Data Link Switching Plus (DLSw+)

- Introduction

  Describes DLSw+ and how to design and configure a DLSw+ network.

- Getting Started

  Describes basic configuration commands required for a DLSw+ network.

- Advanced Features

  Describes advanced DLSw+ features, their benefits, and when and how to use them.

- Customization

  Describes several ways to customize a DLSw+ network.

- Bandwidth Management and Queuing

  Describes how to use bandwidth management and queuing features to enhance performance.

- Designing Hierarchical Networks

  Discusses design considerations for building hierarchical DLSw+ networks.

- Designing Meshed Networks

  Discusses design considerations for building meshed DLSw+ networks.

- RSRB Migration and Multivendor Interoperability

  Compares RSRB and DLSw+ and the benefits of migration from RSRB to DLSw+.

- Using Show and Debug Commands

  Describes how to use show and debug commands.

- Using CiscoWorks Blue: Maps, SNA View, and Native Service Point

  Describes enhanced network management tools available for DLSw+ networks.

- Using DLSw+ with Other Features

  Describes how to use DLSw+ in conjunction with APPN, DSPU concentration, LAN Network, and NCIA.

- Memory Estimates

  Provides details of DLSw+ memory utilization.

- DLSw+ Support Matrix

  Lists DLSw+ features and the Cisco IOS release levels supporting them, by encapsulation type.

Introduction

DLSw+ Design and Implementation describes Data Link Switching Plus (DLSw+) and provides configuration examples to allow you to quickly design and configure simple DLSw+ networks. It 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. This book can be used as a reference only (for configuration examples), as a tuning guide, or as a complete DLSw+ network design guide.

This chapter describes DLSw+ and how to use this manual to design and configure a DLSw+ network. It reviews the key components of the data-link switching (DLSw) features and describes the extensions to the standard that are included in DLSw+. Finally, it recommends how to proceed with designing your 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 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 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 APPN Implementers Workshop (AIW) in the DLSw-related interest group. The current standard is Version 1, which is documented in an informational Request for Comments (RFC), 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 transport the data reliably across an IP backbone.

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

- Establish peer connection
- Exchange capabilities
- Establish circuit

After circuits are established, the standard describes how to control the flow of data between peers.

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 local-area network (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 SAPs, 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–Offers higher-performance transport options when there is enough bandwidth to handle the traffic load without risk of timeouts; in addition, the option to use lower-overhead solutions when bandwidth is at a premium and nondisruptive rerouting is not required
- Modes of operation–Dynamically detects the capabilities of the peer router and operates according to those capabilities

DLSw+ Improved Scalability

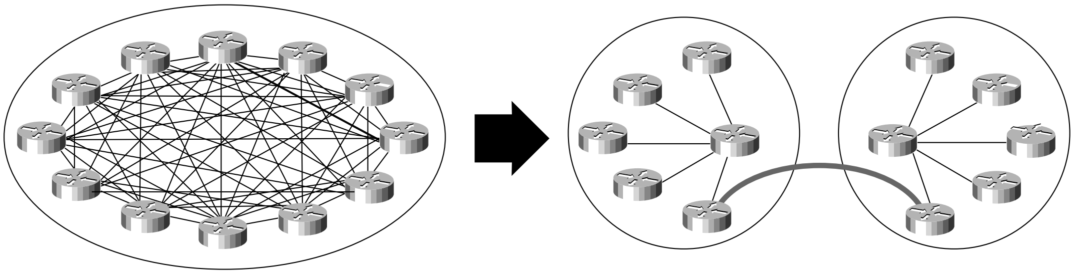
One of the most significant factors that limits the size of LAN internetworks is the amount of explorer traffic that traverses the WAN. There are several optimizations in DLSw+ to reduce the number of explorers.

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 Advanced Peer-to-Peer Networking [APPN] environments), RSRB or standard DLSw implementations require peer connections between every pair of routers. This setup is not only difficult to configure, but 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 is designated to be the *border peer.* 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 1-1). 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 1-1 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.

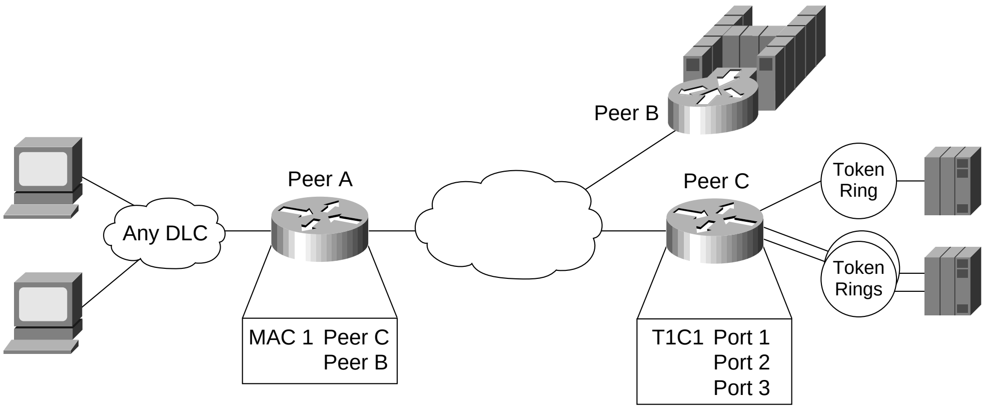
At the destination DLSw+ router, a slightly different procedure is followed using the local cache entries. If the cache entry is fresh, the response is sent immediately. If the cache entry is stale, a single route broadcast TEST frame is sent over the all ports in the cache. If a positive response is 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 1-2. 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 1-2 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 upon 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 of 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" chapter, 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" chapter, 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 CiscoWorks Blue: 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).

The appendixes include memory requirements to assist in network planning and feature, media, and release matrices.

Getting Started

This chapter 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. This section assumes that you are familiar with basic router configuration.

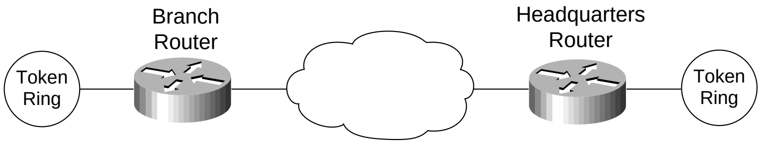
Minimum Required Configuration

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 2-1, 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 2-1 Example of dlsw local-peer and dlsw remote-peer Commands



```
Configuration for Branch Router
dlsw local-peer peer-id 10.2.24.2
dlsw remote-peer 0 tcp 10.2.17.1
```

```
Configuration for Headquarters Router
dlsw local-peer peer-id 10.2.17.1
  promiscuous
```

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 “Advanced Features” chapter.

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

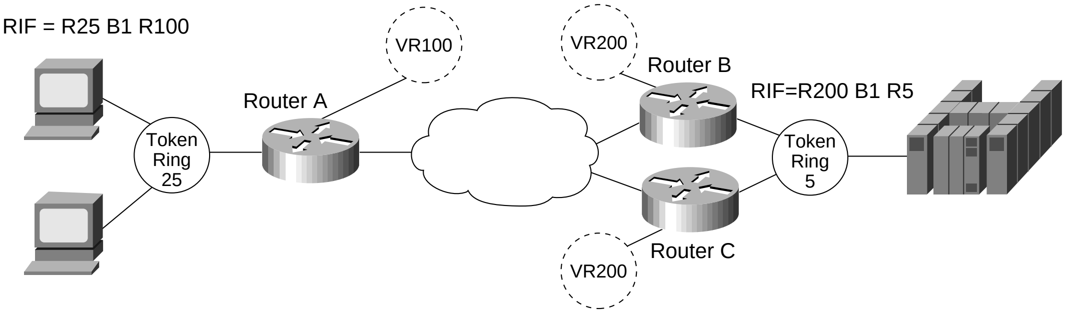
- Token Ring-Define a virtual ring using the **source-bridge ring-group** command and include a **source-bridge** command that tells the router to bridge from the external Token Ring to that virtual ring
- Ethernet–Map a specific Ethernet bridge group to DLSw+ (DLSw+ supports only one Ethernet bridge group)
- 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 chapter provides sample configurations for Token Ring, Ethernet, SDLC, and QLLC.

Token Ring

Figure 2-2 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 2-2 Simple Token Ring DLSw+ Configuration



```
Configuration for Router A
source-bridge ring-group 100
dlsw local-peer peer-id 10.2.17.1
dlsw remote-peer 0 tcp 10.2.24.2
.
.
interface TokenRing0
ring-speed 16
source-bridge active 25 1 100
source-bridge spanning
```

```
Configuration for Router B
source-bridge ring-group 200
dlsw local-peer peer-id 10.2.24.2
   promiscuous
.
.
interface TokenRing0
ring-speed 16
source-bridge active 5 1 200
source-bridge spanning
```

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 2-2, 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.

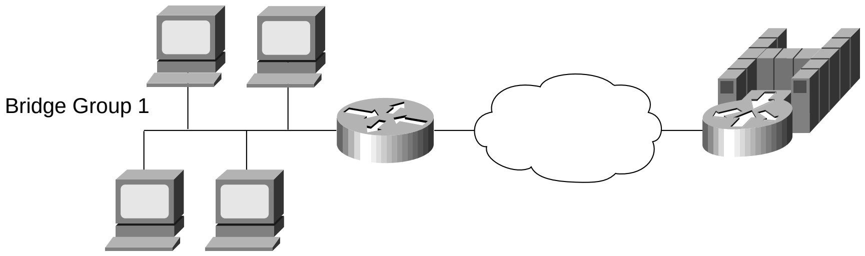
Ethernet

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 2-3, 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

Note Some environments avoid this issue by using MAC addresses consisting of only “magic numbers”—numbers that are the same in canonical and noncanonical formats. These numbers are 00, 18, 24, 3C, 42, 5A, 66, 7E, 81, 99, A5, BD, C3, DB, E7, and FF.

In Figure 2-3, the data is transferred directly to a Cisco router with a CIP, but it could be any DLSw-compliant router, and the upstream SNA end system could reside on any supported media.

Figure 2-3 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
```

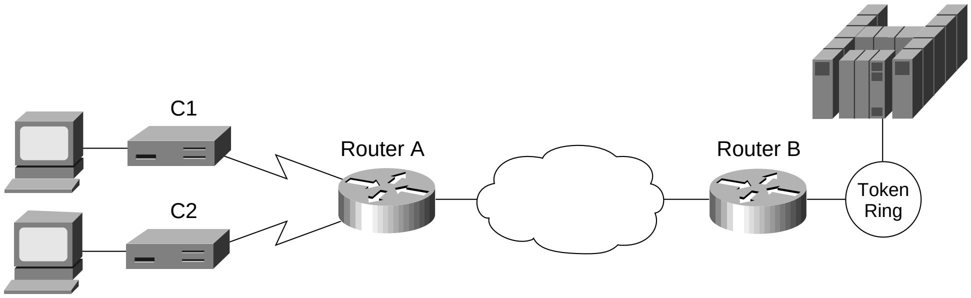
```
source-bridge ring-group 200
dlsw local-peer peer-id 10.2.24.2
   promiscuous
.
.
interface channel 0/1
csna 0100 40
csna 0100 41
int chan 0/2
lan tokenring 0
source-bridge 1000 1 200
adapter 0 4000.0000.0401
adapter 1 4000.0000.0403
```

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 2-4, 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 2-4 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 serial 1
encapsulation sdlc
sdlc role primary
sdlc vmac 4000.3174.1000
sdlc address c2
sdlc xid c2 01767890
sdlc partner 4000.3745.0001 c2
sdlc dlsw c2
```

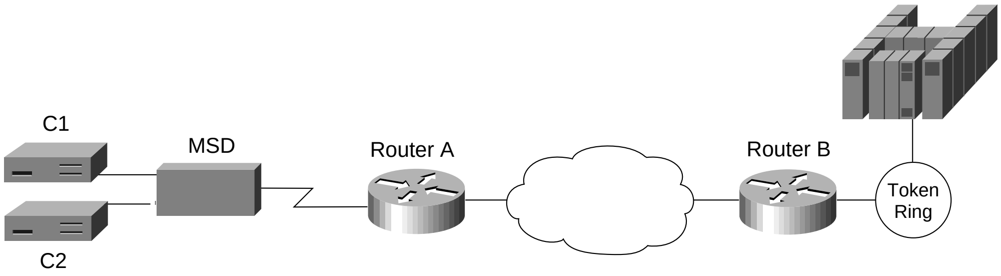
The router can be the secondary end of an SDLC line (for example, when connecting to a FEP over SDLC). In this case, specify **secondary** in the **sdlc role** command, and for PU 2.1 devices, specify **xid-passthru** in the **sdlc address** command.

In Cisco IOS Release 11.0 and later, DLSw+ supports multidrop PU 2.0/2.1. In Figure 2-5, 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 2-5 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 c2 01767890
sdlc partner 4000.3745.0001 c2
sdlc dlsw c1 c2
```

Configuration for Router A, Mixed PU 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 c2 01767890
sdlc partner 4000.3745.0001 c2
dslc dslw c1 c2
```

Configuration for Router A, All PU 2.1

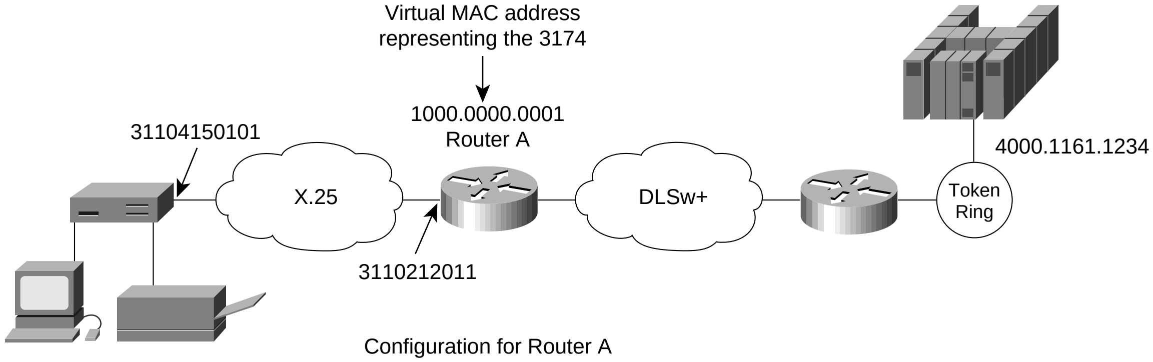
```
interface serial 0
. . .
sdlc role prim-xid-poll
sdlc vmac 4000.3174.0000
sdlc address c1
sdlc partner 4000.3745.0001 c1
sdlc address c2
sdlc partner 4000.3745.0001 c2
sdlc dlsw c1 c2
```

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 (NPSI). 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 2-6 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 2-6 QLLC DLSw+ Configuration to a Single LAN-Attached Upstream Device



Configuration for Router A

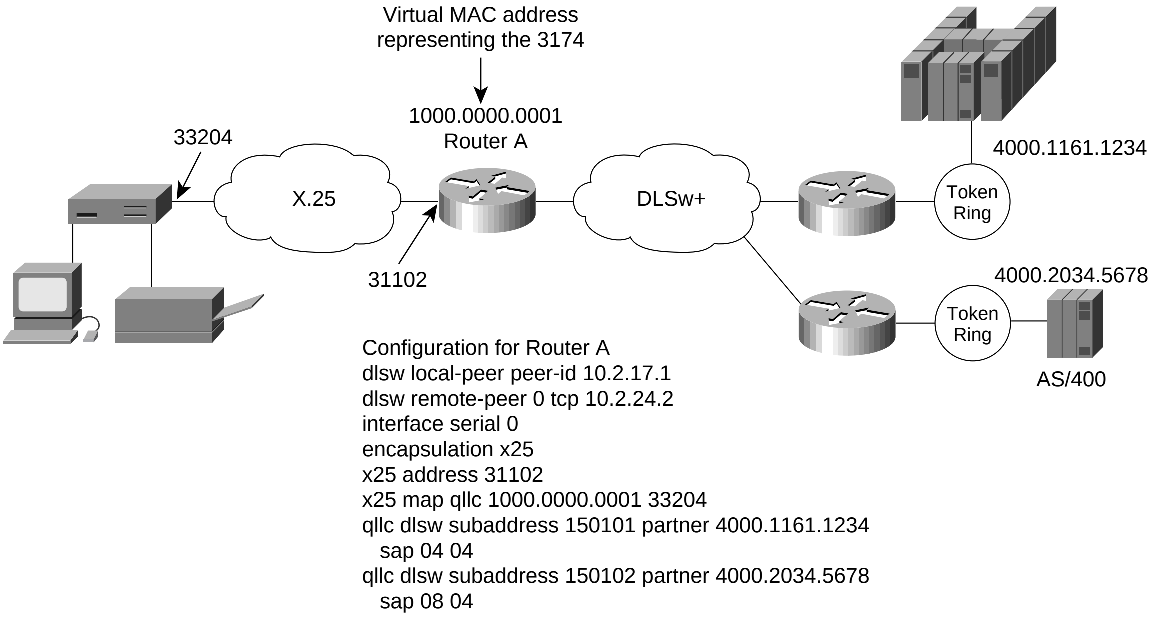
```
dlsw local-peer peer-id 10.2.17.1
dlsw remote-peer 0 tcp 10.2.24.2
interface serial 0
encapsulation x25
x25 address 3110212011
x25 map qllc 1000.0000.0001 31104150101
qllc dlsw partner 4000.1161.1234
```

In Figure 2-7, 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.

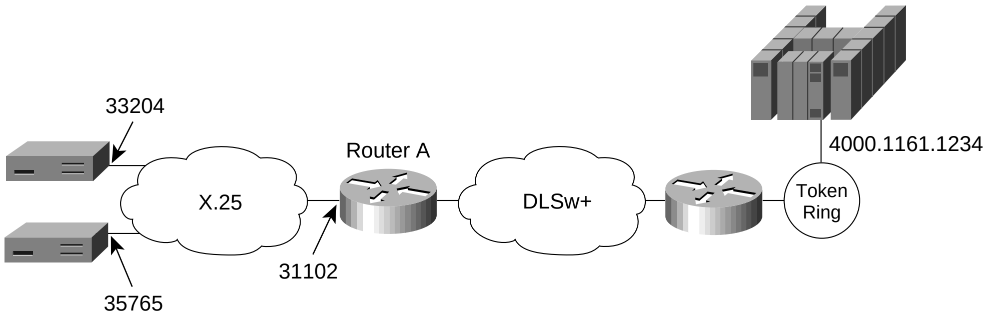
Figure 2-7 QLLC DLSw+ Configuration for Support of Multiple Upstream LAN-Attached Devices



```
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 1000.0000.0001 33204
qllc dlsw subaddress 150101 partner 4000.1161.1234
   sap 04 04
qllc dlsw subaddress 150102 partner 4000.2034.5678
   sap 08 04
```

In Figure 2-8, 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 2-8 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
```

CHAPTER 3

Advanced Features

This chapter describes advanced features of DLSw+, the benefits they provide, and a brief description of when and how to use them. Use this chapter 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. Tuning features are covered in the next chapter.

Load Balancing and Redundancy

If you have multiple central site routers supporting DLSw+ for either load balancing or redundancy, read this section. 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+.

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). Before a cache is populated, 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.

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.

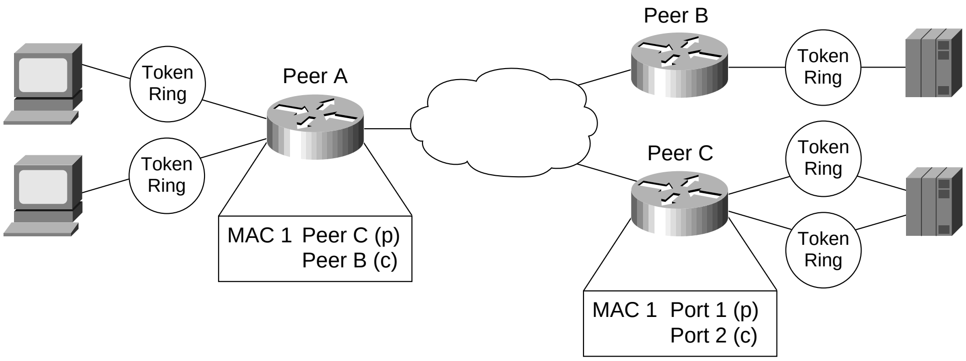
If load balancing is not specified, then the peer selects the first path in the cache and sets up all circuits via that path unless the path is unavailable. The first path in the cache list can be one of the following:

- 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 3-1, 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 [TB].)

Figure 3-1 Possible Configuration and the Resulting Cache Entries Created if All Channel Gateways Illustrated Have the Same MAC Address



```
Configuration for Peer A
dlsw local-peer peer-id 10.2.17.1
dlsw remote-peer 0 tcp 10.2.24.2
dlsw remote-peer 0 tcp 10.2.24.3
```

```
Configuration for Peer B
dlsw local-peer peer-id 10.2.24.3 cost 4
   promiscuous

Configuration for Peer C
dlsw local-peer peer-id 10.2.24.2 cost 2
   promiscuous
dlsw duplicate-path-bias load-balance
```

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.

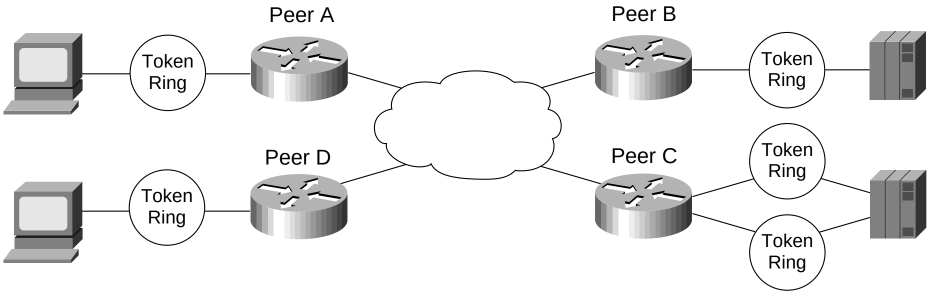
Peer A will forward a positive response to the SNA PU and then establish an end-to-end circuit using Peer C. Peer C is selected because Peer C has a lower cost specified. When the next PU attempts to set up a connection to the same MAC address, it will be set up using Peer C, if available. This is the 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 3-1 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. An important point to note is 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 3-2. 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 [LU] sessions flow over the same circuit that the PU session flows over). In Figure 3-2, Peer A always favors Peer B and Peer D always favors Peer C.

Figure 3-2 Configuration Where Cost Is Specified in the dlsw remote-peer Command Instead of the dlsw local-peer Command



```
Configuration for Peer A
dlsw local-peer peer-id 10.2.17.1
dlsw remote-peer 0 tcp 10.2.24.2 cost 2
dlsw remote-peer 0 tcp 10.2.24.3 cost 4
```

```
Configuration for Peer B
dlsw local-peer peer-id 10.2.24.2
   promiscuous
```

```
Configuration for Peer D
dlsw local-peer peer-id 10.2.18.6
dlsw remote-peer 0 tcp 10.2.24.2 cost 4
dlsw remote-peer 0 tcp 10.2.24.3 cost 2
```

```
Configuration for Peer C
dlsw local-peer peer-id 10.2.24.3
   promiscuous
dlsw duplicate-path-bias load-balance
```

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.

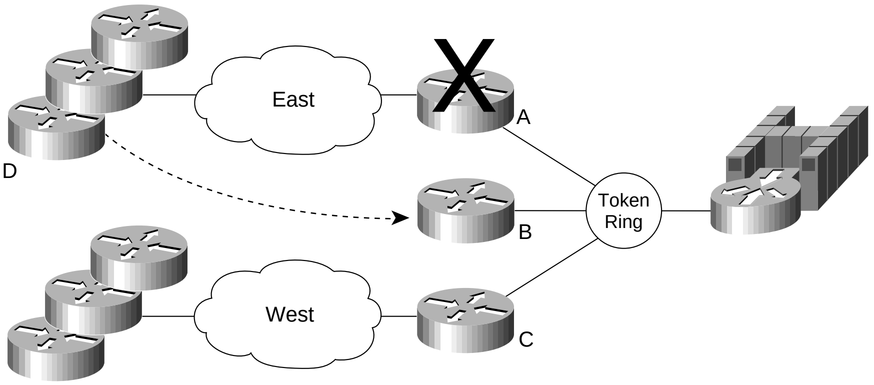
Setting the **dlsw explorer-wait-time** command causes DLSw+ to wait the specified amount of time (for example, one second) before selecting a peer to use for connections. See how to modify timers in the next chapter. This timer can be set in Cisco IOS Release 11.0 and later. Prior to Release 11.0, this 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 3-3 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). Support for all encapsulations will be added in Cisco IOS Release 11.3.

Figure 3-3 How to Use Backup Peers to Enhance Availability in a Large DLSw+ Network



```
Configuration for Router D
dslw local-peer peer-id 10.2.17.1
dslw remote-peer 0 tcp 10.2.24.2                                /* Router A is the primary
dlsw remote-peer 0 tcp 10.2.24.3 backup-peer 10.2.24.2 linger 20  / *Router B to backup Router A
```

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 can be set high enough to allow 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.

Note Backup peers can be used to recover *only* from the loss of a router. They cannot be used to recover from the loss of a mainframe or mainframe channel gateway. The reason is because they are only activated when the primary *peer* fails. To enable automatic recovery from the loss of a mainframe or channel gateway, you must configure multiple active peers.

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, 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 the primary link reaches 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 11 ms to the round trip delay, so its impact is negligible.

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

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 kbps 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 will be supported for DLSw Lite in Cisco IOS Release 11.3.

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
dlsw 11c2-peer listen-sap 04
```

The last statement above will be added in Cisco IOS Release 11.3. It will be required only if other IBM protocols such as APPN or DSPU are running in this router.

Note The **frame-relay map llc2** command will not work on point-to-point sub-interfaces. 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 as follows:

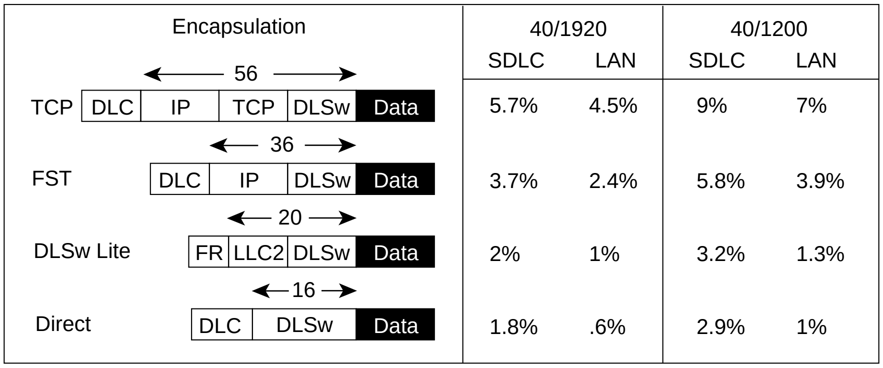
```
dlsw remote-peer 0 frame-relay interface serial 0.1 60
interface s0.1 point-to-point
frame-relay interface-dlci 60
```

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 3-4 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 3-4 do not take into consideration the fact that DLSw+ eliminates keepalive packets and acknowledgments.

Figure 3-4 Frame Format and Per-Packet Overhead of Various Encapsulation Types and Transaction Sizes



Encapsulation		40/1920		40/1200	
		SDLC	LAN	SDLC	LAN
TCP	← 56 → DLC | IP | TCP | DLSw | Data	5.7%	4.5%	9%	7%
FST	← 36 → DLC | IP | DLSw | Data	3.7%	2.4%	5.8%	3.9%
DLSw Lite	← 20 → FR | LLC2 | DLSw | Data	2%	1%	3.2%	1.3%
Direct	←16 → DLC | DLSw | Data	1.8%	.6%	2.9%	1%

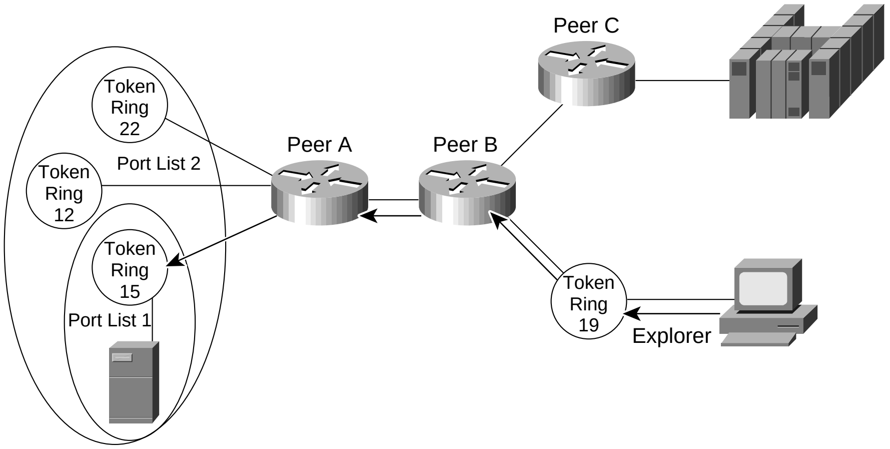
The effective per-packet overhead of DLSw for LAN traffic is lower than SDLC because DLSw+ eliminates the need to carry MAC addresses and RIFs in every frame. DLSw+ does not carry this data because the DLSw+ circuit ID (part of the 16-byte DLSw header) is used for circuit correlation. The overhead of MAC addresses and RIFs can range from 12 to 28 bytes of data. The percentages in Figure 3-4 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 3-5 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 3-5 Ring Lists Used to Limit Broadcast Domains in a DLSw+ Network



```
Configuration for Peer A
dlsw local-peer peer-id 10.2.17.1
dlsw remote-peer 1 tcp 10.2.24.2          /* Peer B is associated with port list 1
dlsw remote-peer 2 tcp 10.2.24.3          /* Peer C is associated with port list 2
dlsw ring-list 1 rings 15
dlsw ring-list 2 rings 22 12 15
```

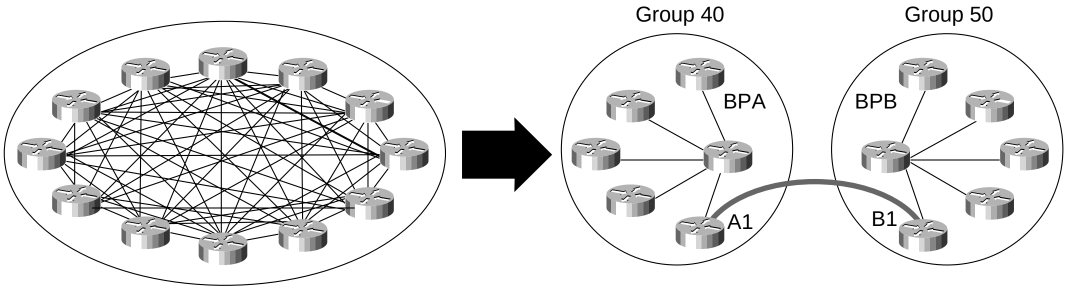
Peer Groups, Border Peers, and On-Demand Peers

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 3-6, 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 3-6 Using Border Peers and Peer Groups to Minimize the Number of Required TCP Connections While Maintaining Full Any-to-Any Connectivity



```
Configuration for Peer A1
dlsw local-peer peer-id 10.2.17.1 group 40 promiscuous
dlsw remote-peer 0 tcp 10.2.24.1
dlsw peer-on-demand-defaults tcp

Configuration for Border Peer A
dlsw local-peer peer-id 10.2.24.1 group 40
     border promiscuous
dlsw remote-peer 0 tcp 10.2.18.2
```

```
Configuration for Peer B1
dlsw local-peer peer-id 10.2.24.3 group 50 promiscuous
dlsw remote-peer 0 tcp 10.2.18.2
dlsw peer-on-demand-defaults tcp

Configuration for Border Peer B
dlsw local-peer peer-id 10.2.18.2 group 50
     border promiscuous
dlsw remote-peer 0 tcp 10.2.24.1
```

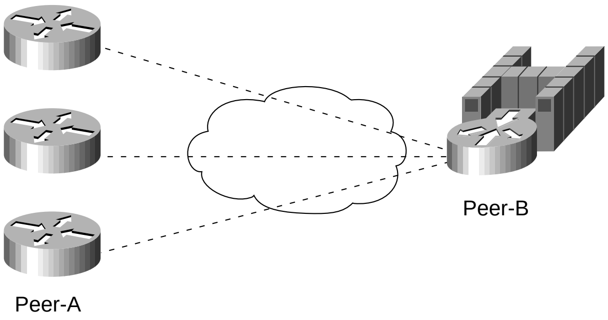
After configuring border peers and peer groups, the same fully meshed connectivity is possible without the overhead. In the “after” network, two peer groups are defined (Group 40 and Group 50). Within each group, one or more peers are configured as border peers. Every peer within Group 40 establishes a peer connection with border peer A (BPA). Every peer within Group 50 establishes a peer connection with border peer B (BPB). The border peers establish a peer connection with each other. When a peer in Group 40 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. BPB, after receiving this explorer, forwards it to every peer in its group. When the resource is found (in this case at B1), a positive reply flows back to the origin (A1) via the two border peers. At this point A1 establishes a direct peer connection to B1. 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** command 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 ten minutes elapse. You can specify the **no-llc** keyword to modify the elapsed time to something other than ten 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 3-7 shows an example of how to use this feature. In Figure 3-7, 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 3-7 DLSw+ Routers Configured to Take Advantage of the Dynamic Peer Feature



```
Configuration for Peer A                        Configuration for Peer B
dlsw local-peer peer-id 10.2.17.1               dlsw local-peer peer-id 10.2.24.3
dlsw remote-peer 0 tcp 10.2.24.2 dynamic            promiscuous
    inactivity 20 dest-mac 4000.3745.0000
```

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* [**inactivity** *seconds* **dmac-out** *mac-address* **tcp-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 tcp** 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 hence 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 zero for all promiscuous peer connections. The **dlsw prom-peer-default** command is similar to the **dlsw peer-on-demand-defaults tcp** 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. Hence, 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 stub 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.

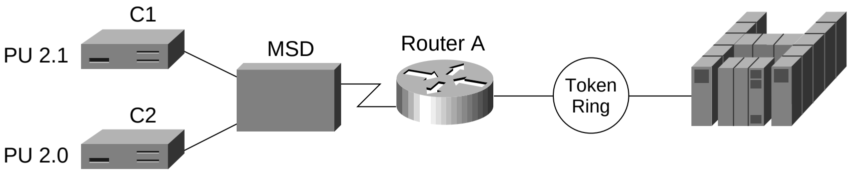
Many NetBIOS implementations use a session keepalive (in addition to a data-link control keepalive) to maintain sessions, so DDR may not work with NetBIOS. (The session level keepalive will keep the dial line up.) To address this issue, a new capability will be added in Cisco IOS Release 11.3. A new command, **dlsw netbios-keepalive-filter**, will filter session keepalives and prevent them from bringing up the WAN link.

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 may 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 3725).

To use local switching, omit **dlsw remote-peer** commands. In the **dlsw local-peer** command, the peer ID is unnecessary. A sample network and configuration is shown in Figure 3-8.

Figure 3-8 Local Switching Configuration in a Mixed PU 2.0 and PU 2.1 Environment



```
Configuration for 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
```

CHAPTER 4

Customization

This chapter describes several ways to customize your DLSw+ network. It includes a description of filtering and static device configuration options, as well as ways to tune network performance by controlling message sizes, timers, and queue depth. Each topic includes the router configuration changes, the effect of the changes, and the benefits that can be derived from the changes. These tuning and customization suggestions are not prerequisites for achieving good performance from DLSw+, but they offer a way to improve overall network performance. They are optional and are unnecessary in many environments.

Read this chapter if you have a very large network (thousands of SNA PUs), a high volume of NetBIOS broadcasts, or a high number of SNA transaction rates (greater than 200 transactions per second).

Note Tuning modifications should only be made with Cisco’s assistance (for example, system buffer tuning).

Filtering

Filtering can be used to enhance the scalability of a DLSw+ network. For example, filtering can be used to:

- Reduce traffic across a WAN link (especially important on very low-speed links and in environments with NetBIOS)
- Enhance the security of a network by controlling access to certain devices

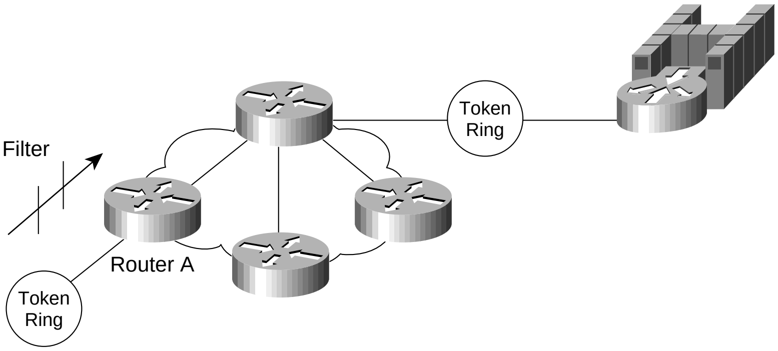
DLSw+ allows you to define access lists that are associated with a particular peer. This capability is powerful because it allows you to decide on a per-site basis what traffic should be allowed to pass over the network. These access lists use standard Cisco filter list syntax.

To filter DLSw+ traffic on a remote peer basis, you must first define an access list containing the resources and the conditions for which you would like the router to pass traffic. You must then associate the access list to a remote peer.

The **dlsw remote-peer** command allows you to point to lists of SAPs, MAC addresses, NetBIOS names, or byte offset filters. You can also simply specify a MAC address in the **dslw remote-peer** command. When these filters are specified, only explorers that pass the access list conditions are forwarded to the remote peer.

Figure 4-1 shows how to use filters to control traffic by protocol or SAP. In this example, the remote peer provides access to SNA resources but blocks all NetBIOS traffic from the WAN. NetBIOS workstations send out large numbers of broadcast frames that can easily overwhelm a low-speed WAN and cause throughput and connectivity problems. To prevent these problems, you can specify an access list as shown in Figure 4-1. The access list numbers can range from 200 to 299. Access lists are applied to peers in the **dlsw remote-peer** command.

Figure 4-1 Using Filtering to Control Traffic by SAP Type



```
Configuration for Router A
access-list 200 permit 0x0000 0x0d0d
dlsw remote-peer 0 tcp 10.17.24.12 lsap-output-list 200
```

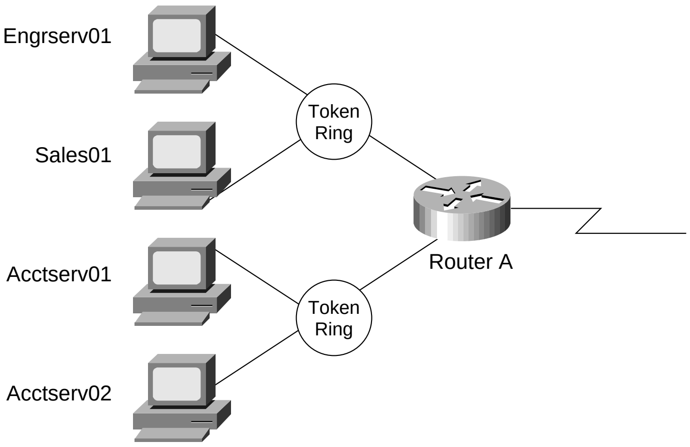
Alternately, to allow NetBIOS and not SNA, specify:

```
   access-list 200 permit 0xf0f0 0x0101
```

Both **access-list** commands can be used to allow only SNA and NetBIOS traffic while blocking other SRB traffic, such as Novell IPX and TCP/IP, from be transmitted across the WAN by DLSw+.

Figure 4-2 shows the configuration required to allow any NetBIOS host with a name starting with “sales” to access the WAN, but not allow any other servers (for example, Engserv01 or Acctserv02) to access the WAN. This can be done for security reasons or to limit the traffic across the WAN link. By applying the access lists to the remote peers instead of the local interfaces, you allow traffic to be locally bridged.

Figure 4-2 Using Filtering to Limit the Broadcasts and Network Access of Individual NetBIOS Servers



Configuration for Router A
netbios access-list host salesfilt permit sales*
dlsw remote-peer 0 tcp 10.17.24.12 host-netbios-out salesfilt
dlsw peer-on-demand-defaults tcp host-netbios-out salesfilt

If you want to prevent this traffic from being forwarded by the router either locally or remotely, you can apply the filter to the Token Ring interface. To apply a NetBIOS access list to an interface, use the following command after the **interface** command:

netbios input-access-filter *filtname*

Use this filter only if you need both local and remote filtering, because it will be applied to all locally bridged traffic and may impact local bridging performance.

Byte filters allow you to filter based on the content of arbitrary fields in a NetBIOS frame. The bytes list name (nblist) is the name of a previously defined NetBIOS bytes access list filter.

```
dlsw remote-peer 0 tcp 10.17.24.12 bytes-netbios-out nblist
```

Another technique to filter traffic is to specify the keyword **dest-mac** in the **dlsw remote-peer** command, which will allow only a single MAC address at the remote peer site to communicate to this local peer. Alternatively, the keyword **dmac-out** lets you specify an access list with multiple MAC addresses.

Static Configuration Options

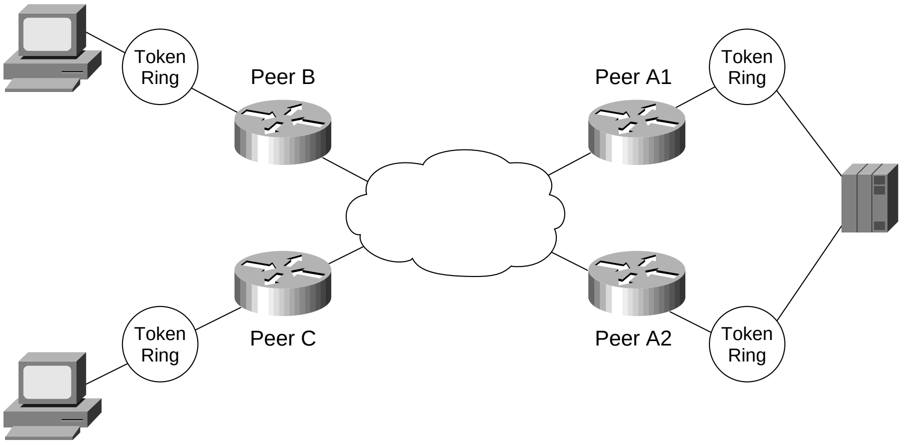
By predefining resources that are accessed frequently, you can minimize broadcast traffic. This traffic can be especially disruptive immediately following a failure of a key resource when every end system attempts to reconnect at the same time. DLSw+ allows you to predefine resources in two ways. You can configure local resources that you want a DLSw+ peer to advertise to other peers, or you can configure static paths that a peer will use to access remote resources.

Advertising Reachability

You can configure reachability of MAC addresses or NetBIOS names with a **dlsw icanreach** command. DLSw+ peers advertise this reachability to remote peers as part of the capabilities exchange. Figure 4-3 illustrates a way to use **dlsw icanreach** commands to prevent remote branches from sending any explorers destined for a mainframe channel gateway across the WAN. In Figure

4-3, two branch offices are shown with routers Peer B and Peer C. At the data center, there are two central site routers, Peer A1 and Peer A2. Both data center routers advertise the reachability of the FEP to the remote routers as part of the capabilities exchange, allowing the branch routers to preload their cache with two paths to the MAC address of the FEP. After a major outage of a FEP or Token Ring, instead of having broadcasts flowing from each remote site, the remote sites will simply reconnect through the appropriate peer.

Figure 4-3 Hierarchical SNA Network Configured to Eliminate the Requirement for Explorers to Find the MAC Address of the FEP or a Mainframe Channel Gateway

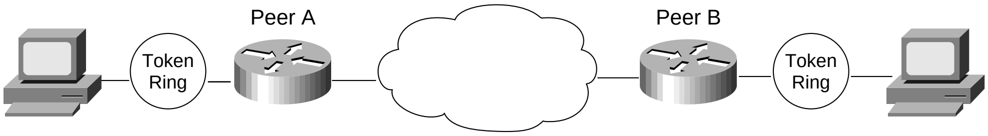


```
Configuration for Peer B
dlsw local-peer peer-id 10.2.17.1
dlsw remote-peer 0 tcp 10.2.24.2
dlsw remote-peer 0 tcp 10.2.24.3

Configuration for Peer C
dlsw local-peer peer-id 10.2.18.1
dlsw remote-peer 0 tcp 10.2.24.2
dlsw remote-peer 0 tcp 10.2.24.3
```

```
Configuration for Peer A1
dlsw local-peer peer-id 10.2.24.2 cost 2
     promiscuous
dlsw icanreach mac-addr 4000.3745.0001

Configuration for Peer A2
dlsw local-peer peer-id 10.2.24.3 cost 4
     promiscuous
dlsw icanreach mac-addr 4000.3745.0001
```

The **dlsw icanreach** command also supports the **mac-exclusive** and **netbios-exclusive** keywords, which indicate that the resources advertised by this peer are the only resources the peer can reach. By specifying **mac-exclusive** or **netbios-exclusive**, you can indicate that the list of specified MAC addresses or NetBIOS names are the *only* ones reachable from a given router. Figure 4-4 shows how **dlsw icanreach netbios-exclusive** can be used to prevent other branch routers from sending explorers for NetBIOS servers other than those advertised.

Figure 4-4 DLSw+ Configured to Advertise Reachability of a Server While Concurrently Advertising that No Other NetBIOS Names Are Reachable



```
Configuration for Peer A
dlsw local-peer peer-id 10.2.17.1
dlsw remote-peer 0 tcp 10.2.18.1
dlsw icanreach netbios-name nysales01
dlsw icanreach netbios-exclusive
```

```
Configuration for Peer B
dlsw local-peer peer-id 10.2.18.1
dlsw remote-peer 0 tcp 10.2.17.1
dlsw icanreach netbios-name lasales01
dlsw icanreach netbios-exclusive
```

Note that if you are using border peers, and remote branch routers do not establish peer connections between them, this reachability information is not exchanged (because the peer connection is not established until *after* the resource is found). When using border peers for branch-to-branch connectivity, sites that communicate frequently can configure direct peer connections and use the **dlsw icanreach** command to preload their cache entries. This eliminates the need to do broadcast searches for frequently accessed resources, but takes advantage of border peer dynamics to find infrequently accessed resources.

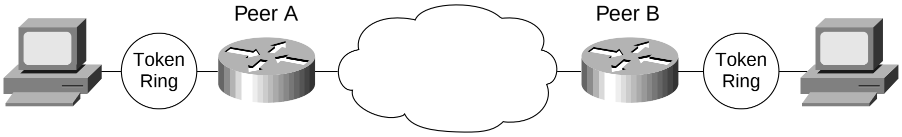
Reachability information learned as part of a capabilities exchange with a remote peer is considered valid as long as that remote peer is active. Multiple central site routers can advertise reachability of the same central site resources. If a remote branch router learns of multiple paths to a central site resource through the capabilities exchange, it will cache up to four paths, and the rules for duplicate path bias apply.

The **dlsw icannotreach saps** command allows you to list SAPs that this router cannot reach locally. This command can be used to advertise to a remote peer that it should not send explorers for certain SAPs (for example, NetBIOS). If there are only a few SAPs that this router can reach, it is probably easier to use the **dlsw icanreach** command.

Defining Static Paths

Static path definition allows a router to set up circuits without sending explorers (the entry is treated as stale until verified, however). The path specifies the peer to use to access a MAC address or NetBIOS name. The remote peer is identified by an IP address or an interface. Path information learned from a static path definition is never deleted. If a static path is not available, the circuit cannot be established. As a result, static paths are more appropriate if only one peer exists that can be used to access a remote node. Figure 4-5 shows how to configure a static path.

Figure 4-5 Configuration for a Static Path Always Used to Reach a Specified MAC Address



```
Configuration for Peer A
disw local-peer peer-id 10.2.17.1
disw remote-peer 0 tcp 10.2.18.1
disw mac-addr 4000.0521.0001 ip-address 10.2.18.1
```

Table 4-1 compares the meaning and use of static path definitions to **icanreach** definitions.

Table 4-1 Comparison of Static Path Configuration and icanreach Configuration

Static Paths	Icanreach
Defines paths to remote resources	Defines reachable local resources
Exclusive does not apply	Includes an exclusive option to minimize unnecessary broadcasts
Not exchanged with capabilities exchange used by this peer	Exchanged with capabilities exchange. Used by remote peers
Never deleted	Deleted from cache when remote peer connection comes down
Single path only	Multiple paths possible

Controlling Transmission Size

Controlling the size of the data transmitted across the network can affect performance in some situations. There are two features in the Cisco IOS software that you need to consider: IP maximum transmission unit (MTU) path discovery and largest frame size.

IP MTU Path Discovery

During peer establishment, peering routers determine the maximum IP frame size to be used for the peer connection. This maximum IP frame size then dictates the maximum number of SNA bytes that can be stored within one IP frame. The default size is 1450 bytes. Therefore, the maximum number of SNA bytes that can be stored within one IP frame is:

1500 - (TCP/IP header + data-link control) = 1450

By increasing the maximum IP frame size, more SNA data can be placed within one TCP frame. This allows you to do the following:

- Increase WAN efficiency by sending large frames
- Decrease the number of TCP acknowledgments
- Reduce router CPU utilization

By specifying IP MTU path discovery, when the peer session is established, each router along the path is queried for its MTU on the output interface. This is done by sending Internet Control Message Protocol (ICMP) echo packets of increasing sizes, with the don’t fragment (DF) bit set. Intermediate routers that do not support that MTU size will respond with an “ICMP packet too big” message. Thus, the originating station knows when it has exceeded the MTU for that path (see RFC 1191 for more information).

By using IP MTU path discovery and by increasing the **ip tcp window-size** setting in each router along the path, you can minimize packet fragmentation. Use the following configuration to set the window size:

ip tcp window-size [*size in bytes*]

Note Setting all MTU sizes to larger values may impact the amount of memory used on the interface card. There is a limited amount of buffer space for the interface cards, and setting the MTU size higher on all interfaces may result in exhausting this memory. More memory will be consumed by buffers if the MTU size is increased. On smaller platforms, such as the Cisco 2500 family of routers, this memory impact may be severe if you only have 2 MB of shared (I/O) memory.

The **ip tcp path-mtu-discovery** command is a global command not specific to an interface. Once this command is active, the maximum IP frame size for a peer connection will be set to the minimum MTU path size on the path of that peer connection. The following is a sample configuration:

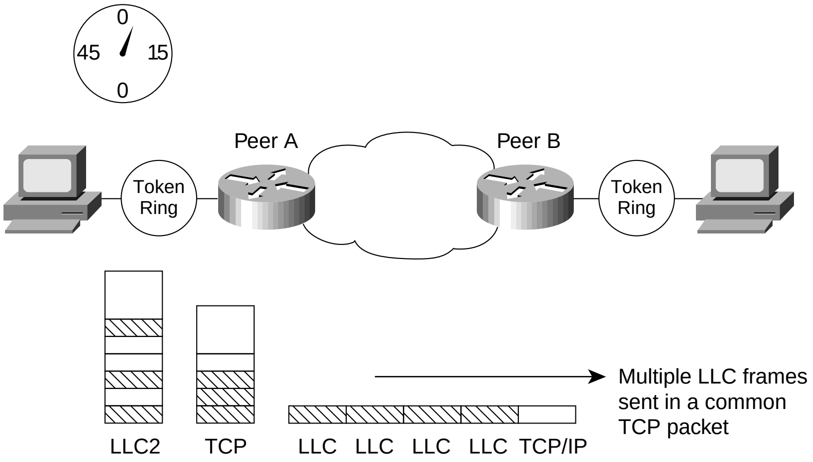
```
Hostname Marulan
enable-password cisco
ip tcp path-mtu-discovery
dlsw local-peer peer-id 172.26.1.1
dlsw remote-peer 0 tcp 172.26.10.1
interface token-ring 0
ip address 172.26.1.1 255.255.255.0
. . .
```

Packet assembly benefits from IP MTU path discovery because during the packet assembly process, more SNA frames can be stored within the TCP frame. For example, if 100 users in a remote location all require 3270 access to the central host, then all SNA request packets will be destined for the same DLSw+ router. During heavy access periods, it is likely that many SNA requests will arrive at the remote router within a short period of time. These multiple SNA frames, all destined for the same host router, can be placed within the same TCP frame, as shown in Figure 4-6. Once the TCP frame is successfully sent to the host router, one TCP acknowledgment can satisfy all the SNA requests.

This packet assembly only occurs during congestion when multiple SNA frames are in the queue. If there is no congestion, it is likely that one SNA packet will map to one TCP frame. DLSw+ will not wait for the multiple packets to arrive in the queue because this would impact end-user response time.

Note When running DLSW+ over low-speed lines (4.8 or 9.6 kbps), an MTU of 576 will provide more consistent response time. Use custom queuing to ensure that SNA gets three times the bandwidth of all other traffic so that an entire screen update is processed at one time.

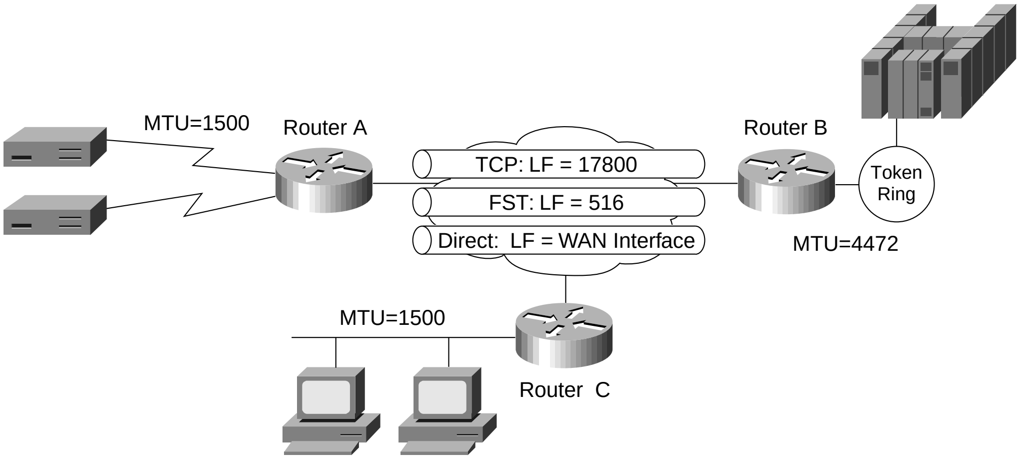
Figure 4-6 DLSw+ Assembles Multiple LLC2 Frames in a Single TCP



Largest Frame Size

When a station is installed on a Token Ring, it can be configured to support a maximum frame size. When this device attempts to connect to its partner (for example, the server, CIP, or FEP), it must send an explorer to locate this device. The originator puts its maximum supported frame size in the explorer. The destination adjusts the maximum frame size before responding. When the response to the explorer is sent, each source-route bridge and each DLSw+ router will query the maximum frame size and adjust as required. When the explorer response reaches the originator, the response will indicate the maximum frame size supported on the entire path. For each explorer, DLSw+ adjusts the maximum frame size to be the minimum of its largest frame size (specified in the **dlsw local-peer** command), the largest frame size of the destination remote peer (specified in the **dlsw remote-peer** command and shared during the capabilities exchange), and the MTU on the local media. The default largest frame size used for remote peers varies by encapsulation type and is shown in Figure 4-7. The default largest frame size in the **dlsw local-peer** command is 17,800.

Figure 4-7 **Default MTU and Largest Frame Sizes for Various Encapsulation Types and Media**



In general, when using TCP encapsulation, you probably will not need to change the largest frame size. However, if using FST (and perhaps if using direct encapsulation), you may need to change the largest frame size. You should change this value only if 1) you know your traffic profile and your output WAN interface MTU and 2) you need to increase throughput.

For example, when using FST, the largest frame default is 516 to ensure that if the packet traverses Ethernet or serial interfaces, you do not exceed 1500 bytes when the DLSw, IP, and data-link control headers are added. If you know your traffic will not traverse an Ethernet LAN, you can increase the largest frame size. You should ensure that the length of the LAN Token Ring packet (less FCS) + 16 (DLSw header) + 20 (IP/FST header) does not exceed the MTU of any interface in the path.

It is meaningful to increase the DLSw+ largest frame size only if the workstations can send larger frames. In this case, by allowing DLSw+ to send larger frames, you will decrease the amount of segmentation required at the workstation. For example, if your message size is 1024 bytes and your maximum frame size on the path is 516 bytes, then the workstation will need to segment the frames. By setting the DLSw+ largest frame size to the next higher valid largest frame to accommodate a 1024-byte information field and all for protocol headers, then the workstation will not need to segment the message.

Set the largest frame size using the following **dlsw local-peer** command:

dlsw local-peer . . . [lf *size***]**

where *size* can be one of the following amounts (bytes):

```
17800
11454
11407
8144
4472
2052
1500
1470
516
```

Timer Settings

There are two types of timer settings: LLC2 idle time and DLSw+ timers.

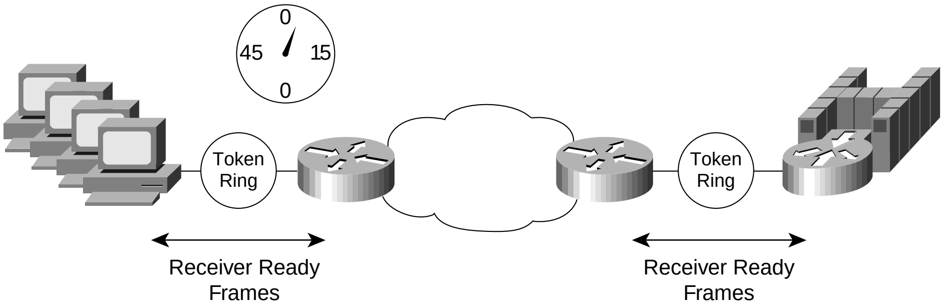
LLC2 Idle Time

LLC2 is a connection-oriented data-link control. Therefore, the end stations involved in the LLC2 connections must periodically check that the LLC2 connection is still active. One way of knowing that a connection is active is by sending and receiving I-frames over the LLC2 connection. Each frame requires an acknowledgment that not only indicates successful receipt of a frame, but also indicates that the connection is still alive. If there is a period of time when no I-frames (in other words, user data) traverse the LLC2 connection, then each workstation must send an LLC2 packet, a receiver ready, to its partner and receive a response to confirm that the LLC2 connection is still operational. The time that the end-stations wait during idle traffic periods before sending a receiver ready frame is called the LLC2 idle time.

Every time the end station sends or receives a frame, it resets its LLC2 idle timer. If the idle timer expires, then the station will send an LLC2 packet to its partner. If there are many thousands of LLC2 sessions, then you will see many LLC2 receiver ready messages traverse the network during idle periods of time.

When a router is locally terminating the LLC2 session, as shown in Figure 4-8, it is the responsibility of the router to adhere to the LLC2 protocol. Thus, during periods of inactivity, the router must send LLC2 requests or acknowledge LLC2 requests from the workstations. This can place an unnecessary load on the router, which can be avoided by increasing the LLC2 idle timer parameter on the LAN segment.

Figure 4-8 LLC2 Receiver Ready Messages Flowing Between End Systems and DLSw+ Routers (One LLC2 Connection at Each DLSw+ Router for Every SNA PU or NetBIOS Session)



A larger LLC2 idle timer value should be implemented when there is a large number of LLC2 sessions. Internal Cisco testing has shown that increasing the LLC2 idle time when supporting 4000 LLC2 sessions decreases the router CPU utilization significantly. The tradeoff is that it will take longer to identify timeout conditions. This is generally a good tradeoff.

A value of 30,000 ms (30 seconds) is suggested, although LLC2 idle time can be increased to as much as 60,000 ms (60 seconds). Use the following syntax to configure this command:

llc2 idle-time *milliseconds*

The maximum value is 60,000. The command to set the LLC2 idle timer is an interface subcommand. Apply it to the appropriate LAN segment. A sample configuration follows:

```
source-bridge ring-group 100
dlsw local-peer peer-id 172.26.1.1
dlsw remote-peer 0 tcp 172.26.10.1
interface token-ring 0
ip address 172.26.1.1 255.255.255.0
source-bridge 3 1 100
llc2 idle-time 30000
. . .
```

DLSw+ Timers

There are several timers in DLSw+ that you can set with a **dlsw timer** command. In general, you do not need to modify these timers. A description of them is included here for completeness along with considerations on the impact of changing them. To change timers, use the following command:

dlsw timer {*timer-type*} *time*

where *time* is specified in seconds or minutes and *timer-type* can be any of the following keywords:

icannotreach-block-time
netbios-cache-timeout
netbios-explorer-timeout
netbios-retry-interval
netbios-verify-interval
sna-cache-timeout
sna-explorer-timeout
sna-retry-interval
sna-verify-interval
explorer-wait-time

The **icannotreach-block-time** is the time the router will mark a resource unreachable after failing in an attempt to find it. While the resource is marked unreachable, searches for that resource are blocked. It is disabled by default. Use this option only if you have excessive explorer traffic and you want to avoid broadcasts for frequently accessed resources that are not currently available or are remote. If used, specify an amount of time that the user is willing to wait for a resource to recover. In some cases (typically in large NetBIOS networks), the NetBIOS station may be up and available, but because of traffic loads, the response may not come back in time. This may cause a peer to consider the station not reachable. If this timer is not specified or set to 0, the user can connect by retrying the command. If the timer is set to 10 minutes, the user cannot connect for 10 minutes.

The **netbios-cache-timeout** is the time that DLSw+ will cache a NetBIOS name location for both the local and remote reachability caches. It defaults to 16 minutes. Setting it lower may cause more broadcasts. Setting it higher increases the chance of having an invalid cache entry. However, for frequently accessed resources, the router will generally delete an invalid cache entry before 16 minutes elapses, so setting this timer to a shorter period of time is probably not necessary.

Cache entries resulting from statically defined reachability paths are never deleted. Cache entries configured using the **dlsw icanreach** command and learned as part of a capabilities exchange are deleted when the associated peer connection is taken down.

The **netbios-explorer-timeout** is the length of time that this router will send explorers to a NetBIOS resource (for LAN resources) or the time DLSw+ will wait for a response before deleting the pending record (for remote resources). It defaults to six seconds. This timer has no impact on when a resource is marked unreachable. Its impact on the LAN is to determine how many retries are sent.

The **netbios-retry-interval** is the interval DLSw+ will wait for a response to a name query or add name query on a LAN before retransmitting the request. The default is one second. Retries will continue to be sent until the NetBIOS explorer timeout is reached (retries are not sent across the WAN).

The **netbios-verify-interval** is the interval between the creation of a cache entry and when the entry is marked stale. If a cache entry is marked stale and a search request comes in for that entry, a directed verify is sent to ensure it still exists. A directed verify is an explorer (for example, NetBIOS NAME-QUERY) sent directly to each cached peer (on the WAN) or a single route explorer sent over every port in the cache (on the LAN). The default is four minutes. Setting this value higher will increase the time it takes for a resource to be found if its cached location is invalid.

The **sna-cache-timeout** is the length of time that DLSw+ will cache the MAC or SAP of an SNA resource before it is discarded. It defaults to 16 minutes. Setting the timer lower may cause more broadcasts. Setting it higher increases the chance of having an invalid cache entry. However, for frequently accessed resources, the router will generally delete an invalid cache entry before 16 minutes elapse, so setting this timer to a shorter period is probably not necessary.

Cache entries resulting from statically defined reachability paths are never deleted. Cache entries configured using the **dlsw icanreach** command and learned as part of a capabilities exchange are deleted when the associated peer connection is taken down.

The **sna-explorer-timeout** is the length of time that this router will send explorers to a NetBIOS (for LAN resources) or the time DLSw+ will wait for a response before deleting the pending record (for remote resources). It defaults to three minutes. This timer has no impact on when a resource is marked unreachable. Its impact on the LAN is to determine how many retries are sent. When using either FST or direct encapsulation without local acknowledgment, this frame is sent over an unreliable mechanism, so it is possible for high volumes of traffic to cause frame drops. In this case, you may want to configure a smaller value for this timer to shorten the time it takes to find resources.

The **sna-retry-interval** is the interval DLSw+ will wait for a response to a TEST or XID request on a LAN before retransmitting the request. The default is 30 seconds.

The **sna-verify-interval** is the interval between the creation of a cache entry and when the entry is marked stale. If a cache entry is marked stale and a search request comes in for that entry, a directed verify is sent to ensure it still exists. A directed verify is a canureach frame sent directly to every cached peer (on the WAN) or a single route explorer sent over every port in the cache (on the LAN). The default is four minutes. Setting this value higher will increase the time it takes for a resource to be found if its cached location is invalid.

The **explorer-wait-time** is the number in seconds that DLSw+ will wait after sending an explorer before picking a peer as the best path. When DLSw+ starts exploring, it waits for *time* seconds before responding to the TEST frame. Setting this timer to one to two seconds will give DLSw+ time to learn all possible peers before selecting the least-cost peer. Do not modify this timer unless you have multiple central site peers, you are using cost to select a preferred peer, and your capable peer will frequently respond first before your preferred peer.

Queue Depths

During congestion, packets might get queued in the router. You can control the depth of certain queues to improve network performance.

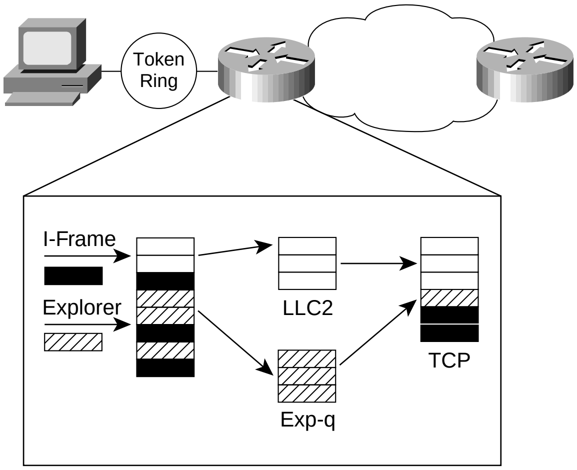
Explorer Queue Depth

Explorers are used to find resources in DLSw+ and on LANs. Explorer caching by DLSw+ helps decrease the steady state explorer load on both the network and on DLSw+ routers. When a DLSw+ router receives an explorer for a cached resource, it either responds locally or sends a directed explorer.

Problems occur when there is an excessive amount of broadcast traffic (known as a broadcast storm) and the explorers arrive at a rate faster than DLSw+ can process them. To address this, you can use the **source-bridge explorerq-depth** command. By using this command, you actually do two things: you create a separate queue for explorer traffic and you limit the number of explorers that can be queued waiting to be processed. Without doing this, a broadcast storm may cause input buffers to fill up with explorer traffic, preventing end-user traffic from getting through.

Figure 4-9 illustrates explorer processing. When an explorer queue is full, any incoming explorers are dropped, causing end systems to retransmit the explorers. By creating a separate SRB explorer queue and limiting its size, you can ensure that explorer traffic does not monopolize DLSw+ buffers.

Figure 4-9 Explorer Processing in a DLSw+ Router



The syntax of the command is:

source-bridge explorerq-depth *depth*

where *depth* is the maximum number of incoming explorers. Once this number is reached, new explorers will be dropped. If you have excessive explorer traffic, set this value to between 10 and 20.

Typically, when there is a explorer storm, most explorers are destined to the same MAC address. Dropping these explorers (when the queue is full) gives the router time to receive the reply to the explorers that were processed and, therefore, obtain a cache hit. Once the cache hit is obtained, the router can respond to the explorers without forwarding them.

Input Hold Queue

This queue is used to hold input frames off the LAN segment (other interface types as well, but we will concentrate on LAN segments) that are waiting to be placed into a system buffer. During peak loads, you may see some buildup (or drops) in this queue. (Use the **show interface** command to get this information. See the chapter "Using Show and Debug Commands" for more information.) Some protocols that are very traffic intensive during startup may require the input hold queue to be increased. Increasing the hold queue enables the router to queue more packets while the router tries to allocate a system buffer.

A good example of this is a startup of APPN sessions. There are many small packets that flow during startup, and it is not unusual to see a buildup in the input hold queue (in other words, the packets come off the Token Ring segment much faster than the router can process them out the WAN ports).

It should be noted that if you see constant drops on the input hold queue, then increasing the input hold queue will not help. There is probably another problem in the network. Increasing the input hold queue can help when there is a transient load (for example, at startup) where the router needs the ability to hold on to a few more packets than normal. This will alleviate packet retransmission and minimize the possibility of further dropped packets.

This command is an interface subcommand. It can be applied to any interface. The syntax of the command is:

hold-queue *length* **[in | out]**

where *length* is the number of buffers that can be stored. The default is 75 input buffers and 40 output buffers. The following is a sample configuration:

```
source-bridge ring-group 100
dlsw local-peer peer-id 172.26.1.1
dlsw remote-peer 0 tcp 172.26.10.1
interface token-ring 0
ip address 172.26.1.1 255.255.255.0
source-bridge 3 1 100
llc2 idle-time 30000
hold-queue 200 in
```

System Buffers

In an SNA environment, dropping system buffers is not good. Consistently dropping buffers will lead to SNA session loss, and therefore, system buffer tuning is required to prevent this situation.

Note This section describes how to diagnose a system buffer problem and to compile enough information so that a Cisco engineer (system engineer or customer engineer) can assist with the system buffer changes. Do not attempt to adjust buffers without assistance.

System buffers come in various sizes (small, middle, large, very large, and huge). The memory used for these buffers is called I/O or shared memory. Low-end routers have one memory location for I/O and another memory location for main memory. High-end routers have one block of memory split into main and I/O.

For process switched traffic, when a packet arrives in the router, it is placed in the smallest size buffer that can accommodate it. If that size buffer is not available and the router can create another buffer quickly enough, it will. Once this new buffer is created, is stays in the pool temporarily but is trimmed back later. This freed memory can then be used to create any other size buffer.

If the router cannot create a buffer in time, a buffer miss will be recorded. If the router cannot create a buffer because there is no more I/O memory available, then a no memory condition is recorded. Not having enough I/O memory available indicates a problem.

Two **show** commands are used to diagnose buffer problems: **show memory** and **show buffers**. The **show memory** command will display the total amount of memory available, memory used, and memory currently available. The **show buffers** command will detail all the buffer information: number of misses, number of no memory conditions, and number of buffers assigned.

If you suspect a memory problem, check the status of your buffers using the **show buffers** command. If you see some buffer misses, do not be alarmed. It is not unusual to see some misses (in other words, if the router has been running for several weeks, you may see that over this time you have 100 misses).

If you view your buffers and see that the miss count is incrementing (by issuing a few **show buffers** commands), then take note of which buffer size is being missed.

Once you have the details of the buffer misses, issue the **show memory** command and take note of the amount of shared (or I/O) memory that is still available. If this value is still larger than 1 MB, it is likely that tuning your buffers will alleviate the buffer misses.

If you note that no memory conditions are occurring (from the **show buffers** command), note the amount of free shared (or I/O) memory (from the **show memory** command). If you find that the amount of free shared memory is almost zero, it is a serious condition. This will occur for one of two reasons:

1. The router needs more I/O memory to accommodate the amount of traffic and flow control requirements
2. You have tuned your buffers and over-allocated in some area and depleted the I/O memory

Once you have gathered this information, open a case with the Cisco Technical Assistance Center, or discuss it with your systems engineer. You should supply the following information:

- Current configuration (issue a **write terminal** command to get this information)
- Description of the symptom (for example, session drops, poor response time, and so forth)
- Output of **show memor**y command (but typically not the whole memory map, just the initial information)
- Output of **show buffers** command (you may want to include the output from multiple **show buffers** commands if you are trying to show an increase in buffer misses; make sure that a **service timestamps log** command has been issued so that the engineer can calculate the misses over time)
- The current Cisco IOS release you are using, which you can determine by issuing a **show version** command

Miscellaneous Customization Options

SRB Explorers

By default, when Cisco’s DLSw+ initiates an explorer, it sends a single route explorer. Most SRB implementations respond to a single route explorer with an all routes explorer so that the best possible path can be selected. If you have an implementation that does not respond to single route explorer with an all routes explorer, you can configure DLSw+ to send explorers as all routes explorers using either the **dlsw allroute-sna** or the **dlsw allroute-netbios** command.

Initial and Maximum Pacing Windows

DLSw+ uses an adaptive pacing flow-control algorithm that automatically adjusts to congestion levels in the network. (This algorithm is described in the “Introduction.”) The default initial pacing window size is 20 and the default maximum pacing window size is 50. Some environments need the ability to adjust this window size. The capability to modify the default window sizes was added in Cisco IOS Release 10.3(14), 11.0(11), 11.1(5), and 11.2.

You may want to set the initial pacing window to a lower value if one side of a connection can send far more data than the other side can receive, for example, if you have a Frame Relay network and the central site router accesses over a T1 link and the remote router accesses over a 56-kbps link. With Cisco IOS Release 11.2, Committed Information Rate (CIR) enforcement provides an alternate way to address this issue. You may want to set the initial or maximum pacing sizes to higher values if one side is frequently waiting for permission to send more traffic and the other side is capable of handling more traffic.

To determine if you should modify either of these defaults, you can use the **show dlsw circuits** command, which will show the current window packets and permitted and granted packets. If the current window shows the maximum of 50 and the permitted and granted packets shows 0 for some time, this indicates that the adaptive pacing has increased to the maximum, but one side is still frequently waiting before it can send more. In this case, you may improve your throughput by increasing the maximum pacing window.

If the current window packet is higher than the initial pacing window but less than the maximum pacing window, and the permitted and granted packet is 0 or very small, it may be a signal that the adaptive pacing algorithm is increasing the window size but is not increasing the window size quickly enough. In this case, you may improve your throughput by increasing the initial pacing window.

If the current window packet is less than the initial pacing window, it may indicate that the receiver cannot absorb traffic as quickly as it can be sent. In this case, you may want to reduce the initial pacing window.

To modify these pacing values, include the following keywords on the **dlsw local-peer** command:

dlsw local-peer . . . [**init_pacing_window** *size*] [**max_pacing_window** *size*]

where *size* can be anything between 1 and 50, but **max_pacing_window** should always be larger than **init_pacing_window**.

CHAPTER 5

Bandwidth Management and Queuing

This chapter describes how you can use Cisco's bandwidth management and queuing features in conjunction with DLSw+ to enhance the overall performance of your network.

Many enterprises run Cisco networks with a mixture of SNA and client/server protocols. If you anticipate that because of your traffic volume or bandwidth limitations, there will be contention for bandwidth, read this chapter. In general, the queuing techniques described in this chapter (with the exception of DLCI prioritization and policy routing) do not even take effect unless there is congestion in the network.

Even if you decide you need to apply some of these queuing techniques, you may not need them everywhere. The output queuing mechanisms described in this chapter can be applied to an individual interface, allowing you to apply queuing to lower-speed access lines while not applying it to higher-speed trunk lines.

Introduction to Cisco IOS Queuing Features

Bandwidth management involves deciding what traffic is highest priority, ensuring that it gets the bandwidth it needs, and deciding how to handle the lower-priority traffic. The Cisco IOS software offers many options for identifying high-priority traffic: protocol, message size, TCP port number, input interface address, LLC SAP, MAC address, SDLC address with serial tunneling (STUN), or LOCADDR.

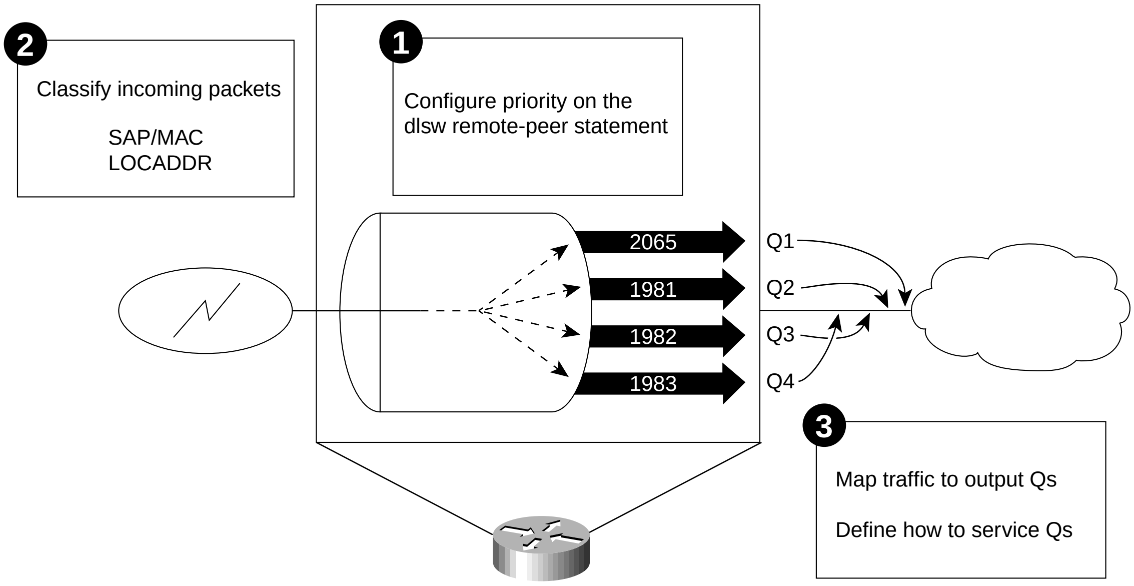
DLSw+ places all SNA and NetBIOS traffic into TCP packets, making it difficult or impossible to identify the traffic by the above characteristics. For that reason, DLSw+ supports opening four separate TCP connections and places traffic directly from the input queue into one of these four pipes based on priority. At the output interface, you can prioritize among these four TCP connections based on their TCP port number.

Once traffic has been assigned to a queue, the Cisco IOS software offers several options for servicing the queues. The key techniques for DLSw+ traffic are *custom queuing* and *priority queuing*. In addition, there is *weighted fair queuing* and *DLCI prioritization*. All of these techniques are described in this chapter.

Figure 5-1 describes the tasks required to configure bandwidth management in a Cisco router. There are three steps:

1. If you chose to distinguish within DLSw+ traffic, to prioritize SNA ahead of NetBIOS, or to prioritize interactive terminal traffic over batch print jobs, you need to include the **priority** keyword in the appropriate **dlsw remote-peer** command. Including this keyword causes DLSw+ to open four TCP connections (identified by ports 2065, 1981, 1982, and 1983). By default, DLSw+ transmits certain traffic over certain TCP connections.
2. The next step is to classify packets on the incoming port and assign the traffic to the appropriate TCP connection. This can be done based on SAP, MAC address, or LOCADDR. If you do Step 1, you must also do Step 2 to have any effect on how the bandwidth is allocated. Step 1 opens the TCP pipes. Step 2 assigns traffic to the pipes.
3. Next, you must assign traffic to the appropriate output queue based on protocol, TCP port number, or message size and then define the queuing technique to be used on the interface (for example, custom queuing or priority queuing). Step 1 and Step 2 may be unnecessary in your environment, but you may still choose to distinguish DLSw+ from other traffic, in which case you need to do Step 3.

Figure 5-1 Tasks Required to Control How Traffic Is Forwarded in Cisco Routers



The queuing of packets only occurs when the total number of outbound packets exceeds the capacity of the outbound link. If a link is not congested, then the router does not need to implement any queuing mechanism, because as soon as it has queued the packet onto the outbound interface, the packet can be sent.

Traffic Classification

The Cisco IOS software supports packet classification by protocol, by TCP port number, by input interface, by message length, or by extended access list. DLSw+ traffic can be classified ahead of other TCP/IP traffic because by default it always uses TCP port number 2065. To classify traffic within DLSw+, specify the **priority** keyword in a **dlsw remote-peer** command.

When **priority** is specified, DLSw+ will automatically activate four TCP connections to that remote peer (ports 2065, 1981, 1982, and 1983). Priority needs to be specified only if you need to prioritize between SNA and NetBIOS, or within SNA by LOCADDR, or MAC or SAP pair (known as SAP prioritization). In addition, this granular packet classification is possible only when TCP encapsulation is selected for a specific remote peer. By default DLSw+ assigns certain traffic to specific TCP ports:

- TCP port 2065 defaults to high priority; in the absence of any other configuration, this port will carry all circuit administration frames (CUR_cs, ICR_cs, contact SSP frames, disconnect SSP frames, XID, ICR_ex), peer keepalives, and capabilities exchange
- TCP port 1981 defaults to medium priority; in the absence of any other configuration, this port will not carry any traffic
- TCP port 1982 defaults to normal priority; in the absence of any other configuration, this port will carry information frames (nonbroadcast datagram frames)
- TCP port 1983 defaults to low priority; in the absence of any other configuration, this port will carry broadcast traffic (CUR_ex, Name_query_ex, SSP DATA/DGRM broadcasts)

Note If you specify **priority** in the **dlsw remote-peer** command and do nothing else, all steady traffic goes in TCP port 1982. If you configure specific traffic to port 2065 (such as all SNA or specific SNA devices), all unspecified traffic goes in TCP port 1982.

You can use classification techniques such as SAP prioritization to change the port assignment of traffic destined for DLSw. However, these techniques have no impact on how the traffic is handled on the output queue. To control how each of the TCP ports is handled on the output queue, you must map the TCP ports to different queue numbers, define the queuing algorithm, and apply that queue list to the output interface.

SAP Prioritization

You can create a priority list that assigns traffic by SAP or MAC address to different TCP ports. You can then apply that list to a LAN interface on a router (support for Ethernet requires Cisco IOS Release 11.0 or later and support for FDDI requires Release 11.2 and a Cisco 7x00). As traffic enters the router, DLSw+ assigns it to a TCP port and passes it to the appropriate output interface.

To provide a fine granularity in the prioritization of packets, the **priority-list** command allows you to specify any combination of destination SAP (DSAP), source SAP (SSAP), destination MAC (DMAC), and source MAC (SMAC). For example, if you want to prioritize all SNA traffic (SAP 04) over NetBIOS traffic (SAP F0), then only the DSAP or SSAP needs to be specified in the command. In contrast, if you want to give precedence to traffic on a particular LLC2 session, then you must specify all four parameters (DSAP, SSAP, DMAC, SMAC) that uniquely identify a LLC2 session. The command syntax is:

sap-priority-list *list-number queue-keyword* [**dsap** *ds*] [**ssap** *ss*] [**dmac** *dm*] [**smac** *sm*] where *list-number* is an arbitrary integer between 1 and 10 that identifies the SAP priority list. The argument *queue-keyword* is a priority queue name or a DLSw+ TCP port name (for example, high, medium, normal, or low).

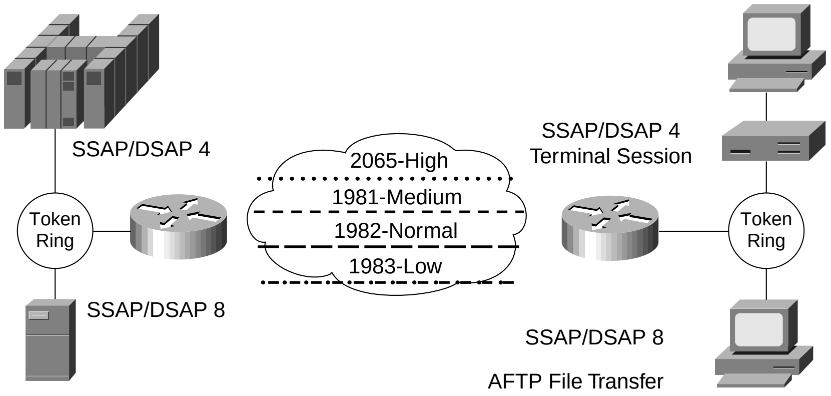
To map a SAP priority list to an Ethernet bridge group (requires Cisco IOS Release 11.0 or later), specify the **sap-priority** keyword on the **dlsw bridge-group** command as follows:

dlsw bridge-group *group-number* **sap-priority** *list*

where *list* identifies the SAP priority list.

In Figure 5-2, SNA batch and SNA interactive traffic are assigned to different TCP ports so that interactive traffic gets preferential service. This is only possible if batch and interactive traffic have different SSAP and DSAP pairs. In this configuration, traffic from SAP 4 is assigned to TCP port number 2065, and traffic from SAP 8 is assigned to TCP port number 1983. Traffic from all other SAPs is placed in TCP port number 1982 by default. Associating the traffic to different TCP ports allows the router to prioritize one type of traffic over the other types. Classifying packets and queuing them to different TCP ports on the input queue *does not* determine how the traffic will be handled on the output queue. The actual prioritization of the TCP ports on the output queue is handled with other commands that will be described later in this chapter.

Figure 5-2 Traffic Assigned to Different DLSw+ TCP Ports Based on SAP



```
sap-priority-list 1 high ssap 4 dsap 4
sap-priority-list 1 low ssap 8 dsap 8
interface TokenRing0
sap-priority 1
```

Another use of SAP prioritization is to give high priority to traffic destined for a FEP by using an output queuing mechanism in conjunction with the following command:

```
sap-priority-list 10 high dmac 4001.3745.0001
```

SAP prioritization only applies for LAN attached devices when using TCP encapsulation to connect to remote peers. SAP prioritization cannot be used in conjunction with LOCADDR prioritization. If both are specified, LOCADDR takes precedence.

LOCADDR Prioritization

LOCADDR is the SNA local address assigned by an SNA boundary network node (PU 4/5) to uniquely identify a dependent SNA LU. (For independent LUs, the LOCADDR is assigned dynamically during session establishment and cannot be used to distinguish between application types.) LOCADDR is carried in the SNA format indicator 2 (FID2) headers that are used when a PU 2.0/2.1 communicates with a PU 4/5.

When DLSw+ is used to transport data between PU 2.0/2.1 and PU 4/5, you can prioritize SNA traffic by LOCADDR. To do this, create a priority list that assigns traffic based on LOCADDR to different TCP ports. Then apply that list to a Token Ring or SDLC interface on a router. As traffic enters the router, DLSw+ assigns it to a TCP port and passes it to the appropriate output interface.

To provide fine granularity in the prioritization of packets, the **locaddr-priority-list** command allows you to prioritize individual LUs. For example, this command lets you prioritize interactive devices ahead of printers.

The command syntax is:

locaddr-priority-list *list-number address-number queue-keyword*

where *list-number* is an arbitrary integer between 1 and 10 that identifies the priority list. The argument *address-number* uniquely identifies an SNA device.

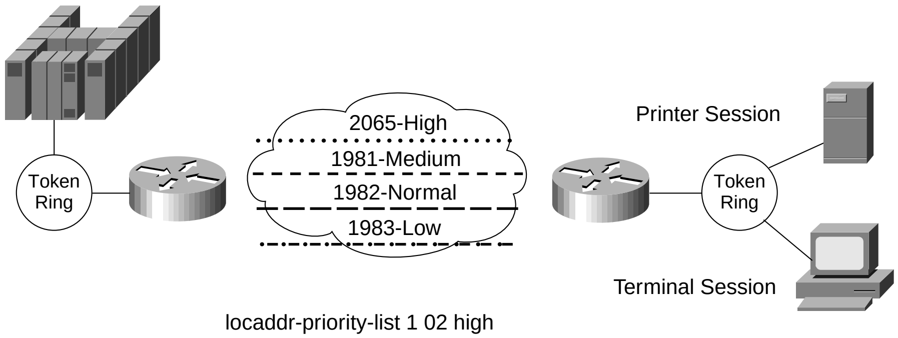
To map a LOCADDR priority list to an Ethernet bridge group (requires Cisco IOS Release 11.0 or later), specify the **locaddr-priority** keyword on the **dlsw bridge-group** command as follows:

dlsw bridge-group *group-number* **locaddr-priority** *list*

where *list* identifies the SAP priority list.

In Figure 5-3, the printer (at LOCADDR 4) is assigned to TCP port 1983. A specific terminal or set of terminals can be assigned to TCP port 2065. All other DLSw+ traffic defaults to TCP port 1982. Classifying packets into different TCP ports on the input queue does not determine how the traffic will be handled on the output queue. The actual prioritization of the TCP ports on the output queue is handled with other commands that will be described later in this chapter.

Figure 5-3 Traffic Assigned to Different DLSw+ TCP Ports Based on LOCADDR



```
locaddr-priority-list 1 02 high
locaddr-priority-list 1 04 low
interface TokenRing0
locaddr-priority 1
```

LOCADDR prioritization applies to dependent LUs attached to DLSw+ via QLLC, SDLC, Token Ring, Ethernet, or FDDI when using TCP encapsulation to communicate with remote peers. LOCADDR prioritization cannot be used in conjunction with SAP prioritization. If both are specified, LOCADDR takes precedence.

Queuing Algorithms

The Cisco IOS software implements four different output queuing algorithms:

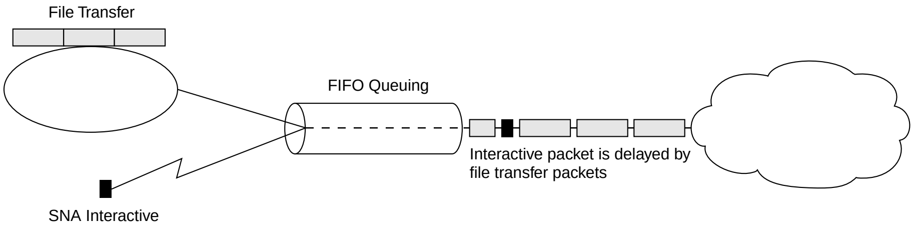
- First in, first out queuing
- Priority queuing
- Custom queuing
- Weighted fair queuing

Each queuing method has advantages and disadvantages. This section describes how each one works and shows configuration examples.

First In, First Out Queuing

This is the simplest and most common interface queuing technique and works well if links are not congested. The first packet to be placed on the output interface queue is the first packet to leave the interface (see Figure 5-4). The problem with first in, first out queuing is that when a station starts a file transfer, it can consume all the bandwidth of a link to the detriment of interactive sessions. The phenomenon is referred to as a packet train because one source sends a “train” of packets to its destination and packets from other stations get caught behind the train. First in, first out queuing is effective for large links that have little delay and minimal congestion.

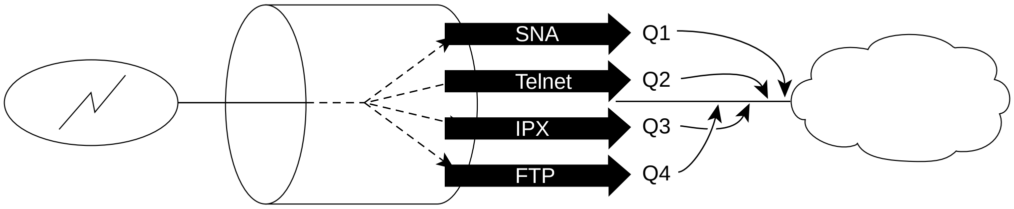
Figure 5-4 Potential Impact of a File Transfer on Interactive Traffic



Priority Queuing

Priority queuing allows network managers to define how they wish traffic to be prioritized in the network. By defining a series of filters based on packet characteristics, traffic is placed into a number of queues; the queue with the highest priority is serviced first, then the lower queues are serviced in sequence (see Figure 5-5). If the highest priority queue is always full, then this queue will continually be serviced and packets from the other queues will queue up and be dropped. In this queuing algorithm one particular kind of network traffic can dominate all others. Priority queuing assigns traffic to one of four queues: high, medium, normal, and low.

Figure 5-5 Priority Queuing Services Traffic on the Highest Priority Queue First



```
Interface Serial1
ip address 20.0.0.1 255.0.0.0
priority-group 1
!
priority-list 1 protocol ip high tcp 2065
priority-list 1 protocol ip medium tcp 23
priority-list 1 protocol ipx normal
priority-list 1 protocol ip low tcp 21
```

In Figure 5-5, the **priority-group** command assigns priority list 1 to Serial1. The **priority-list** command defines the queuing algorithm to be used by queue list 1 and maps the traffic into various queues. Priority queuing is useful when you want to guarantee that the DLSw+ traffic will get through even if it delays other types of traffic. It works best if the DLSw+ traffic is low volume (for example, a small branch with a transaction rate of five to ten transactions per minute), and the number of queues is kept to a minimum (two or three). In this configuration, DLSw+ is in the highest-priority queue, IPX is in the normal queue, and FTP (TCP port 21) is in the lowest-priority queue.

Custom Queuing

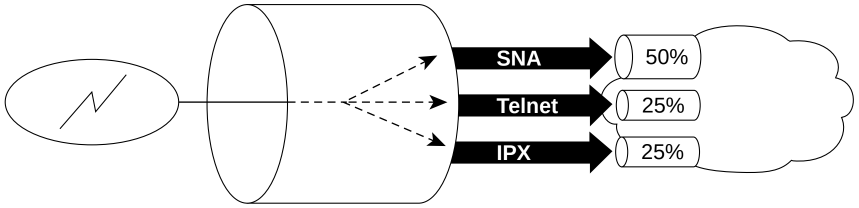
Custom queuing, or bandwidth allocation, reserves a portion of the bandwidth of a link for each selected traffic type. To configure custom queuing, the network manager must determine how much bandwidth to reserve for each traffic type. If a particular type of traffic is not using the bandwidth reserved for it, then other traffic types may use the unused bandwidth.

Custom queuing works by cycling through the series of queues in round-robin order and sending the portion of allocated bandwidth for each queue before moving to the next queue. If one queue is empty, the router will send packets from the next queue that has packets ready to send. Queuing of packets is still first in, first out in nature in each classification (unless APPN is running in the router, in which case the queue is ordered by SNA transmission priority), but bandwidth sharing can be achieved between the different classes of traffic.

In Figure 5-6, custom queuing is configured to take 4000 bytes from the SNA queue, 2000 bytes from the Telnet queue, and 2000 bytes from the IPX queue. This allocates bandwidth in the proportions of 50, 25, and 25 percent. If SNA is not using all its allocated 50 percent of bandwidth, the other queues can utilize this bandwidth until SNA requires it again. The following example shows how to configure custom queuing to allocate bandwidth as illustrated in Figure 5-6:

```
Interface Serial 0
ip address 20.0.0.1 255.0.0.0
custom-queue-list 1
!
queue-list 1 protocol ip 1 tcp 2065
queue-list 1 protocol ip 2 tcp 23
queue-list 1 default 3
queue-list 1 queue 1 byte-count 4000
queue-list 1 queue 2 byte-count 2000
queue-list 1 queue 3 byte-count 2000
```

Figure 5-6 Custom Queuing Removes Specified Byte Count of Traffic from Each Queue in Round-Robin Fashion, Allocating the Bandwidth Proportionally Among the Queues



Custom queuing is commonly used when deploying DLSw+ networks because it allows the network manager to ensure that a guaranteed percentage of the link can be used for SNA, Telnet, and FTP. However, unless the DLSw+ traffic is broken into separate TCP conversations (using SAP or LOCADDR prioritization described earlier), batch SNA transfer or NetBIOS traffic will share the same output queue and may negatively impact interactive SNA response times.

In Cisco IOS Release 11.0, the number of queues available for custom queuing was increased from 10 to 16. The byte counts you should assign to each queue depend upon the bandwidth of the link and the message sizes of the protocols. Byte counts that are too high may adversely skew the performance of custom queuing on low-speed interfaces.

Considerations

When choosing the byte count values for each queue you must consider the following:

- Once the byte count value is exceeded, the frame that is currently being transmitted will be completely sent. Therefore, if you set the byte count to 100 bytes and the frame size of your protocol is 1024 bytes, then every time this queue is serviced, 1024 bytes will be sent, *not* 100 bytes.
- Very large byte counts will produce a “jerky” distribution. That is, if you assign 10,000, 15,000, 20,000, and 25,000 to four queues, each protocol will be serviced nicely when its queue is the one being serviced, but once serviced it may take some time to get back to that queue.
- Window size will also affect the bandwidth distribution. If the window size of a particular protocol is set to one, then that protocol will not place another frame in the queue until it receives an acknowledgment. The custom queuing algorithm will move to the next queue if the byte count is exceeded or there are no frames in that queue. Therefore, with a window size of one, only one frame will be sent each time. If your byte count is set to 2 KB and your frame size is 256 bytes, then only 256 bytes will be sent each time this queue is serviced.
- You need to know the frame size of each protocol. Some protocols, such as IPX, will negotiate the frame size at session startup time.

Determining the Byte Count

To ensure that the actual bandwidth allocation is as close as possible to the desired bandwidth allocation, you must determine the byte count based on each protocol’s frame size. Without doing this, your percentages may not match what you configure.

For example, suppose one protocol has 500-byte frames, another has 300-byte frames, and a third has 100-byte frames. If you want to split the bandwidth evenly across all three protocols, you might chose to specify byte counts of 200, 200, and 200 for each queue. However, that will not result in a 33:33:33 ratio because when the router serviced the first queue, it would send a single 500-byte frame; when it serviced the second queue, it would send a 300-byte frame; and when it serviced the third queue, it would send two 100-byte frames, giving you an effective ratio of 50:30:20. Had you instead specified 1000, 1000, 1000, the router would send two 500-byte frames, five 200-byte frames, and ten 100-byte frames with a bandwidth ratio of exactly 33:33:33.

However, the delay to send 1000 bytes might be too large. Another alternative is to specify 500, 600, 500, which will result in a ratio of 31:38:31 and may be acceptable.

Fortunately, you do not have to use trial and error to determine the correct byte counts. To determine byte counts, follow these steps:

1. Produce a ratio of all frame sizes, dividing all frame sizes by the largest frame size. For example, assume that the frame size for protocol A was 1086 bytes, for protocol B was 291 bytes, and for protocol C was 831 bytes. The ratios would be:

   1086/1086 : 1086/291 : 1086/831

2. Now multiply the results by the percentages of bandwidth you want each protocol to have. In this example we will allocate the following percentages: 20 percent for A, 60 percent for B, and 20 percent for C. This gives us:

   1086/1086(0.2) : 1086/291(0.6) : 1086/831(0.2)

   or

   .2 : 2.239 : 0.261

3. Again, normalize the ratio by dividing each value by the smallest value, that is:

   .2/.2 : 2.239/.2 : .261/.2

   or

   1:11.2:1.3

   This is the ratio of the number of frames that must be sent so that the percentage of bandwidth that each protocol uses is approximately in the ratio of 20, 60, and 20 percent.

4. Note that any fraction in any of the ratio values means that an additional frame will be sent. In the example above, the number of frames sent would be one 1086 byte frame, twelve 291-byte frames, and two 831-byte frames, or 1086, 3492, and 1662 bytes, respectively, from each queue. These are the byte counts you would specify in your custom queuing configuration.

   To determine the bandwidth distribution this represents, first determine the total number of bytes sent after all three queues are serviced:

   (1 x 1086) + (12 x 291) +(2 x 831) = 1086 + 3492 + 1662 = 6240

   Then determine the percentage of the 6240 bytes that was sent from each queue:

   1086/6240, 3492/6240, 1662/6240 = 17.4, 56, and 26.6 percent

As you can see, this is close to the desired ratio of 20:60:20. The resulting bandwidth allocation can be tailored further by multiplying the original ratio of 1:11.2:1.3 by an integer, and trying to get as close to three integer values as possible. For example, if we multiply the ratio by 2, we get 2:22.4:2.6. We would now send two 1086-byte frames, twenty-three 291-byte frames, and three 831 byte frames, or 2172+6693+2493, for a total of 11358 bytes. The resulting ratio is 19:59:22 percent, which is much closer to the desired ratio than we achieved above.

Do not forget that using a very large byte count may cause other problems.

Custom Queuing Configuration

Below is the basic configuration used for custom queuing with SAP prioritization. The following frame sizes were assumed: APPC=59 bytes and 291 bytes, NetBIOS=124 bytes, IP=1086 bytes, and IPX=831 bytes.

```
ssap-priority-list 1 low ssap F0 dsap F0
locaddr-priority-list 1 2 high
locaddr-priority-list 1 3 low
locaddr-priority-list 1 4 medium
source-bridge ring-group 3
dlsw local-peer 136.222.2.
dlsw remote-peer 0 tcp 136.222.1.1 priority
!
interface Ethernet0
ip address 128.207.1.152 255.255.255.0
!
interface Serial0
ip address 136.222.10.2 255.255.255.0
no keepalive
custom-queue-list 3
!
interface Serial1
ip address 136.222.20.2 255.255.255.0
no keepalive
custom-queue-list 3
!
interface TokenRing0
ip address 136.222.2.1 255.255.255.0
ring-speed 16
source-bridge active 2 1 3
source-bridge spanning
sap-priority 1
locaddr-priority 1
llc2 ack-max 7
llc2 ack-delay-time 1
!
router igrp 100
network 136.222.0.0
!
router igrp 109
network 131.108.0.0
!
queue-list 3 protocol ip 1 tcp 2065
queue-list 3 protocol ip 2 tcp 1981
queue-list 3 protocol ip 3 tcp 1982
queue-list 3 protocol ip 4 tcp 1983
queue-list 3 protocol ip 5
queue-list 3 protocol ipx 6
queue-list 3 default 7
queue-list 3 queue 1 byte-count 1200
queue-list 3 queue 4 byte-count 1200
queue-list 3 queue 5 byte-count 1200
queue-list 3 queue 6 byte-count 1200
queue-list 3 queue 7 byte-count 500
```

Weighted Fair Queuing

Weighted fair queuing classifies traffic into conversations and applies priority (or weights) to identified traffic to determine how much bandwidth each conversation is allowed relative to other conversations. Conversations are broken into two categories: those requiring large amounts of bandwidth and those requiring a relatively small amount of bandwidth. The goal is to always have bandwidth available for the small bandwidth conversations and allow the large bandwidth conversations to split the rest proportionally to their weights.

Cisco implements bitwise round-robin fair queuing in Cisco IOS Release 11.0 and later. The prime advantage of fair queuing is that it requires no configuration from the network manager because the router automatically classifies packets passing through an interface into conversations, based on the following:

- TCP/User Datagram Protocol (UDP) port address
- IP source/destination address, protocol type, type of service
- Frame Relay DLCI
- X.25 logical channel number (LCN)
- SRB frame MAC/SAP

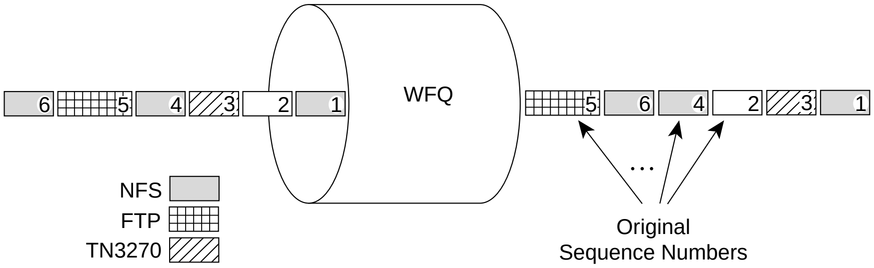
In each case, enough of the packet is checked to break down the streams of packets into separate conversations.

A key disadvantage is that weighted fair queuing does not offer as precise a control over the bandwidth allocation as custom queuing. In addition, in SNA environments, weighted fair queuing typically sees multiple SNA conversations as a single conversation. For example, DLSw+ uses either one or four TCP ports. APPN uses a single LLC2. Hence, instead of SNA interactive sessions moving to the front of the queue, DLSw+ TCP pipes may move to the back of the queue, depending on the number of sessions and quantity of traffic being sent over DLSw+. It is possible, however, to weight certain queues more heavily, which is recommended when using weighted fair queuing in conjunction with DLSw+ or other SNA features. This is covered toward the end of this chapter. In general, do not view weighted fair queuing as an alternative to custom queuing or priority queuing in SNA environments, but simply as a better means of handling default queuing when compared to first in, first out.

In weighted fair queuing, packets between active conversations are reordered so that low-volume conversations are moved forward and high-volume conversations are moved toward the tail of the queue. This reordering results in packet trains being broken up and low-volume conversations receiving preferential service. The high-volume conversations share the delay induced by reordering equally, whereby no one conversation is affected more than another.

In Figure 5-7, packets arrive at the router in the order indicated on the left. They are then reordered according to the size and volume of the three conversations so that the packet from conversation 3 (TN3270) is sent second.

Figure 5-7 Weighted Fair Queuing Reorders Packets on the Output Queue and Packets Within a Single Conversation Are not Reordered



The weighting in weighted fair queuing is currently affected by two mechanisms: IP precedence and Frame Relay discard eligible (DE), forward explicit congestion notification (FECN), and backward explicit congestion notification (BECN). The IP precedence field has values between 0 (the default) and 7. As the precedence value increases, the algorithm allocates more bandwidth to that conversation, which allows it to transmit more frequently.

In a Frame Relay network, the presence of congestion is flagged by the FECN and BECN bits. Once congestion is flagged, the weights used by the algorithm are altered so that the conversation encountering the congestion transmits less frequently.

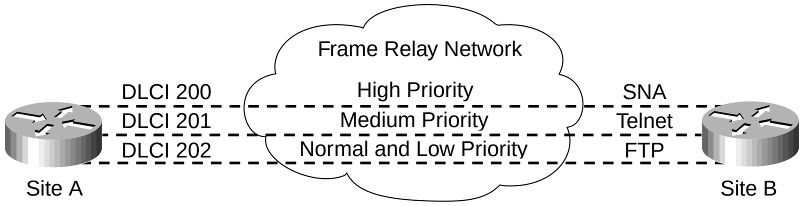
DLCI Prioritization

DLCI prioritization is a process where different traffic types are placed on separate DLCIs so that the Frame Relay network can provide a different CIR for each traffic type. It can be used in conjunction with custom queuing or priority queuing to provide bandwidth management control over the access link to the Frame Relay network. In addition, some Frame Relay switches (for example, the Stratacom IPX, IGX, and BPX/AXIS switches) actually provide prioritization within the Frame Relay cloud based on this priority setting. This feature was introduced in Cisco IOS Release 11.0.

In Figure 5-8, SNA traffic is placed on the first DLCI, Telnet is placed on the second DLCI, and all other traffic is placed on the third DLCI. (The first DLCI number corresponds to high, the second to medium, and so on.) Traffic can be differentiated up to four different DLCIs with this feature. CIRs for each DLCI can then be set to a CIR Be and Bc value appropriate to the characteristics of traffic being sent across the DLCI. The following configuration shows how to use DLCI prioritization to place DLSw+ traffic on DLCI 200, Telnet on DLCI 201, and all other traffic on DLCI 202:

```
Interface Serial0
no ip address
encapsulation frame-relay
!
interface Serial0.200 point-to-point
ip address 20.0.0.1 255.0.0.0
priority-group 2
frame-relay priority-dlci-group 2 200 201 202 202
!
priority-list 2 protocol ip high tcp 2065
priority-list 2 protocol ip medium tcp 23
priority-list 2 default low
```

Figure 5-8 DLCI Prioritization Places SNA, Telnet, and FTP Traffic on Different DLCIs



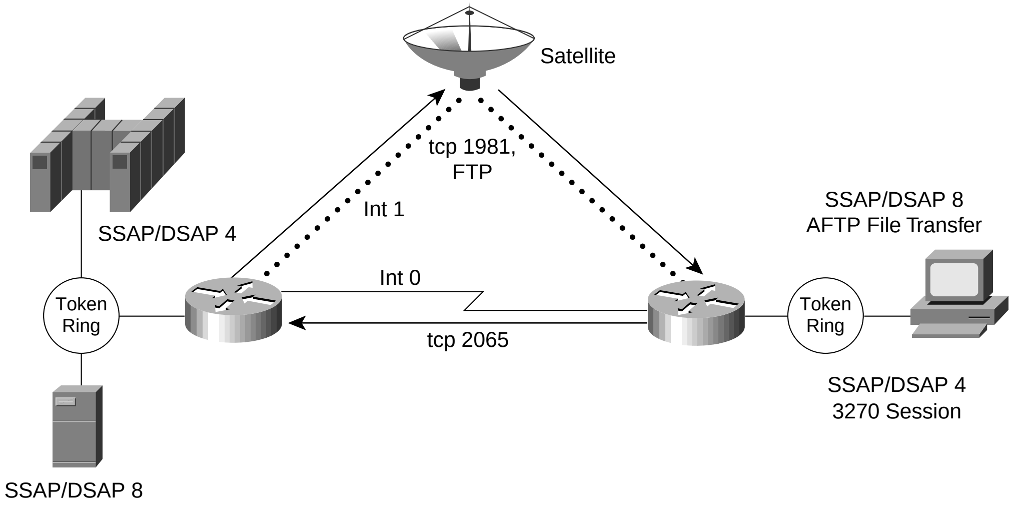
Directing Traffic Flows with Policy Routing

Policy routing is the ability to specify the path that traffic will take through the network or the priority it will receive, based on user-specified parameters.

By using policy routing, a network administrator can control the traffic path, bypassing the normal routing tables. This can be useful where transmission lines between two points have differing characteristics.

In Figure 5-9, there is a low-bandwidth terrestrial link with a low-propagation delay between two points, and a high-bandwidth, high-propagation delay satellite link. The low-bandwidth SNA interactive traffic would be best directed across the terrestrial link, and FTP and SNA file transfers across the satellite link. Policy routing, which was introduced in Cisco IOS Release 11.0, allows you to achieve this.

Figure 5-9 Policy Routing and SAP Prioritization Direct Traffic Across Links That Best Meet the Services Requirements of the Traffic



To achieve the result shown in Figure 5-9, use the following configuration:

```
   source-bridge ring-group 100
   dlsw local-peer peer-id 4.0.0.4
   dlsw remote-peer 0 tcp 5.0.0.5 priority      /* priority keyword opens 4 TCP ports
   interface TokenRing0
   ring-speed 16
   sap-priority 1                               /* maps a sap-priority list to an interface
   source-bridge 1 1 100
   source-bridge spanning
   ip policy route-map test                     /* use policy routing for ip traffic from
                                                   this ring
   sap-priority-list 1 high ssap 4 dsap 4       /* assigns terminal sessions to high
   sap-priority-list 1 low ssap 8 dsap 8        /* assigns AFTP sessions to low
   interface Serial 0
   ip address 20.0.0.1 255.0.0.0
   interface Serial 1
   ip address 30.0.0.1 255.0.0.0
   ip local policy route-map test              /* use policy routing for IP originating in
                                                  this rtr
   access-list 101 permit tcp any any eq 2065   /*permit port 2065 with any ip address
   access-list 102 permit tcp any any eq 1981   /* AFTP traffic (now in tcp 1981)
   access-list 102 permit tcp any eq 20 any     /* FTP traffic
   route-map test permit 3                      /* Defined default path
   set default int serial 0
   route-map test permit 2                      /* Define route map “test” 2
   match ip address 102                         /* all ip addresses that pass filter 102
   set ip next-hop 30.0.0.7
   route-map test permit 1                      /* Define route map “test” 1
   match ip address 101                         /* all ip addresses that pass filter 101
set ip next-hop 20.0.0.6
```

The configuration shows how to use a combination of techniques to prioritize traffic across a WAN. The configuration for policy routing is achieved via route maps. Interface Serial 0 is connected to the terrestrial land line, and Serial 1 is connected to the satellite. Policy routing causes the routing table (which is normally used for forwarding packets) to be ignored and the network administrator's rules to be applied to the forwarding of packets.

You can also use policy routing to determine routing priorities. Policy routing allows you to classify traffic and set the appropriate IP precedence value. In this manner you can sort the network traffic into various types of service at the perimeter of the network and implement those types of service in the core of the network using priority, custom, or weighted fair queuing. As mentioned earlier, the weighting in weighted fair queuing is determined by the value of the IP precedence field. As the precedence value increases, more bandwidth is allocated to that conversation, which allows it to transmit more frequently. This eliminates the need to explicitly classify the traffic at each WAN interface in the core network.

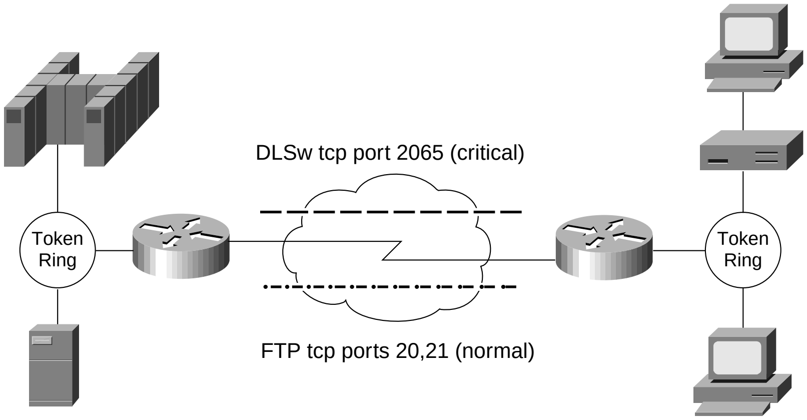
Precedence is a field in the IP header that is used to determine the priority of a packet. Most applications do not set this field so it is typically set to zero. There are eight possible values for the precedence field (see Table 5-1).

Table 5-1 Precedence Field Values

Value	Definition	
Network	Match packets with network control precedence	(7)
Internet	Match packets with internetwork control precedence	(6)
Critical	Match packets with critical precedence	(5)
Flash-override	Match packets with flash override precedence	(4)
Flash	Match packets with flash precedence	(3)
Immediate	Match packets with immediate precedence	(2)
Priority	Match packets with priority precedence	(1)
Routine	Match packets with routine precedence	(0)

By modifying the precedence value, you can increase the amount of bandwidth that weighted fair queuing will allocate to the conversation. For example, by giving DLSw+ traffic a precedence of critical, as shown in Figure 5-10, the DLSw+ conversation (which is all DLSw+ traffic on a given TCP connection) will be given more weight than an FTP conversation going across the same link.

Figure 5-10 Use Policy Routing to Set the Precedence Bits to Give DLSw+ More Weight



If you are using DLSw+ in a weighted fair queuing environment, it is important to configure DLSw+ with more weight, because a single DLSw+ peer connection carries many discrete conversations. Weighted fair queuing only sees one conversation.

The following configuration uses policy routing with weighted fair queuing to set the precedence bits to give DLSw+ more weight:

```
source-bridge ring-group 100
dlsw local-peer peer-id 4.0.0.4
dlsw remote-peer 0 tcp 5.0.0.5
interface Serial 0
ip address 20.0.0.1 255.0.0.0
ip local policy route-map test/* turns on policy routing
access-list 101 permit tcp any any eq 2065/*allows any ip address w/port 2065
route-map test permit 20
match ip address 101/* all ip addresses that pass filter 101
set ip precedence critical
```

Designing Hierarchical Networks

Hierarchical DLSw+ networks are the easiest networks to design and build. They involve minimal routing and are inherently scalable. If you are going to design a hierarchical DLSw+ network, you must answer the following questions:

- How many central site routers are required to handle the traffic load?
- Where is the best place for the central site peer routers?
- How will backup be performed?
- What can be done to minimize explorer traffic and broadcast replication?

This chapter discusses each of these questions and provides information to assist you in making the best decisions for your network. Read this chapter if you are connecting several remote branches to a single primary data center. You may also need to read the chapter “Designing Meshed Networks” if you have frequent branch-to-branch communication among SNA or NetBIOS applications.

Determining the Required Number of Peering Routers

There are many factors involved in determining the number of central site routers required to support a hierarchical network. These factors include the following:

- Number of SNA PUs or concurrent LLC2s to be supported
- Transaction rate at central site and transaction size
- Encapsulation method selected
- Central site routers used for peering
- Number of remote peers connected
- Explorer replication
- Other router processes, such as multiprotocol routing and route table maintenance, compression, and encryption

Number of Devices

The number of SNA PUs is relevant when local acknowledgment is used because each SNA PU has an SDLC or LLC2 connection that must be kept alive by sending messages at regular intervals. These keepalive messages and the timer processing required to determine when to send them is processor intensive. Adjusting LLC2 timers on the routers can help, but in general, on a Cisco 4700 series router assume a maximum of approximately 4000 PUs. Figure 6-1 and Figure 6-2 illustrate the CPU utilization of various routers for varying numbers of PUs and LUs and can be used to approximate the size of the router required. For a more exact calculation, provide the appropriate information to your systems engineer.

Figure 6-1 CPU Usage of Various Routers Assuming TCP Encapsulation and Assuming Each PU Has 10 LUs, Each with 1 Transaction Per Minute

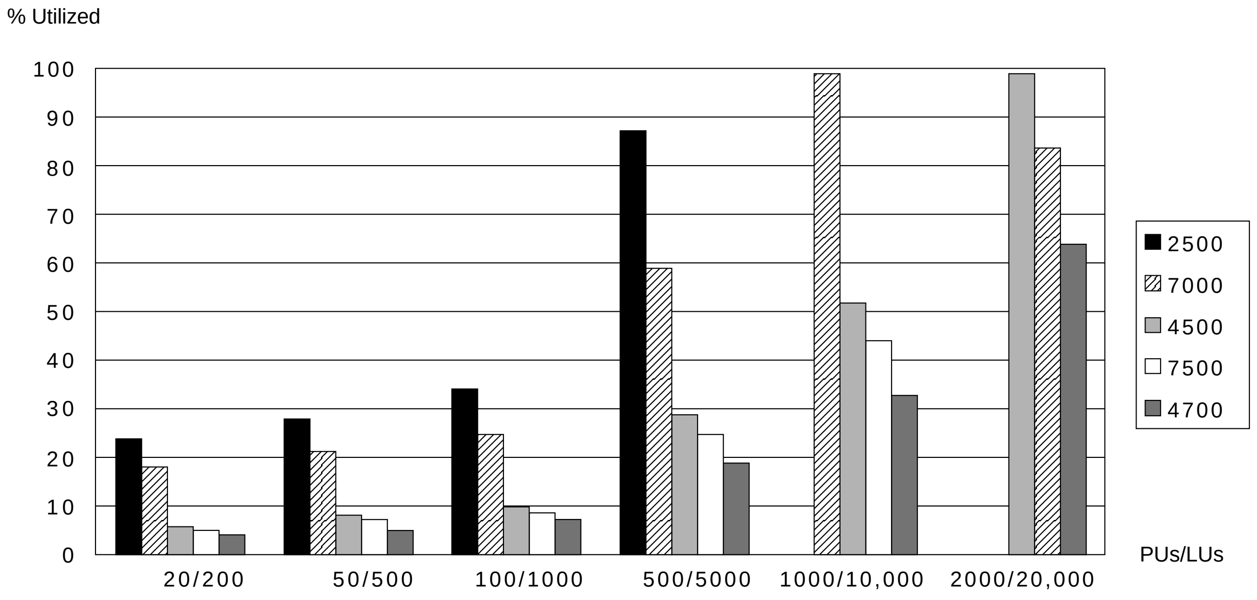
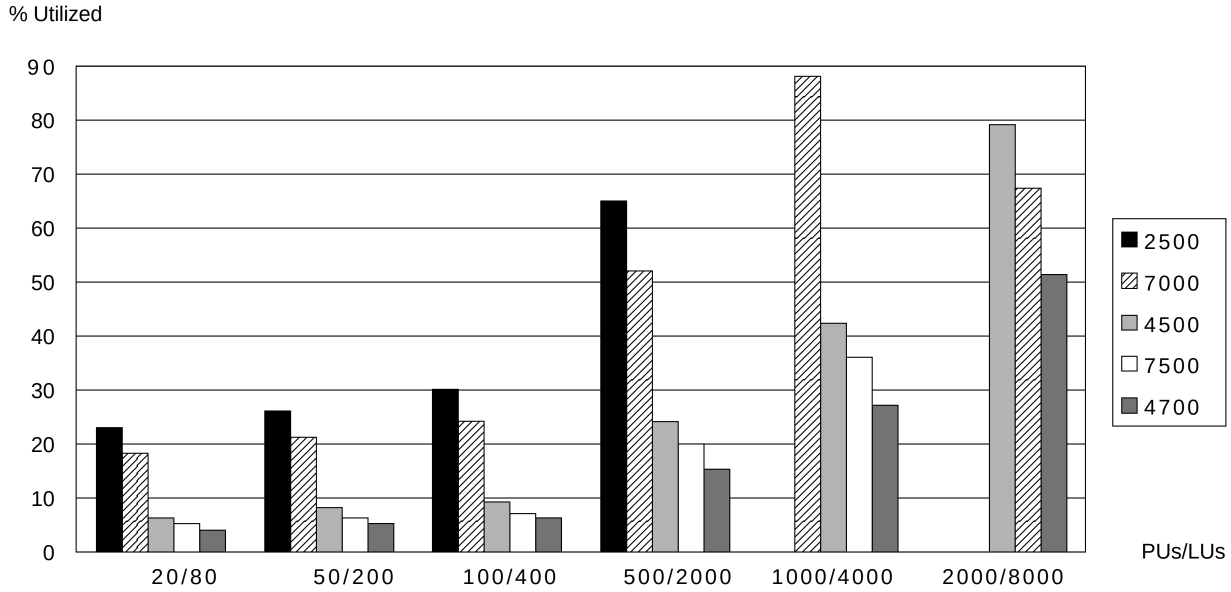


Figure 6-2 CPU Usage of Various Routers Assuming TCP Encapsulation, Transactions of 40 Bytes In and 1000 Bytes Out, and Assuming Each PU Has 4 LUs, Each with 1 Transaction Per Minute



Transaction Rate

The transaction rate also plays a role in determining how many central site routers can be supported. A typical transaction rate is one transaction per LU per minute. By determining the number of LUs per PU on average and the total number of PUs and assuming this transaction rate, you can fairly accurately anticipate the transaction rate of most environments. The transaction size has two components: message size in and message size out (40 bytes in and 1000 bytes out is common). Figure 6-1 and Figure 6-2 illustrate the router utilization with a specific transaction rate and size. Note that the number of PUs has more of an impact than the transaction rate. Varying the LUs is relevant because it changes the transaction rate. If the transaction rate was kept constant as new LUs were added (in other words, fewer transactions per LU as LUs were added), the number of LUs would have no bearing.

Encapsulation Method

The encapsulation method is relevant because different encapsulation methods have different impacts on route processor utilization. Both TCP and LLC2 encapsulation involve local termination of the data-link controls (local acknowledgment) and are process switched. FST and direct encapsulation run in passthrough mode, which means acknowledgments flow end to end. Assuming adequate bandwidth and line quality, these encapsulation types will allow a central site router to support more remote branch routers, because these encapsulations do not support local acknowledgment and require fewer processor cycles. Figure 6-1 and Figure 6-2 assume TCP encapsulation.

Processor Speed

DLSw+ is processor intensive and runs best in a router with a faster route/switch processor (for example, a Cisco 4700, Cisco 7200, or Cisco 7500) rather than a slower processor (for example, a Cisco 4000 or Cisco 7000). Figure 6-1 and Figure 6-2 show the CPU utilization required to support various numbers of PUs and traffic volumes. In Figure 6-1, the transaction size was 40 bytes in and 1000 bytes out. Each PU had 10 LUs, and each LU transmitted at a rate of one transaction per minute. Using these numbers, 500 PUs and 5000 LUs result in 5000/60, or 83 transactions per second at the central site router. The LLC2 idle timer on the Token Ring interface was set to 30 seconds for these tests.

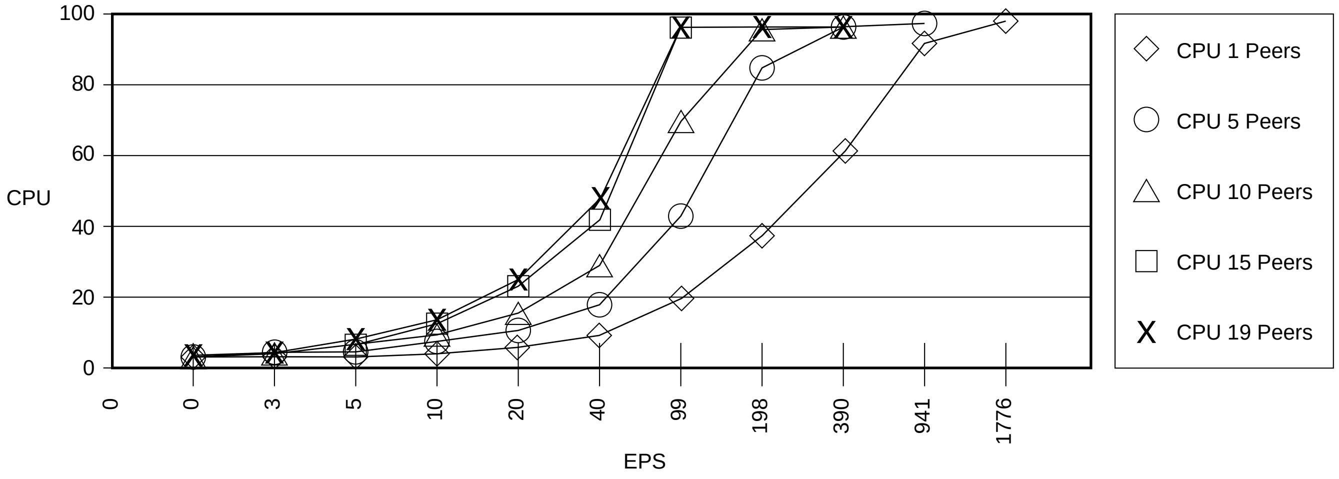
Number of Remote Sites

The number of remote sites that must be connected may have an impact on the number of central site routers required. This number is important when broadcasts must be replicated (see the section “Explorer Replication”). Performance testing shows that the number of peers has no significant impact on router CPU usage if there is no broadcast traffic.

Explorer Replication

Another factor that plays a role in determining the number of central site routers is the amount of explorer replication required. If all the connection requests are remotely initiated and the network is hierarchical, the amount of explorer replication required should be minimal. This assumes appropriate filters are set at the central site to prevent unnecessary explorer propagation (see the chapter “Customization”). When connection requests are initiated at a central site, these requests must be propagated to each remote peer. The number of explorers that can be replicated per second depends on the speed of the route processor. Figure 6-3 illustrates the explorer processing rate of a Cisco 4700. As the number of peers increases, the number of explorers that can be received and replicated per second decreases. For example, if a Cisco 4700 peers to 20 remote peers, it can replicate almost 100 explorers per second to each of the 20 peers. If the Cisco 4700 peers to one router, it can replicate more than 1700 explorers.

Figure 6-3 CPU Utilization of a Central Site Router as the Number of Explorers Per Second Varies and as the Number of Peering Routers Increases (No Caching Is Assumed, and Each Explorer Received Must Be Replicated to Each Remote Peer)



Other Router Processes

Your router may be configured to do more than DLSw+. The NetSys Performance Solver tool will help you size routers that are performing multiple functions, or you can you can approximate the number of routers required based on the amount of CPU you are using for routing functions and what additional load your SNA traffic will place on those routers.

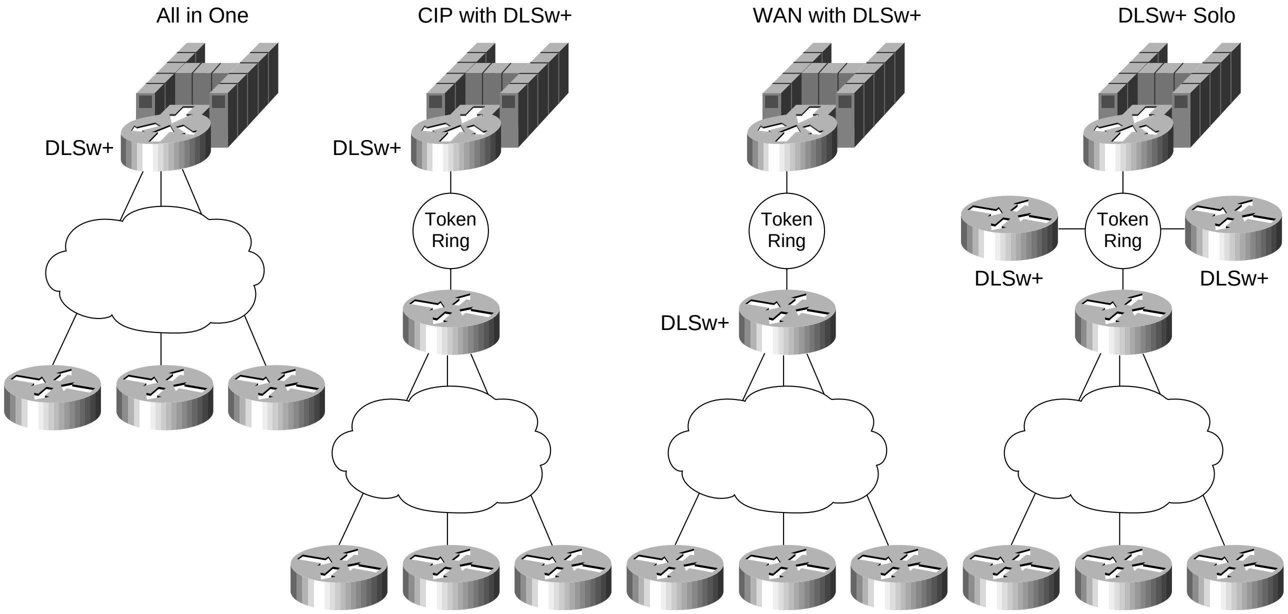
Placement of Peering Routers

After you have determined how many routers you need to support your traffic, you can consider the best place to put the peering routers. There are four alternatives:

1. Peer all remote sites to one or more central site routers that are directly connected to a mainframe over a CIP and directly connected to the WAN over serial ports
2. Peer all remote sites to one or more central site routers that are directly connected to a mainframe over a CIP, but keep WAN function in separate routers
3. Peer all remote sites to direct WAN-attached routers that access the mainframe via a channel gateway such as a Cisco 3745 or another Cisco router with a CIP
4. Peer all remote sites to dedicated DLSw+ routers that are neither WAN connected nor CIP connected

Each of these alternatives is valid and is the best alternative in specific environments. Figure 6-4 illustrates these alternatives.

Figure 6-4 Four Central Site Peering Router Replacement Alternatives



All in One (DLSw+, CIP, and WAN)

Peering to a CIP router that is also a WAN router has the advantage that it requires the smallest number of central site routers. In small networks (30 to 60 branches) that are primarily SNA, this is a reasonable choice.

CIP and DLSw+ Combined

Peering to a CIP router but having a separate WAN router is a good solution for small to medium-sized networks (up to 200 remote branches) with a moderate amount of multiprotocol traffic. This design allows you to segregate multiprotocol broadcast replication from DLSw+ processing. For backup and availability, this solution will typically involve two central site peering routers; one router should be able to handle the load in the event of a failure of the other router.

WAN and DLSw+ Combined

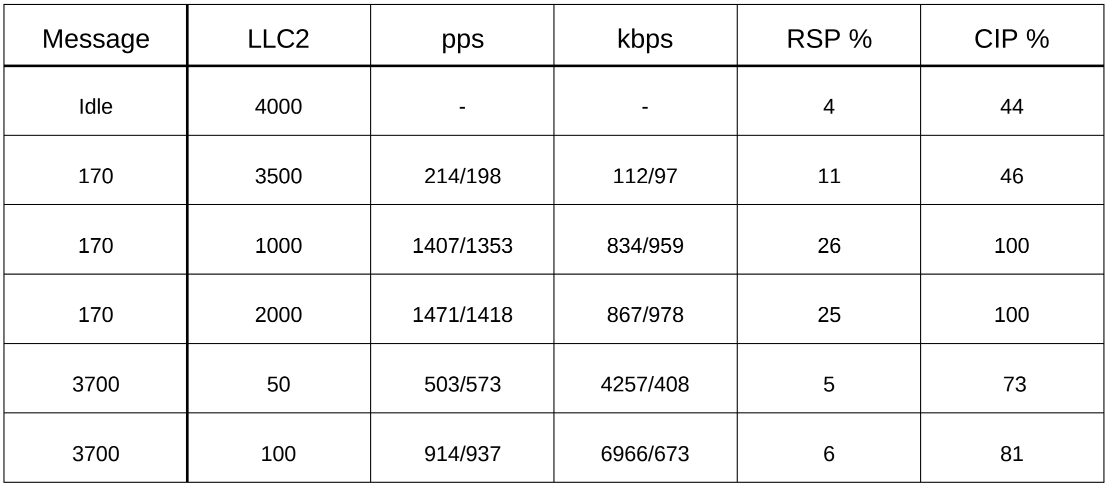
The third solution, peering to the WAN router, is a good solution for medium-sized to large networks that require more than one or two central site routers for DLSw+ processing or that use a channel gateway other than a CIP. To access the channel gateway, you can use SRB over Token Ring (this is adequate for most host traffic) or SRB over FDDI (DLSw+ will support SRB over FDDI in Cisco IOS Release 11.2).

Using WAN and DLSw+ combined, you can segregate the DLSw+ processing from the CIP-attached router and scale the network without buying additional CIP routers. As the network grows beyond the capacity of a single router, you can add Cisco 4700s or Cisco 7200s to handle the capacity. This is more cost effective than adding large Cisco 7500s with CIPs. Because SRB is fast switched, a single Cisco 7500 or Cisco 7000 with a CIP can handle the traffic from four or five Cisco 4500s or three Cisco 4700s using DLSw+. Figure 6-5 illustrates the transaction processing power of a Cisco 7500 and a CIP when using SRB to send traffic from a LAN to the CIP. The message size is noted in the first column, and the number of SNA PUs is indicated in the LLC2 column. The packets per second in and out and thousand bits per second in and out is shown in the next two columns. The Cisco 7500 Route/Switch Processor (RSP) utilization is always relatively low, because the traffic is fast switched off the LAN and to the CIP. The CIP is designed to run at 100 percent for some traffic volume.

To put these traffic volumes in perspective, a transaction rate of approximately 1300 per second would represent the load of 78,000 LUs sending data at a rate of one transaction per LU per minute. The RSP in this example was 26 percent utilized and the CIP was operating at 100 percent. These tests were run using a CIP1. All new CIPs are CIP2 and they support a much higher transaction rate.

This configuration also offers advantages in terms of change management and network availability. By limiting the channel-attached routers to SRB and IP routing, you minimize the requirement for configuration changes or Cisco IOS software upgrades in your channel-attached router. This configuration decreases planned downtime and increases network reliability.

Figure 6-5 CPU Utilization of a Cisco 7500 with a CIP Handling SRB SNA Traffic



Message	LLC2	pps	kbps	RSP %	CIP %
Idle	4000	-	-	4	44
170	3500	214/198	112/97	11	46
170	1000	1407/1353	834/959	26	100
170	2000	1471/1418	867/978	25	100
3700	50	503/573	4257/408	5	73
3700	100	914/937	6966/673	6	81

Dedicated DLSw+

The final alternative separates DLSw+ processing from CIP processing and WAN processing. This is a good solution for large networks with a significant amount of multiprotocol traffic. Although this appears to have the most routers, it may in fact have the same number of routers with the function split across different boxes. The key advantage to this solution is load balancing and backup. If the WAN is a Frame Relay network, a single permanent virtual circuit (PVC) to a central site WAN router will provide connectivity to multiple central site peering routers. This configuration has the same change management and availability advantages as the previous one.

Note FST or TCP encapsulation is required whenever the peering routers are not adjacent, as shown in the CIP with DLSw+ or DLSw+ Solo solutions in Figure 6-4. DLSw Lite and direct encapsulation options assume that the peering routers are adjacent (that is, that DLSw+ is running in the WAN router).

Availability Options

There are several alternatives for building a fault-tolerant network. With DLSw+, recovery from some failures is nondisruptive to the end systems. Recovery from any failure can be dynamic. The following describes recovery scenarios with various features.

Link Recovery

Link failures on the WAN can be recovered by using TCP encapsulation and providing alternate paths (either leased or switched). Local acknowledgment ensures that the router has time to reroute around the link failure without disrupting SNA sessions. Some NetBIOS applications have session-level timers in addition to link-level timers. DLSw+ does not spoof session-level timers, so NetBIOS sessions may drop if there is an outage in the network.

When using FST encapsulation, link failures may or may not be disruptive. Because FST does not offer local acknowledgment, timers may expire before DLSw+ has time to reroute. However, rerouting is dynamic.

When using direct encapsulation, link failures are disruptive but recovery can be automatic. Backup from link failures can be addressed either by having multiple remote peers or by configuring multiple remote peer statements for the same remote peer but specifying a unique path to each one. You can either load balance between them or use cost to cause one path or peer to be preferred over the other, as shown in the following statements:

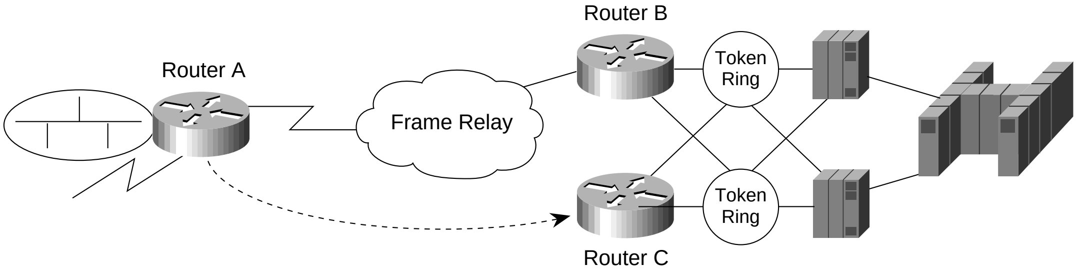
```
dlsw remote-peer 0 frame-relay interface serial 0 22 cost 2
dlsw remote-peer 0 33.33.33.33 cost 4
```

In this example, the first statement describes how to get to a remote peer directly over a Frame Relay link, and the second statement describes how to get to the same remote peer via a TCP path.

Recovery using two peers is illustrated in Figure 6-6.

Note A single central site router can appear as two DLSw+ peers. Simply use two remote peer statements and specify alternate DLCIs or encapsulation methods for accessing each remote peer.

Figure 6-6 DLSw Lite Configuration Providing Dynamic Recovery from the Loss of a Link or Central Site Router



```
Configuration for Router A
dlsw local-peer peer-id 10.2.17.1
dlsw remote-peer 0 interface serial 1 33 cost 1
dlsw remote-peer 0 tcp 10.2.18.2 cost 4
interface serial 1
encapsulation frame-relay
frame-relay map llc2 33
```

```
Configuration for Router B
dlsw local-peer peer-id 10.2.24.3
   promiscuous
duplicate-path-bias load-balance
interface serial 1
encapsulation frame-relay
frame-relay map llc2 17
```

```
Configuration for Router C
dlsw local-peer peer-id 10.2.18.2
   promiscuous
duplicate-path-bias load-balance
```

Central Site Router Recovery

Loss of a central site router is always disruptive, but recovery can be dynamic and immediate. There are two alternatives for recovery: multiple concurrently active central site routers or backup peers.

If remote peers concurrently connect to multiple central site peers, loss of a single central site peer will cause all new sessions to be established over the remaining active central site peers. If remote peers only connect to a single central site peer, you can still specify a backup peer that will be used in the event that the primary peer goes away. Backup peers are supported only for TCP and FST encapsulation.

See the “Advanced Features” chapter for a description of these features and a comparison of the alternatives.

Central Site Mainframe Channel Gateway Recovery

Loss of a central site mainframe channel gateway is always disruptive, but recovery can be dynamic and immediate. The simplest way to recover from this is to have multiple mainframe channel gateways with the same MAC address, each accessible via a different port on a central site router. DLSw+ supports load balancing for up to four ports. This not only addresses availability, it can also spread the traffic across multiple TICs on a FEP to avoid congestion problems. This configuration is commonly known as the duplicate TIC configuration. The same concept can be used in conjunction with a Cisco CIP. DLSw+ allows remote SDLC or Ethernet-attached devices to take advantage of this feature by providing media conversion.

Broadcast Reduction

Because broadcasts impact the processing power of a DLSw+ router, it is important to understand how to eliminate any unnecessary broadcast replication. Some techniques to eliminate replication are filtering, virtual ring numbering, and static device configuration.

Filtering

Filtering unnecessary broadcasts is the best way to minimize explorer replication. DLSw+ will attempt to switch nonrouted multiprotocol traffic if not filtered. The “Customization” chapter describes how to configure filtering to allow only SNA or NetBIOS into DLSw+. Once an access list has been created, it can be applied to the input interface (which would prevent even local forwarding) or to the **dlsw remote-peer** command.

Virtual Ring Numbering

When there are multiple central site DLSw+ routers attached to the same Token Ring segment or SRB LAN, it is possible for broadcast frames to come in from the WAN, be sent out over the physical Token Ring LAN, and be picked up by another DLSw+ router. To prevent that router from retransmitting the frame on the WAN, code the same virtual ring number in all DLSw+ routers attached to the same physical ring or bridged LAN. Normal SRB procedures will prevent broadcasts from being copied on a ring that is already present in the RIF.

Static Device Configuration

Devices can be statically configured in DLSw+. By configuring frequently accessed resources, you can eliminate the need for broadcasts to find those resources.

DLSw+ allows you to statically configure resources (MAC addresses or NetBIOS names) that are local to a DLSw+ peer using a **dlsw icanreach** command. This information is dynamically distributed to all remote peers as part of the capabilities exchange. This feature is extremely useful as a means to advertise reachability of the mainframe channel gateway (FEP or CIP) or key NetBIOS servers. With a few configuration statements at central site routers, you can preload the cache of all the remote peers. When a peer learns of the reachability of an end system via a capabilities exchange, it keeps that information in its cache as long as the peer connection is active. The peer never broadcasts explorers for these resources. If a branch router peers to multiple central site routers, it can learn of multiple ways to access a resource and will cache up to four of them.

Central site routers can also specify the **exclusive** keyword. For example, the command **dlsw icanreach mac-exclusive 4000.3745.0001** tells remote routers that the *only* destination this router can reach is the MAC address of the 3745. This feature can also be used to indicate that certain NetBIOS servers are located at the central site, but not elsewhere. The **exclusive** keyword prevents remote sites from forwarding unnecessary broadcasts.

DLSw+ allows a peer router to advertise when it cannot reach a resource or SAP (this is specified in the **dlsw icannotreach saps** command). One use of this feature is to prevent branch offices from searching the data center for NetBIOS servers. If a DLSw+ peer learns via a capabilities exchange that it cannot reach a resource via a particular peer, it will not send explorers to that peer for that resource.

Note If you are using border peers there are some limitations. Because border peers offer no advantages for hierarchical networks, this chapter assumes they are not being used. See the "Designing Meshed Networks" chapter for a discussion on the implications of border peers and using **dlsw icanreach** configuration commands.

DLSw+ also allows you to statically configure a path to reach a local or remote resource. This is done using a **dlsw mac-addr** or **dlsw netbios-name** command, and it works well if there is only one way to reach a resource and its location will never change. This entry is never deleted from the cache. The "Customization" chapter describes the difference between using static paths and **dlsw icanreach** commands.

Designing Meshed Networks

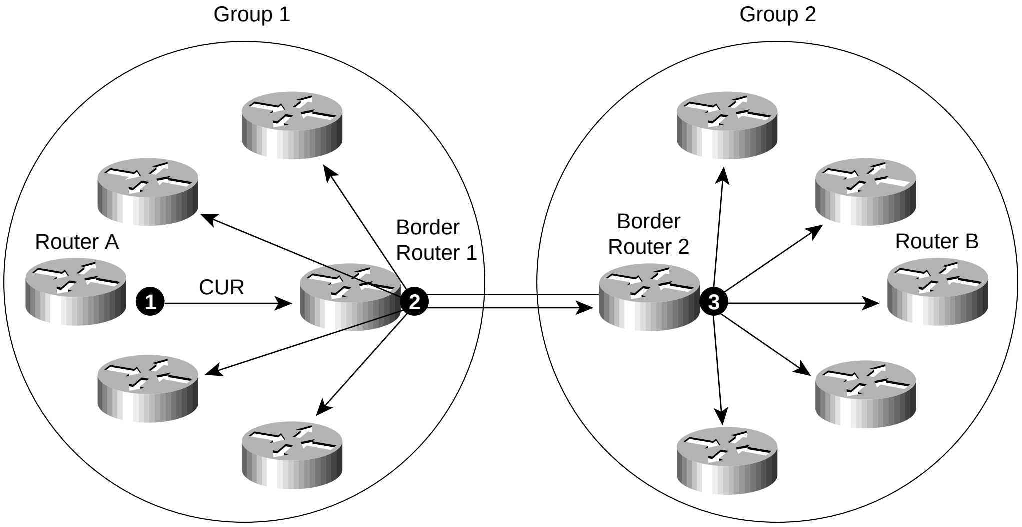
This chapter describes design considerations when using DLSw+ to build a meshed, any-to-any network. It describes the role of border peers and provides guidance on where to place border peers, how to backup border peers, and how to size border peers. It also describes how to size peer groups, how to configure on-demand peers, and how to minimize unnecessary broadcast replication.

The Peer Group Concept

DLSw+ is the only implementation of DLSw that offers features designed specifically to address the issues in building fully meshed networks. These features are border peers, peer groups, and on-demand peers, and they are described in the "Introduction" and the chapter "Advanced Features." Using peer groups you can build enterprise networks that support branch-to-branch communication between either NetBIOS or Advanced Program-to-Program Communications (APPC) applications. Most enterprises do not require fully meshed connectivity, but when it is required, it can be challenging to support. The key reason is that unless resources are statically configured everywhere, extensive broadcast traffic will result. This broadcast traffic can clog access links, and broadcast replication can overburden branch routers.

DLSw+ solves this problem by providing a hierarchical means to dynamically search for branch resources. Instead of a single branch router having to query every other branch router, the branch router sends a single broadcast to its border peer. The border peer propagates the broadcast within its group and to other border peers. Other border peers propagate the broadcast within their group. This method not only minimizes the broadcast replication on each line, it also minimizes the replication work done by any single router. End-to-end TCP connections (called peer-on-demand connections) are set up only when resources are found. Figure 7-1 illustrates explorer processing when border peers are used.

Figure 7-1 Explorer Frames Are Processed by Border Peers



In Figure 7-1, Router A sends a single canureach frame to Border Router 1. Border Router 1 forwards this broadcast to the remaining four routers in Group 1 and to Border Router 2. Border Router 2 forwards the broadcast to all the routers in Group 2. In this way all 10 branch routers receive the canureach frame, but the originating branch router sent only a single copy, and the border routers replicated it only five times.

Explorer Processing When Using Peer Groups

When a **dlsw local-peer** command specifies a group name, that local peer will forward explorer packets only to border peers in its group, configured remote peers in another group, configured remote peers that are not part of any group (for example, non-Cisco routers), or peers in its local cache that match the explorer conditions. You can specifically configure remote peers within the same group, but the local peer still sends explorers only to its border peer. The only exception is that if there are no active border peers within its group, a local peer will forward broadcasts to all configured peers within its group. For example, in Figure 7-1 if Router A is peered to all the routers in Group 1 and to Router B, when Router A gets an explorer frame, it will forward it only to Border Router 1 and to Router B. If Border Router 1 is not active, it will forward the explorer to all the peers in Group 1 in addition to Router B.

There are two reasons you may wish to configure all remote peers within a group. Again, using Figure 7-1 as an example, if all the routers in Group 1 frequently communicate with each other, by configuring **dlsw remote-peer** commands between every pair of routers within the group, the peer connections will always be active. This shortens the connection time for end-user sessions without increasing the broadcast traffic. Connection to less frequently accessed peers in other groups can still be made using on-demand peers. In addition, if there is only a single border peer, it eliminates the single point of failure condition.

DLSw+ does not support cascaded groups. That is, if a border peer from one group forwards an explorer to a border peer in another group, that border peer will not forward the explorer to a third group.

Border peers will forward explorers to local rings in addition to other member peers.

Border Peers

Currently, the sole function of border peers is broadcast replication on behalf of branch routers (or member peers). By concentrating this function in a more powerful distribution or in central site routers, and by distributing the replication function across a number of border peers, you can build networks with fully meshed connectivity without having to put large, powerful routers at every branch. The border peer should be a Cisco 4700, Cisco 7200, or Cisco 7500 series router. The placement of the router is determined by your physical network design. It can reside either at a distribution site or at a central site, depending on where you send your branch traffic enroute to other branch sites.

DLSw+ supports multiple active border peers. If a single group has multiple active border peers, the rules of duplicate path bias apply. Every peer in a group will either load balance across multiple active border peers or will select the border peer with the least cost.

If you are using multiple active border peers, the following rules apply:

- Within a single group, every member peer must peer to every border peer in its group
- All border peers within a group must peer to each other
- All border peers within a group must peer to every border peer in other groups
- Border peers forward explorers to all member peers in their group, all border peers in their group, and to one border peer in every other group

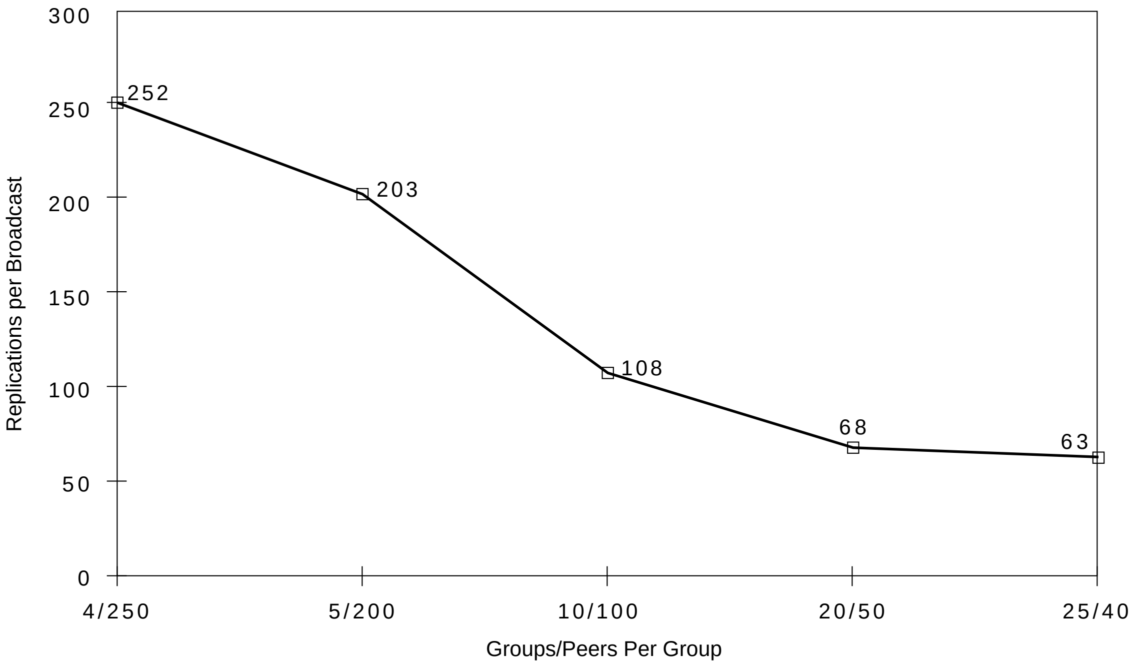
On-Demand Peers

With border peers in place, it is possible for two peers to communicate with each other even though neither has a configuration for the other. This is because they learn about each other via their respective border peers. For that reason, there is a statement that defines how to connect with peers when no **dlsw remote-peer** commands are used. That statement is the **dlsw peer-on-demand-defaults tcp** command, and it can be used to specify the encapsulation, filters, and timers. On-demand peer connections can be made between two peers in the same group or two peers in different groups.

Size of Peer Groups

How you divide your network into groups will determine how much broadcast replication any single router must do. For example, with a 50-branch network, it is possible to use a single peer group as long as the broadcast traffic is not excessive. With a 1000-branch network, a single peer group is not practical (a single broadcast would have to be replicated 999 times). Suppose you designed a network with 4 peer groups, each peer group consisting of 250 routers. With this design, the border peer must replicate every broadcast once for every peer in its group, and once for every other border peer. As shown in Figure 7-2, with 4 groups of 250 routers, there are 3 + 249, or 252, replications per broadcast. With 20 groups of 50 routers, there are 19 + 49, or 68, replications per broadcast. You should design your groups in a manner that ensures your border peers can handle the amount of broadcast replication any single router must perform.

Figure 7-2 Number of Replications Per Broadcast That a Border Peer Must Perform Based on Different Means of Splitting 1000 Branch Routers into Peer Groups



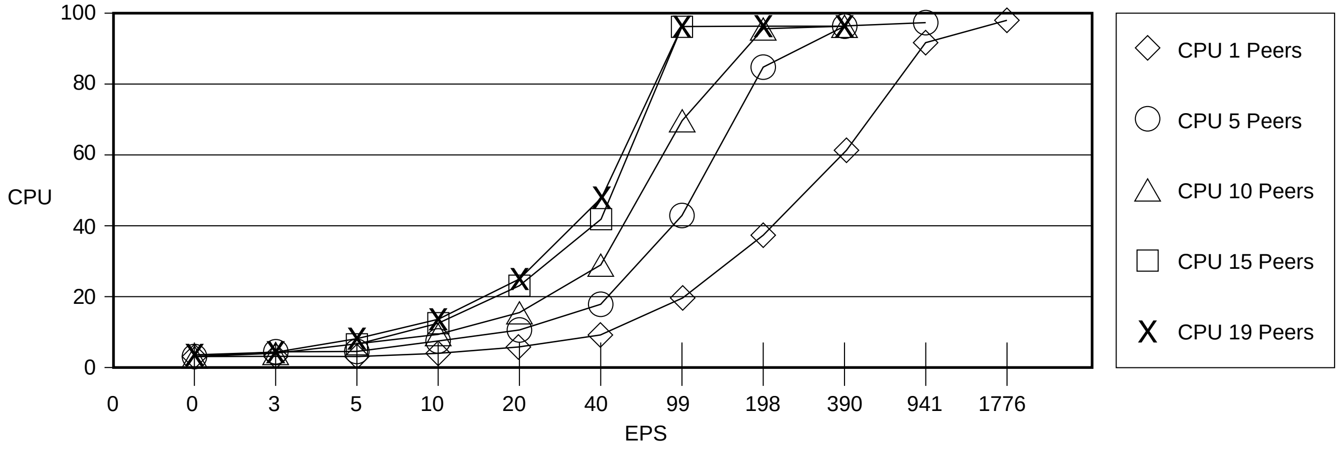
Broadcast Replication/Reduction

Figure 7-3 shows the router utilization based on the number of explorers per second. The practical limit for a Cisco 4700 is 800 to 1000 explorers per second, although it can replicate around 1800 explorers per second if it does nothing else. Assuming a forwarding limit of 800 to 1000 explorers per second, if the Cisco 4700 has 20 peers, it can handle an incoming rate of 40 to 50 explorers per second.

Assume a border peer has 50 peer connections. The worst case scenario for broadcast replication is when a key resource is lost. Every end system will try to find that resource, resulting in at least 50 simultaneous broadcasts (each DLSw+ peer may get multiple explorers but will forward only the first request and queue any duplicates). This would require 50 x 49, or about 2500, explorers. The way to avoid this situation is to preconfigure any resources that an entire enterprise needs to access frequently. Border peers should only be used to find resources that are accessed on an as-needed basis, and not accessed by all branches, all the time.

When you have a hierarchical SNA network and a meshed NetBIOS network, you need to consider the impact of border peers on explorer traffic. You may have occasional branch-to-branch traffic, but the resources that every branch accesses every day are at the central site (the FEP or enterprise servers). Every time a branch router needs to search for the FEP (because the MAC address of the FEP is not in its cache), the branch router sends an explorer to its border peer. The border peer forwards the explorer to every router in its group and every other border peer. Even if a border peer has found the FEP on behalf of another resource, it will forward the explorer everywhere. That is because border peers do not check their cache before forwarding explorers. (In Cisco IOS Release 11.3, border peers will support group caching to address this problem.)

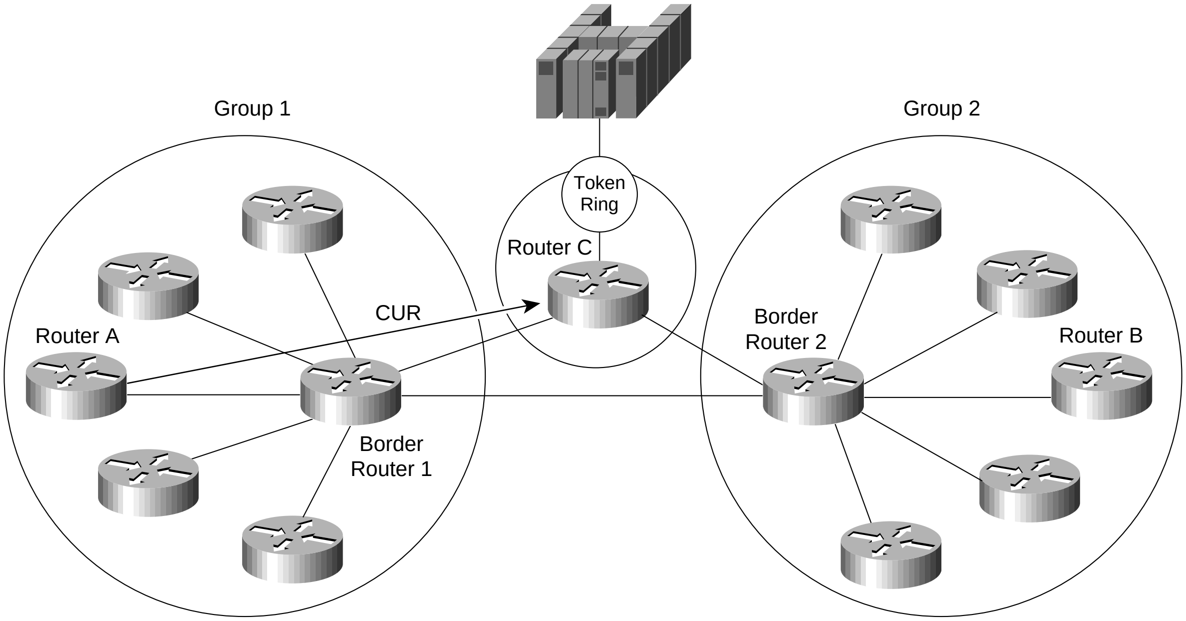
Figure 7-3 CPU Utilization Comparison of a Central Site Router as the Number of Explorers Per Second Varies and as the Number of Peering Routers Increases (No Caching Is Assumed, and Each Explorer Received Must Be Replicated to Each Remote Peer)



As illustrated in Figure 7-4, to avoid unnecessary broadcast forwarding for central site resources, you can configure all the remote branch routers to peer to both their border peer and a data center router (Router C). At Router C, you can configure the reachability of the FEP MAC address in a **dlsw icanreach** command. As soon as the branch router establishes a peer connection with Router C, it will learn that Router C can access the MAC address of the FEP. When Router A receives an explorer for the FEP MAC address, it will first check its cache, where it will find a match (remember, cache entries learned as part of a capabilities exchange are not deleted unless the associated peer connection goes away). Instead of forwarding the explorer to its border peer, it will forward the circuit setup request directly to Router C, avoiding any broadcasts to other branch routers.

Note In this replication scenario, if the border peer and the data center router are the same, specify **dlsw icanreach** in the border peer. In this way, the branch router will preload its cache with the MAC address of the FEP and will always send a directed explorer to the border peer instead of requesting that the border peer do a search on its behalf.

Figure 7-4 Explorers Are Minimized by Peering Branch Routers to Both Their Border Peer and a Central Site Router



RSRB Migration and Multivendor Interoperability

This chapter describes the differences between DLSw+ and RSRB as well as the following issues:

- Why you might want to migrate from RSRB to DLSw+
- What the migration implications are in terms of management, memory, and performance
- How to migrate from RSRB to DLSw+

In addition, this chapter describes interoperability with other RFC 1795 implementations, including valid configuration options.

RSRB and DLSw+ Comparison

RSRB was the predecessor to DLSw+. Created before there were routing standards, RSRB was Cisco's original implementation for transporting LLC2 traffic over an IP network. RSRB has been in existence since 1991 and has been used to build some of the largest integrated SNA and client/server networks in the world. There are thousands of RSRB networks in existence, many with over 1000 RSRB routers. RSRB networks will exist well into the future.

In 1995, however, the first standard for SNA over IP was approved. This standard was created in the AIW and was later documented in RFC 1795. DLSw+ complies with RFC 1795 and provides enhancements that allow DLSw+ networks to scale better and provide better availability than either RSRB or RFC 1795-compliant implementations.

DLSw+ includes functions that were previously provided in several other Cisco features, including RSRB, SDLLC, SR/TLB, Proxy Explorer, and NetBIOS name caching. Most environments using DLSw+ no longer need to configure any of these features.

Why Move to DLSw+?

DLSw+ is Cisco's strategic solution for SNA transport going forward. RSRB will no longer be enhanced. If you need features beyond what RSRB offers, you will likely need to migrate to DLSw+.

DLSw+ addresses several RSRB limitations by including key functions such as local acknowledgment for devices on Ethernet and SDLLC for PU 2.1 devices. In addition, DLSw+ scales better than RSRB, is easier to configure and manage, and provides higher availability with load balancing and backup features. DLSw+ also offers multivendor interoperability. Table 8-1 illustrates the differences between RSRB and DLSw+.

Table 8-1 Comparison of Cisco's RSRB to DLSw+

Benefits	RSRB Features	DLSw+ Features
Performance	IP load sharing Custom and priority queuing	IP load sharing Custom and priority queuing Circuit-level flow control[1] Peer and port load sharing[1]
Availability	Nondisruptive reroute Prevents data-link control timeouts	Nondisruptive reroute Prevents data-link control timeouts Local acknowledgment on Ethernet[1] Backup peers[1] Fault tolerant peers[1]
Scalability	Limited broadcast reduction SRB hop reduction	Broadcast reduction RIF termination[1] Broadcast optimization with peer groups[1]
Flexibility	Media conversion via SDLLC and SR/TLB (2.0) SRB dynamics Transport options (FST, direct) IP and IPX bridging[2] PU 4-to-PU 4 over multipath SRB[2] AST[2] FST between unlike media (SDLLC and SR/TLB[2] RIF passthrough[2] LNM over FST[2]	Media conversion built in (2.0/2.1) SRB dynamics Transport options (FST, direct)[3] DLSw Lite (LLC2 encapsulation)[1] Capabilities exchange[1] Peer biasing with cost[1] SNA DDR[1] Promiscuous peers[1] Multivendor interoperability[1]

[1] Supported by DLSw+ but not by RSRB
[2] Supported by RSRB but not by DLSw+
[3] Supports Token Ring-to-Token Ring only

DLSw+ was designed in a modular fashion to maximize stability and to facilitate new feature additions. The circuit concept in DLSw+ simplifies management. Because DLSw+ is a standard, Cisco's implementation can interoperate with other standard-compliant implementations, protecting your investment in the technology and simplifying network integration of acquired companies. Finally, we anticipate that DLSw+ soon will surpass RSRB as the most commonly employed technique for SNA and client/server integration.

Possible Migration Inhibitors

Some environments will not be able to move from RSRB to DLSw+ at this time. They may require features that either are not present in DLSw+ yet or were added to DLSw+ in a recent release of Cisco IOS software. Table 8-2 shows the DLSw+ features and the Cisco IOS release that is required for that feature. See Appendix B for a more detailed list of DLSw+ features and release availability.

Table 8-2 Cisco IOS Release Support of DLSw+ Features

Feature	Cisco IOS Release Level Required
APPN over DLSw+	11.2
DSPU, Service Point, LAN Network Manager Support	11.1(5)
FST over ATM	11.1(5)
CiscoWorks Blue Maps Support (MIB)	11.1(5)
Show Enhancements	11.0(10), 11.1(4)
Debug Enhancements	11.0(9), 11.1(3)
FST over Frame Relay, Token Ring, FDDI	10.3(12), 11.0(9), 11.1(4)
SNA DDR, MAC Filters	11.1
Dynamic Peers	11.1
DLSw Lite, QLLC Conversion	11.0
Load Balancing	10.3
Ethernet Lack and PU 2.1 SDLLC	10.3
Peer Groups Borders	10.3
On-Demand Peers	10.3
Backup Peers	10.3, 11.1[1]
Promiscuous Peers	10.3

[1] Enhanced

There are a few known RSRB features not currently available in DLSw+. These include:

- Support for duplicate Token Ring paths between FEPs (this support requires RSRB without local acknowledgment because of idiosyncrasies in the FEP Token Ring implementation)
- RIF passthrough
- IP and IPX bridging
- FST between unlike media
- LNM over FST

These features will be considered for a future release of Cisco IOS software, based on customer demand. DLSw+ also does not support automatic spanning tree (AST), which is a protocol used by source-route bridges to determine whether they should forward single route explorers.

Migration Considerations

The first two questions people ask when considering migration are:

1. Does DLSw+ perform as well as RSRB?
2. Does DLSw+ require additional memory?

From a performance standpoint, DLSw+ uses the same or slightly fewer CPU cycles to handle an equivalent amount of traffic. However, DLSw+ uses more memory than RSRB. The key reason more memory is required is that DLSw+ maintains state information for every circuit and caches entries for multiple active paths. Maintaining state information simplifies management, and maintaining cache entries allows better network design. Even with the additional memory requirements, most networks will run well with the default memory that comes with the router and software subset. For example, the Cisco 2500 is a branch router, and in a typical branch with 20 to 40 PUs and LUs, the standard memory configuration that comes with any of the IBM images will run DLSw+ quite well. If you are running RSRB with an older level of the Cisco IOS software, you may want to check if your current routers can support DLSw+ with the memory they have. The Cisco IOS software subset image takes up the bulk of the memory, and the image size has grown over time. The memory required to store the image is the most important part of the equation. (The size of any Cisco IOS software feature set varies by release, so that information is not included here.) If necessary, you can calculate the memory required by DLSw+ from the formulas provided in Appendix A.

When putting in new routers, rather than figuring out the smallest amount of required memory, a common approach is to install enough memory to minimize your chances of having to visit remote sites again. Many enterprises installing Cisco 2500s at remote sites with Cisco IOS Release 11.0 will include 8 MB dynamic RAM and dual bank 8 MB Flash memory. For Cisco 4500s or Cisco 4700s with many peers, many enterprises choose to install the maximum amount of memory (32 MB of box memory and 16 MB of I/O memory) because there are fewer Cisco 4x00s in a typical network and the cost of the additional memory is less of an issue.

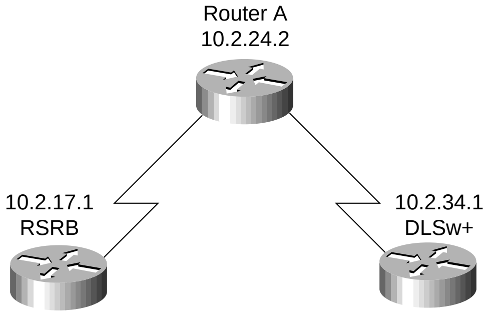
Migrating Steps for a Simple Network

DLSw+ and RSRB can both run in the same router at the same time, but between any two routers you should use one or the other, as shown in Figure 8-1. This allows you to migrate your entire RSRB network to DLSw+ one router at a time. (It is possible to send some traffic—for example, LNM or bridged IP—over RSRB and other traffic over DLSw+.)

To migrate a hierarchical RSRB network to a hierarchical DLSw+ network, do the following:

1. Migrate your routers to Cisco IOS Release 10.3 or later (see Table 8-2 to determine which version of Cisco software you should install to get the features you want).
2. At the data center RSRB routers, configure a **dlsw local-peer** command with **promiscuous** specified. Specify any filters that are required to keep inappropriate traffic (for example, IPX or IP bridged frames) off your DLSw+ network (these filters are described in the “Customization” chapter).
3. Select one remote site and delete the RSRB command (and any related commands such as SDLLC, SR/TLB, Proxy Explorer, and NetBIOS name caching that are no longer required).
4. At that same remote site, add a **dlsw local-peer** command and one or more **dlsw remote-peer** commands. When the configuration at this remote site takes effect, your central site router will use DLSw+ to communicate with this branch and RSRB to communicate with all others. Figure 8-1 shows a sample central site configuration.
5. Repeat Step 4 with the remaining remote sites. When all remote sites are migrated, you can remove the RSRB peer statements from the local peer.

Figure 8-1 Central Site Router Configured to Communicate with Both an RSRB Router and a DLSw+ Router

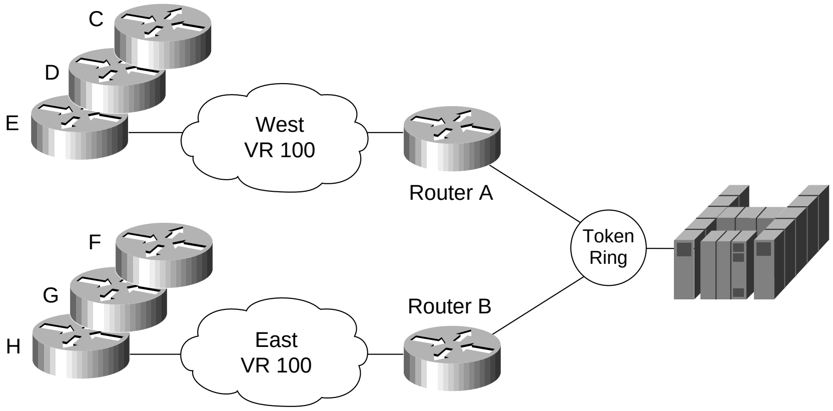


```
Configuration for Router A
source-bridge ring-group 100
dlsw local-peer peer-id 10.2.24.2 promiscuous
source-bridge remote-peer 100 tcp 10.2.24.2
source-bridge remote-peer 100 tcp 10.2.17.1 local-ack

Configuration for RSRB Router
source-bridge ring-group 100    /* must match central VR number
source-bridge remote-peer 100 tcp 10.2.17.1
source-bridge remote-peer 100 tcp 10.2.24.2 local-ack

Configuration for DLSw+ Router
source-bridge ring-group 200    /* no relation to central VR number
dlsw local-peer peer-id 10.2.34.1
dlsw remote-peer 0 tcp 10.2.24.2
```

Migrating to DLSw+ Border Peers from Hierarchical RSRB

Today many RSRB networks are hierarchical to limit broadcasts, but they require any-to-any NetBIOS traffic. Figure 8-2 shows a hierarchical network. Branch routers only peer to the central site router, and all sites use the same virtual ring number to prevent the frames coming in from RSRB Router A from being transmitted out by RSRB Router B.

Figure 8-2 **RSRB Network with a Requirement for Branch-to-Branch Connectivity in Addition to Hierarchical SNA Traffic**



DLSw+ allows the network in Figure 8-2 to support any-to-any communication without an inordinate number of broadcasts. Before you migrate, you need to determine what you want your target network to look like. To support any-to-any communication, you will want to define peer groups, and within each peer group, select one or more border peers. You may decide to make Router A and Router B border peers to address your NetBIOS branch-to-branch traffic. You may also choose to group all the East routers with Router B and all the West routers with Router A. You will need to peer Router A and Router B together (you can use FST over Token Ring in this example) so that the border peers can exchange explorer traffic. (Traffic bridged over the LAN will not be forwarded over the WAN, because both Router A and Router B have the same virtual ring number.) TEST frames destined for the data center will be bridged to the FEP.

Note that in this configuration, not only will NetBIOS explorers be forwarded to all routers in a group, but SNA TEST frames will also be intentionally broadcast everywhere. A simple way to avoid this problem is to configure Router A and Router B with a **dlsw icanreach** command for the FEP (or CIP) MAC. This information will be shared with each remote router as part of the capabilities exchange when the peer connections are initialized. Configuring Router A and Router B with a **dlsw icanreach** command causes remote routers to preload their cache for SNA devices and send only directed broadcasts, not broadcasts that will be retransmitted.

To migrate this network to DLSw+ with border peers, do the following:

1. Migrate to the correct Cisco IOS release.
2. At Router A, configure a **dlsw local-peer** command with **promiscuous** and **border** specified. Also, include a group name. Configure a **dlsw icanreach** command for the FEP MAC.
3. At Router C, delete the RSRB configuration statements and configure a **dlsw local-peer** command with **group** specified and a **dlsw remote-peer** command pointing to Router A. You still have no branch-to-branch communication, but Router C and Router A are now using DLSw+ to communicate.
4. At Router D, delete the RSRB configuration statements and configure a **dlsw local-peer** command with **group** specified and a **dlsw remote-peer** command pointing to Router A. Now Router C and Router D can communicate using Router A as a border peer.
5. Continue the above until all of the first group is migrated. When all remote sites are migrated, you can remove the RSRB peer statements from Router A for routers C, D, and E.

6 At Router B, configure a **dlsw local-peer** command with **promiscuous** and **border** specified. Also, include a group name. Configure a **dlsw icanreach** command for the FEP MAC. Include a **dlsw remote-peer** command for Router A. (Optionally, you can add a **dlsw remote-peer** command for Router B in Router A for symmetry and specify **passive** on that statement. This is a cosmetic change.)

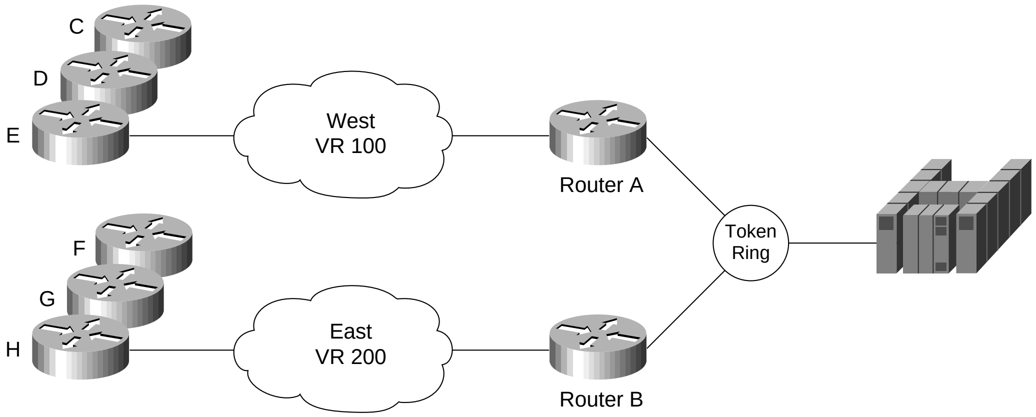
7 Repeat steps 4 through 6 for the routers in the second group, but point the routers to Router B. As soon as the first router in the second group is migrated to DLSw+, branch-to-branch communication between groups will be enabled.

Migrating to Border Peers from Meshed RSRB

Some environments use RSRB for fully meshed networks by careful network design and isolation rings. This section describes how to migrate meshed RSRB to border peers.

In Figure 8-3, all routers in the West peer with all other West routers, and all routers in the East peer with all other East routers. When Router C wants to find a NetBIOS server, it replicates an explorer over each of its three peer connections. When Router A receives the explorer, it puts it on the local Token Ring where it is picked up by Router B. Router B has no way of knowing that the origin is remote, so it replicates the explorer to each of its remote peers. In this way, the broadcast is forwarded, replication in any single router is minimized, and configuration is simplified. This process was the best RSRB had to offer, but it had limitations. All routers within a group had to peer to each other, branch routers had to replicate an explorer once for every router in their group, and the end-to-end session traversed three data links and two TCP connections, complicating management and flow control.

Figure 8-3 **RSRB Network Using Isolation Rings for Branch-to-Branch Connectivity While Concurrently Supporting Hierarchical SNA Traffic**



This network is harder to migrate because of the different virtual route numbers. Both RSRB and DLSw+ use the same virtual ring number, so as long as both are running, explorers will be retransmitted as bridged frames. The steps for migrating this network are:

1 Migrate to the correct Cisco IOS release.

2 At Router A, configure a **dlsw local-peer** command with **promiscuous** and **border** specified. Also, include a group name. Configure a **dlsw icanreach** command for the FEP MAC.

3 At Router C, delete the RSRB configuration statements pointing to Router A (leave in the statements pointing to other branch routers) and configure a **dlsw local-peer** command with **group** specified and a **dlsw remote-peer** command pointing to Router A. Now, Router C and Router A are using DLSw+ to communicate.

4 At Router D, delete the RSRB configuration statements for Router A and Router C, and configure a **dlsw local-peer** command with **group** specified and a **dlsw remote-peer** command pointing to Router A. Now Branch C and Branch D can communicate using Router A as a border peer. Communication to Router E is still with RSRB.

5 At Router E, delete all the RSRB configuration statements and configure a **dlsw local-peer** command with **group** specified and a **dlsw remote-peer** command pointing to Router A. At Router C and Router D, delete the remaining RSRB statements. Now, all routers in the West group can communicate to each other through the border peer (see Figure 8-4). Communication to the East group is still through the Token Ring in the middle.

6 At Router B, configure a **dlsw local-peer** command with **promiscuous** and **border** specified. Also, include a group name. Configure a **dlsw icanreach** command for the FEP MAC.

7 Repeat steps 3 through 5 for the routers in the second group, but point them to Router B.

8 When the final branch in the second group is migrated to DLSw+, add a **dlsw remote-peer** command in Router B for Router A and change the virtual ring number in Router B to match the virtual ring in Router A. When all remote sites are migrated, you can remove all remaining RSRB remote peer statements from Router A and Router B.

9 The final solution is shown in Figure 8-5.

Figure 8-4 RSRB Connections Are Removed and Replaced with DLSw+ Connections Without Loss of Connectivity

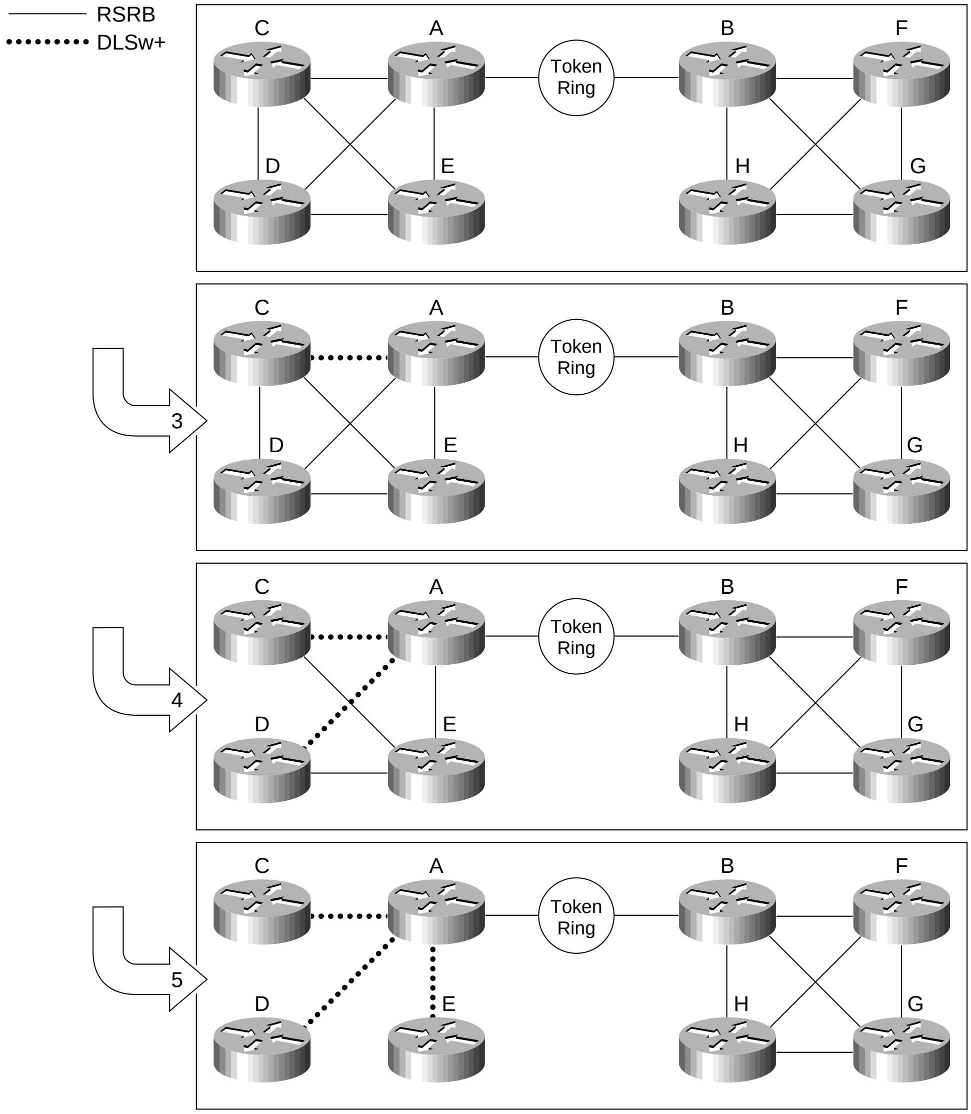
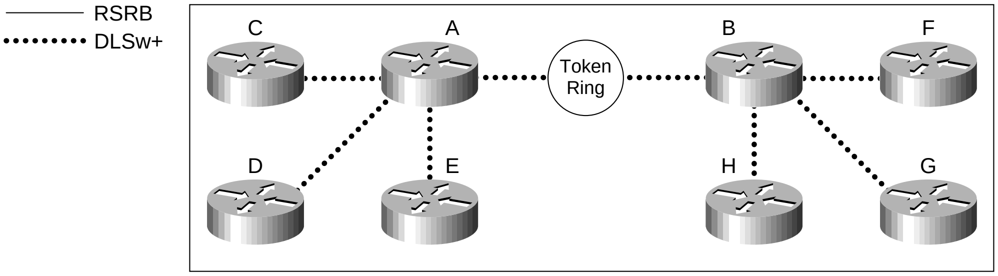


Figure 8-5 Logically Meshed DLSw+ Network



Multivendor Interoperability

You can configure a Cisco DLSw+ router to communicate with a non-Cisco router. However, not all the plus features will be available. Table 8-3 illustrates the features in DLSw+ beyond what the standard offers. Many of the plus features can be used (with some restrictions) even if the peer at the other end is not a Cisco router. These features are noted with an asterisk.

This section details the options you cannot configure and the options that are configurable but somewhat unpredictable. All interoperability testing was done at base-RFC 1795 level only. Cisco has tested interoperability with IBM (both the 6611 and the 2210) and 3Com. Contact Cisco to find out the latest status of this interoperability testing.

Table 8-3 Comparison of DLSw+ and Standard DLSw Features

	DLSw Standard Feature	Additional DLsw+ Features
Performance	IP load sharing	Peer and port load sharing[1]
	Circuit-level flow control	Custom and priority queuing
		Weighted fair queuing
Availability	Nondisruptive rerouting	Backup peers *
	No data-link control timeouts	Fault tolerant peers*
Scalability	Broadcast reduction	Broadcast optimization with peer groups
	Hop count reduction	Ring lists*
Flexibility	Media conversion	Peer biasing with cost *
	SRB dynamics	Diverse transport options
	Capabilities exchange for cache preloading	SNA DDR
		Diverse data-link control media (QLLC, Reverse SDLLC)
		Dynamic peers

[1] Can be used with peers that are non-Cisco routers

The key limitations when building networks with a mix of DLSw standard and DLSw+ routers are:

- You cannot use any encapsulation other than TCP (to non-Cisco routers)

- Non-Cisco routers cannot be border peers or participate in peer groups; in a mixed-vendor environment, you also may not be able to take advantage of load balancing, backup peers, and cost (for these features, it depends on which router is the Cisco router and which one is the non-Cisco router)
- A Cisco router can load balance between two central site non-Cisco routers as long as the Cisco router initiates the canureach exchange; a Cisco router can also locally load balance across all interfaces
- Cisco routers can automatically connect to a backup peer upon loss of a primary peer even if the backup peer is a non-Cisco router; however, the non-Cisco router must either be able to accept a connection from an unknown peer or support something equivalent to the **passive** keyword; the Cisco router will automatically terminate the backup peer connection according to the configuration options
- A Cisco router can establish a dynamic peer connection with a remote non-Cisco peer as long as that remote peer can accept either a connection from an unknown peer or support something equivalent to the **passive** keyword
- Cisco routers can bias remote peer selection based on cost and can support diverse local media

Again, interoperability testing has been limited to RFC 1795 features only, but all of the features should work. Most of these features require that the Cisco router be at the initiating end of the connection.

Local Peer Statements

The following **dlsw local-peer** command keywords are valid because they do not contain information that is sent in a capabilities exchange to a remote router:

dlsw local-peer [**peer-id** *ip-address*] [**lf** *size*] [**keepalive** *seconds*] [**passive**]

[**promiscuous**][**biu-segment**]

The following **dlsw local-peer** command keywords can be configured in a Cisco router, but they should be ignored by non-Cisco standard-compliant routers (they are passed in the capabilities exchange):

dlsw local-peer [**group** *group*] [**border**] [**cost** *cost*]

Remote Peer Statements

For **dlsw remote-peer** commands, you must specify TCP/IP encapsulation. The following keywords are available on this command:

dlsw remote-peer *list-number* **tcp** *ip-address* [**backup-peer** *ip-address*] [**bytes-netbios-out**

bytes-list-name] [**cost** *cost*] [**dest-mac** *mac-address*] [**dmac-output-list** *access-list-number*]

[**host-netbios-out** *host-list-name*] [**lf** *size*] [**linger** *minutes*] [**lsap-output-list** *list*]
[**tcp-queue-max**

size]

These keywords control local filtering, biasing, queue depths, and control when this peer will initiate disconnects with a remote peer.

You should not use the following keywords when you configure the **dlsw remote-peer** commands:

dlsw remote-peer *list-number* **tcp** *ip-address* [**dynamic**] [**inactivity** *minutes*] [**keepalive** *seconds*] [**no-llc** *minutes*] [**priority**] [**timeout** *seconds***]**

The **priority** keyword should not be configured, because it will cause a Cisco router to open four TCP queues, which another vendor's router may not understand or accept. Whether the **dynamic** keyword will work as desired is vendor-dependent. Associated with the dynamic keyword are **inactivity** *minutes* and **no-llc** *minutes***.** SNA DDR relies on **timeout** *seconds* to control when TCP recognizes that it has lost a connection and **keepalive** *seconds* to be set to zero to prevent keepalives from keeping up dial lines. Both keywords will have unpredictable results when used with another vendor's router.

Other DLSw+ Commands

Other DLSw+ configuration commands that can be used include:

- **dlsw ring-list**
- **dlsw mac-addr**
- **dlsw netbios-name**
- **dlsw icanreach**
- **dlsw duplicate-path-bias**

Using Show and Debug Commands

This chapter describes how to use show and debug commands to monitor DLSw+ and to troubleshoot. Certain situations may require external equipment (such as protocol analyzers) to understand what is happening in the network environment. DLSw+ is designed to minimize such situations and to provide tools that offer sufficient information in the majority of cases.

In addition to describing DLSw+ **show** and **debug** commands, this chapter describes **show** and **debug** commands for related feature sets that can be useful in finding problems in DLSw+ environments.

Finally, this chapter includes examples that describe how to use the tools to find and correct problems in the network. The examples provide insight into the correct methodology to find and resolve DLSw+ problems.

DLSw+ Show Commands

DLSw+ provides several **show** commands that allow you to display relevant information about DLSw+ routers, circuits, peers, and reachability:

- **show dlsw capabilities**
- **show dlsw circuits**
- **show dlsw fastcache**
- **show dlsw local-circuit**
- **show dlsw peers**
- **show dlsw reachability**

The following sections explain each of these commands.

Show DLSw Capabilities

To display the capabilities of the local DLSw+ peer or remote peer use the **show dlsw capabilities** command. DLSw+ capabilities are always exchanged as part of the peer initiation process. They can also be exchanged (called a “run-time capabilities exchange”) in an active peer session if something changes. Without any keywords, the **show dlsw capabilities** command shows the capabilities learned from each remote peer. Keywords can be used to specify a particular peer for which capabilities should be shown or to display the capabilities the local peer will advertise to any remote peers.

This command may be useful in determining whether peers support certain features. This may be of particular importance when dealing with DLSw+ features, such as border peering.

The syntax of the **show dlsw capabilities** command follows:

show dlsw capabilities [**interface** *type number* | **ip-address** *ip-address* | **local**]

Syntax Description

interface—Interface used to access a remote peer (direct/LLC2 encapsulation)

ip-address—IP address of a remote peer (FST or TCP encapsulation)

local—Specifies the local DLSw+ peer

Figure 9-1 shows sample output from a **show dlsw capabilities** command issued for a local peer.

Figure 9-1 Output from a show dlsw capabilities Command Issued for a Local Peer

```
milan#sh dls cap local
DLSw: Capabilities for local peer
 vendor id (OUI):         ‘00C’ (cisco)
 version number          : 1
 release number          : 0
 init pacing window      : 20
 unsupported saps        : none
 num of tcp sessions     : 1
 loop prevent support    : no
 icanreach mac-exclusive : no
 icanreach netbios-excl. : no
 reachable mac addresses : none
 reachable netbios names : none
 cisco version number    : 1
 peer group number       : 0
 border peer capable     : no
 peer cost               : 3
 biu-segment configured  : no
 current border peer     : none
 version string          :
Cisco Internetwork Operating System Software IOS™ GS Software (GS7-K-M),
Experimental Version 11.1(10956) [sbales 139]
Copyright (c) 1986-1996 by cisco Systems, Inc.
Compiled Thu 30-May-96 09:12 by sbales8
```

Show DLSw Circuits

To display information about the end-to-end sessions using DLSw+, use the **show dlsw circuits** command. When using TCP encapsulation (or Frame Relay direct encapsulation with local acknowledgment), DLSw+ locally terminates the data-link connection. That is, acknowledgment and keepalive frames are exchanged locally between the end station and the data-link switch while data frames flow directly from one end station to the other. This allows the session to remain active even if the path between the DLSw+ peers suffers a short period of unavailability.

To provide this service, the two DLSw+ peers through which the session is established must keep administrative information about the session (for example, sequence numbers), and the peers must remain synchronized (for instance, if one peer receives a disconnect request from its end station, it must tell the other peer so that it can disconnect from the remote end station). This record keeping and synchronization is accomplished using DLSw+ circuits.

The **show** commands have been changed in Cisco IOS Releases 11.0(10.1) and 11.1(3.3) to allow users to display circuits at a particular end-station address or SAP. This simplifies and speeds up problem isolation and resolution.

In an environment where two devices are in session across a TCP DLSw+ cloud, you should see corresponding circuits on the DLSw+ peers, and the state should be CONNECTED. There is another state called CKT_ESTABLISHED, which indicates that the routers have set up the circuit successfully, but that the end stations have not yet initiated their sessions across that circuit. This message could be indicative of any number of problems, including problems with XID exchanges or devices for which a VARY ACT command has not been issued from VTAM.

Note that when using FST peers (or direct encapsulation peers not using local acknowledgment), the data-link connection is not locally terminated. The RIF (if Token Ring) is terminated, but the data-link connection is end to end, and you will not see circuits established for sessions across DLSw+ FST or direct peers not using local acknowledgment.

The syntax of the **show dlsw circuits** command follows:

show dlsw circuits [**detail**] [*0-255*] | **mac-address** *address* | **sap-value** *value*]

Syntax Description

detail—Display full remote circuit details (this keyword can be specified in conjunction with any of the following keywords to minimize the volume of data returned; only one of the following keywords can be specified)

*0-255***—**Display the circuit with this key index

mac-address—Display the remote circuits using a specific MAC

sap-value—Display all remote circuits using a specific SAP

Figure 9-2 shows sample output from a **show dlsw circuits** command. In this example, the router had only a single circuit, so **detail** was specified without any qualifiers. For key routers at a central site, to minimize the output, you may chose to omit the detail if you are listing all of the active circuits.

Figure 9-2 Output from a show dlsw circuits Command

```
milan#show dlsw circuits detail
Index   local addr(lsap)remote addr(dsap) state
194-00  0800.5a9b.b3b2(F0)0800.5ac1.302d(F0) CONNECTED
        PCEP: 995AA4UCEP: A52274
        Port:To0/0peer 172.18.15.166(2065)
        Flow-Control-Tx CW:20, Permitted:28;Rx CW:22, Granted:25
        RIF = 0680.0011.0640
```

Show DLSw Fastcache

The **show dlsw fastcache** command allows you to display the cache being used by DLSw+ when FST or direct (passthrough) encapsulation is used. Using DLSw+ with FST peers or direct encapsulation peers (without local acknowledgment enabled) allows you to use the router's fast-switching capabilities, improving throughput and reducing the load on the router's CPU. To do this, a fast-switching cache must be built. The first frame between two end stations will be process switched, and during this process an entry will be made in the fast-switching cache so that subsequent frames between those end stations may be fast switched.

You can view the fast-switching cache that DLSw+ has created—this information can be useful in determining what path specific data is taking, or to help determine whether traffic is flowing between two specific stations.

Figure 9-3 shows the output from **show dlsw fastcache** command.

Figure 9-3 Output from a show dlsw fastcache Command

```
milan#show dlsw fastcache
peer              local-macremote-mac l/r sap rif
FST 172.18.15.166  0800.5a9b.b3b2  0800.5ac1.302d
F0/F0  0680.0011.0640
```

Show DLSw Local Circuits

Starting with Cisco IOS Release 11.1, local conversion via DLSw+ became a configurable option. Prior to this, to convert between diverse data-link protocols, a user had to have two routers peered to each other, each with one of the media types.

With the local conversion feature, now this can be done within a single router and with no remote peers required. DLSw+ supports local conversion between SDLC or QLLC and LLC2, and between SDLC and QLLC. To do the data-link conversion, the router must keep state information similar to that described in the section "Show DLSw Circuits." The router creates a circuit, but in this case both halves of the circuit are maintained on the same router. This information can be collected through the **show dlsw local-circuit** command. You can specify all local circuits or you can qualify the search in one of the arguments.

show dlsw local-circuit [*0-63*] | [**mac-address** *address*] | [**sap-value** *value*]

Syntax Description

0-63—Display the local circuit with this key index

mac-address—Display all local circuits using the specified MAC

sap-value—Display all local circuits using the specified SAP

Figure 9-4 shows the output of this command.

Figure 9-4 Output from a show dlsw local-circuit Command

```
milan#show dlsw local-circuit
 key      mac-addr   sap    state        port rif
58-00  4000.1234.56c1 04 CONNECTED      Se3/7 --no rif--
             PCEP: A4BB04  UCEP: A4BA04
       4001.3745.1088 04 CONNECTED      To0/0 08B0.A041.0DE5.0640
             PCEP: 995A18  UCEP: A4BA04
59-00 4000.1234.56c2  04 CONNECTED      Se3/7 --no rif--
             PCEP: A4C290  UCEP: A4C190
       4001.3745.1088 04 CONNECTED      To0/0 08B0.A041.0DE5.0640
             PCEP: A4B7A4  UCEP: A4C190
```

Show DLSw Peers

The **show dlsw peers** command allows you to show the status of remote peers. With the exception of local circuits, nothing happens in DLSw+ without remote peer connections. If the peer is not in CONNECT status, no data traffic will be able to flow between end stations that are trying to traverse the peer connection.

In addition to the state of the peer, the **show dlsw peers** command will tell you what kind of peer this is—either configured, promiscuous, or peer-on-demand, which is created when border peers are used. To show the status of a remote peer, use the **show dlsw peers** command with the following syntax:

show dlsw peers [**interface** *type number* | **ip-address** *ip-address*]

Syntax Description

interface—Interface used to access a remote peer (direct encapsulation)

ip-address—IP address of a remote peer (FST or TCP encapsulation)

Figure 9-5 shows the output of the **show dlsw peers** command.

Figure 9-5 Output from a show dlsw peers Command

```
milan#show dlsw peers
Peers                state    pkts_rx   pkts_tx   type drops   ckts
TCP    uptime
TCP 172.18.15.166  CONNECT     26086     8400    conf     0      1
0 00:03:42
```

Show DLSw Reachability

You can use the **show dlsw reachability** command to determine which SNA or NetBIOS DLSw+ end stations a router has in its cache. DLSw+ checks the reachability cache when it is trying to initiate a session to determine if it already knows the correct peer or port to use for this session. It also checks this cache when attempting to send traffic that is not session-based (that is, connectionless) across DLSw+. If DLSw+ does not know where a particular destination address is, it queries other peers that it knows about. When it does learn how to reach a destination, DLSw+ keeps that information for a specific amount of time in an effort to reduce the broadcast traffic on the network.

Reachability tables can become large. To make the table more useable, **show** commands have been changed in the later maintenance releases to allow you to search the reachability table for a particular MAC address or NetBIOS name. This simplifies problem isolation and diagnosis on a particular station (or a particular protocol). These changes are currently available in Cisco IOS Releases 11.0(10.1) and 11.1(3.3).

You can use the **show dlsw reachability** command to show the entire reachability cache, or use one of the keywords to show a portion of the reachability cache. Reachability is usually the second item to check when troubleshooting a connection that will not come up. First, check the peer to ensure that it is connected, because no traffic will flow over a disconnected peer. Then check the reachability. You should see that one of the devices is FOUND LOCAL and that the other is FOUND REMOTE (and vice-versa on the other peer). If the status of one of the resources is SEARCHING, VERIFY, or not present, there may be a problem in the data path between that device and its nearest DLSw+ peer. If not found, it may imply that the cache entry has timed out.

To determine which end stations are in a router’s cache use the **show dlsw reachability** command with the following syntax:

show dlsw reachability [**mac-address** [*address*]] [**netbios-names** [*name*]]

Syntax Description

mac-address—Displays all addresses in the reachability cache, or the path to a specific MAC

netbios-names—Displays all NetBIOS names in the reachability cache, or the path to a specific name

Figure 9-6 shows the output from a **show dlsw reachability** command.

Figure 9-6 Output from a show dlsw reachability Command

```
milan#show dlsw reachability
DLSw MAC address reachability cache list
Mac Addr        status    Loc.    peer/port             rif
0800.5a9b.b3b2  FOUND     LOCAL   TokenRing0/0     06B0.0011.0640
0800.5ac1.302d  FOUND     REMOTE  172.18.15.166(2065)

DLSw NetBIOS Name reachability cache list
NetBIOS Name    status    Loc.    peer/port             rif
paulo01s        FOUND     REMOTE  172.18.15.166(2065)
vito01r         FOUND     LOCAL   TokenRing0/0     06B0.0011.0640
```

Other Useful Show Commands

Show Interface

The **show interface** command can be useful to determine whether the data path between the end station and the DLSw+ router is active. In addition, for SDLC interfaces this command provides information about individual devices (SDLC addresses) on a single-drop or multidrop line. Figure 9-7 shows the output from a **show interface** command.

Figure 9-7 Output from a show interface Command for a Serial Interface Supporting SDLC

```
Serial3/7 is up, line protocol is up
 Hardware is cxBus Serial
 Description: sdlc config to MVS
 MTU 4400 bytes, BW 1544 Kbit, DLY 20000 usec, rely 255/255, load 1/255
 Encapsulation SDLC, loopback not set
   Router link station role: PRIMARY (DCE)
   Router link station metrics:
     slow-poll 10 seconds
     T1 (reply time out) 3000 milliseconds
     N1 (max frame size) 12016 bits
     N2 (retry count) 20
     poll-pause-timer 10 milliseconds
     poll-limit-value 1
     k (windowsize) 7
     modulo 8
     sdlc vmac: 4000.1234.56--
 sdlc addr C1 state is CONNECT
     cls_state is CLS_IN_SESSION
```

```
    VS 4, VR 4, Remote VR 4, Current retransmit count 0
    Hold queue: 0/200 IFRAMEs 20/20
    TESTs 0/0 XIDs 0/0, DMs 0/0 FRMRs 0/0
    RNRs 228/0 SNRMs 1/0 DISC/RDs 0/0 REJs 0/0
    Poll: set, Poll count: 0, chain: C2/C2
sdlc addr C2 state is CONNECT
    cls_state is CLS_IN_SESSION
    VS 4, VR 6, Remote VR 4, Current retransmit count 0
    Hold queue: 0/200 IFRAMEs 20/14
    TESTs 0/0 XIDs 0/0, DMs 0/0 FRMRs 0/0
    RNRs 357/0 SNRMs 1/0 DISC/RDs 0/0 REJs 0/0
    Poll: clear, Poll count: 0, ready for poll, chain: C1/C1
Last input never, output 00:00:00, output hang never
Last clearing of “show interface” counters never
Output queue 0/40, 0 drops; input queue 0/75, 0 drops
5 minute input rate 0 bits/sec, 10 packets/sec
5 minute output rate 0 bits/sec, 10 packets/sec
    1663 packets input, 8248 bytes, 0 no buffer
    Received 0 broadcasts, 0 runts, 0 giants
    0 input errors, 0 CRC, 0 frame, 0 overrun, 0 ignored, 0 abort
    1673 packets output, 6832 bytes, 0 underruns
    0 output errors, 0 collisions, 2 interface resets
    0 output buffer failures, 0 output buffers swapped out
    1 carrier transitions
    RTS down, CTS up, DTR up, DCD up, DSR up
```

Show IP Route

The **show ip route** command can be useful in determining why a peer is not reaching a CONNECT state. Often what appears to be a DLSw+ problem turns out to be an IP routing problem preventing one router from reaching the other. Using the **show ip route** command or executing an **extended ping** command to the remote peer address from the local peer address can help ensure that the problem is not an IP connectivity problem.

Show Source Bridge

Local SRB is used to get Token Ring frames into DLSw+. The **show source-bridge** command can be used to see if this local SRB has been correctly set up in the ring group. It will also indicate if there are large numbers of SRB drops at the interface level. Large numbers of drops could indicate a problem or could simply indicate the presence of an access list.

Show Bridge

When using transparent bridging as an entry into the DLSw+ cloud (for example, end stations on Ethernet), the **show bridge** command allows you to determine if the router knows the MAC address of the end stations and, if so, whether they were determined from the correct interface.

Show LLC2

When locally terminating LAN sessions, DLSw+ establishes LLC2 sessions with the LAN-attached end stations. The **show llc2** command is useful in monitoring the state of these LLC2 sessions. Figure 9-8 shows the output of a **show llc2** command.

Figure 9-8 Output from a show llc2 Command for a LAN Interface Supporting SDLC

```
milan#show llc
LLC2 Connections: total of 2 connections
TokenRing0/0 DTE: 4001.3745.1088 4000.1234.56c1 04 04 state NORMAL
   V(S)=9, V(R)=12, Last N(R)=9, Local window=7, Remote Window=127
   akmax=3, n2=8, Next timer in 2300
   xid-retry timer      0/0      ack timer       0/1000
   p timer              0/1000   idle timer      2300/10000
   rej timer            0/3200   busy timer      0/9600
   akdelay timer        0/100    txQ count       0/200
TokenRing0/0 DTE: 4001.3745.1088 4000.1234.56c2 04 04 state NORMAL
   V(S)=8, V(R)=9, Last N(R)=8, Local window=7, Remote Window=127
   akmax=3, n2=8, Next timer in 2504
   xid-retry timer      0/0      ack timer       0/1000
   p timer              0/1000   idle timer      2504/10000
   rej timer            0/3200   busy timer      0/9600
   akdelay timer        0/100    txQ count       0/200
```

Show TCP

When using TCP encapsulation, one TCP session (or more) is opened between the TCP peers. The **show tcp** command shows information about that session, including information about longest and average round-trip timers. This could be useful in finding WAN congestion or routing protocol issues that cause performance problems at the end station.

Other

There are many other commands that may be useful in certain environments, including:

- **show frame-relay map**
- **show frame-relay pvc**
- **show lnm station**
- **show interfaces accounting**

DLSw Debug Commands

Although it is possible to turn on all DLSw+ debugging, this may result in far more information than is needed in any particular situation and will make it more difficult to analyze the debug output. When possible, try to determine which debug is needed and turn on as little debug as possible. In addition, remember that it is advisable to use any router debugging only at the direction of Cisco engineers; it is possible to hang a router with too much debug, particularly if the router is running at high-CPU utilization. The following statement illustrates the syntax of the **debug dlsw** command:

debug dlsw [**core** | **local-circuit** | **peers** | **reachability**]

Syntax Description

core—Collects information about (remote) circuit events and flow control

local-circuit—Collects information about local circuit events

peers—Collects information about peer events

reachability—Collects information about explorer traffic and reachability

Core Debugging

The DLSw+ core is the engine responsible for establishing and maintaining remote circuits. If possible, specifying the index of the circuit you wish to debug will cut down on the amount of output you get. However, if you want to watch a circuit initially come up, this is not an option. The syntax of the **debug dlsw core** command is:

debug dlsw core [*0-255* | **flow-control** | **messages** | **state** | **xid**]

Syntax Description

*0-255***—**Limits debug output to circuits with this key index

flow-control—Limits DLSw+ debug to core flow control

messages—Limits DLSw+ debug to core messages

state—Limits DLSw+ debug to core finite state machine state transitions

xid—Limits DLSw+ debug to core XID command/response bit tracking

Core flow-control debugging will provide information about congestion in the WAN or at the remote end station. If the WAN or remote station is congested, DLSw+ will send receiver not ready frames on its local circuits, throttling data traffic on established sessions and giving the congestion an opportunity to clear.

Core message debugging allows you to view specific packets being received by DLSw+ from one of its peers or from a local medium via common layer services interface (CLSI).

Core state debugging allows you to see when the state of a circuit changes. This command is especially useful when attempting to determine why a session is not establishing or why it is being disconnected.

Core XID debugging allows you to track the XID state machine, which the router uses to track XID commands and responses used in negotiations between end stations prior to the establishment of a session.

Local Circuit Debugging

Local circuit debugging is comparable to core debugging for circuits that are established on a single router (Cisco IOS Release 11.1 and later). The same type of information in the complete set of debug DLSw+ core options is available with debug. The syntax of the **debug dlsw local-circuit** command is:

debug dlsw local-circuit *0-63*

Syntax Description

*0-63***—**Key index for a specific local circuit

Peer Debugging

Peer debugging is useful in determining why a DLSw+ peer is not reaching CONNECT state or why a peer in CONNECT state is being torn down. This debug is particularly useful in debugging problems related to border peers and peer-on-demand peers. The syntax of the **debug dlsw peers** command is:

debug dlsw peers [**interface** *interface* | **ip-address** *ip-address*]

Syntax Description

interface—Interface used to reach a remote peer (direct encapsulation only)

ip-address—IP address of a remote peer (TCP or FST encapsulation only)

Figure 9-9 shows the output from a **debug dlsw peers** command during a normal peer connect sequence, displayed from the router that initiated the peer connection.

Figure 9-9 Output from a debug dlsw peers Command Issued at the Initiating Router During a Normal Peer Connection

```
DLSw: action_a() attempting to connect peer 172.18.15.166(2065)
DLSw: action_a(): Write pipe opened for peer 172.18.15.166(2065)
DLSw: peer 172.18.15.166(2065), old state DISCONN, new state WAIT_RD
DLSw: passive open 172.18.15.166(11018) -> 2065
DLSw: action_c(): for peer 172.18.15.166(2065)
DLSw: peer 172.18.15.166(2065), old state WAIT_RD, new state CAP_EXG
DLSw: CapExId Msg sent to peer 172.18.15.166(2065)
DLSw: Recv CapExId Msg from peer 172.18.15.166(2065)
DLSw: Pos CapExResp sent to peer 172.18.15.166(2065)
DLSw: action_e(): for peer 172.18.15.166(2065)
DLSw: Recv CapExPosRsp Msg from peer 172.18.15.166(2065)
DLSw: action_e(): for peer 172.18.15.166(2065)
DLSw: peer 172.18.15.166(2065), old state CAP_EXG, new state CONNECT
DLSw: dlsw_tcpd_fini() for peer 172.18.15.166(2065)
DLSw: dlsw_tcpd_fini() closing write pipe for peer 172.18.15.166
DLSw: action_g(): for peer 172.18.15.166(2065)
DLSw: closing write pipe tcp connection for peer 172.18.15.166(2065)
DLSw: peer_act_on_capabilities() for peer 172.18.15.166(2065)
```

Figure 9-10 shows the output from a **debug dlsw peers** command during a normal peer connect sequence, displayed from the router that received the peer connection request.

Figure 9-10 Output from a debug dlsw peers Command Issued at the Receiving Router During a Normal Peer Connection

```
DLSw: passive open 172.18.15.166(11020) -> 2065
DLSw: action_b(): opening write pipe for peer 172.18.15.166(2065)
DLSw: peer 172.18.15.166(2065), old state DISCONN, new state CAP_EXG
DLSw: CapExId Msg sent to peer 172.18.15.166(2065)
DLSw: Recv CapExId Msg from peer 172.18.15.166(2065)
DLSw: Pos CapExResp sent to peer 172.18.15.166(2065)
DLSw: action_e(): for peer 172.18.15.166(2065)
DLSw: Recv CapExPosRsp Msg from peer 172.18.15.166(2065)
DLSw: action_e(): for peer 172.18.15.166(2065)
DLSw: peer 172.18.15.166(2065), old state CAP_EXG, new state CONNECT
DLSw: peer_act_on_capabilities() for peer 172.18.15.166(2065)
DLSw: dlsw_tcpd_fini() for peer 172.18.15.166(2065)
DLSw: dlsw_tcpd_fini() closing write pipe for peer 172.18.15.166
DLSw: action_g(): for peer 172.18.15.166(2065)
DLSw: closing write pipe tcp connection for peer 172.18.15.166(2065)
```

Figure 9-11 shows the output from a **debug dlsw peers** command during a normal peer disconnect sequence.

Figure 9-11 Output of a debug dlsw peers Command During Normal Peer Disconnect

```
DLSw: action_d(): for peer 172.18.15.166(2065)
DLSw: aborting tcp connection for peer 172.18.15.166(11015)
DLSw: peer 172.18.15.166(2065), old state CONNECT, new state DISCONN
```

Reachability Debugging

Reachability debugging allows you to see when entries are added to the DLSw+ reachability cache, when they are deleted from this cache, and when the core is able to find a destination MAC address or NetBIOS name in the cache (thereby avoiding a broadcast). If all this information is required, the **verbose** keyword should be specified.

debug dlsw reachability [**error** | **verbose**] [**netbios** | **sna**]

Syntax Description

error—Show only reachability errors

verbose—Show reachability event detail

netbios—Show only reachability events for NetBIOS

sna—Show only reachability events for SNA

The **verbose** keyword provides a great deal of information, so two subsets of verbose reachability debugging are available: error or event. Event debugging (default behavior if neither **verbose** nor **error** is specified) provides information only about events resulting in a state change, events that are not errors but are somewhat out of the ordinary, and errors. If only the errors are desired, the **error** keyword can be used. In normal operation, **error** should produce output only in rare situations (for example, low-memory conditions).

In a further effort to allow the user to minimize output, either the **sna** or **netbios** keywords can be specified in addition to one of the other keywords. If one is specified, only reachability debug will be produced if it was caused by that traffic protocol (or any traffic that DLSw+ cannot link to a specific protocol, such as TEST frame). If neither **sna** nor **netbios** is specified, debug will not check which protocol a message is related to before printing it.

The debug example in Figure 9-12 shows that DLSw+ is receiving TEST frames on the Ethernet interface. DLSw+ will put the source address into the reachability cache (if it is not already there). The status of SEARCHING here indicates that DLSw+ is already trying to resolve the destination MAC address. This router has already sent one canureach frame to its peers, so there is no need to send another. Had the status been NOT_FOUND, this DLSw+ peer would have sent a canureach frame to all of its peers. Had it been FOUND (in other words, there was already an entry in the reachability cache), the DLSw+ peer would have used that information to respond to the request or to forward the frame toward the destination (depending on whether the cache entry is fresh or stale).

Figure 9-12 Output from a debug dlsw reachability Command

```
CSM: Received CLSI Msg : TEST_STN.Ind dlen: 47 from TokenRing0/0
CSM:   smac c000.0000.0050, dmac 0800.5a54.ee59, ssap 4 , dsap 0
CSM: test_frame_proc: ws_status = SEARCHING
CSM: sending TEST to Serial3/7
CSM: Received CLSI Msg : TEST_STN.Ind dlen: 47 from TokenRing0/0
CSM:   smac c000.0000.0306, dmac 4000.0000.0308, ssap 4 , dsap 0
CSM: test_frame_proc: ws_status = SEARCHING
```

Other Useful Debug Commands

Other useful debug commands include:

- **debug source-bridge**
- **debug sdlc**
- **debug clsi**

Debug Examples

The following examples show how the **debug** commands can be used to pinpoint the cause of a problem.

Problem 1: No machines from a remote site can reach the central site. The peer at the remote site has IP address 172.18.15.156.

Action 1: Checking the output from the **show dlsw peers** command, we see:

```
Peers:                 state     pkts_rx  pkts_tx type drops ckts
TCP uptime
 TCP 172.18.15.156    DISCONN          0        0 conf     0    0
- -
```

Action 2: We can use **debug dlsw peers** command to determine the problem:

```
DLSw: action_a() attempting to connect peer 172.18.15.156(2065)
DLSw: action_a(): Write pipe opened for peer 172.18.15.156(2065)
DLSw: peer 172.18.15.156(2065), old state DISCONN, new state WAIT_RD
DLSw: dlsw_tcpd_fini() for peer 172.18.15.156(2065)
DLSw: tcp fini closing connection for peer 172.18.15.156(2065)
DLSw: action_d(): for peer 172.18.15.156(2065)
DLSw: peer 172.18.15.156(2065), old state WAIT_RD, new state DISCONN
DLSw: Not promiscuous - Rej conn from 172.18.15.166(2065)
```

Diagnosis: Attempts to open peer 172.18.15.156 are not successful. DLSw+ received an open request from 172.18.15.166, but DLSw+ rejected it because that peer was not defined. Upon investigation, we determine that the peer that we have defined was entered incorrectly and should be 172.18.15.166, which is the device attempting to peer to us. After changing this address, the peer connects:

```
Peers:                  state    pkts_rx  pkts_tx type drops ckts
TCP uptime
 TCP 172.18.15.166       CONNECT       2        2 conf     0 0
  00:24:27
```

Problem 2: SDLC-attached devices are unable to reach the host. Milan is the peer at the remote site where the SDLC devices reside.

Action 1: Issuing the **show dlsw peers** command tells us the peer is up:

```
milan#sh dlsw peers
Peers:                state      pkts_rx     pkts_tx  type   drops  ckts
TCP uptime
 TCP 172.18.15.166    CONNECT          9         140  conf       0  0
  00:02:10
```

Action 2: Issuing the **show dlsw circuits** tells us no circuits are up:

```
milan#show dlsw circuit
milan#
```

Action 3: Issuing a **show interfaces** command tells us the state of the SDLC addresses is USBUSY, which indicates that we have successfully connected to the downstream SDLC devices:

```
milan#show interfaces 3/7
Serial3/7 is up, line protocol is up
  Hardware is cxBus Serial
  Description: sdlc config to MVS
  MTU 4400 bytes, BW 1544 Kbit, DLY 20000 usec, rely 255/255, load
1/255
 Encapsulation SDLC, loopback not set
   Router link station role: PRIMARY (DCE)
   Router link station metrics:
     slow-poll 10 seconds
     T1 (reply time out) 3000 milliseconds
     N1 (max frame size) 12016 bits
     N2 (retry count) 20
     poll-pause-timer 10 milliseconds
     poll-limit-value 1
     k (windowsize) 7
     modulo 8
     sdlc vmac: 4000.1234.56--
 sdlc addr C1 state is USBUSY
     cls_state is CLS_STN_CLOSED
     VS 0, VR 0, Remote VR 0, Current retransmit count 0
     Hold queue: 0/200 IFRAMEs 29/18
     TESTs 0/0 XIDs 0/0, DMs 0/1 FRMRs 0/0
     RNRs 620/0 SNRMs 3/0 DISC/RDs 1/0 REJs 0/0
     Poll: clear, Poll count: 0, ready for poll, chain: C2/C2
 sdlc addr C2 state is USBUSY
     cls_state is CLS_STN_CLOSED
     VS 0, VR 0, Remote VR 0, Current retransmit count 0
     Hold queue: 0/200 IFRAMEs 37/26
     TESTs 0/0 XIDs 0/0, DMs 0/0 FRMRs 0/0
     RNRs 730/0 SNRMs 7/0 DISC/RDs 2/0 REJs 0/0
     Poll: set, Poll count: 0, chain: C1/C1
 Last input never, output 00:00:00, output hang never
 Last clearing of “show interface” counters never
 Output queue 0/40, 0 drops; input queue 3/75, 0 drops
 5 minute input rate 0 bits/sec, 40 packets/sec
 5 minute output rate 0 bits/sec, 40 packets/sec
     12740307 packets input, 25482189 bytes, 0 no buffer
     Received 0 broadcasts, 0 runts, 0 giants
     0 input errors, 0 CRC, 0 frame, 0 overrun, 0 ignored, 0 abort
     12740340 packets output, 25487483 bytes, 0 underruns
     0 output errors, 0 collisions, 5 interface resets
     0 output buffer failures, 0 output buffers swapped out
     3 carrier transitions
     RTS down, CTS up, DTR up, DCD up, DSR up
```

Action 4: By checking the configuration, we determine that these devices are defined to reach a partner at MAC address 4001.3745.1088:

```
milan#write terminal
...
!
interface Serial3/7
 description sdlc config to MVS
 mtu 4400
 no ip address
 encapsulation sdlc
 no keepalive
 clockrate 9600
 sdlc role primary
 sdlc vmac 4000.1234.5600
 sdlc N1 12016
 sdlc address C1
 sdlc xid C1 05DCCCC1
 sdlc partner 4001.3745.1088 C1
 sdlc address C2
 sdlc xid C2 05DCCCC2
 sdlc partner 4001.3745.1088 C2
 sdlc dlsw C1 C2
!
. . .
```

Action 5: Issuing the **show dlsw reachabiilty mac-address** command tells us DLSw+ has not been able to find this address:

```
milan#show dlsw reachability mac-address 4001.3745.1088
DLSw MAC address reachability cache list
Mac Addr        status     Loc    peer/port            rif
4001.3745.1088  SEARCHING  REMOTE
```

Action 6: Issuing the **show dlsw reachability mac-address** address at the FEP-attached router (bolzano) tells us the remote peer is still searching for this resource:

```
bolzano#show dlsw reachability mac-address 4001.3745.1088
DLSw MAC address reachability cache list
Mac Addr        status     Loc.    peer/port           rif
4001.3745.1088  SEARCHING  LOCAL
```

Action 7: We know this is a Token Ring-attached FEP, yet issuing the **show source-bridge** command tells us that no Token Ring interfaces are set up for SRB:

```
bolzano#show source-bridge
Global RSRB Parameters:
 TCP Queue Length maximum: 100

Ring Group 100:
  No TCP peername set, TCP transport disabled
   Maximum output TCP queue length, per peer: 100
  Rings:
```

Diagnosis: After adding the **source-bridge** statement to interface Token Ring 0, we again issue the **show source-bridge** command and see:

```
bolzano#show source-bridge

Local Interfaces:                   receive             transmit
       srn bn trn r p s n max hops  cnt:bytes
cnt:bytes     drops
To0    222  6 100 *   f   7  7  7    23:6562              0:0
0

Global RSRB Parameters:
 TCP Queue Length maximum: 100

Ring Group 100:
 No TCP peername set, TCP transport disabled
  Maximum output TCP queue length, per peer: 100
 Rings:
  bn: 6 rn: 222 local ma: 4000.3060.0458 TokenRing0
fwd: 0

Explorers: ------- input -------        ------- output -------
         spanning   all-rings   total  spanning all-rings
```

```
total
To0             0          0        0         0          0
0
 Local: fastswitched 19        flushed 0     max Bps 38400

        rings     inputs          bursts     throttles
output drops
           To0        19               0             0
0
```

The SDLC circuits have come up:

```
bolzano#show dlsw circuits
Index   local addr(lsap)        remote addr(dsap) state
250-00  4001.3745.1088(04)      4000.1234.56c1(04) CONNECTED
        Port:To0         peer 172.18.15.157(2065)
        Flow-Control-Tx CW:20,  Permitted:29; Rx CW:20, Granted:32
        RIF = 08B0.A041.0DE6.0640
251-00  4001.3745.1088(04)      4000.1234.56c2(04) CONNECTED
        Port:To0        peer 172.18.15.157(2065)
        Flow-Control-Tx CW:20, Permitted:31; Rx CW:20, Granted:32
        RIF = 08B0.A041.0DE6.0640
```

Problem 3: This case is similar to the last case, but one remote SDLC device comes up, while the other remote device does not. Milan is the router attached to the remote SDLC devices.

Action 1: Issuing the **show dlsw peers** command tells us the peer is up:

```
milan#show dlsw peers
Peers:                state     pkts_rx   pkts_tx  type drops ckts
TCP uptime
 TCP 172.18.15.166   CONNECT        561       420  conf     0    2
0 00:26:42
```

Action 2: The **show dlsw reachability mac-address** command (specifying the MAC address of the FEP) tells us that reachability is all right:

```
milan#show dlsw reachability mac-address 4001.3745.1088
DLSw MAC address reachability cache list
Mac Addr        status    Loc.   peer/port             rif
4001.3745.1088  FOUND     REMOTE 172.18.15.166(2065)
```

Action 3: The **show dlsw circuits mac-address** command tells us that only one of the two circuits is connected:

```
milan#show dlsw circuit mac-address 4001.3745.1088
Index    local addr(lsap)   remote addr(dsap)  state
250-00   4000.1234.56c1(04) 4001.3745.1088(04) CONNECTED
251-00   4000.1234.56c2(04) 4001.3745.1088(04) CKT_ESTABLISHED
```

The state of CKT_ESTABLISHED tells us that there is a data path between the two devices over which a session could be established, but that session has not yet connected (in this case, no SABME/UA exchange has occurred).

Action 4: Issuing a **show debug dlsw core** command provides the following output:

```
milan#debug dlsw core state
DLSw core state debugging is on
milan#
DLSw: START-FSM (251-00): event:DLC-Id state:CKT_ESTABLISHED
DLSw: core: dlsw_action_f()
DLSw: END-FSM (251-00): state:CKT_ESTABLISHED->CKT_ESTABLISHED
DLSw: START-FSM (251-00): event:DLC-Id state:CKT_ESTABLISHED
DLSw: core: dlsw_action_f()
DLSw: END-FSM (251-00): state:CKT_ESTABLISHED->CKT_ESTABLISHED
DLSw: START-FSM (251-00): event:WAN-XID state:CKT_ESTABLISHED
DLSw: core: dlsw_action_g()
DLSw: END-FSM (251-00): state:CKT_ESTABLISHED->CKT_ESTABLISHED
DLSw: START-FSM (251-00): event:DLC-Id state:CKT_ESTABLISHED
DLSw: core: dlsw_action_f()
DLSw: END-FSM (251-00): state:CKT_ESTABLISHED->CKT_ESTABLISHED
DLSw: START-FSM (251-00): event:DLC-Id state:CKT_ESTABLISHED
DLSw: core: dlsw_action_f()
DLSw: END-FSM (251-00): state:CKT_ESTABLISHED->CKT_ESTABLISHED
DLSw: START-FSM (251-00): event:DLC-Id state:CKT_ESTABLISHED
DLSw: core: dlsw_action_f()
DLSw: END-FSM (251-00): state:CKT_ESTABLISHED->CKT_ESTABLISHED
DLSw: START-FSM (251-00): event:DLC-Id state:CKT_ESTABLISHED
DLSw: core: dlsw_action_f()
DLSw: END-FSM (251-00): state:CKT_ESTABLISHED->CKT_ESTABLISHED
DLSw: START-FSM (251-00): event:WAN-XID state:CKT_ESTABLISHED
DLSw: core: dlsw_action_g()
DLSw: END-FSM (251-00): state:CKT_ESTABLISHED->CKT_ESTABLISHED
DLSw: START-FSM (251-00): event:DLC-Id state:CKT_ESTABLISHED
DLSw: core: dlsw_action_f()
DLSw: END-FSM (251-00): state:CKT_ESTABLISHED->CKT_ESTABLISHED
```

Diagnosis: We see that DLSw+ is seeing and passing XIDs from both the SDLC-attached device and the FEP, yet the FEP is not attempting to initiate the session. This is often an issue with something in the XID (most commonly the IDBLK/IDNUM).

Action 5: Checking the configuration at milan, we see that the XID defined for use on the router is 05DCCCCC:

```
milan#write terminal
. . .
!
interface Serial3/7
 description sdlc config to MVS
 mtu 4400
no ip address
 encapsulation sdlc
 no keepalive
 clockrate 9600
 sdlc role primary
 sdlc vmac 4000.1234.5600
 sdlc N1 12016
 sdlc address C1
 sdlc xid C1 05DCCCC1
 sdlc partner 4001.3745.1088 C1
 sdlc address C2
 sdlc xid C2 05DCCCCC
 sdlc partner 4001.3745.1088 C2
. . .
```

Action 6: Checking the configuration in VTAM, we see that the XID is supposed to be 05DCCCC2. There is no way to see what is defined in VTAM from the router; this must be obtained from the host. After changing this value, the session comes up:

```
milan#conf t
Enter configuration commands, one per line. End with CNTL/Z.
milan(config)#int s 3/7
milan(config-if)#sdlc xid c2 05DCCCC2
milan(config-if)#^Z
milan#show dlsw circuit
Index   local addr(lsap)   remote addr(dsap)  state
250-00  4000.1234.56c1(04) 4001.3745.1088(04) CONNECTED
251-00  4000.1234.56c2(04) 4001.3745.1088(04) CONNECTED
```

These examples are not meant to be an exhaustive list of the things that can go wrong and how to detect them. However, these are fairly useful in demonstrating how to use the available tools to attack and diagnose any DLSw+ connectivity issue.

Using CiscoWorks Blue: Maps, SNA View, and Native Service Point

This chapter describes how to use some of the enhanced network management tools available with DLSw+. It shows the kind of information available with these tools, provides some sample output, and describes the prerequisites to using these tools with DLSw+.

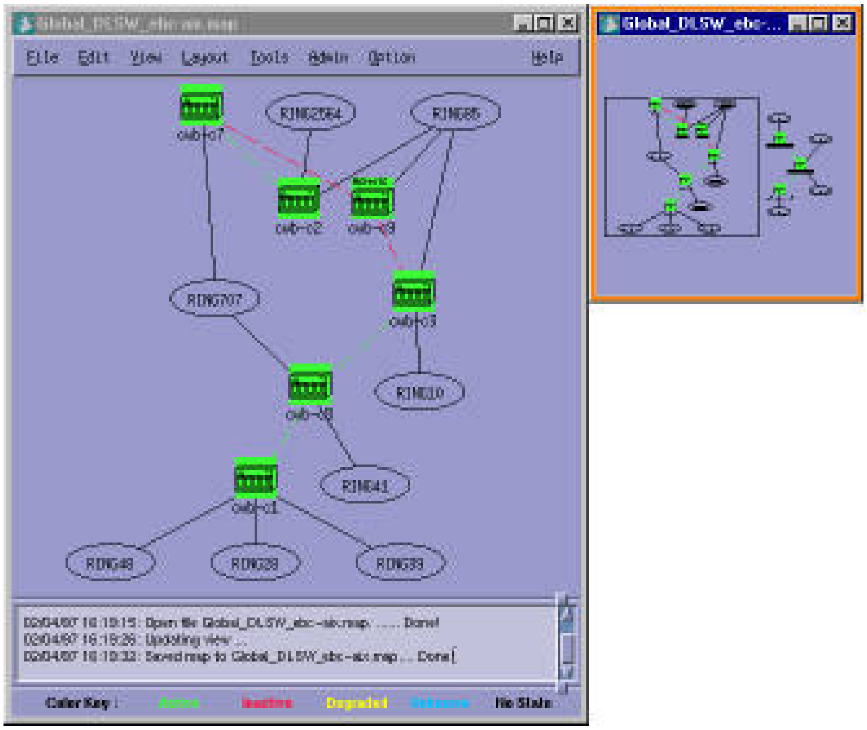
CiscoWorks Blue Maps provides a logical view of the portion of your router network relevant to DLSw+ (there is a similar tool for RSRB and APPN). CiscoWorks Blue SNA View adds to the information provided by Maps by correlating SNA PU and LU names with DLSw+ circuits and DLSw+ peers. CiscoWorks Blue Native Service Point support allows you to manage your router network from the mainframe console using IBM's NetView or Sterling's SOLVE:Netmaster.

Using Maps

Quite often, the challenge with network management is that there is too much information, not too little. Maps addresses this problem by providing information relevant to the problem you are trying to solve in an easy-to-use, graphical interface. Maps for DLSw+ requires Cisco IOS Release 11.1(6.4). DLSw+ routers must be IP-addressable to be viewed using Maps. Currently, you must use either FST or TCP encapsulation to show peers.

The Global DLSw map, shown in Figure 10-1, is the first screen you come to when you start Maps. This map is derived by polling DLSw+ routers. To minimize polling traffic, you optionally can define a set of key routers that are centrally located and peer to all remote routers. When key routers are specified, they are the only routers polled to generate the global map. Other routers are polled only when required to obtain requested information.

Figure 10-1 Global Map View for DLSw+ in Maps



```
02/04/97 16:19:15: Open file Global_DLSW_abc-ab.map ...... Done!
02/04/97 16:19:26: Updating view ...
02/04/97 16:19:32: Saved map to Global_DLSW_abc-ab.map ... Done!
```

Key routers are polled periodically to obtain updated information about peer connections and circuits. Making central site routers peer routers reduces polling overhead while allowing full visibility. To mark a router key, go to Edit and select Key Devices.

The Global DLSw map shows only the DLSw+ routers, not the IP-addressable entities that are not transporting SNA or NetBIOS. By eliminating the “noise,” this screen allows you to more quickly pinpoint potential problem areas. DLSw+ peer connections are shown with color-coded lines connecting the routers. The locator window allows you to hone in on the relevant DLSw+ routers.

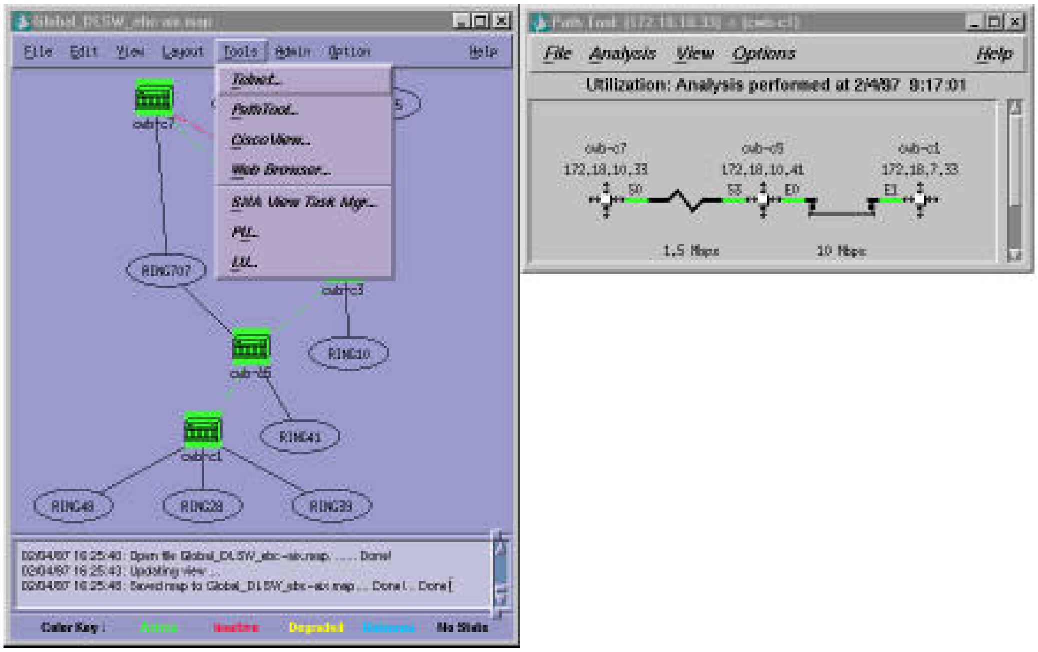
When you click on an individual DLSw+ peer in this screen, the popup window allows you to access:

- FocusView—A view of the DLSw+ network from a particular router
- Information—For example, the version number of DLSw+ running in this router, its uptime, the number of active peer connections, and the number of active circuits
- Statistics—A list of remote peers connected to this peer, and then statistical information about the peer connection
- Circuit List—A list of all circuits established through this peer (valid for TCP or LLC2 encapsulation only)

As shown in Figure 10-2, when you click on a router, you can also use the Tools menu at the top of the screen to Telnet to the router. For instance, you can select CiscoView from the Tools menu to see the physical connections and state of the router. Finally, you can access Path Tool from the Tools menu. If SNA View is installed, you can also access the PU/LU Filter screen and view NetView Logs. SNA View will be covered more later in this chapter.

When you request a circuit list, Maps polls the selected router. The router looks at the circuit list in the MIB and obtains the MAC/SAP pair for the next n circuits and the state of those circuits. (As you will see later on, if SNA View is installed, the router can then use the correlation logic to associate the circuit with a PU name). The circuit list shows the state of the circuits and provides a simple way to search for the circuit or group of circuits for which you need more information. All circuit information requires Maps Release1.1.

Figure 10-2 Tools Integration

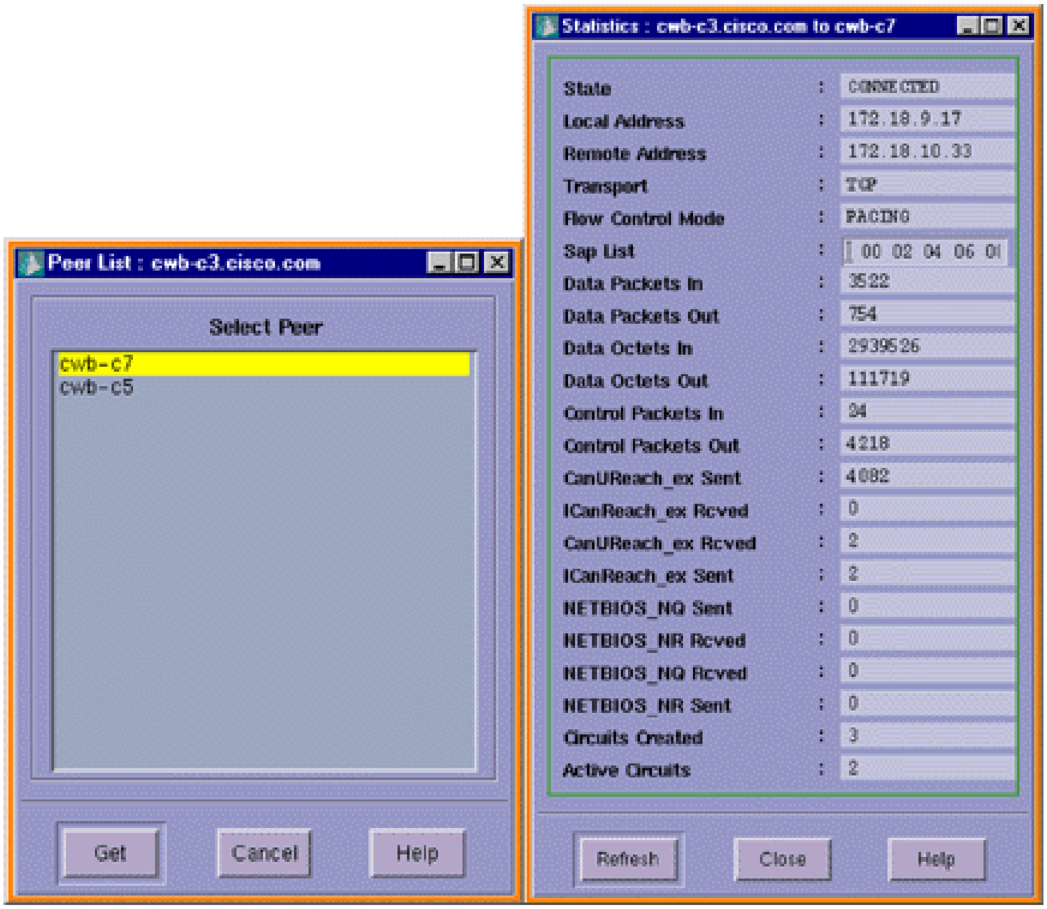


```
02/04/97 16:25:40 : Open file Global_DLSW_abc-ab.map. ..... Done!
02/04/97 16:25:43 : Updating view ...
02/04/97 16:25:48 : Saved map to Global_DLSW_abc-ab.map ... Done! .. Done[
```

Utilization: Analysis performed at 2/4/97 9:17:01

From the list of remote peers, you can select a specific remote peer and get the statistics for that peer. This includes data packet counts, explorer packet counts, and circuit counts for that peer. When you select a peer and request statistics, that peer router is polled and its MIB is checked. The output of this is shown in Figure 10-3. This screen may illustrate, for example, that a specific peer has a large number of circuits active or that it is processing a large number of NetBIOS broadcasts.

Figure 10-3 Sample View of Remote Peer Statistics

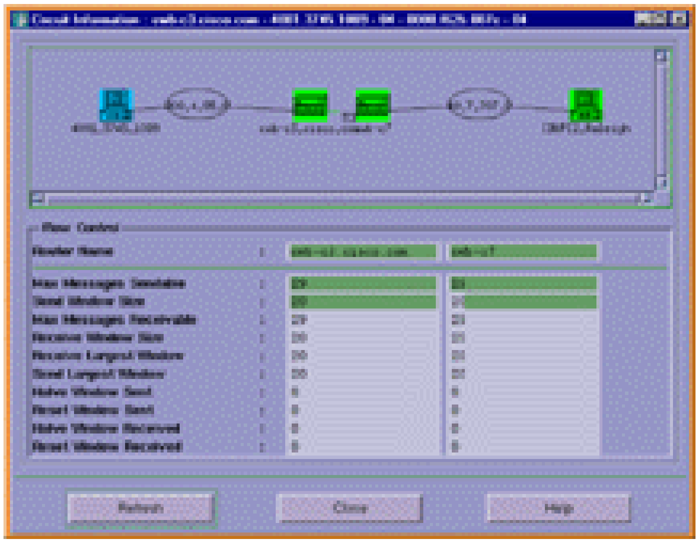

Peer List : cwb-c3.cisco.com
Select Peer
cwb-c7
cwb-c5
Get
Cancel
Help
Statistics : cwb-c3.cisco.com to cwb-c7
Refresh
Close
Help

State : CONNECTED
Local Address : 172.18.9.17
Remote Address : 172.18.10.33
Transport : TCP
Flow Control Mode : PACING
Sap List : 00 02 04 06 0
Data Packets In : 3522
Data Packets Out : 754
Data Octets In : 2939526
Data Octets Out : 111719
Control Packets In : 34
Control Packets Out : 4218
CanUReach_ex Sent : 4082
ICanReach_ex Rcved : 0
CanUReach_ex Rcved : 2
ICanReach_ex Sent : 2
NETBIOS_NQ Sent : 0
NETBIOS_NR Rcved : 0
NETBIOS_NQ Rcved : 0
NETBIOS_NR Sent : 0
Circuits Created : 3
Active Circuits : 2

From the circuit list described earlier, you can access specific circuit information. The Circuit Information screen, shown in Figure 10-4, shows the circuit path comprised of the data links at the end point and the peer connection in the middle. If the data link is Token Ring, the screen illustrates RIF. (Token Ring is shown as an oval. SDLC, FDDI, and Ethernet are indicated with a straight line and labeled either SDLC or LLC.) The state of each element on the path is illustrated with color coding, allowing you to rapidly pinpoint problems.

If you are running SNA View, you will also get the name of any PU 2.0 or PU 2.1. For PU 4s or mainframe channel gateways, the MAC address is shown. The detailed circuit information illustrates flow-control data. Large discrepancies between the sending counts of one router and the receive counts of the other could indicate a problem.

Figure 10-4 Sample View of Circuit Information Screen



Router Name	:	cwb-c3.cisco.com	cwb-c7
Max Messages Sendable	:	29	20
Send Window Size	:	20	20
Max Messages Receivable	:	29	20
Receive Window Size	:	20	20
Receive Largest Window	:	20	20
Send Largest Window	:	20	20
Halve Window Sent	:	0	0
Reset Window Sent	:	0	0
Halve Window Received	:	0	0
Reset Window Received	:	0	0

Using SNA View

SNA View allows you to correlate DLSw+ circuits and peer connections with PU and LU names. SNA View has a mainframe component that queries VTAM to build a database of SNA PUs and LUs and maintains this database by capturing VTAM messages that indicate state changes.

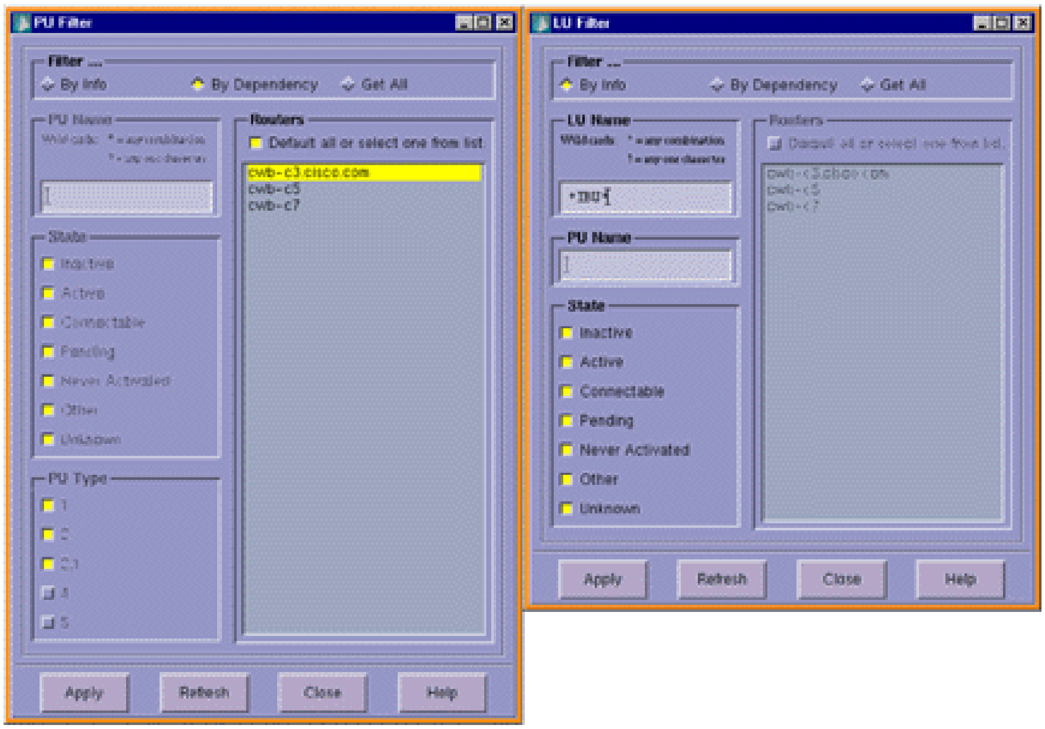
SNA View requires VTAM 4.1 and NetView 1.3 or Sterling's SOLVE: Netmaster 3.1. SNA View runs on MVS, but not VM or VSE. Obviously, most AS/400 environments do not have VTAM; therefore, SNA View does not provide PU or LU information for AS/400 environments. (SNA View for DLSw+ and RSRB requires VTAM, but SNA View for APPN does not.)

To access information about PUs and LUs in your network, use the PU or LU Filter screen. To get to this screen, select Tools from the main menu and then select PU or LU from the pull-down menu. There are two ways to access information about SNA PUs and LUs from this screen.

1. By Info—You can request information about a specific PU or LU, or a list of PUs or LUs known to VTAM, by specifying the resource name, specifying a partial name with wildcards, or requesting the entire list; the PU name to MAC/SAP correlation requires either TCP or LLC2 encapsulation, because this information is derived from circuit lists in the router; neither FST or direct encapsulation tracks circuits
2. By Dependency—You can request the PUs and LUs associated with a specific router; the dependency view requires that the peer transport be TCP

Figure 10-5 shows a PU and an LU Filter screen. If you click on By Info, you would enter the PU information on the left side of the PU Filter screen. If you click on By Dependency, you would select a router on the right side of the PU Filter screen.

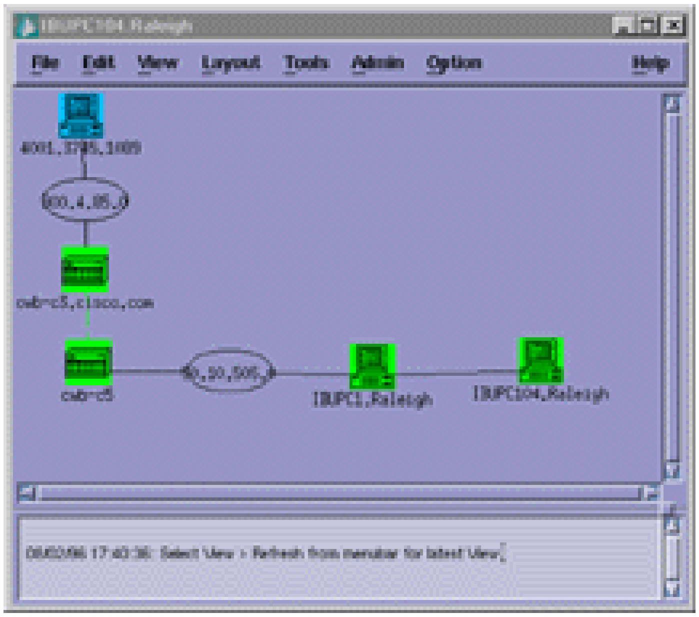
Figure 10-5 PU and LU Filter Screens



From this screen, you will get a list of PUs or LUs. From the list of PUs or LUs that match the previous criteria, you can activate or deactivate SNA devices from this screen. Because the activate or deactivate command kicks an activation or deactivation request to VTAM, you can now use your SNMP console to control any SNA resource (PU 2.x or LU), including those that are not part of your DLSw+ network.

By selecting a PU or LU from the list, you can get information about how that PU or LU is connected if it is connected to a DLSw+ router. This is shown in Figure 10-6.

Figure 10-6 LU Dependency Screen



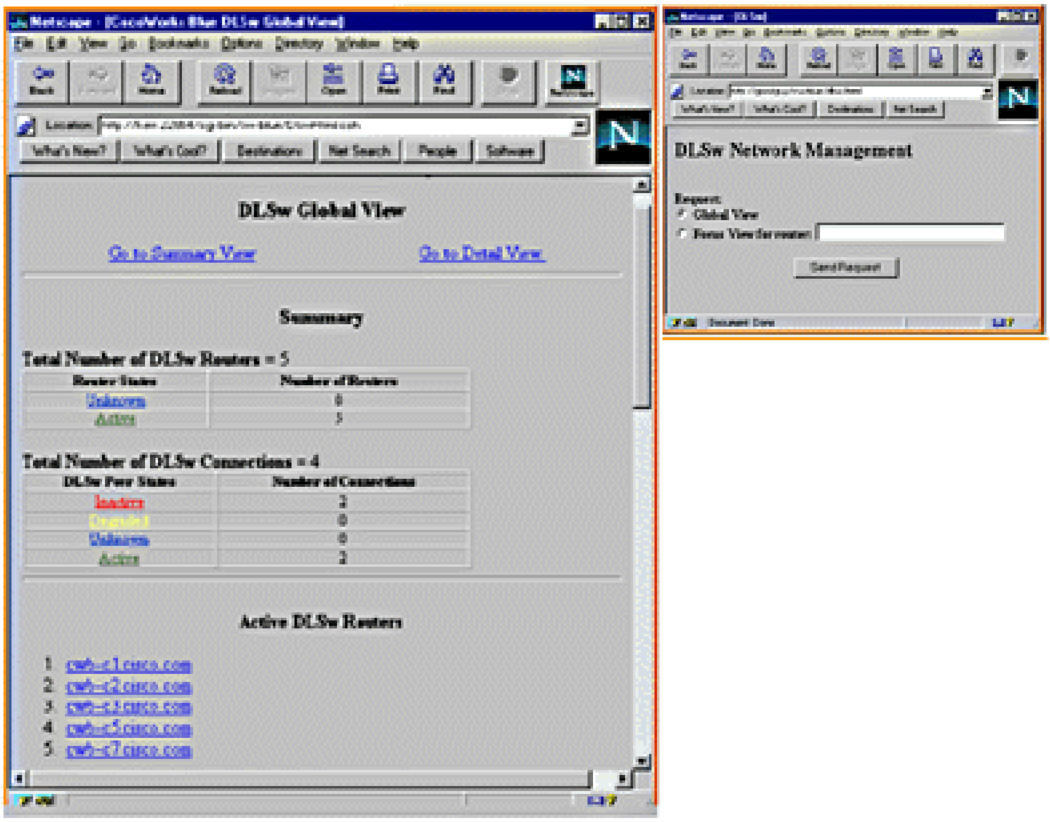
08/02/96 17:40:36: Select View > Refresh from menubar for latest View.

This screen uses color to indicate problem areas in the network. From this screen, you can also select a DLSw+ peer and use Path Tool to trace the path between two DLSw+ peers. To get the IP path between cwb-c3 and cwb-c5 in Figure 10-6, you would click on one of the routers and select Path Tool from the Tools pull down menu.

Managing DLSw from the Web

Most of the data available through CiscoWorks Blue Maps and SNA View can also be accessed through Web browsers, providing an easy-to-use interface that can be used from any PC desktop. With this capability, users no longer need to be seated at or using X-software into the network management system. There must still be at least one UNIX platform somewhere in the management environment. Through the Web browser interface from office or home, users can retrieve information about their DLSw, APPN, and RSRB networks on a platform of their choice. For example, as shown in Figure 10-7, the user might want to see a tabular snapshot of the state of the DLSw+ network.

Figure 10-7 Web Browser View of DLSw+ Network Status


Netscape - [CiscoWorks Blue DLSw Global View]
File Edit View Go Bookmarks Options Directory Window Help
Back Forward Home Reload Images Open Print Find Stop
Location:
What's New? What's Cool? Destinations Net Search People Software

DLSw Global View

Go to Summary View   Go to Detail View

Summary

Total Number of DLSw Routers = 5

Router Status	Number of Routers
Unknown	0
Active	5

Total Number of DLSw Connections = 4

DLSw Peer Status	Number of Connections
Inactive	2
Degraded	0
Unknown	0
Active	2

Active DLSw Routers

1. cwb-c1.cisco.com
2. cwb-c2.cisco.com
3. cwb-c3.cisco.com
4. cwb-c5.cisco.com
5. cwb-c7.cisco.com

DLSw Network Management

Request:
● Global View
○ Focus View for router:

Using Native Service Point

With Cisco IOS Release 11.0, every Cisco router that shipped with the IBM software feature set also shipped with service point capability. Service point capability allows a Cisco router to communicate directly to either IBM's NetView or Sterling's SOLVE:Netmaster.

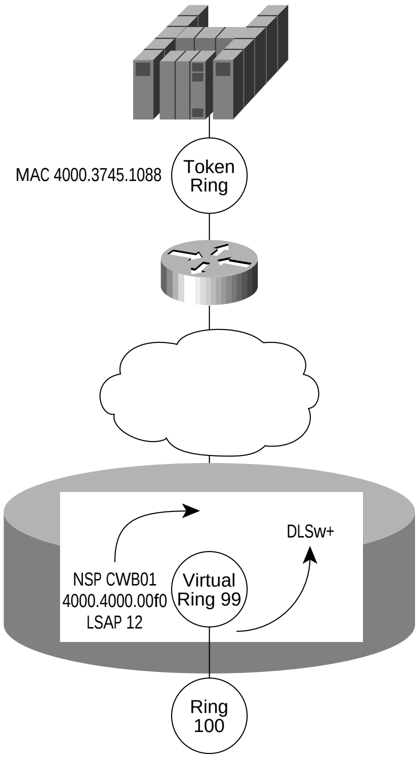
To use this feature, you must configure your router to use the service point function. You must also configure VTAM to recognize the service point router as a PU.

When VTAM initializes, it activates a system services control points (SSCP)-to-PU session between itself and each router that has been configured to use the service point function. NetView or SOLVE:Netmaster uses this SSCP-to-PU session to establish a service point session with the router. NetView and SOLVE:Netmaster then use this service point interface to send commands to the router or receive messages from the router.

If you configure service point in your routers, your router will send SNA alerts directly to VTAM, greatly improving visibility from your mainframe network management applications. For example, if you are using DLSw+ for media conversion from SDLC to Token Ring, by using service point you can view SDLC alerts at your Hardware Monitor screen even though your SNA resources now appear as if they were Token Ring attached to VTAM. To use service point in conjunction with DLSw+ requires Cisco IOS Release 11.1(5). Figure 10-8 shows a sample configuration.

To configure service point, you must define the SNA PU in the router (using the **dspu host** command), define a virtual MAC address for this SNA PU (4000.4000.00f0 was used in Figure 10-8), and use the virtual data-link control (VDLC) statement to connect the service point traffic to the DLSw+ virtual ring.

Figure 10-8 Configuration of Service Point over DLSw+



```
source-bridge ring-group 99

dlsw local-peer peer-id 150.10.20.1
dlsw remote-peer 0 tcp 150.10.20.2
sna vdlc 99 4000.4000.00f0/* define LAN address of Native Service Point
sna vdlc enable-host lsap 12/* define SAP for Native Service Point
dspu host cwb01 xid-snd 06500001 rmac 4000.3745.1088 rsap 4 lsap 12
sna vdlc start cwb01
interface TokenRing0
source-bridge 100 1 99
```

To further enhance network management, you can use Cisco’s application Native Service Point. Native Service Point Release 2.0 requires MVS VTAM 3.4 and IBM NetView 1.3 or Sterling SOLVE:Netmaster 3.1.

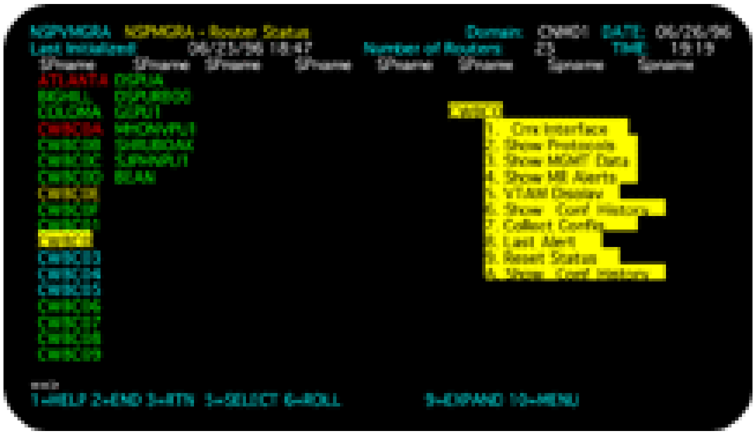
Native Service Point allows you to access the command line interface of a Cisco router from your NetView or SOLVE:Netmaster console. Using Native Service Point, you can issue any command from your NetView or SOLVE:Netmaster console that you can issue from a Telnet interface to the router. You can configure the router, issue **show** commands, and issue **debug** commands.

Native Service Point is an ideal solution for SNA environments that are just beginning to deploy multiprotocol networks and have NetView or SOLVE:Netmaster expertise but not Simple Network Management Protocol (SNMP) expertise. It minimizes training and equipment costs for network management while providing enough function to easily maintain a small network of 50 or fewer routers. The router interface through Native Service Point is more user friendly than a Telnet interface (with features such as command retrieval and the ability to store output to a virtual sequential access method [VSAM] data base).

For networks with more than 50 routers, you may prefer to use this tool in conjunction with CiscoWorks SNMP tools to allow control over groups of routers for such labor-intensive tasks as configuration downloads and software upgrades.

Figure 10-9 shows the main Native Service Point screen. This screen lists each Cisco router configured with NSP (each router has an SNA PU appearance to VTAM). The router status is indicated with color. Selecting a router causes the popup menu to appear, which allows you to either access the router's command line interface or gather information about the router.

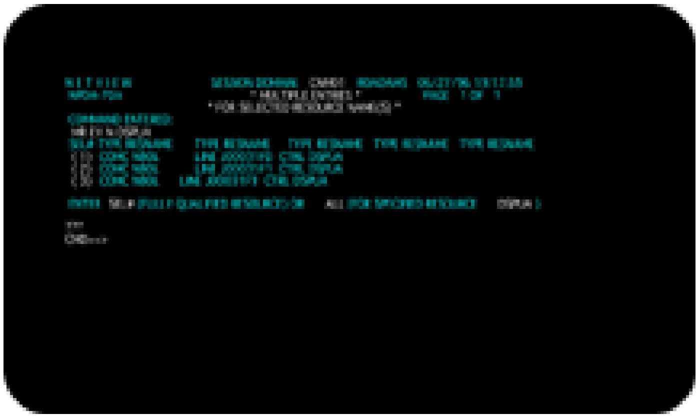
Figure 10-9 Main Cisco Native Service Point Screen at NetView



```
NSPVMGRA  NSPMGRA - Router Status                    Domain:  CNM01  DATE:  06/26/96
Last Initialized:     06/25/96 18:47      Number of Routers:   23      TIME:    19:19
 SPname      SPname  SPname     SPname   SPname   SPname    Spname    Spname
ATLANTA  DSPUA
BIGHILL  DSPUR800
COLOMA   GLPU1                                    CWBC02
CWBC0A   MHONVPU1                                   1.  Cns Interface
CWBC0B   SHRUBOAK                                   2.  Show Protocols
CWBC0C   SJPVNPU1                                   3.  Show MGMT  Data
CWBC0D   BEAN                                       4.  Show MR Alerts
CWBC0E                                              5.  VTAM Display
CWBC0F                                              6.  Show   Conf  History
CWBC01                                              7.  Collect Config
CWBC02                                              8.  Last Alert
CWBC03                                              9.  Reset Status
CWBC04                                              A.  Show   Conf  History
CWBC05
CWBC06
CWBC07
CWBC08
CWBC09

===>
1=HELP 2=END 3=RTN  5=SELECT 6=ROLL             9=EXPAND 10=MENU
```

Figure 10-10 shows the Hardware Monitor alert panel for a router, reached from option 4 (Show MR Alerts) on the previous figure. Each line of data represents an alert that was detected against the selected router. By selecting a number next to one of these alerts, you can see more details about the alert, such as router connectivity to host and performance statistics.

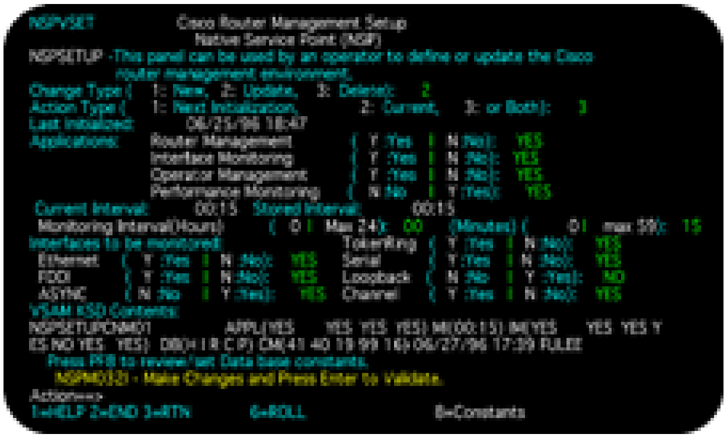
Figure 10-10 Hardware Monitor Screen with Alert Data Forwarded from a Cisco Router



```
N E T V I E W            SESSION DOMAIN: CNM01  [illegible]  06/27/96 [illegible]
NPDA-70H                       * MULTIPLE ENTRIES *            PAGE  1 OF  1
                        * FOR SELECTED RESOURCE NAME(S) *
COMMAND ENTERED:
[illegible]
SEL# TYPE RESNAME     TYPE RESNAME     TYPE RESNAME   TYPE RESNAME   TYPE RESNAME
( 1) CONC [illegible]     LINE [illegible]  CTRL DSPUA
( 2) CONC [illegible]     LINE [illegible]  CTRL DSPUA
( 3) CONC [illegible]   LINE [illegible]  CTRL DSPUA

ENTER SEL# (FULLY QUALIFIED RESOURCE) OR  ALL (FOR SPECIFIED RESOURCE    DSPUA )

???
CMD==>
```

Figure 10-11 shows the Native Service Point setup screen, which you can use for trend analysis. You can select to monitor specific routers or specific interfaces on those routers. You determine how often you want polling to occur and where you want the data to be archived. You can monitor router memory and cycle utilization as well as traffic and error statistics.

Figure 10-11 Native Service Point Tailored for Trend Analysis



```
NSPVSET            Cisco Router Management Setup
                     Native Service Point (NSP)
NSPSETUP -This panel can be used by an operator to define or update the Cisco
          router management environment.
Change Type (   1: New,  2: Update,    3: Delete):    2
Action Type (   1: Next Initialization,          2: Current,    3: or Both):    3
Last Initialized:      06/25/96 18:47
Applications:    Router Management          ( Y :Yes  | N :No):   YES
                 Interface Monitoring       ( Y :Yes  | N :No):  YES
                 Operator Management        ( Y :Yes  | N :No):   YES
                 Performance Monitoring    ( N :No    | Y :Yes):    YES
 Current Interval:       00:15  Stored Interval:        00:15
 Monitoring Interval(Hours)        (  0 |  Max 24):  00    (Minutes) (      0 |  max 59):  15
Interfaces to be monitored:              TokenRing  ( Y :Yes  | N :No):   YES
 Ethernet    ( Y :Yes  | N :No):   YES   Serial     ( Y :Yes  | N :No):   YES
 FDDI        ( Y :Yes  | N :No):   YES   Loopback   ( N :No   | Y :Yes):   NO
 ASYNC      ( N :No    | Y :Yes):    YES  Channel    ( Y :Yes  | N :No):   YES
VSAM KSD Contents:
NSPSETUPCNM01               APPL(YES     YES YES YES) MI(00:15) IM(YES     YES YES Y
ES NO YES  YES)  DB(H I R C P) CM(41 40 19 99 16) 06/27/96 17:39 FULEE
   Press PF8 to review/set Data base constants.
    NSPM032I - Make Changes and Press Enter to Validate.
Action===>
1=HELP 2=END 3=RTN          6=ROLL                     8=Constants
```

Using DLSw+ with Other Features

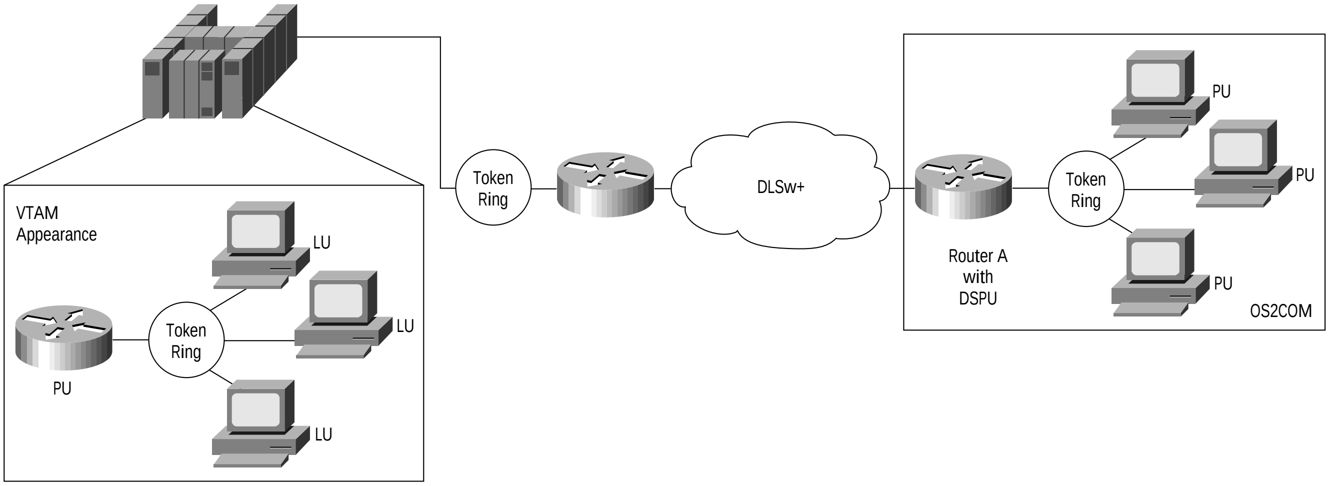
This chapter describes how to use DLSw+ in conjunction with other Cisco IOS software features: APPN, DSPU concentration, LAN network manager (LNM), and NCIA. It briefly describes these features, discusses why you would want to run these features in the same router with DLSw+, and provides some sample configurations.

Using DLSw+ with DSPU Concentration

With Cisco IOS Release10.3, the Cisco IOS software includes support for a feature known as DSPU concentration. In Release 11.1(5), DSPU concentration can be used in conjunction with DLSw+ in the same router. (In Release 11.2, DSPU concentration ships as part of the IBM base.)

DSPU concentration consolidates multiple DSPUs into a single upstream PU appearance with multiple LUs, as shown in Figure 11-1. This can be useful when remote branches have 20 to 30 PUs per branch, a common occurrence if using client software that has both a PU and LU appearance. With 300 or more branch offices, this equates to 6000 to 9000 PUs in the network. DSPU concentration can reduce this to 300 PUs.

Figure 11-1 DSPU Concentrates SNA PUs



By reducing the number of PUs (in this case, from 9000 to 300), you accomplish several things:

- Minimize NCP and VTAM memory requirements
- Speed up disaster recovery, because fewer SSCP_PU sessions need to be reestablished
- Minimize the number of central site routers required

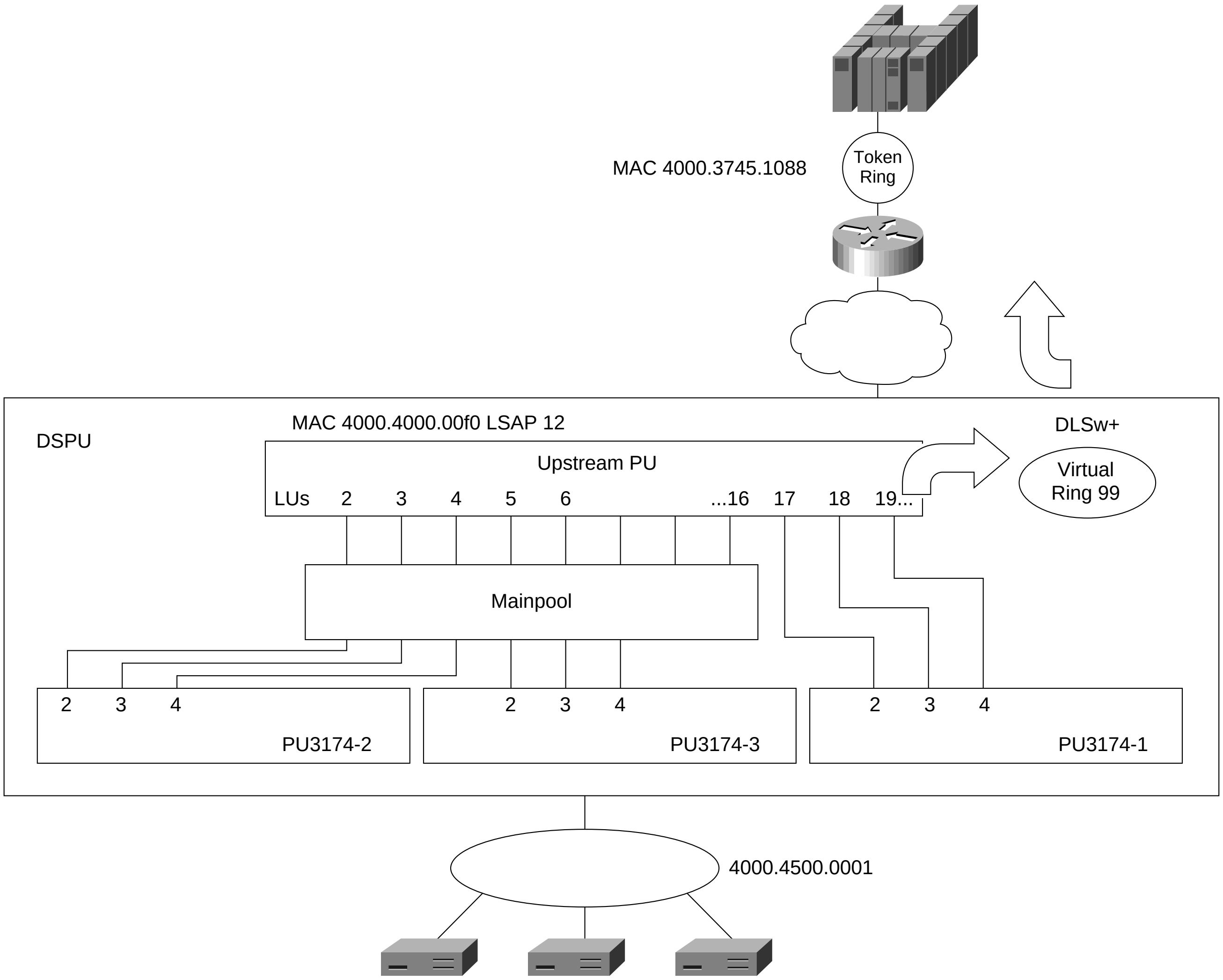
This last advantage is key in scaling a large DLSw+ network. As discussed in the chapter "Designing Hierarchical Networks," the number of PUs is a key factor in sizing your central site routers. That is because each SNA PU has a data-link control connection to its adjacent PU. DLSw+ (when using either TCP or LLC2 encapsulation) is involved in maintaining all of these data-link control connections. Under typical SNA traffic loads, to support a network with 300 branches and 9000 PUs requires three to four central site routers (depending on traffic volumes and allowing for backup). To support a 300-branch network with 300 PUs will typically require no more than two central site routers. Thus, some environments can benefit greatly from using DSPU concentration at remote sites. DSPU concentration can also be used when connecting large numbers of PUs (hundreds of PUs) in distribution sites, with the goal of scaling the network at a lower cost.

To configure DSPU, you need to do the following:

- Map DSPU to DLSw+ using the **vdlc** command
- Define the upstream PU (the one that VTAM will see) and assign it an XID, virtual MAC address, and link SAP
- Define the LU pool
- Define the downstream resources (PUs and LUs); you can either specifically map downstream resources to an upstream LU, or you can use a pool of LUs

Figure 11-2 shows a sample configuration.

Figure 11-2 DSPU Concentration Example



```
/* Configure DLSw+ */
source-bridge ring-group 99
dlsw local-peer 0 tcp 150.10.20.1
dlsw remote-peer 0 tcp 150.10.20.2
/* Configure DSPU to use VDLC upstream */
dspu vdlc 99 4000.4000.00f0/* MAC addr of dspu on virtual ring 99*/
dspu vdlc enable-host lsap 12 /* DSPU sap for upstream communications*/
/* Configure host */
dspu host fep1 xid-snd 06500001 rmac 4000.3745.1088 rsap 4 lsap 12
dspu vdlc start fep1 /* start communication to fep1 over vdlc*/
/* Configure host LUs in a pool*/
dspu pool mainpool host fep1 lu 2 16
```

The **dspu host** command defines an upstream PU with which this DSPU will communicate. The XID specified will be used when connecting with this host. The remote MAC is the MAC address used to reach this host (typically the FEP MAC).

Each upstream PU attaches to a single host. There can be multiple upstream PUs in a single router, allowing communication to multiple hosts. A single dynamic pool can be used to attach to multiple hosts.

The **dspu pool** command defines a range of host LUs in an LU pool. Dynamically defined LUs (that is, LUs allocated from the pool) contend for the available upstream LU resources.

The following configuration defines the three PUs shown in Figure 11-2:

```
/* Configure downstream PUs/LUs*/
dspu pu pu3174-1 xid-rcv 05d00001 rmac 4000.3174.0001 rsap 4 lsap 8
dspu lu 2 4 host fep1 17
dspu pu pu3174-2 rmac 4000.3174.0002 rsap 4 lsap 8
dspu lu 2 4 pool mainpool
dspu pu pu3174-3 xid-rcv 05d00003
dspu lu 2 4 pool mainpool
dspu default-pu
dspu lu 2 4 pool mainpool
/* Enable downstream PU connection*/
interface TokenRing0
rmac 4000.4500.0001
dspu enable-pu lsap 8
dspu start pu3174-1 /* activates pu3174-1*/
```

The pu3174-1 configuration statement specifies the XID, remote MAC, and remote SAP to be used by this PU. Because all three keywords are specified, a connect-in resource must match all three parameters to be associated with this definition. (Connect-in means the PU initiates the host connection, typically with a TEST frame on a LAN.) The pu3174-1 will be allocated three dedicated LUs beginning with LU 17. It will access fep1. The pu3174-2 only requires a match in the remote MAC and remote SAP. It is allocated three LUs from the pool. The pu3174-3 only requires a matching XID. It is also allocated three LUs from the pool. If a PU connects in and does not match any of these PUs, it can use the default PU definition. The pu3174-1 is the only PU that is activated by DSPU. The local SAP defined in the downstream 3174s must be 08 to communicate with the DSPU in the router and must specify MAC address 4000.4500.0001.

The MAC address specified after the interface command is the MAC address by which downstream resources will know this DSPU. If DLSw+ were downstream (instead of Token Ring, as in this example), the DSPU virtual MAC address would be specified instead.

DSPU concentration has been available since Cisco IOS Release 10.3 for Token Ring and RSRB. Support for Frame Relay requires Release 11.0. SDLC, Ethernet, and QLLC support was added in Release 11.1. Support in the same router as DLSw+ requires Release 11.1(5).

Using DLSw+ with APPN

DLSw+ has supported APPN since Release 10.3. However, running APPN and DLSw+ in the same router has only been supported since Release 11.2. This section discusses why and where APPN is required, under what circumstances you would run APPN and DLSw+ in the same router, and how to configure it.

What Is APPN

APPN is an SNA architecture that defines how peer nodes communicate. It differs from subarea SNA in several ways:

- APPN does not have a hierarchical structure; there is no concept of upstream or downstream resources, primary or secondary roles are negotiated, and all network nodes have control points
- End systems understand the network architecture and are not on the periphery or boundary of SNA; therefore, SNA Class Of Service (COS) extends to the desktop, and best paths through the network can be determined directly from the end systems
- APPN is more dynamic; both directory and topology information is determined dynamically with minimal configuration requirements
- With High-Performance Routing (HPR), APPN will dynamically reroute around link failures without disrupting SNA sessions
- APPN is more open; the APPN architecture is enhanced in an open forum (the AIW), and as a result, the routing function is available on more, lower-cost, multifunction platforms, such as multiprotocol routers

Where and Why to Use APPN

When APPN was first defined in the mid-1980s, it supported LU 6.2 applications only. Because most applications were 3270 applications, there was limited migration toward APPN in corporate networks. In VTAM 4.2, however, VTAM implemented a feature known as dependent LU server (DLUS). When used in conjunction with dependent LU requester (DLUR), this feature allows you to use APPN for any application in your network.

APPN is not for every network, but a percentage of SNA networks will implement APPN in some portion of their network. Where APPN is implemented depends on the problem you are trying to address:

- APPN in the data center

  In multihost environments, APPN allows you to reduce costs and enhance performance by minimizing your dependency on FEPs and NCP software, while migrating to a Cisco CIP. (If migrating to an IBM 3746, you may still want to use Cisco's APPN implementation in the data center for high-performance DLUR processing.) Also, APPN in the data center allows you to enhance SNA application availability by taking advantage of the capabilities of an IBM parallel sysplex complex. By limiting APPN to the data center, you minimize cost and avoid any scalability issues, while still getting the benefit of routing directly to the correct mainframe.

- APPN in the backbone

  APPN allows you to reduce your FEP costs while maintaining SNA functionality. In multiple data center environments, you can use Cisco routers with APPN functionality where you have FEPs today to allow SNA routing and COS at distribution sites. The Cisco routers cost less, are easier to maintain, support more diverse LAN and WAN media, and can optionally be used to support multiprotocol traffic. By limiting APPN to the backbone, you minimize cost and avoid any scalability issues, while still getting the benefit of routing directly to the correct data center.

- APPN to the branch

  APPN may be required at the branch if you have LU 6.2 imaging or other high-bandwidth applications as well as interactive SNA running in the same SNA device. In this case, APPN can protect existing interactive SNA traffic from bandwidth-intensive SNA applications by taking SNA COS all the way to the branch.

- APPN to offload AS/400s

  In this environment, APPN can offload network processing from your AS/400s.

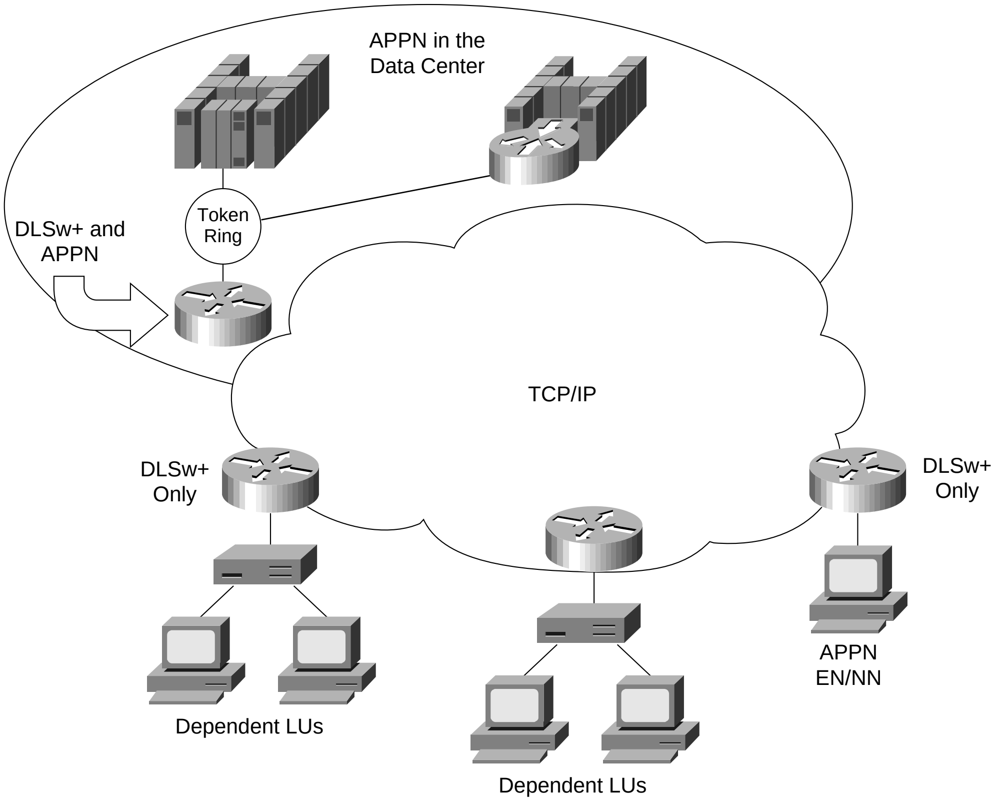
Because the focus of this design guide is DLSw+, the next few pages will describe each of the above environments and why DLSw+ may also be required. This section describes how the Cisco IOS software supports APPN and DLSw+ running in the same router. There is no specific configuration required if APPN is not running in the same router as DLSw+. DLSw+ sees the Cisco APPN router as a PU 2.1.

APPN in the Data Center

With VTAM 4.2, APPN has become a viable option for enterprise networks with legacy 3270 applications. In addition, new IBM FEPs can be purchased with native APPN support. IBM's strategy for high availability in the data center (using a parallel sysplex complex) requires APPN. As a result, many data centers in the future will migrate to APPN.

APPN in conjunction with DLSw+ is required if the data center routers support APPN and there are remote branches that access these data center routers using DLSw+, as shown in Figure 11-3.

Figure 11-3 APPN in the Data Center

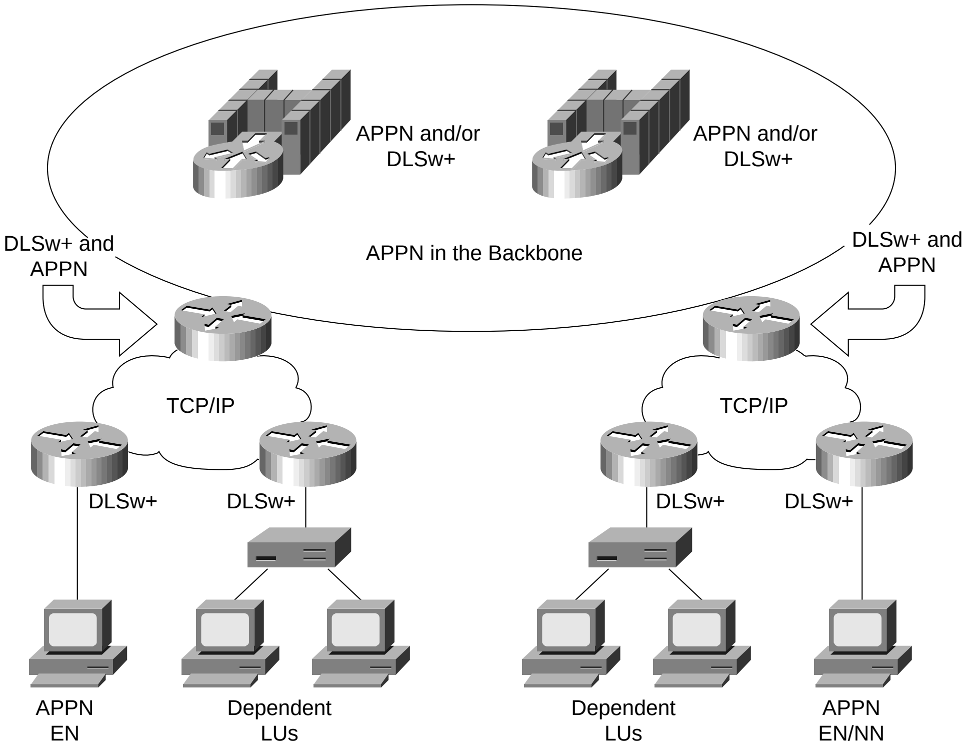


By supporting DLSw+ in the same router as APPN, you can use DLSw+ to get to the data center and you can use Cisco's APPN implementation in the data center to handle SNA routing to the correct SNA application host and to handle DLUR processing.

APPN in the Backbone

If you are migrating from a network with remote FEPs to a multiprotocol network using Cisco routers, you can use Cisco's APPN feature to provide the same SNA COS and routing that your FEPs provided. You do not need to put APPN everywhere, because you did not have SNA function everywhere before. You can use DLSw+ to connect remote sites to your APPN backbone, as shown in Figure 11-4.

Figure 11-4 APPN in the Backbone



APPN to the Branch

In the subarea world, SNA COS applies only between PU 4s and PU 5s. Outbound, unidirectional COS was added to the NCP in a later release, but a PU 2.0 in a subarea network does not understand COS. Most likely, if your SNA network performs well today, and you are not adding new, bandwidth-intensive applications, you do not need to put APPN in the branch to support COS.

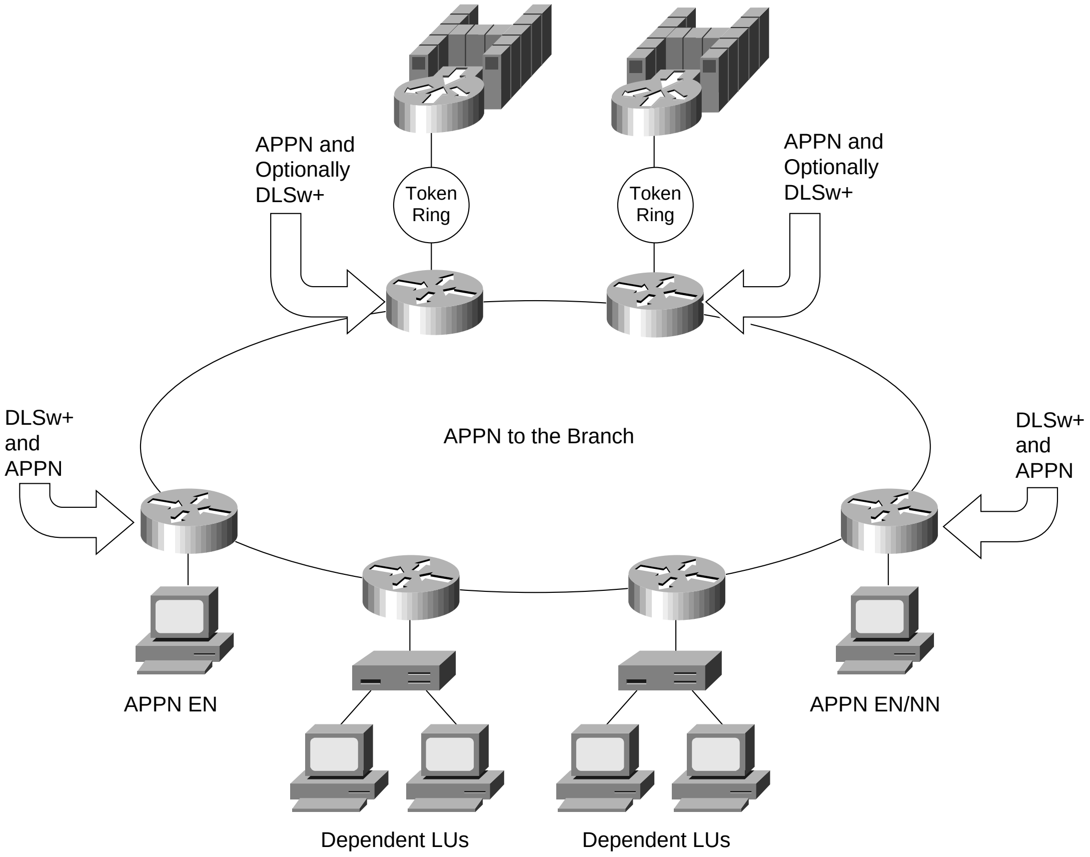
However, if you are adding new SNA applications, such as imaging applications, you may have a new requirement for providing SNA COS to the branch. APPN at the branch addresses this requirement. The Cisco IOS software supports native APPN across any transport. Because of scalability problems inherent to APPN, if you have a large network (more than 100 branch offices), you should work with Cisco consulting engineers to ensure that your network design will scale correctly.

Some environments will chose to run APPN over DLSw+ from the branch. The key reasons enterprises run APPN over DLSw+ in this environment are:

- DLSw+ provides nondisruptive rerouting around link failures and dynamic nondisruptive fall back to primary links when they recover
- DLSw+ provides DDR over circuit-switched links
- IP is the backbone protocol of choice in your network

By using custom queuing in conjunction with APPN, you can protect SNA traffic from multiprotocol traffic while concurrently prioritizing SNA traffic according to SNA COS. Figure 11-5 shows an example of APPN over DLSw+ from the branch.

Figure 11-5 APPN to the Branch



Configuration Details

To configure APPN to run over DLSw+, you need to do the following:

- Configure DLSw+
- Configure the APPN control point
- Configure APPN to transport data over VDLC and define its virtual MAC address
- Configure link stations to adjacent APPN nodes

The following is a sample configuration showing APPN over DLSw+:

```
Source-bridge ring-group 100
dlsw local-peer peer-id 172.18.3.111
dlsw remote-peer peer-id 172.18.3.125
. . .
interface token-ring 0
source-bridge 1 1 100
appn control-point NETA.NNA
 dlus NETA.CPAC
 dlur
 complete
appn port VPORT vdlc
 vdlc 100 vmac 4444.5555.6666 /* mac address of NNA
 complete
appn link-station TONNA
 port VPORT
 lan-dest-address
 1111.2222.3333 /* LAN address of adjacent NN complete
```

Currently, the Cisco IOS software supports APPN Intermediate Session Routing (ISR) only. HPR will be supported in Cisco IOS Release 11.3. However, there are other HPR implementations available, so you may need to run HPR over DLSw+ before HPR is available in the Cisco IOS software.

At this time, there are restrictions when running HPR over DLSw+ (either standard DLSw or Cisco's DLSw+). When an APPN HPR network node initializes, it brings up an LLC2 connection with its adjacent node using its APPN SAP, typically 0x04. It uses this SAP for its control point session with the adjacent node, and DLSw+ will properly handle this traffic. (It establishes a circuit denoted by the unique MAC/SAP pair.)

However, when HPR sends and receives end-system traffic, it uses connectionless unnumbered information (UI) frames with the HPR SAP, typically 0xC8. Because there is no circuit associated with this new SAP, DLSw+ does not know how to process the frames. The UI frames can either get broadcast or dropped. (There is also a MAC address cache that can send the frames to the right place, but this cache times out.)

Some end-system implementations allow the HPR SAP to match the APPN SAP. For those implementations, DLSw+ can transport HPR. The IBM OS/2 Communications Manager, however, always uses distinct SAPs, and therefore, DLSw+ will not transport HPR traffic originating from OS/2 Communications Manager.

Using DLSw+ with LAN Network Manager

IBM's LAN Network Manager is a management tool used to manage Token Ring media attachment units (MAUs) and Token Ring adapters. It uses a proprietary protocol to communicate with agent software in source-route bridges and in Cisco routers to obtain the status of the Token Ring network and to send commands to Token Ring-attached devices.

When using DLSw+ with LAN Network Manager, your LAN Network Manager displays will be more meaningful if you use the same virtual ring number everywhere.

There are no special configuration requirements to use LAN Network Manager in conjunction with DLSw+. The following configurations will work.

Local router configuration:

```
source-bridge ring-group 100
dlsw local-peer peer-id 172.18.4.001
interface TokenRing0
ring-speed 16
source-bridge 7 9 100
source-bridge spanning
```

Remote router configuration:

```
source-bridge ring-group 100
dlsw local-peer peer-id 172.18.3.111
dlsw remote-peer peer-id 172.18.4.001
interface TokenRing0
ring-speed 16
source-bridge 2 1 100
source-bridge spanning
```

Using DLSw+ with NCIA

NCIA Phase II—otherwise known as RFC 2114, DLSw Client Access Protocol—is an architecture developed by Cisco and submitted to the Internet Engineering Task Force (IETF) to address the requirement for SNA application access over IP backbones. The IETF specification refers to the architecture as the DLSw Remote Access Protocol (DRAP). NCIA differs from TN3270 because with NCIA the client actually runs SNA software. WithTN3270, the client only runs IP software. With NCIA, the SNA traffic is encapsulated in IP at the client and transported back to an NCIA server. This means that NCIA supports native 3270, LU 0, and LU 6.2 applications; has no printer limitations; requires no keyboard mapping; and supports full SNA management.

Because the client is sending only TCP/IP traffic, NCIA Phase II is an excellent solution for environments with a mix of SNA applications and a requirement to migrate to a TCP/IP backbone. NCIA Phase II clients can access NCIA servers over any TCP/IP path, including Serial Line Internet Protocol (SLIP) or Point-to-Point Protocol (PPP) dial connections. In many environments NCIA Phase II is an excellent complement to TN3270 solutions where there is an application mix. Cisco IOS Release 11.2 implements NCIA Server. Wall Data's client software is currently available.

Desktop DLSw is similar to NCIA Phase II. With desktop DLSw, client vendors support RFC 1795 and use it to encapsulate SNA in TCP/IP at the desktop. Desktop DLSw, however, has some scalability issues that are addressed by NCIA Server. Table 11-1 contrasts TN3270, NCIA, and desktop DLSw application and client support. Table 11-2 contrasts NCIA Server, desktop DLSw, and TN3270 for network design.

NCIA Server runs in a Cisco router and connects to NCIA clients using TCP/IP. From the NCIA server back to the data center, you can use DLSw+, DSPU, or RSRB. This section only describes using DLSw+. Figure 11-6 shows how an NCIA client attaches to DLSw+. NCIA Server shipped with Cisco IOS Release 11.2.

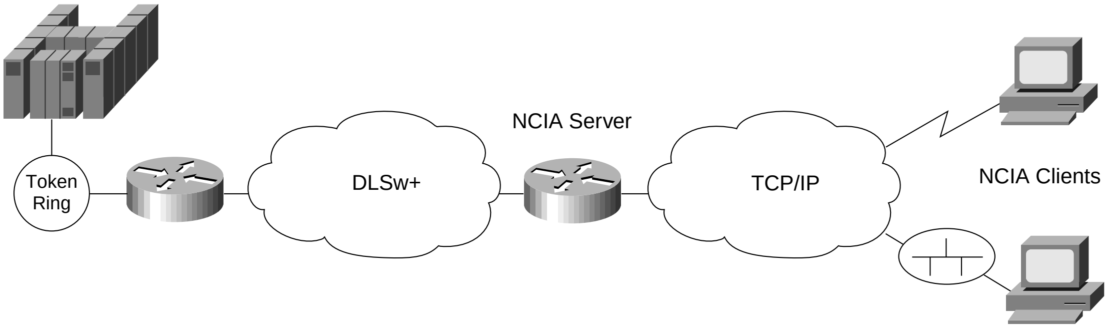
Table 11-1 Comparison of NCIA Server, Desktop DLSw, and TN3270 with Respect to Desktop and Applications

Desktop/ Application Considerations	NCIA Server	Desktop DLSw	TN3270
Stack at Desktop	SNA PU and TCP/IP	SNA PU and TCP/IP	TCP/IP
NetBIOS Support	No	Yes	No
LU 0 and LU 6.2	Yes	Yes	No
Host Types	Mainframe, PC, AS/400	Mainframe, PC, AS/400	Mainframe
Printer Stream	Depends on client	Depends on client	LU 1 and LU 3
Client Vendors	Wall Data	Eicon, Wall Data	Many
Desktop Configuration	PU plus LUs	PU plus LUs	Dynamic LUs

Table 11-2 Comparison of NCIA Server, Desktop DLSw, and TN3270 with Respect to Network Design

Network Design Considerations	NCIA Server	Desktop DLSw	TN3270
Header Overhead (in addition to TCP/IP)	12 (data)	16 (data)	0
Scalability Limit (assuming Cisco solution)	Number of PUs per router (3000-4000 clients per central site router without DSPU concentration)	Number of peers per router (200-400 clients per central site router is a practical limit)	Transaction rate and size (16,000 TN3270 clients per CIP at one transaction/ client/minute, 200 in/800 out)
Cisco Router Solution	NCIA Server	Data center DLSw+ router	CIP/7x00 with TN3270 Server
Network Size	Medium, or large when combined with DSPU	Small unless combined with DSPU in distribution sites	Large

Figure 11-6 NCIA Server Used with DLSw+

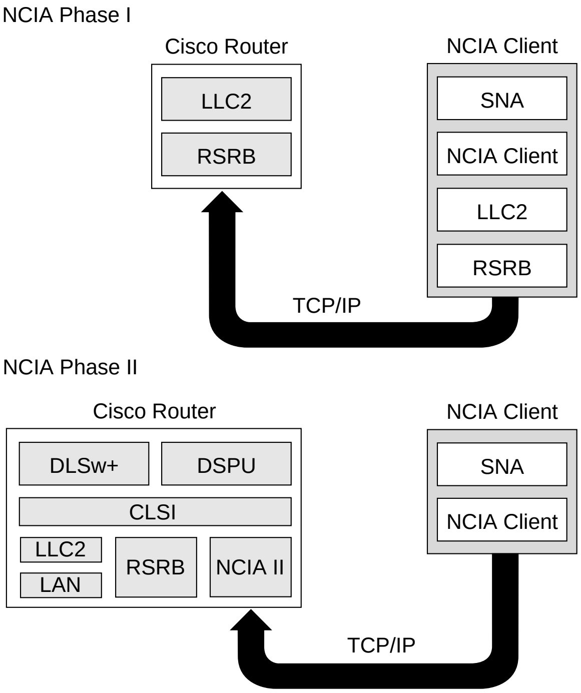


The following is a sample configuration for NCIA Server:

```
source-bridge ring-group 44
dlsw local-peer peer-id 172.22.12.130
dlsw remote-peer 0 tcp 172.22.12.131
nica server 1 172.22.12.130 1000.2000.3000 1000.1000.4000 128
!
interface Ethernet0
ip address 172.22.12.130 255.255.255.0
!
interface TokenRing0
no ip address
ring-speed 16
source-bridge 77 4 44
```

NCIA Server is defined in the NCIA Phase II architecture. An earlier Cisco architecture, implemented by several vendors (Stampede, Wall Data, and Attachmate), used Cisco's RSRB on the client. This architecture works well for small networks and works in conjunction with DSPU concentration for larger networks. NCIA Phase II simplifies configuration and enhances scalability. Figure 11-7 compares NCIA Phase I and NCIA Phase II. NCIA Phase I does not interoperate with DLSw+.

Figure 11-7 NCIA II Compared to NCIA I



APPENDIX A

Memory Estimates

This appendix provides details of DLSw+ memory utilization. This information may be useful if you are upgrading from an older version of Cisco IOS software and want to determine if you can run a newer level of software in conjunction with DLSw+.

In general, if you are installing DLSw+ in new Cisco routers, it is best to install enough memory so that you minimize your chances of having to visit remote sites. Many enterprises running Cisco 2500s at remote sites with Cisco IOS Release 11.0 will install 8 MB dynamic RAM and dual bank 8 MB Flash memory. For central site Cisco 4500s or Cisco 4700s with many peers, many enterprises choose to install the maximum amount of memory (32 MB of box memory and 16 MB of I/O memory). There are fewer Cisco 4x00s in a typical network, and the cost of the additional memory is not much of an issue.

Main Memory

The following can be used to calculate memory requirements:

Number of TCP connections x 84

+

Number of concurrent LLC2 connections x 838

+

Number of SNA cache entries x ZZ

+

Number of NetBIOS cache entries x YY

where ZZ is 178 for SNA entries in Cisco IOS Release 10.3 / 11.0 and is 234 for SNA entries in Cisco IOS Release 11.1, and where YY is 188 for NetBIOS entries in Cisco IOS Release 10.3 / 11.0 and is 244 for NetBIOS entries in Cisco IOS Release 11.1

I/O Memory

You can also estimate buffer size requirements for DLSw+ with the following formula:

Number of TCP connections x [max TCP window size[1] + (TCP queue size[2] x buffer size[3])]

+

Number of concurrent LLC2 connections x [LLC2 maximum window size x MTU]

Remember that if you specify the **priority** keyword in a **dlsw remote-peer** command, four TCP connections are established. An LLC2 corresponds to a circuit in all cases except DLSw+ local switching, where each circuit requires memory for two LLC2s. None of the above formulas account for non-DLSw+ traffic.

[1] Default is 20 K
[2] Default is 100
[3] Buffer size is size of the buffer that will fit the LAN interface MTU

DLSw+ Support Matrix

The tables in this appendix provide a description of the DLSw+ features, in what releases they are supported, and for what encapsulation types they are supported. In general, TCP/IP encapsulation provides the maximum functionality, but many features are still available if using other encapsulation types.

Table B-1 SSP (Router-to-Router) Transport Options

Media	TCP/IP	FST (Requires Release 11.1 for RSP Support)	Direct /Passthrough (Requires Release 11.1 for RSP Support)	DLSw Lite (Direct/Lack)
Serial HDLC	10.3	10.3	10.3	No
Frame	10.3	10.3(13), 11.0(9),11.1(4)	11.0	11.0
ATM	10.3	11.1(5)	No	No
FDDI	10.3	10.3(12), 11.0(9),11.1(4)	No	No
Token Ring	10.3	10.3(12), 11.0(9),11.1(4)	No	No
Ethernet	10.3	10.3	No	No
SMDS	10.3	11.0(12), 11.1(7)[1]	No	No
X.25	10.3	No	No	No
PPP	10.3	No	No	No

[1] Cisco 7500, 4500, 4000, and 2500 series routers only

Table B-2 Media Conversion Options

Media/PU Type	TCP/IP	FST (Requires Release 11.1 for RSP Support)	Direct / Passthrough	DLSw Lite (Direct/Lack)
Token Ring-to-Token Ring: PU 4/5-to-PU 2.x PU 2.1-to-PU 2.1 PU 4-to-PU 4 for single FEP RIF only PU 5-to-PU 5	10.3	10.3	10.3	11.0
Ethernet-to-Ethernet	10.3	No	No	11.0

Table B-2 Media Conversion Options (Continued)

Media/PU Type	TCP/IP	FST (Requires Release 11.1 for RSP Support)	Direct / Passthrough	DLSw Lite (Direct/Lack)
SDLC-to-SDLC: PU 4/5-to-PU 2.x/1 PU 2.1-to-PU 2.1	10.3	No	No	11.0
Token Ring-to-Ethernet	10.3	No	No	11.0
Token Ring-to-SDLC[1]	10.3	No	No	11.0
Token Ring-to-QLLC[1]	11.0	No	No	11.0
Ethernet-to-QLLC[1]	11.0	No	No	11.0
SDLC-to-QLLC[1]	11.0	No	No	11.0
SRB/FDDI-to-Token Ring/Ethernet/FDDI (Cisco 7x00 only)	11.2	No	No	11.2
TB FDDI	11.1	No	No	11.2
CIP/CSNA	11.0	11.0	11.0	11.0

[1] Media conversion support is for PU 4/5-to-PU 2.x or PU 2.1-to-PU 2.1 only

Table B-3 Features and Their Supported Transports

Features	TCP/IP	FST (Requires Release 11.1 for RSP Support)	Direct/ Passthrough	DLSw Lite (Direct/Lack)
Dynamic Peers[1]	11.1	No	No	No
SNA DDR[2]	11.1	No	No	No
Border Peers	10.3	10.3	No	No
On-Demand Peers[3]	10.3	10.3	No	No
Backup Peers	10.3 / 11.1 enhanced	10.3 / 11.1 enhanced	No	No
LNM, DSPU, or NSP over DLSw+ (both running in same router)	11.1(5)	No	No	11.1(5)
APPN over DLSw+ (both running in same router)	11.2	No	No	11.2
Payload Compression	11.1	No	No	11.1
Maps Support (MIB)	11.1(5)	11.1(5)	No	No
SNA View PU Correlation	11.1(5)	No	No	11.1(5), name to MAC/SAP pair only
Multidrop 2.1	11.0	No	No	11.0
80D5 Encapsulation (global only)	11.0	No	No	11.0

Table B-3 (Continued)Features and Their Supported Transports

Features	TCP/IP	FST (Requires Release 11.1 for RSP Support)	Direct/ Passthrough	DLSw Lite (Direct/Lack)
Bridging of IP/IPX (RSRB feature only)	No	No	No	No

[1] Configured remote peers that are only connected when required

[2] Allows remote peers to maintain a peer connection without requiring peer keepalives; this feature was designed for circuit switched networks where peer keepalives would keep the circuit up

[3] Remote peers that are dynamically learned via border peers and only connected when required

Table B-4 Features Independent of Encapsulation Type

Features	Release Level Required
Show Enhancements	11.0(10), 11.1(4)
Debug Enhancements	11.0(9), 11.1(3)
Load Balancing and Fault Tolerance Across Multiple Active Peers	10.3
Cost	10.3
Promiscuous and Passive Peers[1, 2]	10.3

[1] Peering to routers that are not preconfigured

[2] Configured remote peers for which this local peer will not initiate a peer connection

Table B-5 Local DLSw+ Media Conversion Support (For Single Router DLSw+ Configurations)

Media	Release Level Required
SDLC-to-Token Ring	11.1
SDLC-to-Ethernet	11.1
Token Ring-to-Ethernet	No
QLLC-to-SDLC	11.1
QLLC-to-LLC2	11.1
Token Ring-to-Token Ring	No