifying LLC2 encapsulation on a Frame Relay line:

```
dlsw remote-peer 0 frame-relay interface serial 01 33
frame-relay map llc2 33
```

Note The **frame-relay map llc2** command will not work on point-to-point subinterfaces. Instead, you must provide the DLCI number in the **frame-relay interface-dlci** command and specify the same DLCI number in the **dlsw remote-peer frame relay** command.

The following is a sample **dlsw remote-peer** command for point-to-point subinterfaces:

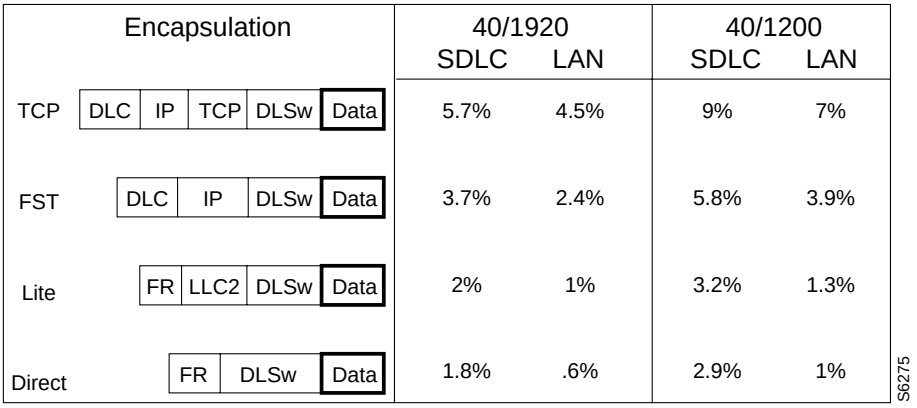
```
dlsw remote-peer 0 frame-relay interface serial 0.1 60
interface s0.1 point-to-point
frame-relay interface-dlci 60
```

Encapsulation Overhead

Different types of encapsulation incur different amounts of overhead on a per-frame basis. But with TCP and LLC2, local acknowledgment and keepalive traffic are removed from the WAN, reducing the number of packets. Also, techniques like payload or header compression and packing multiple SNA frames in a single TCP packet can further reduce the overhead. The percentage of overhead created by DLSw depends on the encapsulation method used.

Figure 7-14 illustrates the frame format for TCP, FST, DLSw Lite, and direct encapsulation. The percentage shown is the amount of overhead assuming SNA transactions of 40 in, 1920 out (a screen refresh) and 40 in, 1200 out. With smaller transactions the overhead is larger. The TCP encapsulation numbers are worst-case numbers because they assume that each SNA path information unit (PIU) is encapsulated in a separate TCP packet. In fact, if there is more than one SNA PIU in the output queue, multiple frames will be encapsulated in a single TCP packet, reducing the overhead. The percentages in Figure 7-14 do not take into consideration the fact that DLSw+ eliminates keepalive packets and acknowledgments.

Figure 7-14 Frame Format and Per-Packet Overhead of Various Encapsulation Types and Transaction Sizes



The effective per-packet overhead of DLSw for LAN traffic is lower than SDLC because DLSw+ eliminates the need to carry MAC addresses and RIFs in every frame. DLSw does not carry this data because the DLSw circuit ID (part of the 16-byte DLSw header) is used for circuit correlation. The overhead of MAC addresses and RIFs can range from 12 to 28 bytes of data. The percentages in Figure 7-14 assume the minimum overhead (no RIF).

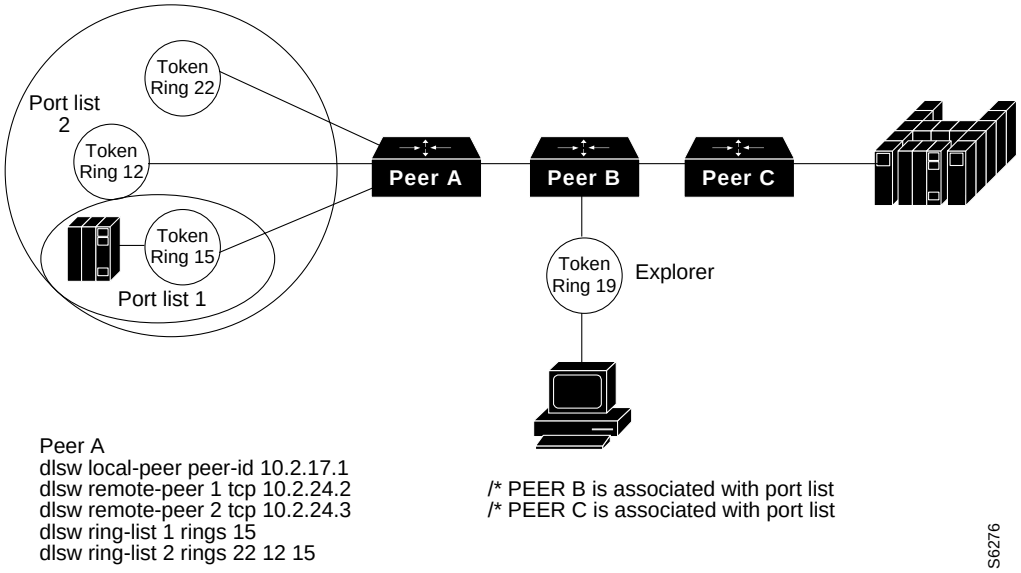
Port Lists

Port lists allow you to create virtual LANs (VLANs) or broadcast domains in a DLSw+ network. Using port lists, you can control where broadcasts are forwarded. For example, in Figure 7-15, there are three rings at the distribution site (where Peer A resides).

All the rings have SNA end systems, but Ring 15 is the only ring with NetBIOS servers. The branch with Peer B needs access to the NetBIOS servers on Ring 15, but does not need access to other rings. Port lists allow you keep all broadcasts from Peer B off Rings 12 and 22 (and prevent Peer B from communicating with devices on Rings 12 or 22).

You can distinguish among different Token Ring ports and serial ports using port lists, but all Ethernet ports are treated as a single entity (Ethernet bridge group).

Figure 7-15 Ring Lists Used to Limit Broadcast Domains in a DLSw+ Network



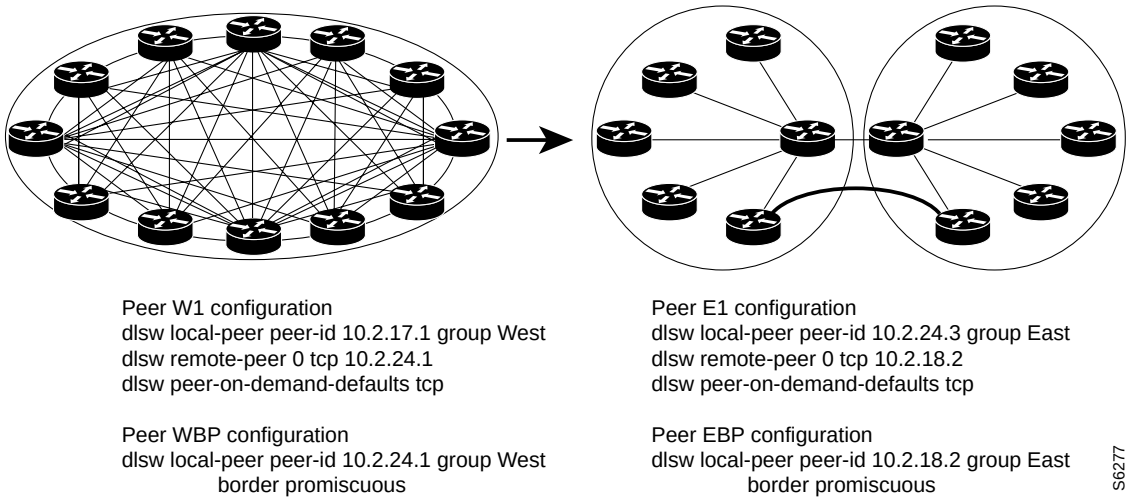
Peer Groups, Border Peers, and On-Demand Peers

Peer groups and border peers can be used to minimize the number of peer connections required for any-to-any communication. Prior to the introduction of border peers, any two DLSw routers that required connectivity needed a peer connection active at all times. This peer connection is used to find resources and to carry circuit traffic. In a fully meshed network of n routers, this requires nx(n-1)/2 TCP connections. This is complex to configure and can result in unnecessary explorer traffic. To address this issue, DLSw+ supports the concept of peer groups and border peers. Peer groups are arbitrary groups of routers with one or more designated border peers. Border peers form peer connections with every router in their group and with border peers in other groups. The role of a border peer is to forward explorers on behalf of other routers.

Use peer groups and border peers only when you need branch-to-branch communication between NetBIOS or APPN end systems. For more information on this feature, read the chapter “Designing Meshed Networks.”

In Figure 7-16, the “before” network shows the required TCP connections for fully meshed connectivity without using border peers. Without border peers, any time a router wants to find a resource that is not in its cache, it must create an explorer frame and replicate it for each TCP connection. This creates excessive explorer traffic on the WAN links and processing load on the router.

Figure 7-16 Using Border Peers and Peer Groups to Minimize the Number of Required TCP Connections While Maintaining Full Any-to-Any Connectivity



After configuring border peers and peer groups, the same fully meshed connectivity is possible without the overhead. In the "after" network, two peer groups are defined (West Group and East Group). Within each group, one or more peers are configured as border peers. Every peer within the West Group establishes a peer connection with the west border peer (WBP). Every peer within the East Group establishes a peer connection with east border peer (EBP). The border peers establish a peer connection with each other. When a peer in the West Group wants to find a resource, it sends a single explorer to its border peer. The border peer forwards this explorer to every peer in its group and to every other border peer. The EBP, after receiving this explorer, forwards it to every peer in its group. When the resource is found (in this case at E1), a positive reply flows back to the origin (W1) via the two border peers. At this point W1 establishes a direct peer connection to E1. Peer connections that are established via border peers without the benefit of preconfiguration are called peer-on-demand connections. The rules for establishing on-demand peers are defined in the **dlsw peer-on-demand-defaults tcp** commands in each router.

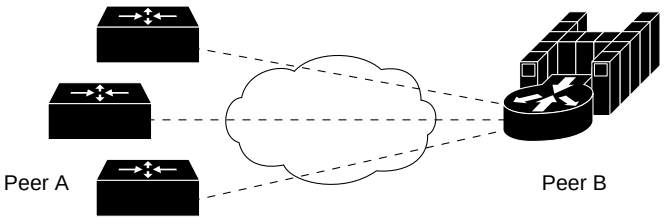
Dynamic Peers

Dynamic peers (available in Cisco IOS Release 11.1 and later) are configured remote peers that are connected only when there are circuits using them. When a **dlsw remote-peer** command specifies **dynamic**, the remote peer is activated only when an end system sends an explorer frame that passes all the filter conditions specified in the **dlsw remote-peer** command. Once the dynamic peer connection is established, the explorer is forwarded to the remote peer. If the resource is found, a circuit is established and the remote peer will remain active until all circuits using that remote peer terminate and five minutes elapse. You can specify the **no-llc** keyword to modify the elapsed time to something other than five minutes. Optionally, the remote peer can be configured to disconnect when there is no activity on any of the circuits for a prespecified amount of time (inactivity timer).

Filters that minimize how many explorers are sent to a remote peer can be included in **dlsw remote-peer** commands. In the case of dynamic peers, these filters are also used to prevent the dynamic peer from being activated. The remote peer statement allows you to point to lists of SAPs, MAC addresses, NetBIOS names, or byte offset filters. You can also specify a MAC address on the **dlsw remote-peer command** for a dynamic peer, in which case that remote peer is activated only when there is an explorer for the specified MAC address. Figure 7-17 shows an example of how to use this feature. In Figure 7-17, the dynamic peer is only established if an explorer frame is received

that is destined for the MAC address of the FEP. After the peer connection is established, if there is no activity on this peer connection for 20 minutes, the peer connection and any circuits using the connection are terminated because **inactivity 20** was specified.

Figure 7-17 DLSw+ Routers Configured to Take Advantage of the Dynamic Peer Feature



Peer A configuration
dlsw local-peer peer-id 10.2.17.1
dlsw remote-peer 0 tcp 10.2.24.2 dynamic
inactivity 20 dest-mac 4000.3745.0000

Peer B configuration
dlsw local-peer peer-id 10.2.24.3

S6278

When to Use Dynamic Peers

Use dynamic peers if you have a large network but do not require all remote sites to be connected at the same time. By using dynamic peers, you can minimize the number of central site routers needed to support the network. You can also use dynamic peers for occasional communication between a pair of remote sites. Dynamic peers differ from on-demand peers because they must be preconfigured. Finally, for small networks, dynamic peers can be used to dial out during error recovery.

SNA Dial-on-Demand Routing

SNA Dial-on-Demand Routing (DDR) refers to the ability for DLSw+ to transfer SNA data over a dial-up connection and automatically drop the dial connection when there is no data to send. The SNA session remains active. To use SNA DDR, configure the following on the **dlsw remote-peer** command:

dlsw remote-peer *list-number* **tcp** *ip-address* **dynamic keepalive** ***0*** **timeout** *seconds*

The **dynamic** keyword is optional but recommended because it will prevent the remote peer connection from being established unnecessarily. The **dynamic** option is described in the previous section and can be used in conjunction with the **dmac-out** or **dmac-output-list** options on the **dlsw remote-peer** command to ensure that peer connections are only brought up when desired (for example, when a device is trying to locate the FEP).

The **keepalive** keyword is required. DLSw+ locally acknowledges SNA (or more precisely, SDLC or LLC2) traffic, so no data-link control acknowledgments or receiver ready frames will bring up the dial connection. However, DLSw+ peers send peer keepalives to each other periodically, and these keepalives will bring up the dial connection. The **keepalive** option refers to how often DLSw+ peers send peer keepalives to each other. If you set this to zero, no keepalives will be sent and, therefore, the peer keepalive will not keep the dial line up. You must specify **keepalive 0** in *both* peers; that is, either you must specify the remote peers at both the local and remote DLSw+ routers, or you must use the **prom-peer-default** command to set **keepalive** to zero for all promiscuous peer connections. The **prom-peer-default** command has the same options as the **peer-on-demand-defaults** command and is available in the later maintenance release of all DLSw+ releases.

The keepalive parameter refers to how often DLSw+ peers send peer keepalives to each other. If you set this to zero, no keepalives are sent, and the peer keepalive will not keep the dial line up. This parameter must be specified in *both* peers, which means that you must either specify the remote peers at both the local and remote DLSw+ routers, or you must use the **dlsw prom-peer-default** command to set keepalive to 0 for all promiscuous peer connections. The **dlsw prom-peer-default** command is similar to the **dlsw peer-on-demand-defaults** command and is available in the later maintenance releases of all DLSw+ releases.

The **timeout** keyword is recommended. Without peer keepalives, DLSw+ is dependent on TCP timers to determine when the SNA session has come down. TCP will only determine that it has lost a partner if it does not get an acknowledgment after it sends data. By default, TCP may wait up to 15 minutes for an acknowledgment before tearing down the TCP connection. Therefore, when **keepalive 0** is specified, you should also set the **timeout** keyword, which is the number of seconds that TCP will wait for an acknowledgment before tearing down the connection. Timeout should be long enough to allow acknowledgments to get through in periods of moderate to heavy congestion, but short enough to minimize the time it takes to recover from a network outage. SNA data-link control connections typically wait 150 to 250 seconds before timing out.

Other Considerations

In addition to preventing keepalive traffic from bringing up the Integrated Services Digital Network (ISDN) lines, you need to worry about routing updates. In hub and spoke environments, to prevent route table updates from bringing up the dial connections, use static routes. Alternatively, you can use Routing Interface Protocol (RIP) Version 2 or on-demand routing for IP routing from the dial-up branches to the central site. On-demand routing (ODR) is a mechanism that provides minimum-overhead IP routing for sub sites. Define RIP Version 2 or on-demand routing on the ISDN interface of the central router as passive mode. Then redistribute RIP Version 2 or ODR routes into the main routing protocol (Enhanced Interior Gateway Routing Protocol [IGRP] or Open Shortest Path First [OSPF]). This allows you to have multiple routers at the central site for load balancing or redundancy. Whichever router receives the call from the remote site will have the route installed dynamically. At the remote site, the routing protocol (RIP or ODR) must be denied from the dialer list.

For meshed topologies, you can minimize routing table updates by using a distance-vector protocol such as RIP or IGRP in combination with Cisco's snapshot routing feature. Snapshot routing prevents regular routing updates from bringing up the ISDN connection. The changes in routing tables are sent either when the link is opened by end-user traffic or at a regular configurable interval. Snapshot routing supports not only IP routing updates, but also Novell's IPX routing and SAP updates.

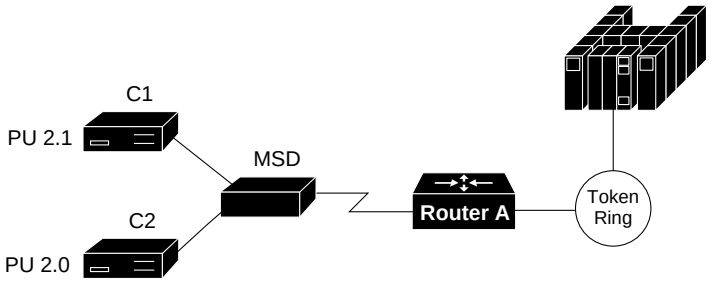
Many NetBIOS implementations use a session keepalive (in addition to a data-link control keepalive) to maintain sessions, so DDR may not work with NetBIOS. (The session level keepalive will keep the dial line up.)

Local Switching

Local switching (available in Cisco IOS Release 11.1 and later) allows a single router to provide media conversion between SDLC and Token Ring and between QLLC and LAN. This is useful in environments that need simplified SNA network design and improved availability. For example, by converting SDLC to Token Ring, fewer FEP expansion frames are required; moves, adds, and changes are easier; and recovery from a FEP or Token Ring interface coupler (TIC) failure can be automatic (by using duplicate TIC addresses). Local switching can be used to connect SDLC devices directly to a Cisco router with a CIP card. Local switching can also be used over a WAN where the remote branch has SNA devices on LANs, but the central site FEP still requires serial connectivity (for example, when the FEP is a Cisco 3725 router).

To use local switching, omit **dlsw remote-peer** commands. In the **dlsw local-peer** command, the peer ID is unnecessary. A sample network and its configuration are shown in Figure 7-18.

Figure 7-18 Local Switching Configuration in a Mixed PU 2.0 and PU 2.1 Environment




```
Peer A Router A
dlsw local-peer
interface serial 0
…
sdlc role primary
sdlc vmac 4000.3174.0000
sdlc address c1 xid-poll
sdlc partner 4000.3745.0001 c1
sdlc address c2
sdlc xid c2 01767890
sdlc partner 4000.3745.0001 c2
sdlc dlsw c1 c2
```

S6279