Reference Guide

DLSw+ Design and Implementation Guide

Volume 1 in the Cisco IBM Internetworking Design Guide Series

Cisco Systems

Cisco Systems

Corporate Headquarters
Cisco Systems, Inc.
170 West Tasman Drive
San Jose, CA 95134-1706
USA
www.cisco.com
Tel: 408 526-4000
800 553-NETS (6387)
Fax: 408 526-4100

European Headquarters
Cisco Systems Europe
11, Rue Camille Desmoulins
92782 Issy-les-Moulineaux
Cedex 9
France
www-europe.cisco.com
Tel: 33 1 58 04 60 00
Fax: 33 1 58 04 61 00

Americas Headquarters
Cisco Systems, Inc.
170 West Tasman Drive
San Jose, CA 95134-1706
USA
www.cisco.com
Tel: 408 526-7660
Fax: 408 527-0883

Asia Pacific Headquarters
Cisco Systems Australia, Pty., Ltd
Level 9, 80 Pacific Highway
P.O. Box 469
North Sydney
NSW 2060 Australia
www.cisco.com
Tel: +61 2 8448 7100
Fax: +61 2 9957 4350

Cisco Systems has more than 200 offices in the following countries and regions. Addresses, phone numbers, and fax numbers are listed on the

Cisco Web site at www.cisco.com/go/offices

Argentina • Australia • Austria • Belgium • Brazil • Bulgaria • Canada • Chile • China PRC • Colombia • Costa Rica • Croatia
Czech Republic • Denmark • Dubai, UAE • Finland • France • Germany • Greece • Hong Kong SAR • Hungary • India • Indonesia
Ireland • Israel • Italy • Japan • Korea • Luxembourg • Malaysia • Mexico • The Netherlands • New Zealand • Norway • Peru
Philippines • Poland • Portugal • Puerto Rico • Romania • Russia • Saudi Arabia • Scotland • Singapore • Slovakia • Slovenia • South Africa
Spain • Sweden • Switzerland • Taiwan • Thailand • Turkey • Ukraine • United Kingdom • United States • Venezuela • Vietnam • Zimbabwe

Tables

About this Guide

DLSw+ Design and Implementation Guide 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.

Cisco documentation and additional literature are available in a CD-ROM package, which ships with your product. The Documentation CD-ROM, a member of the Cisco Connection Family, is updated monthly. Therefore, it might be more current than printed documentation. To order additional copies of the Documentation CD-ROM, contact your local sales representative or call customer service. The CD-ROM package is available as a single package or as an annual subscription. You can also access Cisco documentation on the World Wide Web at www.cisco.com,www-china.cisco.com, or www-europe.cisco.com.

If you are reading Cisco product documentation on the Web, you can submit comments electronically. Click **Feedback** in the toolbar, select **Documentation**, and click **Enter the feedback form**. After you complete the form, click **Submit** to send it to Cisco. We appreciate your comments.

Intended Audience

This document is for anyone who wants to learn more about Cisco's data center solutions. It begins with an overview that is appropriate for all audiences. It also includes design guidelines and sample configurations appropriate for network designers, systems engineers, consulting engineers, and network support personnel. The document assumes familiarity with networking and Cisco routers, but does not assume mastery of either.

Examples of key configuration commands are shown to aid in understanding a particular configuration. However, this document does not contain the exact and complete configurations. This information is available and regularly updated in Cisco.com and in the Cisco product documentation. Cisco.com is Cisco's primary, real-time support system and is accessible at www.cisco.com.

Document Structure

This document contains the following chapters and appendixes:

- *Introduction*—Describes DLSw+ and how to use this manual to design and configure a DLSw+ network
- *Getting Started*—Describes the basic configuration commands required for a DLSw+ network
- *Advanced Features*—Describes advanced features of DLSw+, the benefits they provide, and a brief description of when and how to use them

- *Customization*—Describes several ways to customize your DLSw+ network
- *Bandwidth Management and Queuing*—Describes how you can use Cisco's bandwidth management and queuing features in conjunction with DLSw+ to enhance the overall performance of your network
- *Designing Hierarchical Networks*—Describes design considerations when using a hierarchical DLSw+ network
- *Designing Meshed Networks*—Describes design considerations when using DLSw+ to build a meshed, any-to-any network
- *RSRB Migration and Multivendor Interoperability*—Describes the differences between remote source-route bridging (RSRB) and DLSw+; the reasons you would migrate from RSRB to DLSw+; the migration implications in terms of management, memory, and performance as well as the steps to migrate from RSRB to DLSw+
- *Using* **show** and **debug** *Commands*—Describes how to use **show** and **debug** commands to monitor DLSw+ and to troubleshoot
- *Using CiscoWorks Blue: Maps, SNA View, and Internetwork Status Monitor*—Describes some of the enhanced network management tools available with DLSw+
- *Using DLSw+ with Other Features*—Describes how to use DLSw+ in conjunction with other Cisco IOS® Software features: SNA Switching Services (SNASw), DSPU, LAN Network Manager (LNM), and native client interface architecture (NCIA)
- *DLSw+ Ethernet Redundancy Feature*—Describes how the DLSw+ Ethernet Redundancy feature works and how to configure it and provides sample configurations
- *Memory Estimates*—Provides details of DLSw+ memory utilization
- *DLSw+ Support Matrix*—Provides a description of the DLSw+ features, in what releases they are supported, and for what encapsulation types they are supported
- *Ethernet DLSw+ Redundancy*—Discusses network design issues in a DLSw+ environment with Ethernet-attached end systems

Credits

Many people have contributed to this document. We wish to thank the following people for their contributions:

- Bob Allegretti
- Scott Bales
- Jon Beck
- Donna Kidder
- Steven Koretsky
- Debbie Morrison
- Diantha Pinner

Cisco.com

Cisco.com is Cisco's primary, real-time support channel. Maintenance customers and partners can self-register on Cisco.com to obtain additional information and services.

Available 24 hours a day, 7 days a week, Cisco.com provides a wealth of standard and value-added services to Cisco customers and business partners. Cisco.com services include product information, product documentation, software updates, release notes, technical tips, the Bug Navigator, configuration notes, brochures, descriptions of service offerings, and download access to public and authorized files.

Cisco.com serves a wide variety of users through two interfaces that are updated and enhanced simultaneously: a character-based version and a multimedia version that resides on the Web. The character-based Cisco.com supports Zmodem, Kermit, Xmodem, FTP, and Internet e-mail, and it is excellent for quick access to information over lower bandwidths. The Web version of Cisco.com provides richly formatted documents with photographs, figures, graphics, and video, as well as hyperlinks to related information.

You can access Cisco.com in the following ways:

- www.cisco.com
- www-europe.cisco.com
- www-china.cisco.com
- Telnet: cco.cisco.com
- Modem: From North America, 408 526-8070; from Europe, 33 1 64 46 40 82. Use the following terminal settings: VT100 emulation; databits: 8; parity: none; stop bits: 1; and connection rates up to 28.8 kbps.

If you are a network administrator and need personal technical assistance with a Cisco product that is under warranty or covered by a maintenance contract, contact Cisco's Technical Assistance Center (TAC) at 800 553-2447, 408 526-7209, or tac@cisco.com. To obtain general information about Cisco Systems, Cisco products, or upgrades, contact 800 553-6387, 408 526-7208, or cs-rep@cisco.com.

Introduction

This chapter describes Data-Link Switching Plus (DLSw+) and how to use this manual to design and configure a DLSw+ network. It reviews the key components of the 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 WAN. The end systems can attach to the network over Token Ring, Ethernet, Synchronous Data Link Control (SDLC), Qualified Logical Link Control (QLLC), or Fiber Distributed Data Interface (FDDI). See Appendix B, "DLSw+ Support Matrix," for details. 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, and PU 2.1-to-PU 2.1 (any supported data-link control). See Appendix B for details about DLSw+ connectivity.

Note: DLSw+ provides support for redundant paths between PU 4 devices in Cisco IOS® Release 12.0 and later with the DLSw+ RIF Passthru feature. See the "Advanced Configuration" chapter for more details. Prior to Release 12.0, 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.

DLSw Version 1 Standard

The DLSw standard was defined at the Advanced Peer-to-Peer Networking (APPN) Implementers Workshop (AIW) in the DLSw-related interest group. The DLSw Version 1 standard 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 service access points (SAPs), and the number of TCP sessions supported. Media Access Control (MAC) address lists and NetBIOS name lists can also be exchanged at this time, and if desired, a DLSw partner can specify that it does not want to receive certain types of search frames. It is possible to configure the MAC addresses and NetBIOS names of all resources that will use DLSw and thereby avoid any broadcasts. After the capabilities exchange, the DLSw partners are ready to establish circuits between SNA or NetBIOS end systems.

Establish Circuit

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

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

Cisco's DLSw+ implementation caches 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. It is important to have flow control for data traffic and because UDP Unicast handles explorer traffic, it is exempt from the flow control described here. Flow control is handled by a windowing mechanism that can dynamically adapt to buffer availability and end-station flow-control mechanisms.

Note: Cisco's DLSw+ implementation also uses TCP transmit queue depth to handle flow control.

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 Version 2 Standard

The Version 2 standard is documented in RFC 2166. It includes RFC 1795 and adds the following enhancements:

- IP multicast
- UDP unicast
- Enhanced peer-on-demand routing feature
- Expedited peer connection

Users implement DLSw+ Version 2 for scalability if they are using multivendor DLSw devices with an IP multicast network. DLSw Version 2 requires complex planning because it involves configuration changes across an IP network.

IP Multicast

Multicast service avoids duplication and excessive bandwidth of broadcast traffic because it replicates and propagates messages to its multicast members only as necessary. It reduces the amount of network overhead in the following ways:

- Avoids the need to maintain TCP SSP connections between two DLSw peers when no circuits are available
- Ensures that each broadcast results in only a single explorer over every link

DLSw Version 2 is for customers who run a multicast IP network and do not need the advantages of border peering.

UDP Unicast

DLSw Version 2 uses UDP unicast in response to an IP multicast. When address resolution packets (CANUREACH_ex, NETBIOS_NQ_ex, NETBIOS_ANQ, and DATAFRAME) are sent to multiple destinations (IP multicast service), DLSw Version 2 sends the response frames (ICANREACH_ex and NAME_RECOGNIZED_ex) via UDP unicast.

Enhanced Peer-on-Demand Routing Feature

DLSw Version 2 establishes TCP connections only when necessary and the TCP connections are brought down when there are no circuits to a DLSw peer for a specified amount of time. This method, known as peer-on-demand routing, was recently introduced in DLSw Version 2, but has been implemented in Cisco DLSw+ border peer technology since Cisco IOS Release 10.3.

Expedited TCP Connection

DLSw Version 2 efficiently establishes TCP connections. Previously, DLSw created two unidirectional TCP connections and then disconnected one after the capabilities exchange took place. With DLSw Version 2, a single bidirectional TCP connection is established if the peer is brought up as a result of an IP multicast/UDP unicast information exchange.

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.
- *Management*—Works with enhanced network management tools such as CiscoWorks Blue Maps, CiscoWorks SNA View, and CiscoWorks Blue Internetwork Status Monitor (ISM).

DLSw+ Improved Scalability

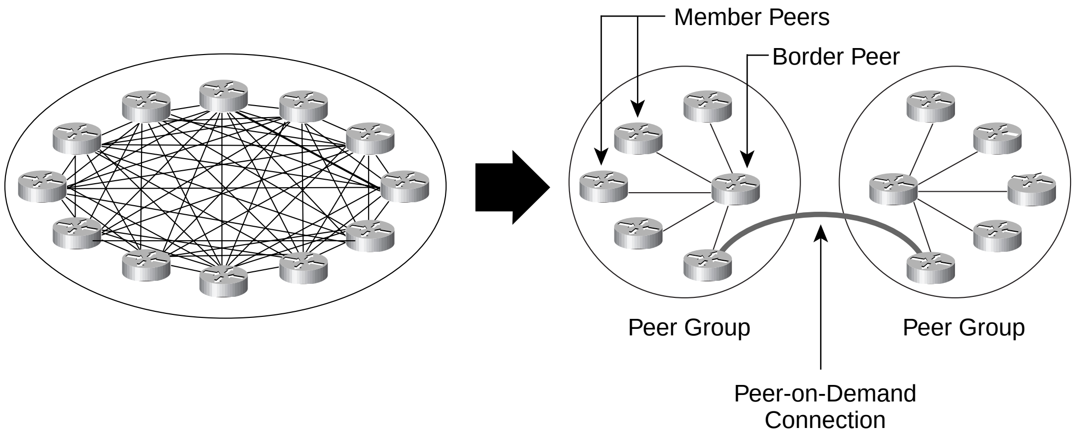
One of the most significant factors that limits the size of LAN internetworks is the amount of explorer traffic that traverses the WAN. There are several optimizations in DLSw+ to reduce the number of explorers.

Peer Group Concept

Perhaps the most significant optimization in DLSw+ is a feature known as *peer groups*. Peer groups are designed to address the broadcast replication that occurs in a fully meshed network. When any-to-any communication is required (for example, for NetBIOS or APPN environments), RSRB or standard DLSw implementations require peer connections between every pair of routers. This setup is not only difficult to configure, 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 and 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 checks its local, remote, and group cache for any reachability information before forwarding the explorer. If no match is found, the border peer forwards the explorer on behalf of the peer group member. If a match is found, the border peer sends the explorer to the appropriate peer or border peer. This setup eliminates duplicate explorers on the access links and minimizes the processing required in access routers.

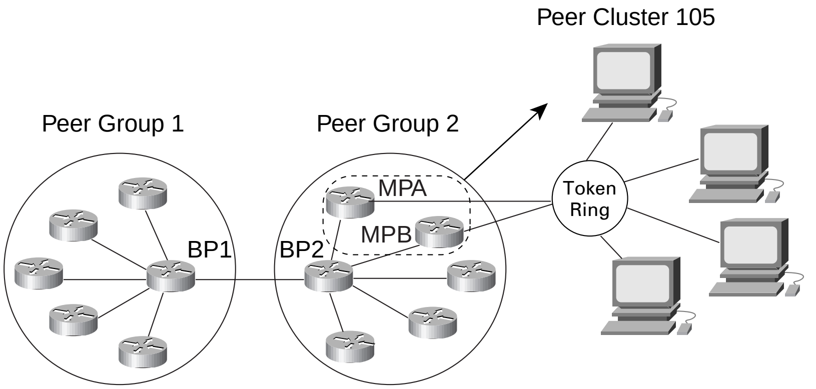
Figure 1-1   Peer Group Concept Simplifies and Scales Any-to-Any Networks



When 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.

You can further segment routers within the same peer group that are serving the same LAN into a *peer cluster.* This segmentation reduces explorers because the border peer recognizes that it only has to forward an explorer to one member within a *peer cluster* (see Figure 1-2).

Figure 1-2   Peer Cluster Feature Reduces Explorer Traffic



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. When the explorer response is received at the originating DLSw+ router, all subsequent explorers are handled based on the newly acquired information. This eliminates the start-of-day explorer storm that many networks experience.

UDP Unicast

The UDP unicast feature eliminates unnecessary congestion caused by retransmission of explorers and unnumbered information frames because it sends those frames via UDP rather than TCP. Cisco's DLSw+ introduced the UDP unicast feature prior to the release of DLSw Version 2 in Cisco IOS Release 11.2(6)F. One difference between the two enhancements is that the Release 11.2(6)F UDP unicast feature requires that a TCP connection exist before packets are sent via UDP. Because the TCP session is up and capabilities are exchanged, the peers know exclusive reachability information that permits them to further reduce the explorer load on the network. DLSw Version 2 sends UDP/IP multicast and unicast before the TCP connection exists, which further propagates explorers.

NetBIOS Dial-on-Demand Routing

To further reduce the amount of traffic on a WAN, DLSw+ filters the NetBIOS session alive packets that are sent periodically over a WAN in a dial-on-demand routing (DDR) environment. These session alive packets do not require a response, do not impede proper data flow, and keep the DDR interfaces up. NetBIOS DDR reduces the number of unwanted per-packet charges that occur in a DDR network.

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). The user configures the **sna-verify-interval** to determine the length of time a router waits before marking the cache entry stale. See the "Customization" chapter. If any peer does not respond, it is deleted from the list. This may result in reordering the cache. The user configures the **sna-cache-timeout** interval to determine the amount of time that cache entries are maintained before they are deleted. See the "Customization" chapter.

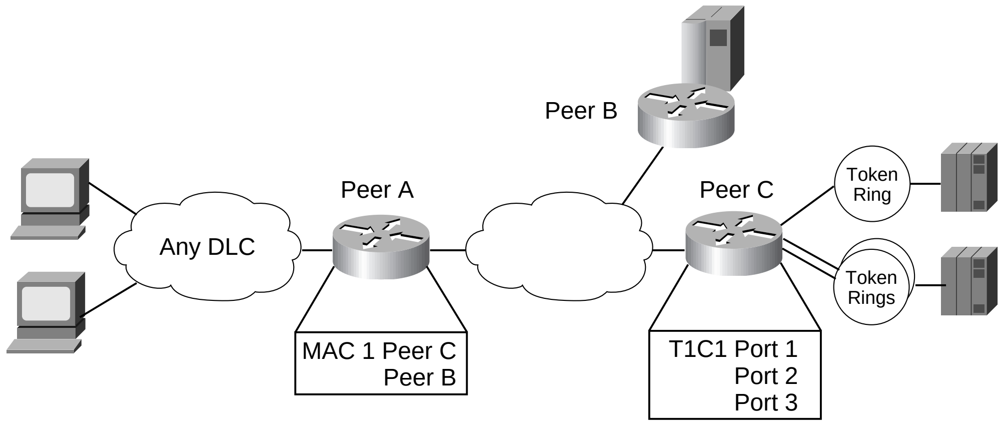
At the destination DLSw+ router, a slightly different procedure is followed using the local cache entries. If the cache entry is fresh, the response is sent immediately. If the cache entry is stale, a single route broadcast TEST frame is sent over the all ports in the cache. If a positive response is received, an ICANREACH frame is sent to the originating router. TEST frames are sent every 30 seconds The user configures these timers. See the "Customization" chapter.

Alternately, when there are duplicate paths to the destination end system, you can configure load balancing. DLSw+ alternates new circuit requests in either a round-robin or *enhanced* load balancing fashion through the list of capable peers or ports. If round-robin is configured, the router distributes the new circuit in a round-robin fashion, basing its decision on which peer or port established the last circuit. If enhanced load balancing is configured, the router distributes new circuits based on existing loads and the desired ratio. It detects the path that is underloaded in comparison to the other capable peers and assigns new circuits to that path until the desired ratio is achieved.

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 1-3. 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 1-3 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. DLSw is tied to TCP/IP. Cisco supports five 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.
- *TCP/IP with RIF Passthru*—Transports SNA and NetBIOS traffic across WANs where the RIF is not terminated. This solution allows multiple active paths between FEPs.
- *Fast Sequence Transport (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 High-Level Data Link Control (HDLC) or point-to-point Frame Relay (DLSw+ Lite) 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 Frame Relay connection where local acknowledgment and reliable transport are important, but where nondisruptive rerouting around link failures is not required. DLSw Lite uses the RFC 1490 encapsulation of Logical Link Control, type 2 (LLC2). It is a form of Direct encapsulation.

DLSw+ Modes of Operation

Cisco has been shipping IBM internetworking products for many years. There is a substantial installed base of Cisco routers running RSRB today. Therefore, it is sometimes preferable 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 caching (the ability to determine if a destination is on a LAN before sending CANUREACH frames across a WAN), explorer firewalls, and media conversion.

DLSw+ Management Features

DLSw+ supports several network management tools that enables network administrators to more easily troubleshoot and manage their network. For details on the supported applications, see the "Using CiscoWorks Blue: Maps, SNA View, and Internetwork Status Monitor" chapter.

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 "RSRB Migration and Multivendor Interoperability" chapter.

The "Using Show and Debug Commands" and "Using CiscoWorks Blue: Maps, SNA View, and Internetwork Status Monitor" chapters describe network management capabilities available with DLSw and should be read by all DLSw+ users.

The "DLSw+ Ethernet Redundancy Feature" chapter describe how to provide redundancy in an Ethernet environment. It describes how the feature works and how to configure it and provides sample configurations. If you are running an earlier release than 12.0(5)T, you should read Appendix C, "Ethernet Redundancy," for a discussion on network design issues in a DLSw+ environment with Ethernet-attached end systems.

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

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

CHAPTER 2

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

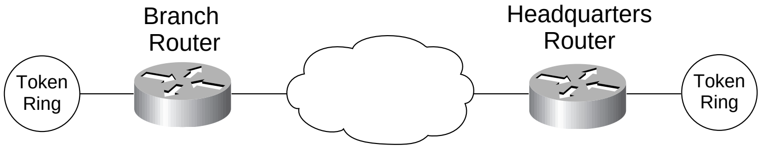
Configuring DLSw+ on most networks is not difficult. Every router that supports DLSw+ must have a **dlsw local-peer** command; **dlsw remote-peer** commands are optional, but usually at least one side of a peer connection must configure a remote peer. If a DLSw+ peer configuration omits **dlsw remote-peer** 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+.

Note: SR/TLB is required, however, when doing local translation between Ethernet and Token Ring.

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 loopback address configured via the **interface loopback 0 command** 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+:

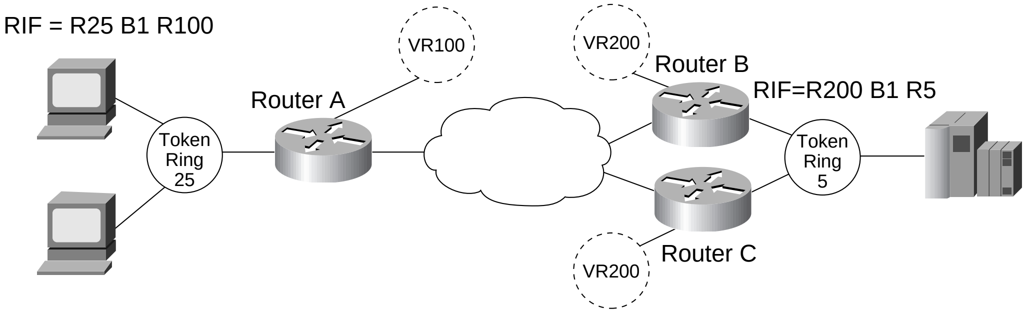
- *Token Ring*—Define a virtual ring using the global **source-bridge ring-group** command and include **a source-bridge** command that tells the router to bridge from the external Token Ring to that virtual ring on the interface
- *Ethernet*—Map a specific Ethernet bridge group to DLSw+
- *SDLC*—Define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

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 global **source-bridge ring-group** command that specifies a virtual ring number. In addition, you must include an interface **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



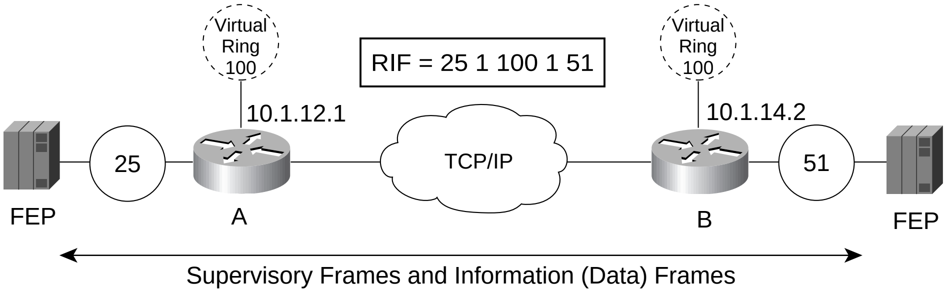
```
Configuration for Router A
source-bridge ring-group 100
dlsw local-peer peer-id 10.2.17.1
dlsw remote-peer 0 tcp 10.2.24.2
.
.
interface TokenRing0
   ring-speed 16
   source-bridge active 25 1 100
   source-bridge spanning
```

```
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 interface **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.

DLSw+ also supports TCP/IP with RIF Passthru, which means the RIF is not terminated and the entire source-route bridged path appears in the RIF. This solution is used to allow multiple active paths between FEP's. For proper configuration of the TCP/IP with RIF Passthru, the virtual ring numbers must match between the DLSw+ peers, however, the Token Ring numbers must be unique (see Figure 2-3).

Figure 2-3 DLSw+ Network Configured with TCP/IP RIF Passthru



```
Router A
source-bridge ring-group 100
dlsw local-peer peer-id 10.1.12.1
dlsw remote-peer 0 tcp 10.1.14.2 rif-passthru 100
interface tokenring 0
   ring-speed 16
   source-bridge 25 1 100

Router B
source-bridge ring-group 100
dlsw local-peer peer-id 10.1.14.2
dlsw remote-peer 0 tcp 10.1.12.1 rif-passthru 100
interface tokenring 0
   ring-speed 16
   source-bridge 51 1 100
   source-bridge spanning
```

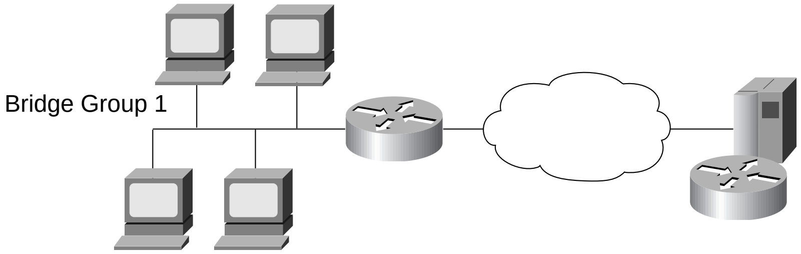
Ethernet

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 2-4, 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 IBM 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 2-4, 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 2-4 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
```

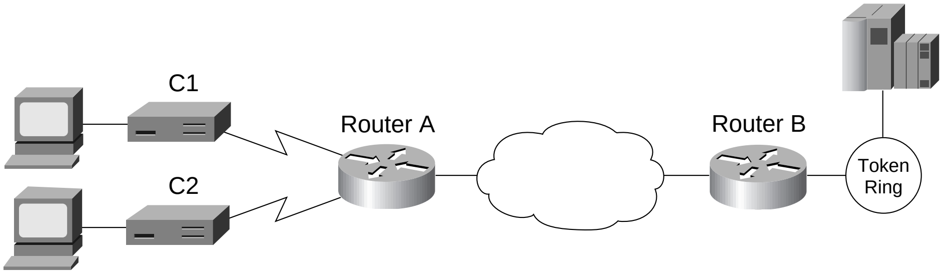
SDLC

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.

Note: With SDLC scenarios, DLSw+ is recommended over serial tunnel (STUN) except when configuring PU 4-to-PU 4 SDLC-attached devices.

In Figure 2-5, 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 2-5 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
```

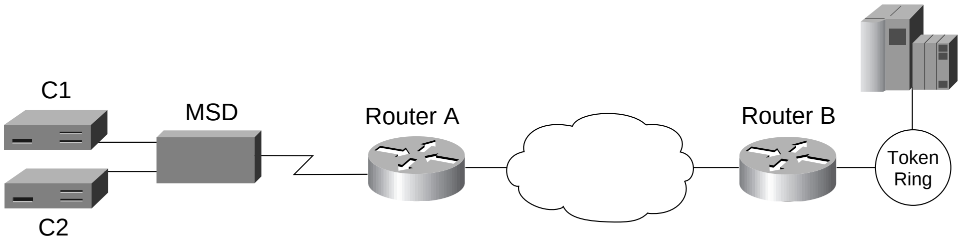
The router can be the secondary end of an SDLC line (for example, when connecting to a FEP over SDLC). In this case, specify **secondary** in the interface **sdlc role** command, and for PU 2.1 devices, specify **xid-passthru** in the interface **sdlc address** command.

In Cisco IOS Release 11.0 and later, DLSw+ supports multidrop PU 2.0/2.1. In Figure 2-6, the multidrop PU 2.0 configuration includes an interface **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 interface **sdlc role** command. For PU 2.0 devices, you must code the IDBLK and IDNUM in the interface **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 2-6 Multidrop SDLC DLSw+ Configuration



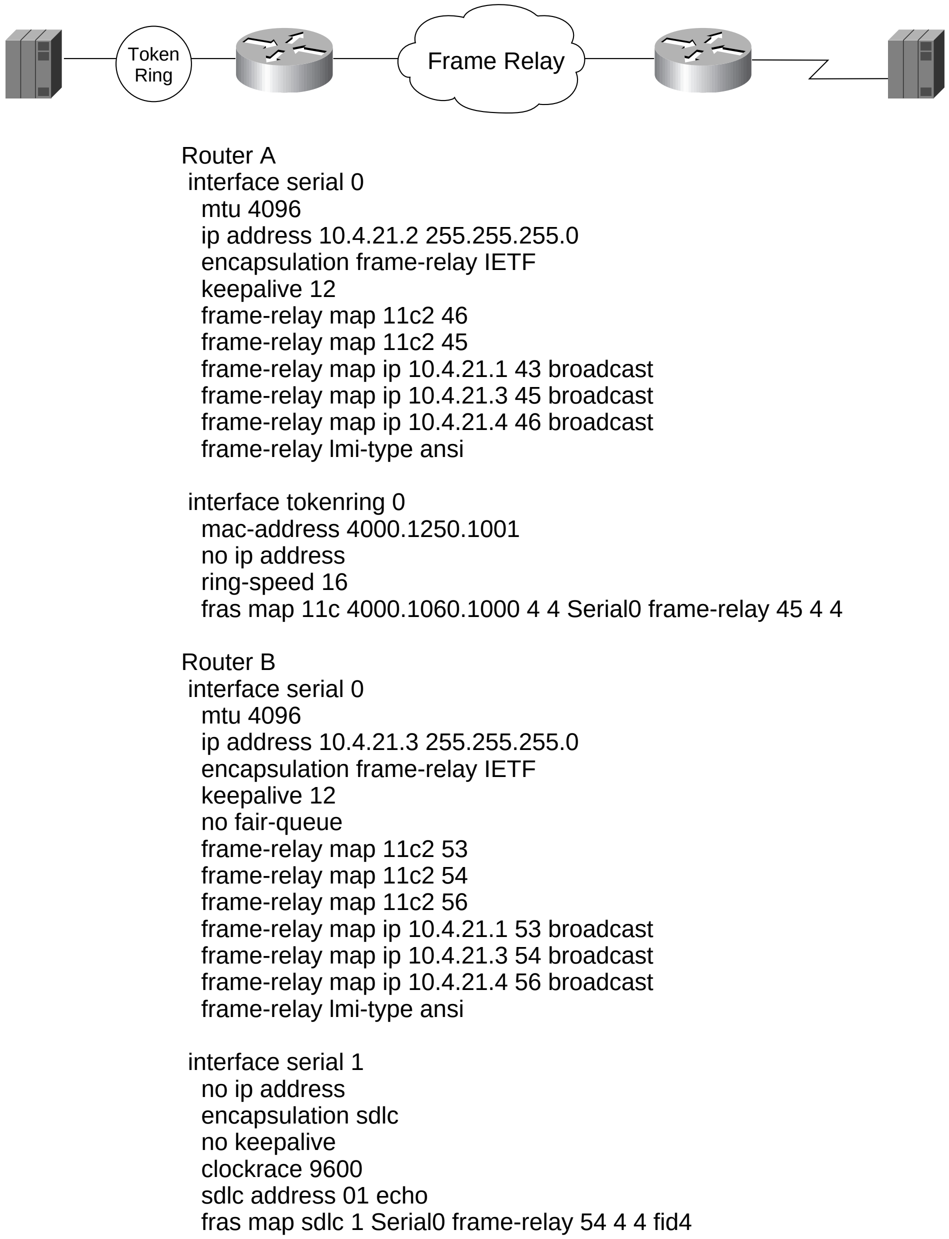
```
Configuration for Router A
Both C1 and C2 are PU 2.0

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
```

In Cisco IOS Release 11.3 and later, DLSw+ supports LLC2 to SDLC between PU 4 devices (see Figure 2-7).

Figure 2-7 LLC2-to-SDLC DLSw+ Configuration



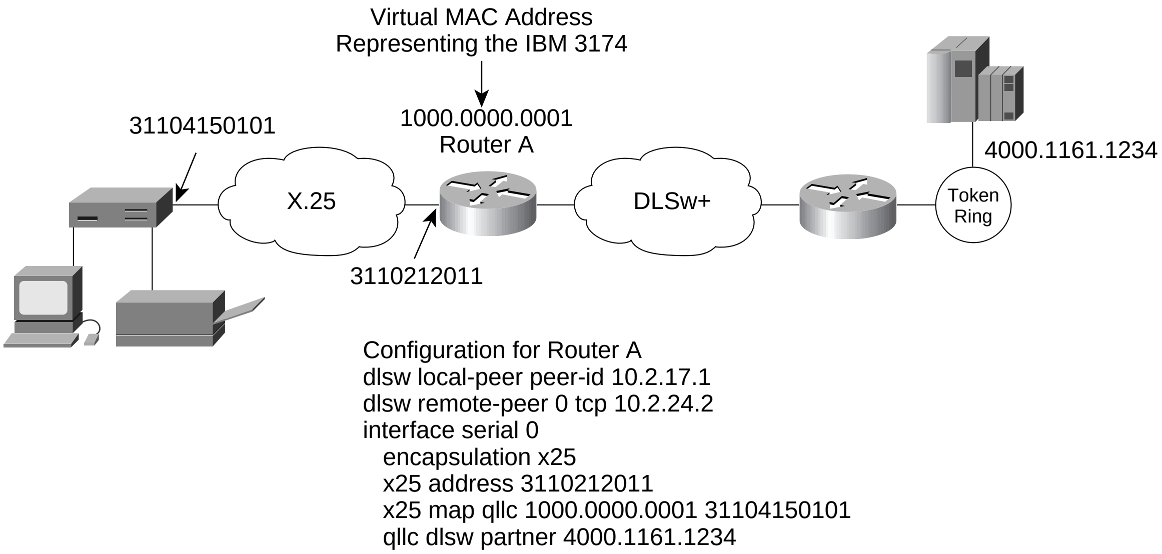
```
Router A
 interface serial 0
   mtu 4096
   ip address 10.4.21.2 255.255.255.0
   encapsulation frame-relay IETF
   keepalive 12
   frame-relay map 11c2 46
   frame-relay map 11c2 45
   frame-relay map ip 10.4.21.1 43 broadcast
   frame-relay map ip 10.4.21.3 45 broadcast
   frame-relay map ip 10.4.21.4 46 broadcast
   frame-relay lmi-type ansi

 interface tokenring 0
   mac-address 4000.1250.1001
   no ip address
   ring-speed 16
   fras map 11c 4000.1060.1000 4 4 Serial0 frame-relay 45 4 4

Router B
 interface serial 0
   mtu 4096
   ip address 10.4.21.3 255.255.255.0
   encapsulation frame-relay IETF
   keepalive 12
   no fair-queue
   frame-relay map 11c2 53
   frame-relay map 11c2 54
   frame-relay map 11c2 56
   frame-relay map ip 10.4.21.1 53 broadcast
   frame-relay map ip 10.4.21.3 54 broadcast
   frame-relay map ip 10.4.21.4 56 broadcast
   frame-relay lmi-type ansi

 interface serial 1
   no ip address
   encapsulation sdlc
   no keepalive
   clockrace 9600
   sdlc address 01 echo
   fras map sdlc 1 Serial0 frame-relay 54 4 4 fid4
```

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 2-8 shows DLSw+ being used to allow remote devices to connect to a DLSw+ network over an X.25 public packet switched network. In this example, all QLLC traffic is addressed to destination address 4000.1161.1234, which is the MAC address of the FEP. The remote X.25-attached IBM 3174 is given a virtual MAC address of 1000.0000.0001. This virtual MAC address is mapped to the X.121 address of the IBM 3174 (31104150101) in the X.25-attached router.

Figure 2-8 QLLC DLSw+ Configuration to a Single LAN-Attached Upstream Device



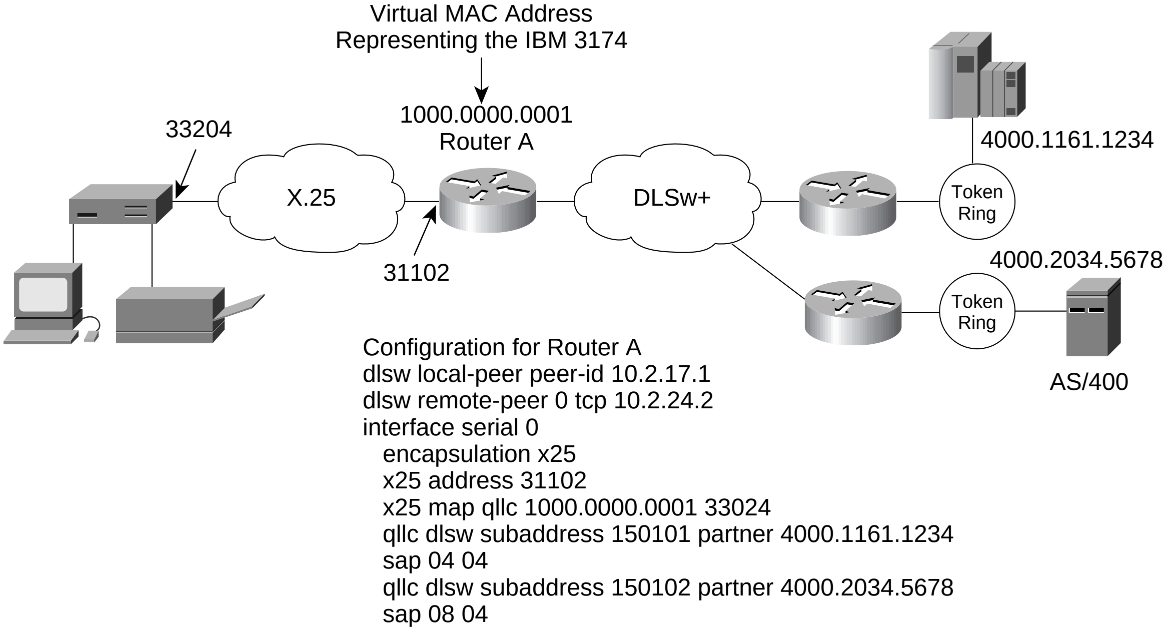
```
Configuration for Router A
dlsw local-peer peer-id 10.2.17.1
dlsw remote-peer 0 tcp 10.2.24.2
interface serial 0
   encapsulation x25
   x25 address 3110212011
   x25 map qllc 1000.0000.0001 31104150101
   qllc dlsw partner 4000.1161.1234
```

In Figure 2-9, a single IBM 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 IBM 3174 appears to the FEP as a Token Ring-attached resource with MAC address 1000.0000.0001. The IBM 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 IBM 3174 appears to the AS/400 as a Token Ring-attached resource with MAC address 1000.0000.0001. The IBM 3174 uses a source SAP of 08 when communicating with the AS/400.

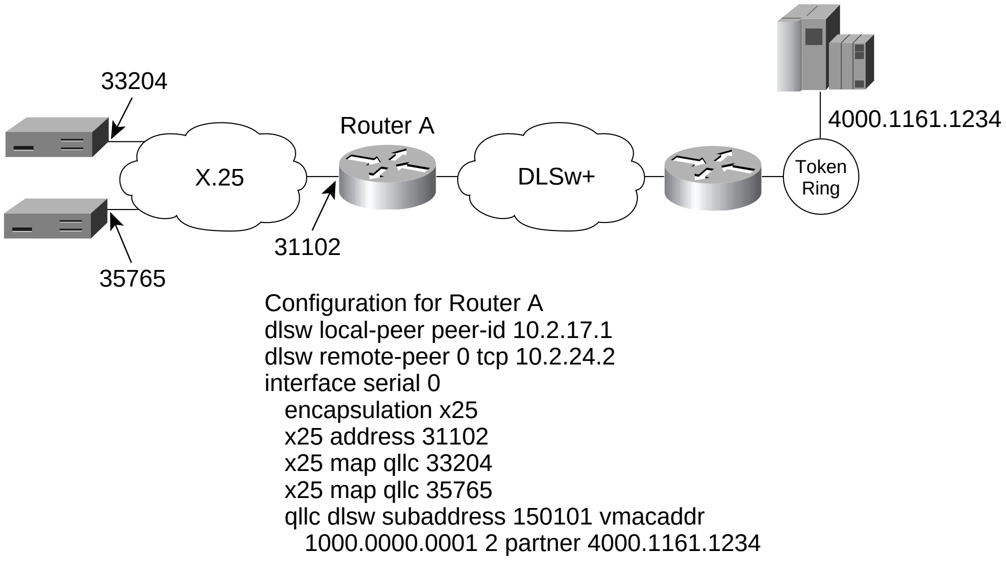
Figure 2-9 QLLC DLSw+ Configuration for Support of Multiple Upstream LAN-Attached Devices



```
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 33024
   qllc dlsw subaddress 150101 partner 4000.1161.1234
   sap 04 04
   qllc dlsw subaddress 150102 partner 4000.2034.5678
   sap 08 04
```

In Figure 2-10, two X.25 resources want to communicate over X.25 to the same FEP. In the router attached to the X.25 network, every X.25 connection request for X.121 address 31102150101 is directed to DLSw+. The interface **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 2-10 QLLC DLSw+ Configuration Supporting 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 when and how to use them. Use this chapter to determine which options you want to implement 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 and how they affect the different phases of operation.

Load balancing in these cases do 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 or unless **dynamic** is specified and a peer will wait until it has traffic to send). The routers then exchange their capabilities. Included in the capabilities exchange are any resources configured in the global **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 the type of load balancing, if any, is specified.

Fault-Tolerant Mode

If load balancing is not specified, DLSw+ handles multiple paths in fault-tolerant mode. In normal operations, a 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:

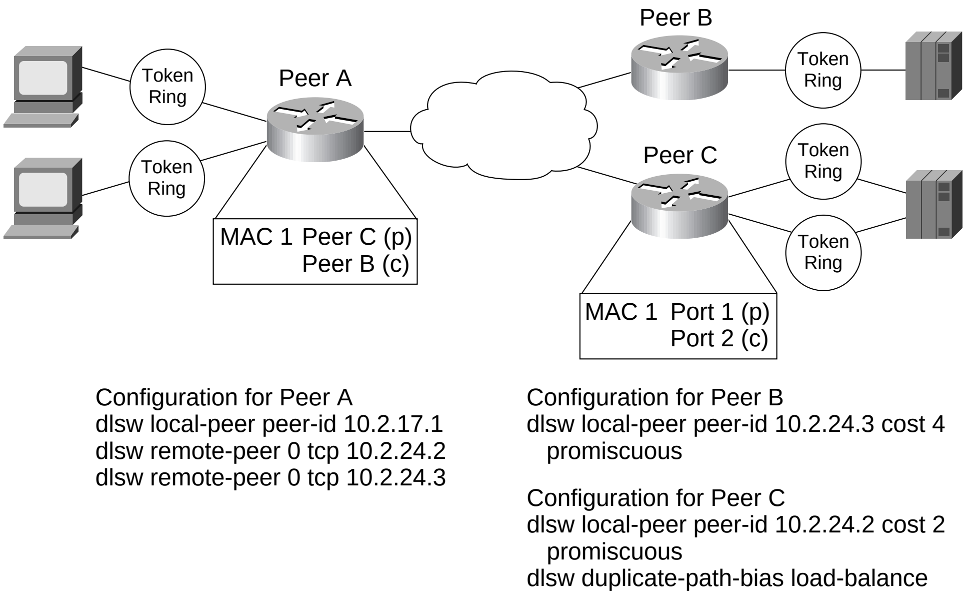
- Peer from which the first positive response was received
- Peer with the least cost
- Port over which the first positive response was received

Configuring Cost

Although configuring cost on a peer is not required, it can be used to control which path the sessions use. 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 cost configured on the **dlsw remote peer** command overrides any cost value learned from another devices **dlsw local peer** command.

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 source-route bridging (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.

Note: The output of the **show dlsw capabilities** command displays the cost learned from the other devices rather than what is actually configured on the local peer. For example, the output of the **show dlsw capabilities** command on Peer A will show a cost value of 4 for remote peer B and a cost value of 2 for remote peer C. To determine the cost configured on the local device, issue the **show running configuration** command on the local peer.

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.

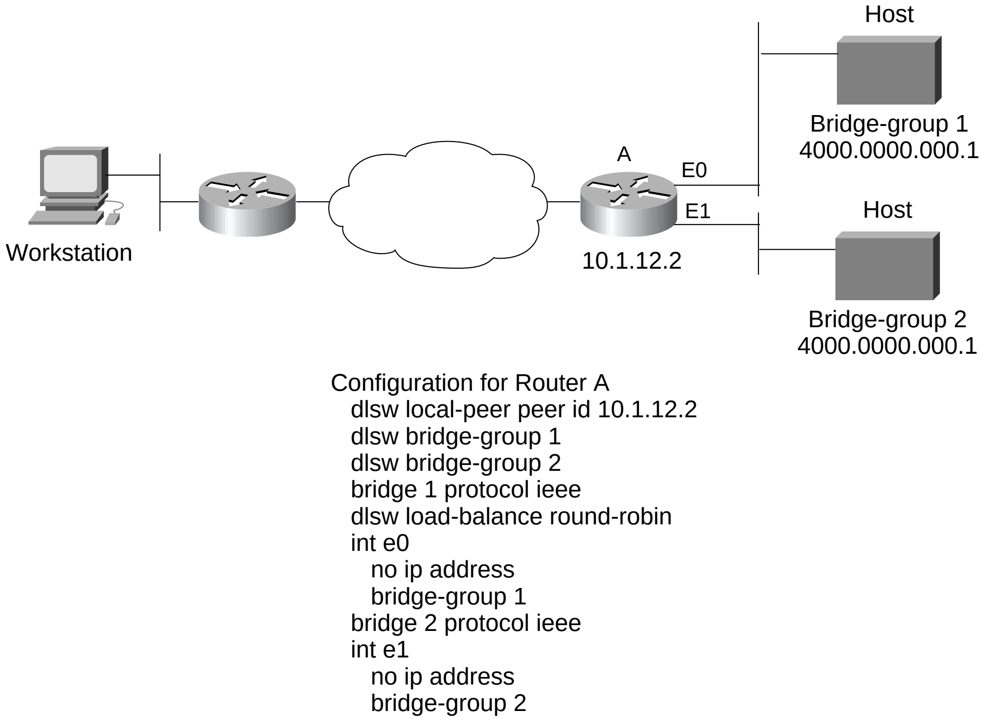
Peer A will forward a positive response to the SNA PU and then establish an end-to-end circuit using Peer C. Peer C is selected because Peer C has a lower cost specified. When the next PU attempts to set up a connection to the same MAC address, it will be set up using Peer C, if available.

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 the **round-robin** keyword in the **dlsw load-balance** command. Each new SNA PU will use the next path in the list in a round-robin fashion. See the "Load Balancing Mode" section for more details.

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). On Ethernet, however, duplicate MAC addresses for channel gateways will only work if you have a unique bridged Ethernet segment that are not bridged together, either in the router (by putting them in the same bridge group) or by a separate bridge. (Token Ring networks can rely on SRB for loop prevention.) You can locally load balance if the Ethernet segments are not bridged together in the router (by putting them in the same bridge group) or by a separate bridge group. This configuration is especially beneficial when you want to use duplicate network interface card (NIC) addresses with Ethernet. In 3-2, Router A has two reachability cache entries for MAC address 4000.000.0001 pointing to different bridge groups. Router A will load balance link-establish requests in round-robin mode because it is configured with the **dlsw load-balance round-robin** command.

Figure 3-2 DLSw+ Doing Local Load Balancing with Duplicate NIC Addresses in an Ethernet Environment

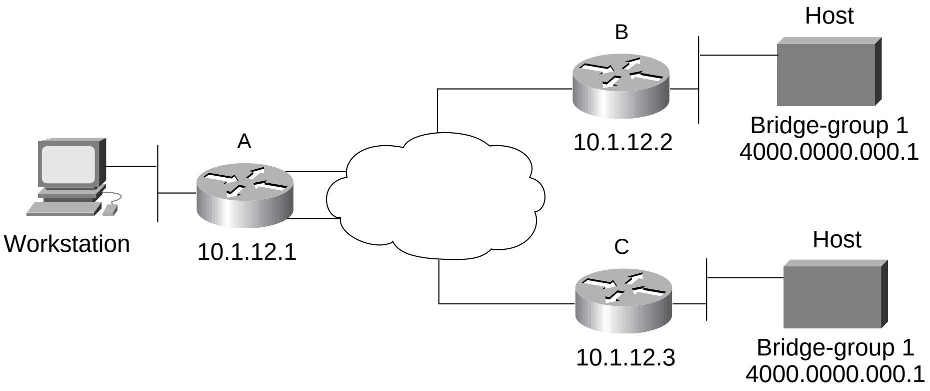


```
Configuration for Router A
   dlsw local-peer peer id 10.1.12.2
   dlsw bridge-group 1
   dlsw bridge-group 2
   bridge 1 protocol ieee
   dlsw load-balance round-robin
   int e0
      no ip address
      bridge-group 1
   bridge 2 protocol ieee
   int e1
      no ip address
      bridge-group 2
```

Alternately, if you are running Cisco IOS Release 12.1 or later, you can enable the Ethernet Redundancy feature. See the “Ethernet Redundancy” chapter.

You can do remote load balancing with duplicate NIC addresses in an Ethernet environment if the host devices with the duplicate NIC address are not sharing the same Ethernet LAN segment. In 3-3, Router A has two reachability cache entries for MAC address 4000.000.0001 pointing to different peer addresses. Note that the bridge groups are the same.

Figure 3-3 DLSw+ Doing Remote Load Balancing with Duplicate NIC Addresses in an Ethernet Environment

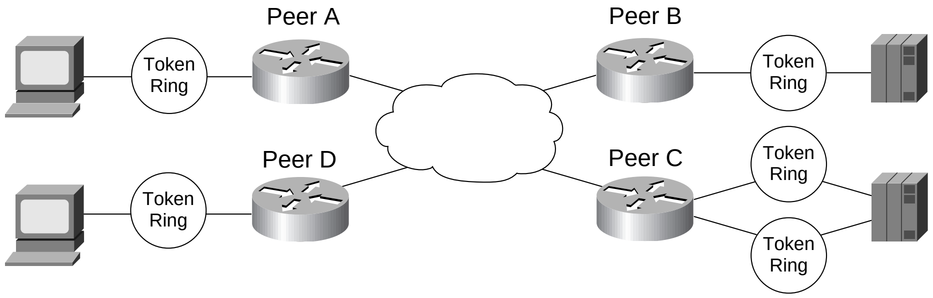


```
Configuration for Router A
   dlsw local-peer peer id 10.2.12.1
   dlsw remote-peer 0 tcp 10.1.12.2 circuit weight 10
   dlsw remote-peer 0 tcp 10.1.12.3 circuit weight 10
   dlsw remote-peer 0 tcp 10.2.20.1
   dlsw load-balance circuit-count
   dlsw timer explorer-wait-time 100

Configuration for Router B
   dlsw local-peer peer id 10.1.12.2 cost 2
   dlsw bridge-group 1
   bridge 1 protocol ieee
   interface e1
      no ip address
      bridge-group 1

Configuration for Router C
   dlsw local-peer peer id 10.1.12.3 cost 2
   dlsw bridge-group 1
   bridge 1 protocol ieee
   interface e1
      no ip address
      bridge-group 1
```

Router A will load balance equally between Peer B and C because the **dlsw load-balance circuit-count** command is configured and because equal values are specified in the **circuit-weight** of the **dlsw remote-peer** commands. See the "Load Balancing" section of this chapter for more details on how to configure load balancing.

An alternate way to specify cost is to use the **dlsw remote-peer** command as shown in Figure 3-4. 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 3-4, Peer A always favors Peer B and Peer D always favors Peer C.

Figure 3-4 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 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 B
dlsw local-peer peer-id 10.2.24.2
  promiscuous

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

Load Balancing Mode

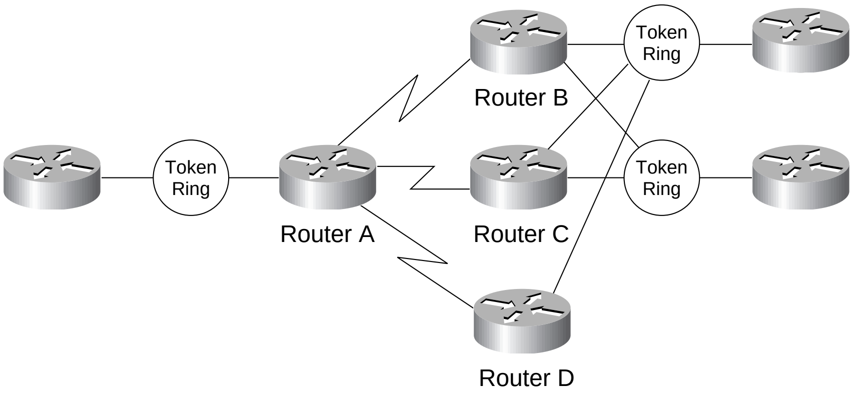
If the **dlsw load-balance** command is configured, DLSw+ load balances multiple paths based on whether the **round-robin** or **circuit-count keyword** is selected. If **round-robin** is specified, a peer distributes new circuits in a round-robin fashion to the capable peers (peers that have the lowest or equal cost specified) in the cache. DLSw+ load balances between two paths with the lowest cost if they are the lowest known. For example, in Figure 3-5, suppose the user wants to load balance between Routers B and C and that each are configured with a cost of 3. If Router D has a cost of 2, however, all circuits will establish through Router D because it is the lowest known cost.

In Figure 3-5, a workstation on the left sends an explorer packet to Peer A. 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 and Peer C receive a positive reply to the explorer and send a positive response back to Peer A. Peer A records that it has two peers it can use to reach the MAC address of the SNA device.

Peer A forwards a positive response to the workstation and then establishes an end-to-end circuit using Peer C or Peer B (depending on the first response received). Peer A distributes any new circuits between Peer B and Peer C in a round-robin fashion.

If, for example, Router B fails, all SNA sessions are terminated and reestablished through Router C. When Router B becomes available again, the sessions remain active on Router C (despite Router B's recovery.) Router A distributes any new circuits in a round-robin mode between Router B and Router C.

Figure 3-5 DLSw+ with Enhanced Load Balancing/Round-Robin Mode



```
Router A
  dlsw local-peer peer-id 10.2.19.1
  dlsw remote-peer 0 tcp 10.2.24.2
  dlsw remote-peer 0 tcp 10.2.19.5
  dlsw remote-peer 0 tcp 10.2.20.1
  dlsw load-balance round-robin
  dlsw timer explorer-wait-time 100

Router B
  dlsw local-peer peer-id 10.2.24.2 cost 1 promiscuous

Router C
  dlsw local-peer peer-id 10.2.19.5 cost 1 promiscuous

Router D
  dlsw local-peer peer-id 10.2.20.1 promiscuous
```

Note: In Cisco IOS Release 12.0(3)T, the **dlsw load-balance** command replaced the **dlsw duplicate-path-bias load-balance** command. Although the **dlsw duplicate-path-bias load-balance** command continues to be accepted, it is converted to the new command if the configuration is displayed or saved. As with the **dlsw duplicate-path-bias load-balance** command, how the cache entries are used depends on which keyword (**round-robin** or **circuit count**) is specified.

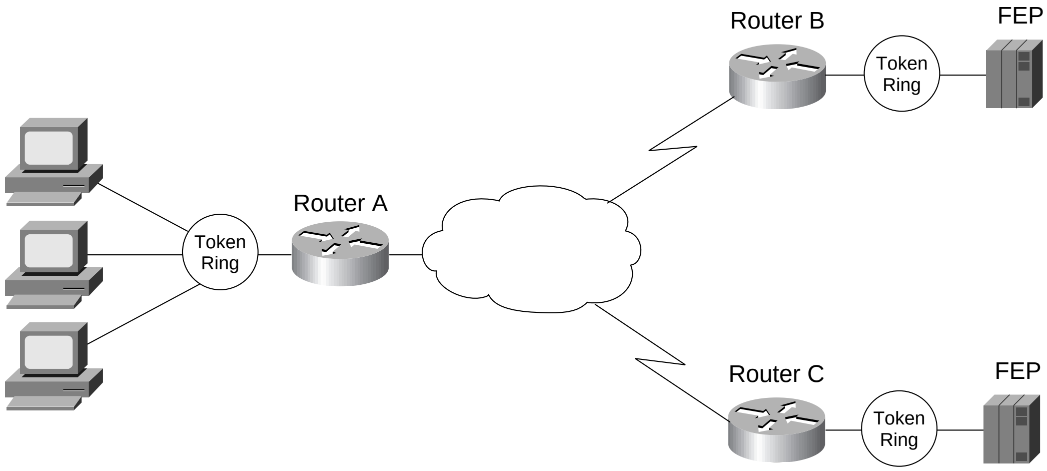
If **circuit count** is specified, then the Enhanced Load Balancing feature is configured. Each new circuit gets distributed based on existing loads and the desired ratio. The user assigns a circuit-weight value to the local peer and to each remote capable peer to create a desired ratio among the peers. The DLSw+ Enhanced Load Balancing feature calculates the difference between the desired and the actual ratio of circuits being used on a peer. It detects the path that is underloaded in comparison to the other capable peers and assigns new circuits to that path until the desired ratio is achieved.

The Enhanced Load Balancing feature load balances among peers and local ports; however, the circuit weight can only be applied to peers. DLSw+ distributes the traffic among local ports in a round-robin fashion, even if the user configures them for circuit weight.

In Figure 3-6, Router A, B, and C have a *circuit weight* of 10. In this case, there is 1:1 ratio between Router B and Router C and therefore, Router A knows that Router B and Router C should be handling the same number of circuits. Router A distributes the circuits evenly between Router B and Router C.

If, for example, Router B fails, all SNA sessions are terminated and reestablished through Router C. When Router B becomes available again, the sessions remain active on Router C (despite Router B’s recovery.) Router A distributes any new circuits to router B until it has the same number of circuits as Router C, achieving the 1:1 ratio with Router C. Compare this to the round-robin method where Router A would alternate the new circuits between Router B and Router C, resulting in an imbalance in circuit distribution.

Figure 3-6 DLSw+ with Enhanced Load Balancing/Circuit Count Mode



```
Router A
   dlsw local-peer peer-id 10.2.19.1
   dlsw remote-peer 0 tcp 10.2.24.2 circuit weight 10
   dlsw remote-peer 0 tcp 10.2.20.1 circuit weight 10
   dlsw load-balance circuit-count
   dlsw timer explorer-wait-time 100

Router B
   dlsw local-peer peer-id 10.2.24.2 cost 1 promiscuous

Router C
   dlsw local-peer peer-id 10.2.19.5 cost 1 promiscuous
```

Suppose the configuration was the following:

```
Router A
dlsw local-peer peer-id 10.2.19.1
dlsw remote-peer 0 tcp 10.2.24.2 circuit weight 20
dlsw remote-peer 0 tcp 10.2.20.1 circuit weight 10
dlsw load-balance circuit-count
dlsw timer explorer-wait-time 100

Router B
dlsw local-peer peer-id 10.2.24.2 cost 1 promiscuous

Router C
dlsw local-peer peer-id 10.2.19.5 cost 1 promiscuous
```

In this case, there is 2:1 ratio between Router B and Router C because they are configured with a circuit-weight of 20 and 10, respectively. Router A learns that Router B should be handling twice as many circuits as Router C. Router A checks how many circuits it has with each peer and makes its decision based on a 2:1 ratio.

If, for example, Router B fails, all SNA sessions are terminated and reestablished through Router C. When Router B becomes available again, the sessions remain active on Router C (despite Router B's recovery.) Router A distributes any new circuits to Router B until it achieves the 2:1 ratio between Router B and Router C.

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 timers explorer-wait-time** command causes DLSw+ to wait the specified amount of time 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.

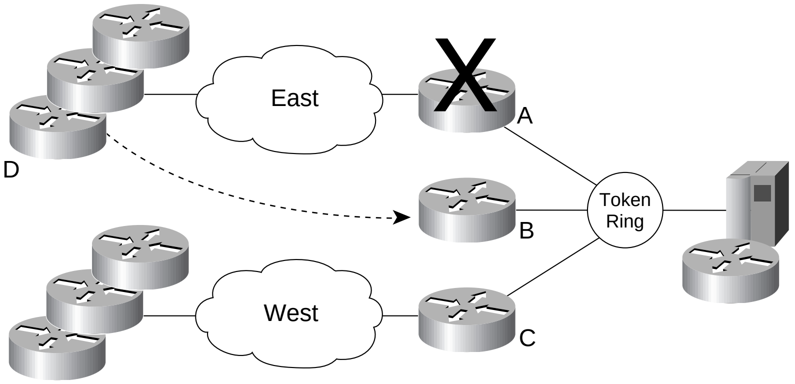
Backup Peers

Failure

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 3-7 illustrates how to configure a backup peer. To use backup peers, any encapsulation method used to access the primary peer will work.

Figure 3-7 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

Configuration for Router A
dlsw local-peer peer-id 10.2.24.2 promiscuous

Configuration for Router B
dlsw local-peer peer-id 10.2.24.3 promiscuous

Configuration for Router E
dlsw local-peer peer-id 10.2.18.1 promiscuous
dlsw remote-peer 0 tcp 10.2.24.5 backup 10.2.24.2----➤ Router B to backup Router C
```

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 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 reestablish through Router B. When Router A becomes available again, all new sessions are established through Router A, but sessions active on Router B remain on Router B until the linger timer expires. No new circuits are brought up on Router B during the linger period. Setting the **linger** keyword to 0 causes sessions on Router B to remain active until they terminate on their own. Omitting the **linger** keyword causes all sessions to drop immediately from Router B as soon as Router A recovers. 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.

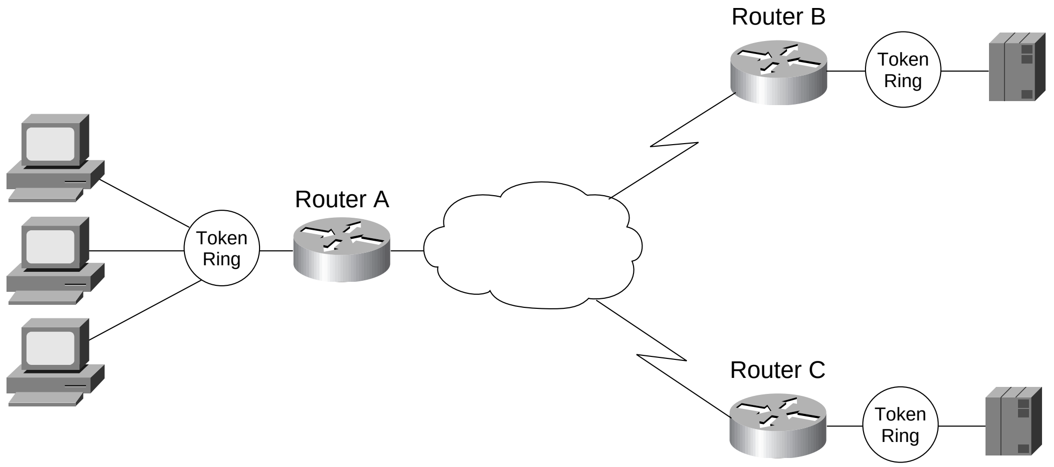
By using a backup peer statement in each remote branch instead of concurrently peering to two routers, a single backup router at a central site can easily back up any other central site router. There is no work on a backup router until a primary router fails.

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.

Summary of Availability Options

DLSw+ provides several options to enhance availability: load balancing, redundancy, and backup peers. Each option affects the distribution of circuits depending on the phase of operation (start-up, normal, failure, recovery from failure). Table 3-1 summarizes each availability option in DLSw+ and its effect on circuit distribution during the different phases of operation. The coordinate values (x/y) represent the number of circuits on Router B and Router C (Router B/Router C) in Figure 3-8.

Figure 3-8 Sample DLSw+ Configuration for Table 3-1



```
Router A
   dlsw local-peer peer-id 10.2.19.1
   dlsw remote-peer 0 tcp 10.2.24.2 circuit weight 10
   dlsw remote-peer 0 tcp 10.2.20.1 circuit weight 10
   dlsw load-balance circuit-count
   dlsw timer explorer-wait-time 100

Router B
   dlsw local-peer peer-id 10.2.24.2 cost 1 promiscuous

Router C
   dlsw local-peer peer-id 10.2.19.5 cost 1 promiscuous
```

Table 3-1 Sample Circuit Distribution in DLSw+

	Start up (Initial 8 Circuits)	Failure (Number of Circuits When Router A Fails)	Failure (Distribution of Recovered Circuits)	Recovery from Failure (Number of Circuits After Router A Recovers)	Normal (8 New Circuits)	Normal (8 New Circuits)
Backup Scenario	8/0	0/0	0/8	8/0[1]	16/0	24/0
Fault-Tolerant Mode	8/0[2]	0/0	0/8	0/8	8/8	16/8
Load Balancing Round-Robin	4/4	0/4	0/8	0/8	4/12	8/16
Enhanced Load Balancing Circuit Count	4/4	0/4	0/8	0/8	8/8	12/12
Cost Configured	8/0	0/0	0/8	0/8	8/8	16/8

[1] If linger option is set to 0, the 8 circuits will stay with Router B.

[2] Assuming Router A is the first path in the cache

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, TCP/IP with RIF Passthru, 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 offers nondisruptive rerouting around link failures. With TCP encapsulation, you can take advantage of DDR 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 Series 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 milliseconds 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. See Appendix B, “DLSw+ Support Matrix,” for supported media types. Load balancing across multiple WAN links or IP paths is possible because TCP resequences traffic.

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

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

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

TCP/IP with RIF Passthru Encapsulation

TCP/IP with RIF Passthru is an option used to support multiple active paths between FEPs. It disables local acknowledgment and provides the packet with a complete end-to-end RIF, which replaces any state-based information that is normally required by DLSw+ to route packets. It is process switched and offers nondisruptive rerouting around link failures.

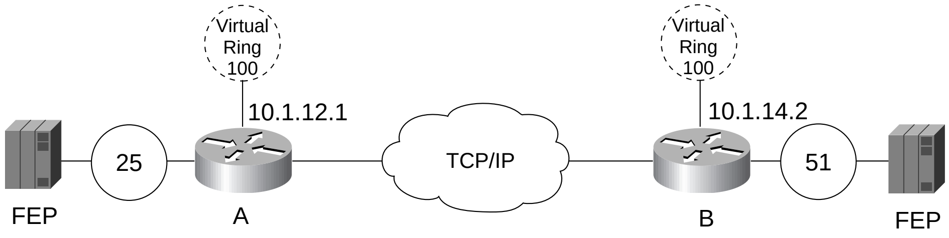
It is supported only when the end systems reside on Token Ring and are configured for SRB. See Appendix B, “DLSw+ Support Matrix,” for details.

TCP/IP with RIF Passthru supports activating a FEP via an NCP if the image is already loaded on its internal disk; otherwise, remote load is not supported. The following features are not supported with the DLSw+ RIF Passthru feature:

- Border peers
- Peer-on-demand peers
- Dynamic peers
- Backup peers

Prior to this feature, the design in Figure 3-9 was not supported:

Figure 3-9 Unsupported DLSw+ Configuration Prior to TCP/IP with RIF Passthru



The following is a sample **dlsw remote-peer** command specifying TCP/IP with RIF Passthru encapsulation:

```
dlsw remote-peer 0 tcp 10.2.24.5 rif-passthru 100
```

The virtual ring numbers must match between the DLSw+ peers. The Token Ring numbers, however, should be uniquely different throughout the network

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 does not support all media types. See Appendix B "DLSw+ Support Matrix" for details. FST will reroute around link failures, but rerouting may be disruptive because of LLC2 time-outs. 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 does not support all media types. See Appendix B, "DLSw+ Support Matrix," for details. 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
int s1
 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 only 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 does not support all media types. See Appendix B, "DLSw+ Support Matrix," for details. DLSw Lite is process switched and processes approximately the same traffic volume as TCP encapsulation.

With DLSw Lite, link failures are disruptive. Availability can be achieved by having multiple active central site peers, which allows for dynamic, but disruptive, recovery from the loss of either a link or a central site peer. Backup peers are 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
int s1
 frame-relay map llc2 33
```

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:

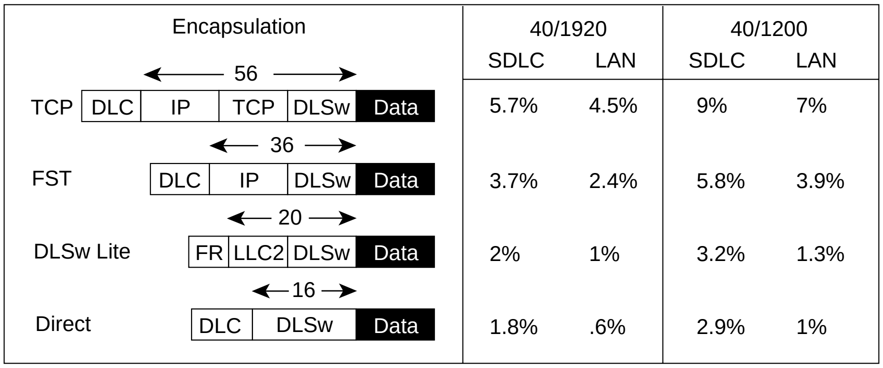
```
dlsw remote-peer 0 frame-relay interface serial 0 60
  interface s0.1 point-to-point
   frame-relay interface-dlci 60
```

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 3-10 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 3-10 do not take into consideration the fact that DLSw+ eliminates keepalive packets and acknowledgments.

Figure 3-10 Frame Format and Per-Packet Overhead of Various Encapsulation Types and Transaction Sizes



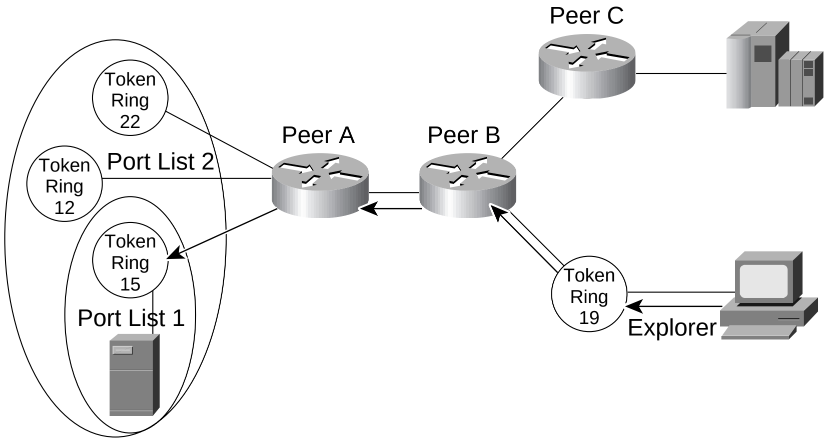
Encapsulation		40/1920		40/1200	
		SDLC	LAN	SDLC	LAN
TCP	← 56 → DLC | IP | TCP | DLSw | Data	5.7%	4.5%	9%	7%
FST	← 36 → DLC | IP | DLSw | Data	3.7%	2.4%	5.8%	3.9%
DLSw Lite	← 20 → FR | LLC2 | DLSw | Data	2%	1%	3.2%	1.3%
Direct	← 16 → DLC | DLSw | Data	1.8%	.6%	2.9%	1%

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 3-10 assume the minimum overhead (no RIF).

Port Lists

Port lists allow you to create broadcast domains in a DLSw+ network. Using port lists, you can control where broadcasts are forwarded. For example, in Figure 3-11 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 3-11 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
```

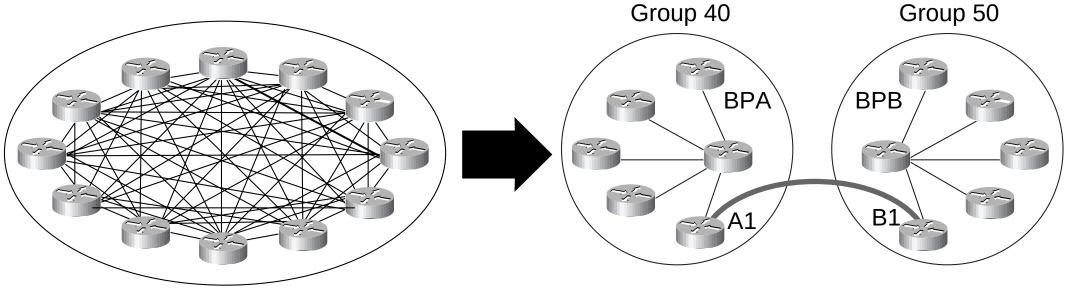
Peer Groups, Border Peers, Peer Group Clusters, and On-Demand Peers

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.

The border peer's functionality was enhanced in Cisco IOS Release 11.3 with the Border Peer Caching feature. The border peers check their local, remote and group cache before forwarding explorers to other routers. The local cache gives the border peer reachability information on its local data-link control. If the border peer finds that it can reach a destination via its local cache, then it does not forward the explorer to other peers. The remote cache gives the border peer reachability information within its own peer group. If the border peer finds it can reach a destination via its remote cache, then it forwards the explorer only to that peer. The group cache gives a border peer reachability information about other peer groups to which it does not belong. If the border peer finds it can reach a destination via its group cache, then it sends the explorer only to a border peer in that specific group.

In Figure 3-12, 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 3-12 Using Border Peers and Peer Groups to Minimize the Number of Required TCP Connections while Maintaining Full Any-to-Any Connectivity



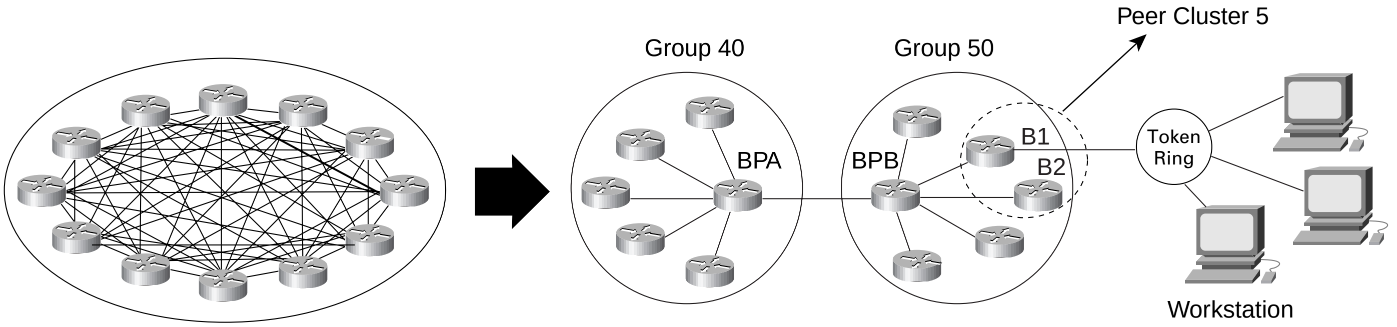
```
Configuration for Peer A1
dlsw local-peer peer-id 10.2.17.1 group 40 promiscuous
dlsw remote-peer 0 tcp 10.2.24.1
dlsw peer-on-demand-defaults tcp

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 checks its local, remote and group cache. If the resource is located in one of its caches, then the border peer forwards the explorer to the destination. If the resource is not located in the border peer’s cache, 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. Use peer groups and border peers only when you need branch-to-branch communication between NetBIOS or APPN end systems.

The Peer Group Cluster feature was introduced in Cisco IOS Release 12.0(3)T to further minimize explorer replication in border peer networks. Peer group clusters can be used in very large border peer networks where multiple routers within a peer group are serving the same LAN. DLSw+ “clusters” DLSw+ peers that are connected to the same LAN into logical groups. When the multiple peers are defined in the same peer cluster, the DLSw+ border peer does not forward explorers to more than one member within the same peer cluster. In Figure 3-13, member peers B1 and B2 are serving the same Token Ring LAN and have been configured into Peer Cluster 5.

Figure 3-13 Using Border Peers, Peer Groups, and Peer Clusters to Minimize the Number of Required TCP Connections while Maintaining Full Any-to-Any Connectivity



```
Configuration for Peer B2
dlsw local-peer peer-id 10.2.24.5 group 50 promiscuous cluster 5
dlsw remote-peer 0 tcp 10.2.18.2
dlsw peer-on-demand-defaults tcp
```

```
Configuration for Peer B1
dlsw local-peer peer-id 10.2.24.3 group 50 promiscuous cluster 5
dlsw remote-peer 0 tcp 10.2.18.2
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
```

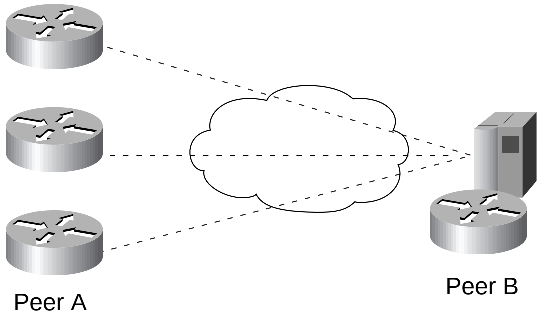
```
Configuration for Border Peer B
dlsw local-peer peer-id 10.2.18.2 group 50
   border promiscuous
dlsw remote-peer 0 tcp 10.2.24.1
```

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. When 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 remains 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** *minutes*).

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 3-14 shows an example of how to use this feature. In Figure 3-14, 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 3-14 DLSw+ Routers Configured to Take Advantage of the Dynamic Peer Feature



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

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

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 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 bring up the dial connection. However, DLSw+ peers send peer keepalives to each other periodically, and these keepalives do 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 are sent and, therefore, the peer keepalive does 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 does 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 only determines 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 waits 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 (ODR) for IP routing from the dial-up branches to the central site. ODR is a mechanism that provides minimum-overhead IP routing for stub sites. Define RIP Version 2 or ODR 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 [EIGRP] 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 has 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.

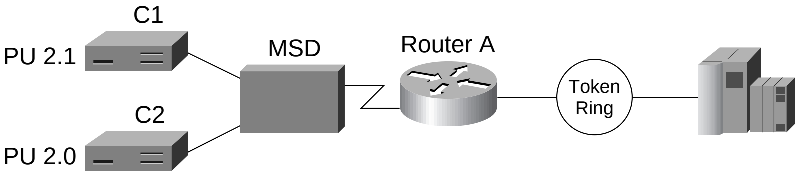
Many NetBIOS implementations use a session keepalive (in addition to a data-link control keepalive) to maintain sessions, so DDR may not work with NetBIOS. (The session level keepalive will keep the dial line up.) To address this issue, a new capability was added in Cisco IOS Release 11.3. A new command, **dlsw netbios-keepalive-filter**, filters 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 can also be used over a WAN where the remote branch has SNA devices on LANs, but the central site FEP still requires serial connectivity (for example, when the FEP is an IBM3725).

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 3-15.

Figure 3-15 Local Switching Configuration in a Mixed PU 2.0 and PU 2.1 Environment



```
Configuration for Router A
dlsw local-peer
source-bridge ring group 100
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
interface tokenring 0
   source bridge 1 1 100
   source bridge spanning
```

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

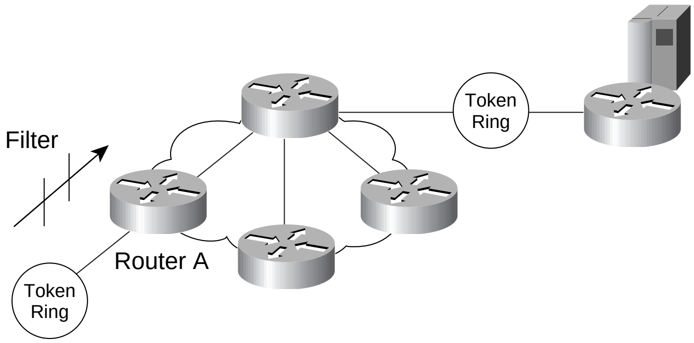
DLSw+ allows you to define access lists that are associated with a particular peer. This capability is powerful because it allows you to decide on a per-site basis what traffic should be allowed to pass over the network. These access lists use standard Cisco filter access list syntax.

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 **dlsw 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:

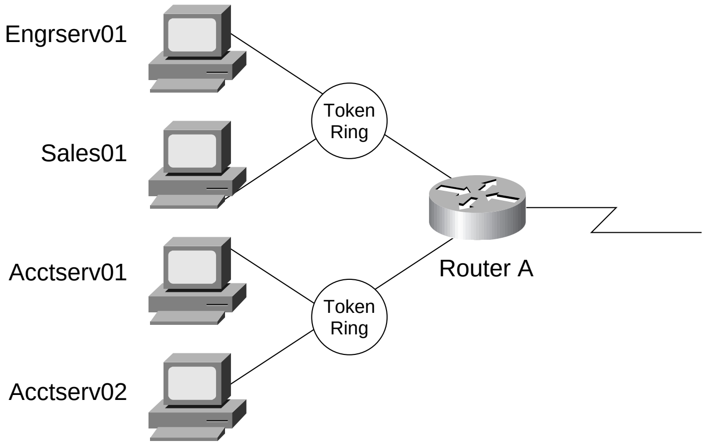
```
access-list 200 permit 0xf0f0 0x0101
```

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+.

Note: By default, DLSw+ handles all protocols not being routed by the router in which it resides.

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 host name
```

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.

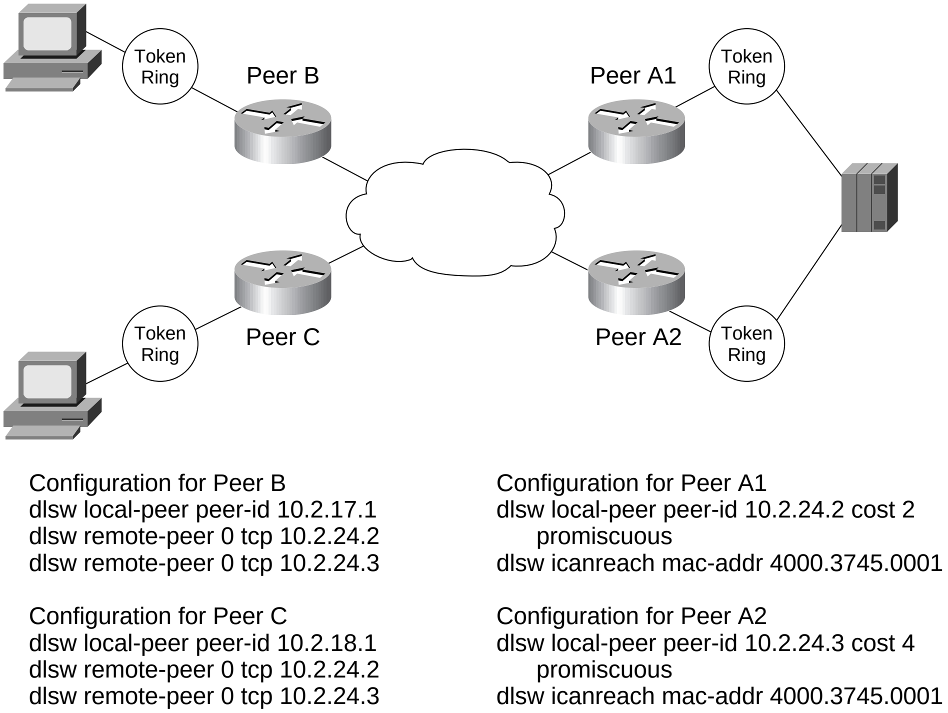
Static Configuration Options

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

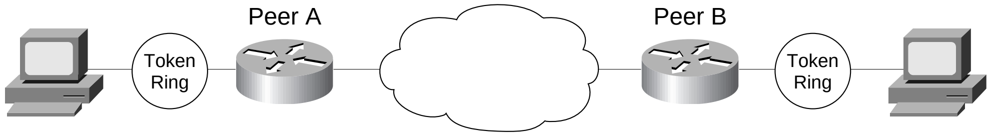


```
Configuration for Peer B
dlsw local-peer peer-id 10.2.17.1
dlsw remote-peer 0 tcp 10.2.24.2
dlsw remote-peer 0 tcp 10.2.24.3

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.

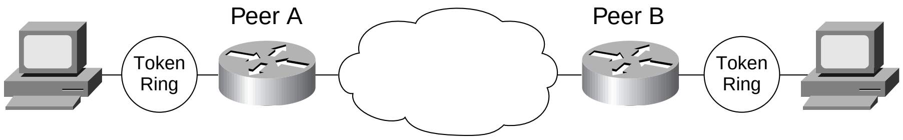
Reachability information learned as part of a capabilities exchange with a remote peer is considered valid as long as that remote peer is active. Multiple central site routers can advertise reachability of the same central site resources. If a remote branch router learns of multiple paths to a central 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 saps** 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 local or 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
Multiple static paths possible	Multiple paths possible

Setting Transmission Unit 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: largest frame size and IP maximum transmission unit (MTU) path discovery.

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 queries the maximum frame size and adjusts as required. When the explorer response reaches the originator, the response indicates 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 Table 4-2. The default largest frame size in the **dlsw local-peer** command is 17,800.

Table 4-2 Default MTU and Largest Frame Sizes for Various Encapsulation Types and Media

Encapsulation Type	DLSw+ LF Default Values	IP/MTU Fragmentation	Orientation of WAN Transmission
TCP	17800	Yes	Byte stream
FST	516	No	Packet
Direct	MTU set on local interface	No	Packet

The largest frame size and the MTU interplay differently depending on the encapsulation type.

In general, when using TCP encapsulation, you probably do not need to change the largest frame size because TCP fragments the frame size according to the MTU. For example, if the LF is smaller than the MTU, then TCP fragments each packet and sends them in sections across the WAN. If the LF is larger than the MTU, then individual packets are placed into the TCP/IP frame.

If using FST encapsulation, you might need to change the largest frame size. You should change this value only if you know your traffic profile and your output WAN interface MTU and 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. If the LF exceeds an MTU of an interface along the path, the packet is dropped and the session does not establish.

If using Direct encapsulation, you probably do not need to change the largest frame size because Direct encapsulation uses the MTU on the local interface to negotiate the LF size. As a result, the LF size never exceeds the MTU of an interface.

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 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 needs 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 does 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
```

IP MTU Path Discovery

When IP MTU path discovery is configured, peering routers determine the maximum IP frame size to be used for the TCP peer connection during peer establishment. 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

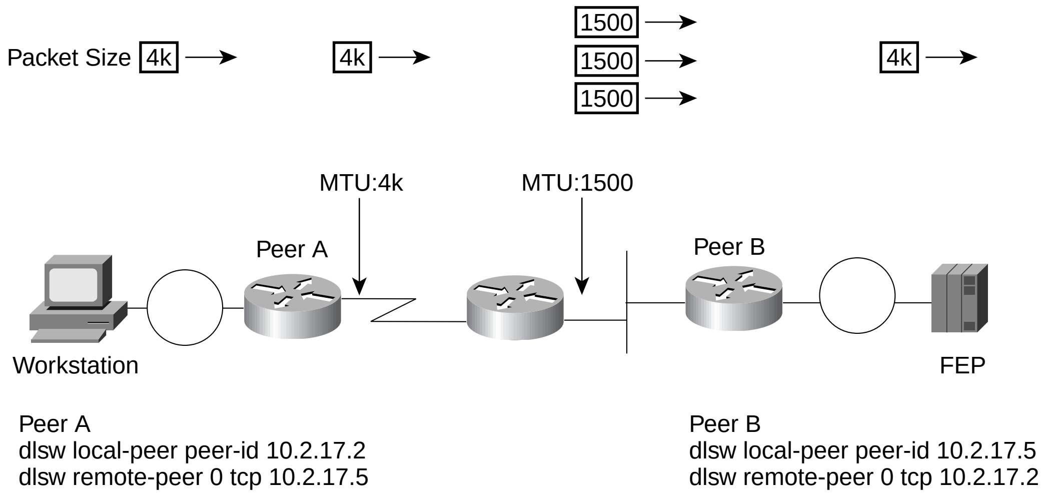
On the other hand, if bandwidth is not an issue, setting the IP frame size to a smaller number can improve response time.

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).

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 is 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. When 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. Figure 4-6 and Figure 4-7 show how the packet size is affected when a network is configured with and without MTU path discovery.

Figure 4-6 DLSw+ Design without IP MTU Path Discovery

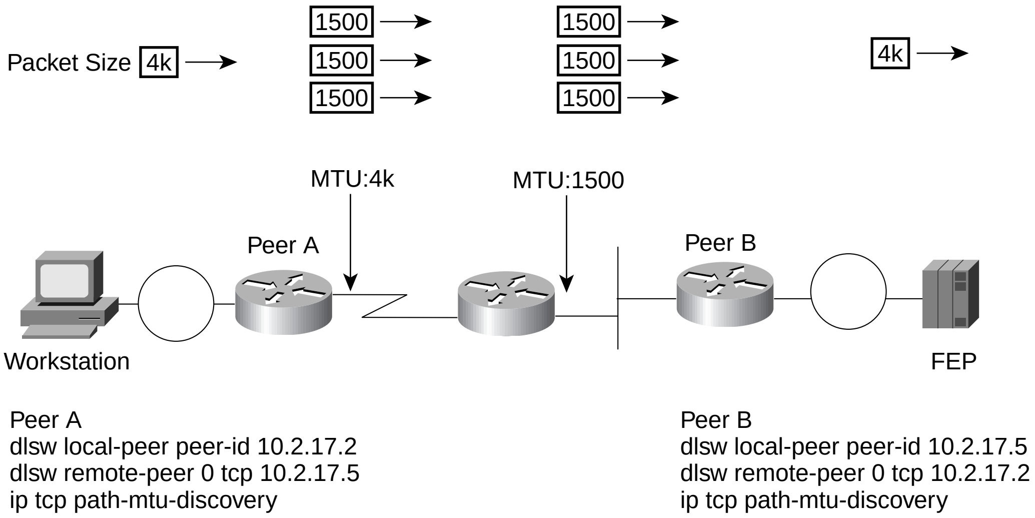


```
Peer A
dlsw local-peer peer-id 10.2.17.2
dlsw remote-peer 0 tcp 10.2.17.5
```

```
Peer B
dlsw local-peer peer-id 10.2.17.5
dlsw remote-peer 0 tcp 10.2.17.2
```

Figure 4-7 shows a DLSw+ network with IP MTU path discovery configured. IP MTU path prevents the packet from being fragmented at the IP layer. Fragmenting the packet at the IP layer causes problems because packets at this layer do not have the valuable TCP header packet identifier information, such as priority or destination. In Figure 4-7, the IP packet is automatically set to 1500 before being sent on the network, based on the minimum MTU size on the path of that peer connection. In Figure 4-6, however, the packet gets fragmented in route at the IP layer because the 4000 packet size is too large for the MTU set on the intermediate IP router.

Figure 4-7 DLSw+ Design with IP MTU Path Discovery



```
Peer A
dlsw local-peer peer-id 10.2.17.2
dlsw remote-peer 0 tcp 10.2.17.5
ip tcp path-mtu-discovery
```

```
Peer B
dlsw local-peer peer-id 10.2.17.5
dlsw remote-peer 0 tcp 10.2.17.2
ip tcp path-mtu-discovery
```

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 are 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. When 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+ does 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 provides 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.

Timer Settings

There are two types of timer settings: LLC2 timers and DLSw+ timers. The only timer discussed in this section is the LLC2 timers.

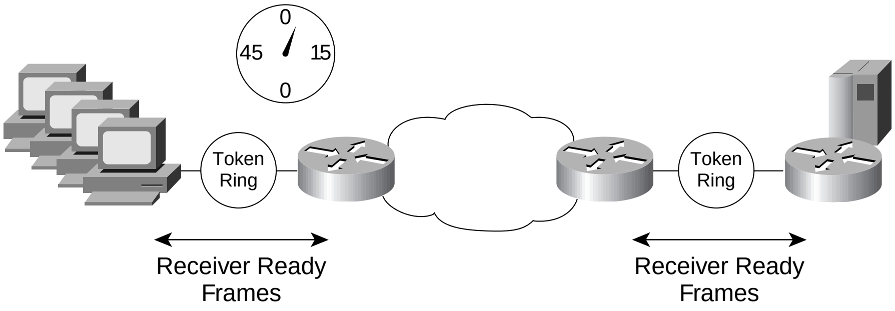
LLC2 Idle Time

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 sends an LLC2 packet to its partner. If there are many thousands of LLC2 sessions, then 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. Increasing the LLC2 idle time when supporting 4000 LLC2 sessions decreases the router CPU utilization significantly. The trade-off is that it takes longer to identify time-out conditions. This condition is generally a good trade-off.

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

```
llc2 idle-time milliseconds
dlsw bridge-group idle-time milliseconds
```

The **dlsw bridge-group idle-time** command affects the LLC2 idle time on a transparent bridge interface. Although it is possible to configure the LLC2 idle time on an Ethernet interface, the timer is not affected. The LLC2 timer does affect the Ethernet interface, however, when the DLSw+ Ethernet Redundancy feature is enabled because transparent bridging is not configured.

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:

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

The **explorer-delay-time** is the time the router waits before responding to explorers. Use this option on the router that is load balancing traffic. Because the DLSw+ peer selects its new circuit paths from within its reachability cache, the peer performing the load balancing needs enough time to receive all the explorer responses. The valid range is 1 to 5 minutes. It defaults to 0.

The **icannotreach-block-time** is the time the router marks 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+ caches 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 generally deletes 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 sends explorers to a NetBIOS resource (for LAN resources) or the time DLSw+ waits for a response before deleting the pending record (for remote resources). It defaults to 6 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-group-cache** is the length of time NetBIOS entries stays in the group cache. Use this option only on routers configured to be border peers. The valid range is 1 to 86000 seconds. It defaults to 240 seconds (4 minutes).

The **netbios-retry-interval** is the interval DLSw+ waits for a response to a name query or add name query on a LAN before retransmitting the request. The default is 1 second. Retries 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 4 minutes. Setting this value higher increases 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+ caches 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 generally deletes 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 sends explorers to a NetBIOS (for LAN resources) or the time DLSw+ waits for a response before deleting the pending record (for remote resources). It defaults to 3 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-group-cache** is the length of time SNA entries stay in the group cache. Use this option only on routers configured to be border peers. The valid range is 1 to 86000 seconds. It defaults to 240 seconds (4 minutes).

The **sna-retry-interval** is the interval DLSw+ waits 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 4 minutes. Setting this value higher increases 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+ waits 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 1 to 2 seconds gives 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 frequently responds first before your preferred peer.

When configuring the Ethernet Redundancy feature, you can use the **dlsw transparent timers** command to set the timeout value that the master router waits for all requests for a circuit before giving permission to a router to take a circuit. You can create separate timeout values for SNA and NetBIOS sessions. The default NetBIOS value is 400 milliseconds and the default SNA value is 1000 milliseconds. It is not one of the options in the **dlsw timer** command, rather it is a separate command.

Queue Depths

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

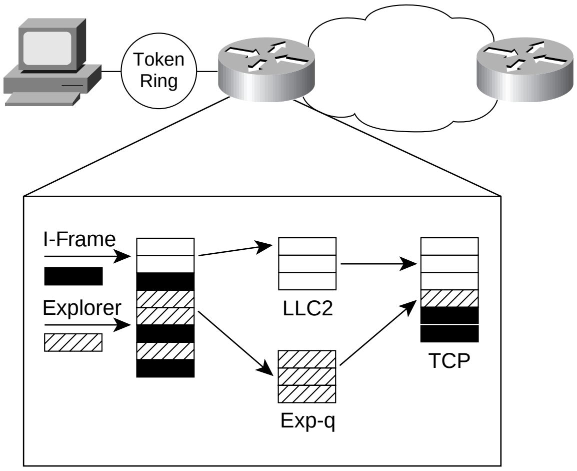
Explorer Queue Depth

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 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. Dropping these excessive explorers also minimizes CPU utilization.

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 limiting the size of the SRB explorer queue, 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. When this number is reached, new explorers are 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. When the cache hit is obtained, the router can often respond to the explorers without forwarding them.

Input Hold Queue

This queue is used to keep track of input frames off the LAN segment (other interface types as well, but here we will concentrate on LAN segments) that are awaiting system processing. 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 simultaneously process more packets from a particular interface.

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 does 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 alleviates packet retransmission and minimizes 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`**

Where *length* is the number of buffers that can be stored. The default is 75 input 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 leads 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 does. When 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 is 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 displays the total amount of memory available, memory used, and memory currently available. The **show buffers** command details 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.

When 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 occursfor one of two reasons: either the router needs more I/O memory to accommodate the amount of traffic and flow control requirements, or you have tuned your buffers and over-allocated in some area and depleted the I/O memory.

When you have gathered this information, open a case with the Cisco Technical Assistance Center (TAC), 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 memory** 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. Be sure to track how much time was allowed to lapse between outputs.
- 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 shows 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.

Note: The queuing mechanisms described in this chapter apply only to TCP encapsulation.

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 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.

When 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.

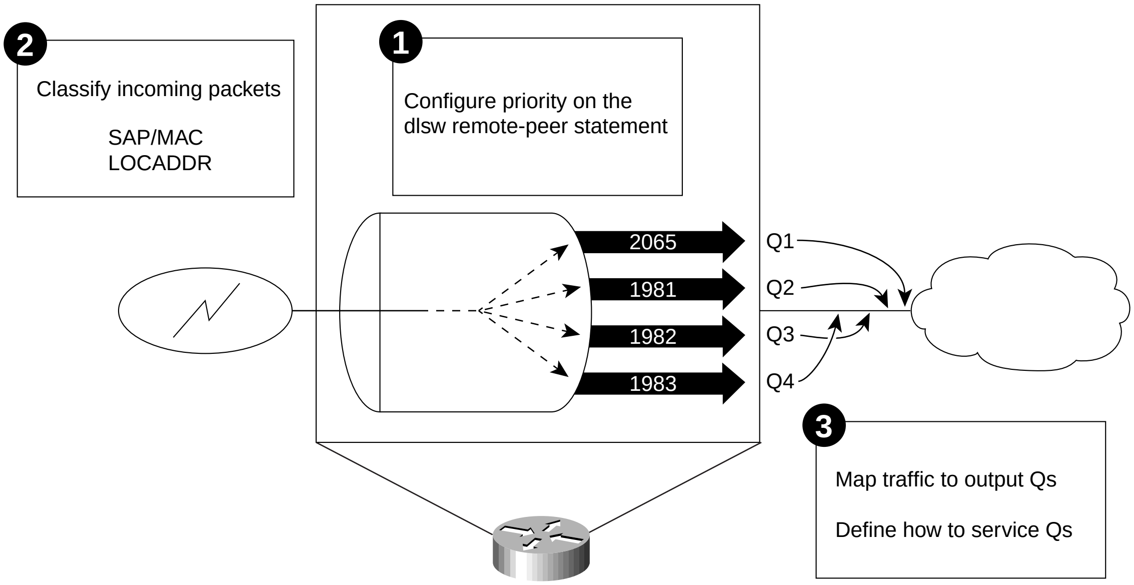
Figure 5-1 describes the tasks required to configure bandwidth management in a Cisco router. There are three steps:

Step 1. If you choose 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.

Step 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.

Step 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, and 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+ automatically activates 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 carries 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 does not carry any traffic
- TCP port 1982 defaults to normal priority; in the absence of any other configuration, this port carries information frames (nonbroadcast datagram frames)
- TCP port 1983 defaults to low priority; in the absence of any other configuration, this port carries broadcast traffic (CUR_ex, Name_query_ex, SSP DATA/DGRM broadcasts)

Note: You can configure specific traffic to go into either port 2065, 1981, or 1983. If you specify **priority** in the **dlsw remote-peer** command and do nothing else, all data traffic goes in TCP port 1982 and 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). 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).

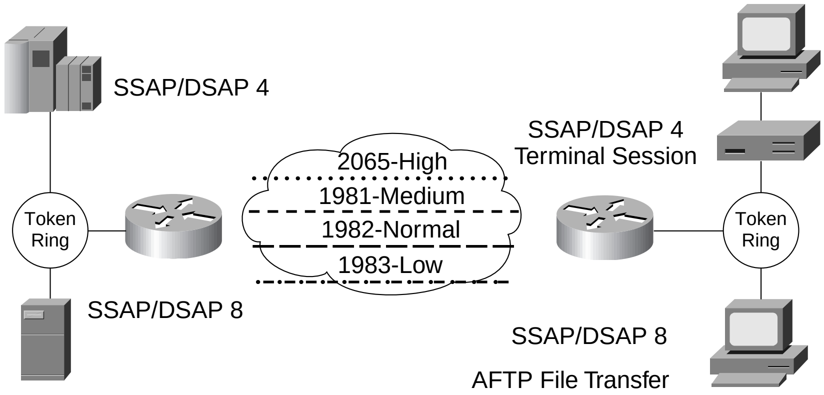
To map a SAP priority list to an Ethernet bridge group (requires Cisco IOS Release 11.0 or later), specify the **sap-priority** keyword on the **dlsw bridge-group** command as follows:

```
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 is 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.

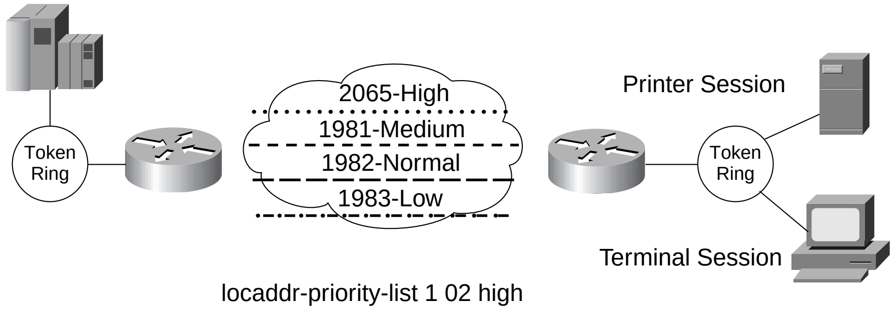
To map a LOCADDR priority list to an Ethernet bridge group (requires Cisco IOS Release 11.0 or later), specify the **locaddr-priority** keyword on the **dlsw bridge-group** command as follows:

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 is 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.

SNA ToS

DLSw+ type of service (ToS) is another method for providing granularity and ensuring prioritization for your SNA traffic. It maps SNA Class of Service (COS) to IP ToS, ensuring priority is preserved across the IP network. When DLSw+ is used in conjunction with APPN, SNA ToS maps APPN COS to IP ToS, and preserves SNA COS across an IP network.

When the **priority** option on the **dlsw remote-peer** command is specified, DLSw+ automatically activates four TCP connections to the remote peer, sets IP Precedence values and assigns traffic to specific ports according to the rules defined in Table .

Table 5-1 TCP Port-to-IP Precedence Default Mapping

TCP Port	DLSw+ Priority Queue	IP Precedence Value	IP Precedence Value
2065	High	Critical	5
1981	Medium	Flash override	4
1982	Normal	Flash	3
1983	Low	Immediate	2

The default precedence values can be overridden using the **dlsw tos map** command or by using policy-based routing. In the following example, the medium-priority traffic is remapped to immediate IP precedence; normal-priority traffic is mapped to priority IP precedence; and low-priority traffic is mapped to routine IP precedence (the high-priority traffic remains at critical precedence):

```
dlsw tos map high 5 medium 2 normal 1 low 0
```

After opening the four pipes and separating the traffic into individual queues, apply weighted fair queuing to service the queues.

When DLSw+ is used in conjunction with APPN, ToS maps APPN COS to IP ToS and preserves SNA COS across an IP network. For example, SNA batch can be prioritized higher than SNA interactive traffic. When the **priority** keyword is specified on the **dlsw remote-peer** command, DLSw+ automatically activates four TCP connections to the remote peer, sets IP precedence values and assigns traffic to specific ports according to the rules defined in Table 5-1.

Table 5-2 APPN COS to IP ToS Mapping

APPN Mode Names	SNA Transmission Priority	TCP Port	Priority Queue	IP Precedence	Precedence Numeric Value
CPSNASVCMGR	Network	2065	High	Critical	5
#INTER	High	1981	Medium	Flash override	4
#CONNECT	Medium	1982	Normal	Flash	3
#Batch	Low	1983	Low	Immediate	2

APPN and DLSw+ must be running in the same router for APPN COS to IP ToS mapping to occur. ToS only applies to TCP and FST encapsulation types. When using FST encapsulation, ToS marks all DLSw+ traffic with IP precedence “network.” The user cannot alter these default mappings.

Queuing Algorithms

The Cisco IOS Software implements four different output queuing algorithms:

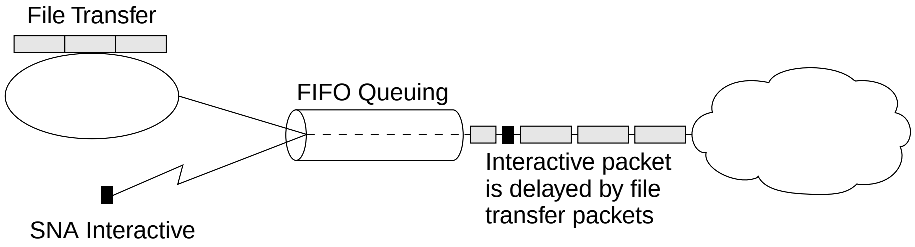
- First in, first out queuing
- Priority queuing
- Custom queuing
- Weighted fair queuing

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. It is the default queuing mechanism for any interface with more than 2 MB bandwidth. 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.

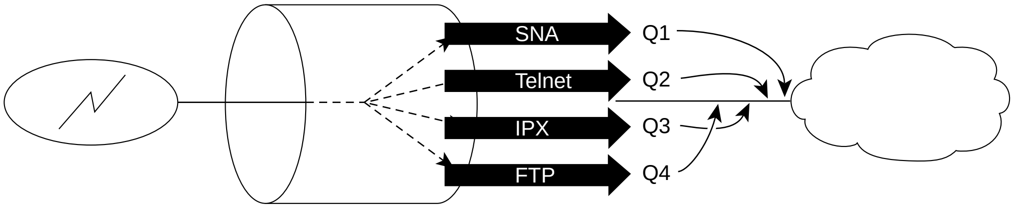
Figure 5-4 Potential Impact of a File Transfer on Interactive Traffic



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 is continually serviced and packets from the other queues queue up and are 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, Telnet (TCP port 23) is in the medium queue, IPX is in the normal queue, and FTP (TCP port 21) is in the lowest-priority queue.

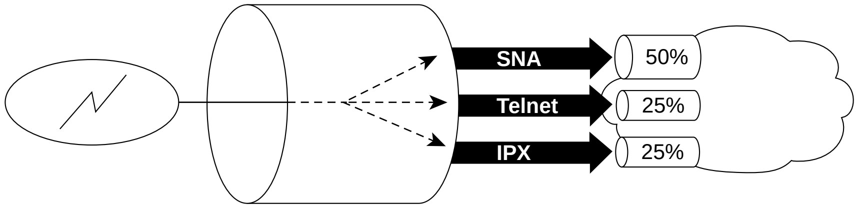
Custom Queuing

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 sends 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 default 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.

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



```
Interface Serial0
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
```

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 shares 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:

- When the byte count value is exceeded, the frame that is currently being transmitted is 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 are sent, *not* 100 bytes.
- Very large byte counts produce a "jerky" distribution. That is, if you assign 10,000, 15,000, 20,000, and 25,000 to four queues, each protocol is serviced nicely when its queue is the one being serviced, but after it is serviced, it may take some time to get back to that queue.
- Window size also affects the bandwidth distribution. If the window size of a particular protocol is set to one, then that protocol does not place another frame in the queue until it receives an acknowledgment. The custom queuing algorithm moves 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 is sent each time. If your byte count is set to 2 KB and your frame size is 256 bytes, then only 256 bytes are sent each time this queue is serviced.
- You need to know the frame size of each protocol. Some protocols, such as IPX, 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 does 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:

Step 1. Produce a ratio of all frame sizes, dividing into 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

Step 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

Step 3. 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.

Step 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

Following is a basic configuration used for custom queuing with SAP prioritization:

```
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 peer-id 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
  !
  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
```

The default byte count for queues 2 and 3 is 1500 even though it does not appear in the configuration.

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/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

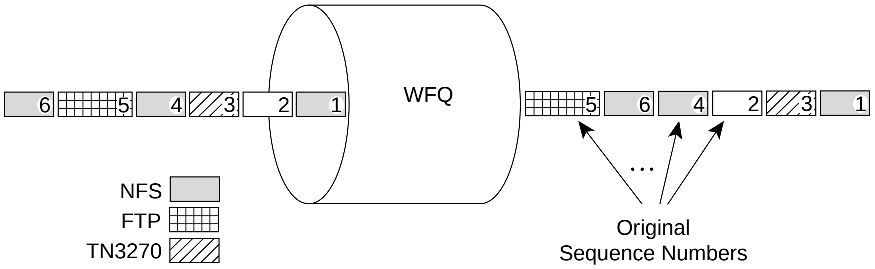
In each case, enough of the packet is checked to break down the streams of packets into separate conversations.

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 favorably, which is recommended when using weighted fair queuing in conjunction with DLSw+ or other SNA features. This topic 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. When congestion is flagged, the weights used by the algorithm are altered so that the conversation encountering the congestion transmits less frequently.

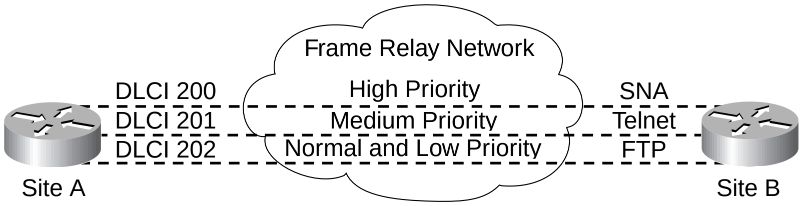
DLCI Prioritization

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. Priority queuing provides bandwidth management control over the access link to the Frame Relay network. Frame Relay switches (for example, the Stratacom IPX, IGX, and BPX/AXIS switches) provide prioritization within the Frame Relay cloud. In other words, the DLCI does not prioritize the traffic, it separates the traffic so that Frame Relay can prioritize it based on the DLCI number. 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



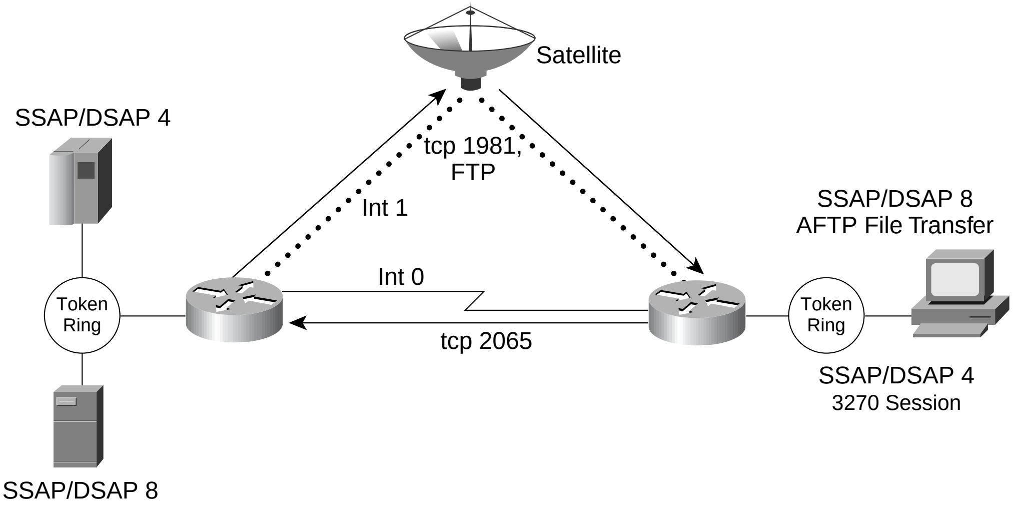
Directing Traffic Flows with Policy Routing

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 8---------->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 5-3).

Table 5-3 Precedence Field Values

Value	Definition	
Network	Match packets with network control precedence[1]	(7)
Internet	Match packets with internetwork control precedence[1]	(6)

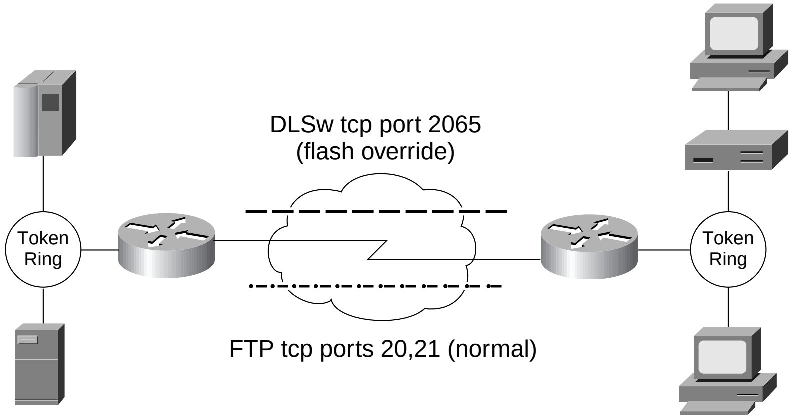
Table 5-3 Precedence Field Values (Continued)

Value	Definition	
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)

1. Reserve this value for network critical traffic

By modifying the precedence value, you can increase the amount of bandwidth that weighted fair queuing allocates 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) is given higher priority 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+ higher priority:

```
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 flash-override
```

RSVP Bandwidth Reservation

Resource Reservation Protocol (RSVP) bandwidth reservation allows DLSw+ to reserve network bandwidth for TCP connections between DLSw+ peers. The user specifies the amount of RSVP reserved bandwidth in the following ways:

- *Globally*—When the user configures the **dlsw rsvp** command, DLSw+ uses these values for initiating RSVP to all its peers. After RSVP is globally enabled, the user must enable RSVP on specific peers. The user can retain the *average-bit-rate* and *maximum-burst* values set in the **dlsw rsvp** command or the user can override these values for any particular peer connection (remote, promiscuous, or peer-on-demand) by configuring the **dlsw remote peer**, **dlsw prom-peer-defaults**, or **dlsw peer-on-demand-defaults** command.
- *Per peer*—When the user configures the **dlsw remote peer tcp** command, DLSw+ configures the RSVP parameters specifically for this peer connection.
- *Type of peer connection*—When the user configures either the **dlsw peer-on-demand-defaults** or **dlsw prom-peer defaults** command, DLSw+ uses the configured RSVP parameters for peer-on-demand and promiscuous peer connections, respectively.

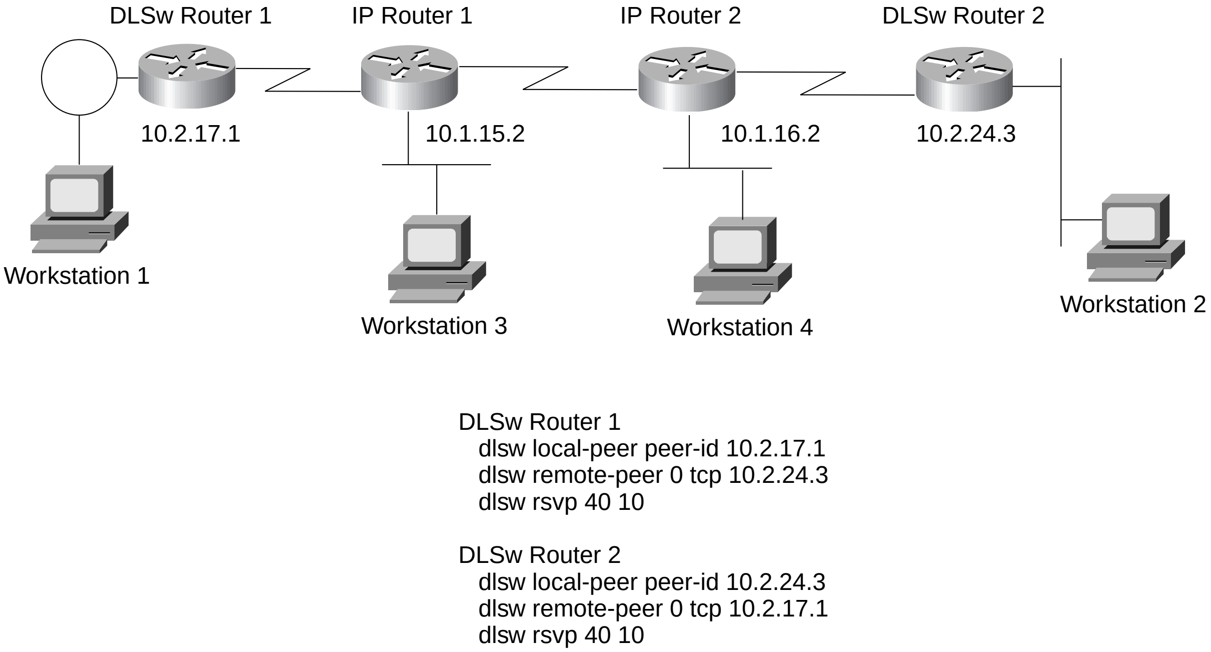
In any of these situations, the user turns off RSVP by setting the *average-bit-rate* or *maximum-burst* values to 0.

Because RSVP requires both a sender and a receiver, the DLSw+ RSVP bandwidth reservation feature must be implemented on both devices of a DLSw+ connection. However, RSVP does not need to be configured on all devices that are in the IP routed path between two DLSw+ peers. In this type of configuration the devices in the middle must support only IP RSVP; they do not need to be configured for the new DLSw+ RSVP bandwidth reservation feature or of DLSw+. The devices between the peers prioritize the IP packets belonging to the DLSw+ session according to the IP ToS settings. If, however, the devices in the middle do not support IP RSVP, end-to-end bandwidth is not guaranteed.

In the case of priority peers, RSVP bandwidth reservation is done only for the highest priority connection to the peer (TCP port 2065). If the user configures priority queuing and RSVP on the same peer, the user must ensure that the RSVP designated traffic is assigned to the highest priority TCP peer connection.

If the users change the *average-bit-rate* or *maximum-burst* settings without removing the existing RSVP bandwidth reservation, a message warns the users that they are removing the existing reservation and that they need to request a new reservation with new values.

In Figure 5-11, DLSWRTR 1 and DLSWRTR 2 are configured for the DLSw+ RSVP bandwidth reservation feature with an average bit rate of 40 and a maximum-burst rate of 10.



```
DLSw Router 1
   dlsw local-peer peer-id 10.2.17.1
   dlsw remote-peer 0 tcp 10.2.24.3
   dlsw rsvp 40 10

DLSw Router 2
   dlsw local-peer peer-id 10.2.24.3
   dlsw remote-peer 0 tcp 10.2.17.1
   dlsw rsvp 40 10
```

Figure 5-11 Sample Configuration of DLSw+ with RSVP

There following **show** commands are useful in verifying the DLSw+ RSVP feature:

- **show ip rsvp request—**Verifies whether the RSVP RESV messages for DLSw+ are sent all the way through the RSVP network to the remote peer
- **show ip rsvp reservation—**Verifies that the RSVP bandwidth reservations are in place for the DLSw+ peers
- **show ip rsvp sender—**Verifies that the RSVP PATH messages for DLSw+ are sent and that the feature is working correctly

To disable the DLSw+ RSVP bandwidth reservation feature for all peers, issue the global configuration **no dlsw rsvp** command. Setting the *average-bit-rate* and *maximum burst* values to 0 in the **dlsw remote peer tcp**, **dlsw prom-peer defaults**, and **dlsw peer-on-demand defaults** commands turns off RSVP for a particular peer connection. The reservations made by the DLSw+ RSVP commands can be deleted by the global RSVP commands (for example, **no ip rsvp reservation**).

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 "Designing Meshed Networks" chapter 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 One Transaction per Minute

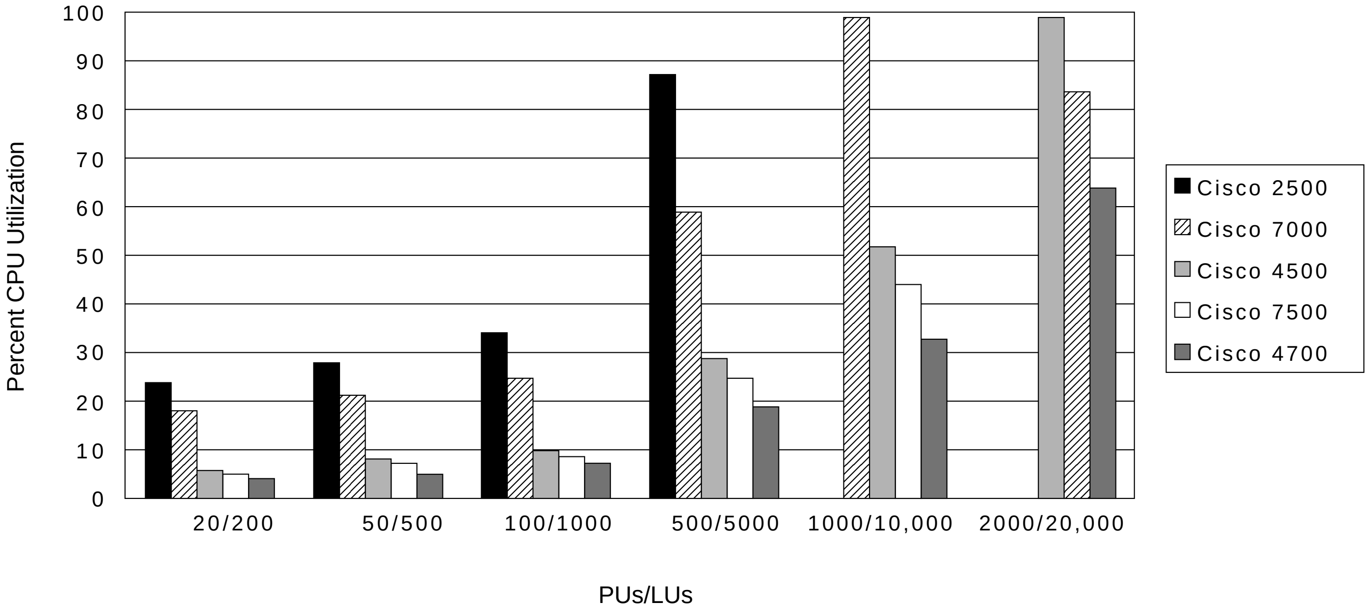
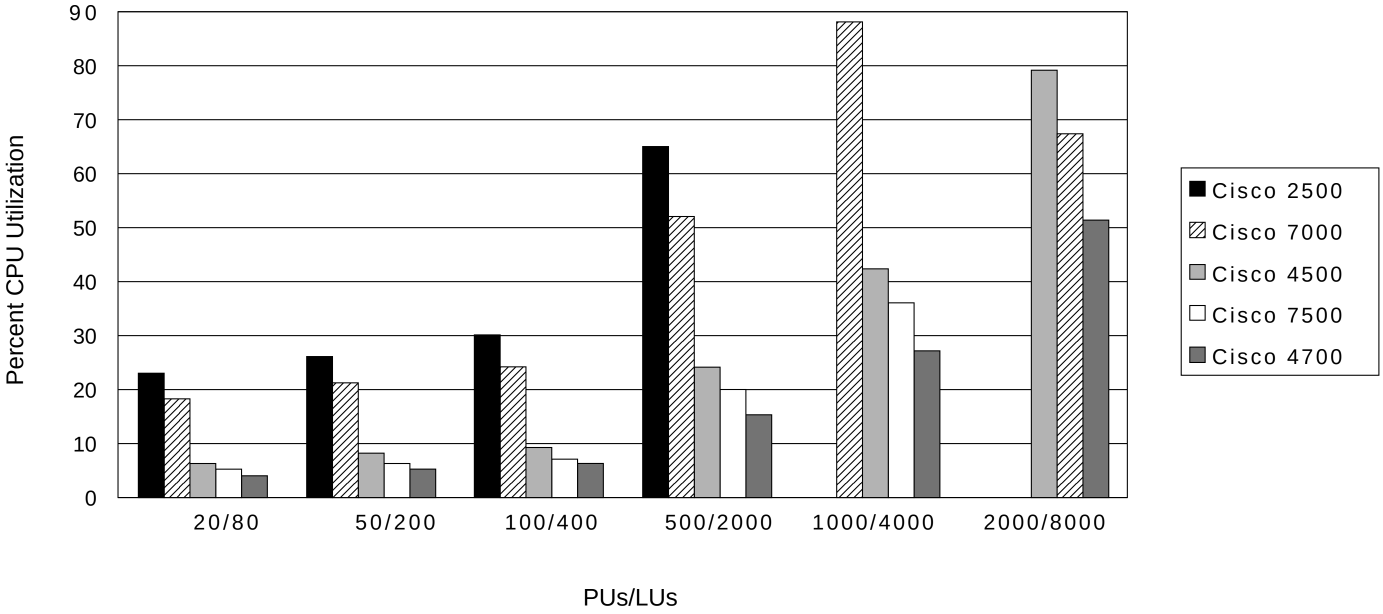


Figure 6-2 CPU Usage of Various Routers Assuming TCP Encapsulation, Transactions of 40 Bytes in and 1000 Bytes Out, and Assuming Each PU has 4 LUs, Each with One Transaction per Minute



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 passthru 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, 7200, or 7500 Series router) rather than a slower processor (for example, a Cisco 4000 or 7000 Series router). 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.

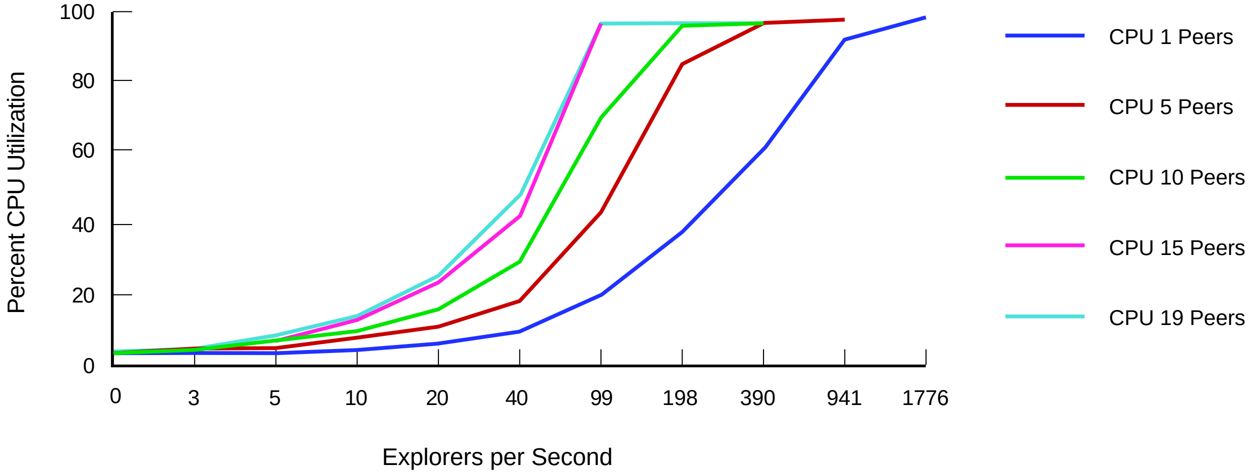
Number of Remote Sites

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 “Customization” chapter). 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 Series router. 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 Series router peers to 20 remote peers, it can replicate almost 100 explorers per second to each of the 20 peers. If the router 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.

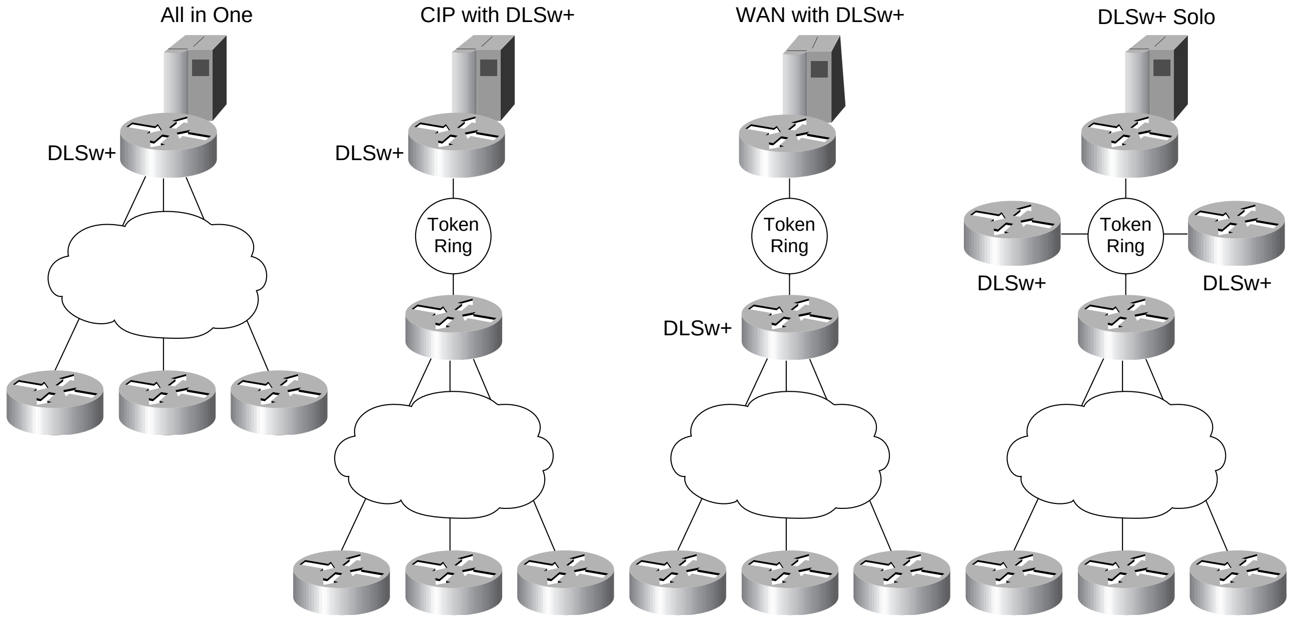
Placement of Peering Routers

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 an IBM 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 4700 or 7200 Series routers to handle the capacity. This is more cost effective than adding large Cisco 7500 Series routers with CIPs. Because SRB is fast switched, a single Cisco 7500 or 7000 Series router with a CIP can handle the traffic from four or five Cisco 4500s or three Cisco 4700s using DLSw+. Figure 6-4 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.

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 provides 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.

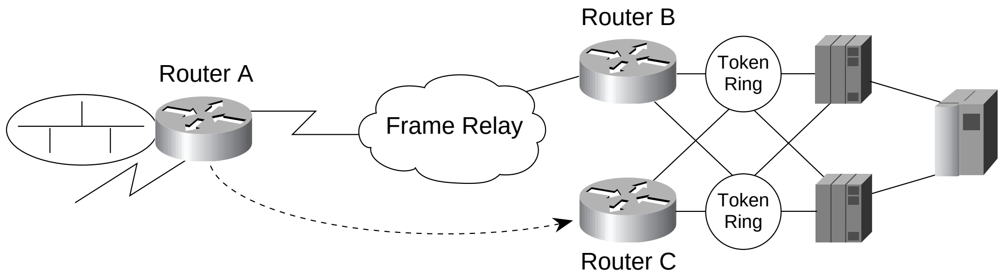
When using direct encapsulation, link failures are disruptive but recovery can be automatic. Backup from link failures can be addressed either by having multiple remote peers or by configuring multiple remote peer statements for the same remote peer but specifying a unique path to each one. You can either 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 6-5.

Figure 6-5 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 causes 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.

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+ attempts to switch nonrouted multiprotocol traffic if not filtered. The "Customization" chapter describes how to configure filtering to allow only SNA or NetBIOS into DLSw+. When 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 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 commands **dlsw icanreach mac-address 4000.3745.0001** and **dlsw icanreach mac-exclusive** tell remote routers that the *only* destination this router can reach is the MAC address of the IBM 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.

CHAPTER 7

Designing Meshed Networks

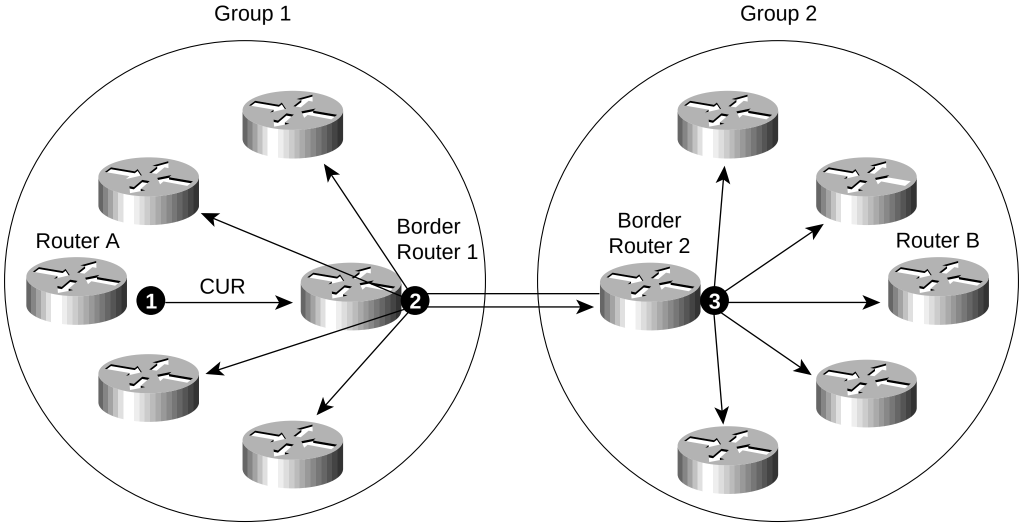
This chapter describes design considerations when using DLSw+ to build a meshed, any-to-any network. It describes the role of border peers and provides guidance on where to place border peers, how to backup border peers, and how to size border peers. It also describes how to size peer groups, how to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

The Peer Group Concept

DLSw+ 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 "Advanced Features" chapters. Using peer groups you can build enterprise networks that support branch-to-branch communication between either NetBIOS or Advanced Program-to-Program Communication (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 results. 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 checks its local, remote and group cache before forwarding the explorer. If the resource is not found, 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 checks its local, remote and group cache. If the resource is found in its cache, the border peer forwards the explorer to the known destination. If the resource is not found, then it 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 each border router replicated it only five times.

DLSw+ further reduces explorer replication with the Peer Group Cluster feature, introduced in Cisco IOS Release 12.0(3)T. If multiple routers are serving the same LAN within the same peer group, they can be "clustered." This classification further reduces explorer replication because the border peer does not send an explorer to more than one router serving the same LAN within a peer group.

Explorer Processing When Using Peer Groups

When a **dlsw local-peer** command specifies a group number, that local peer forwards explorer packets only to border peers in its 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 forwards broadcasts to all configured peers. 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 forwards it only to Border Router 1. If Border Router 1 is not active, it forwards 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 are always 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 for connectivity within the group.

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 does not forward the explorer to a third group.

Border peers will forward explorers to local interfaces 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, 7200, or 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. Every peer in a group will forward explorers to one of the border peers in its group. If the border peers are configured with different costs, then the member peer will select the border peer with the lowest cost. If the border peers are configured with same cost, then the member peer will select the border peer with which it had the most recent active peer connection. Border peers do not perform load balancing.

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

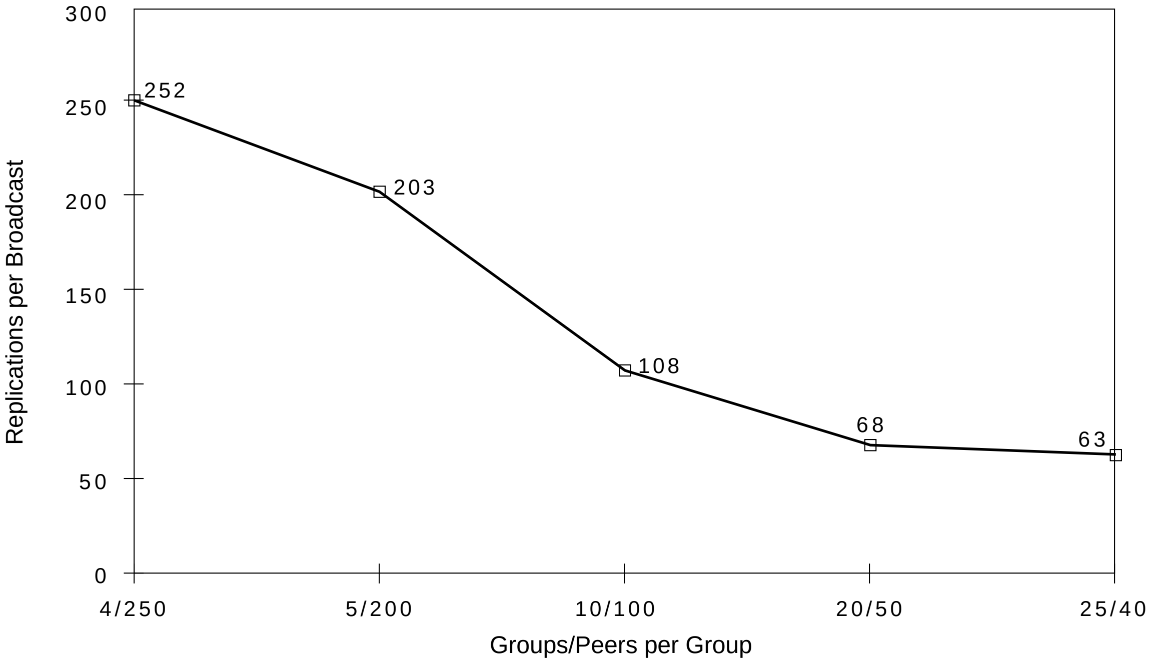
On-Demand Peers

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 determines 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 four 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 four 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



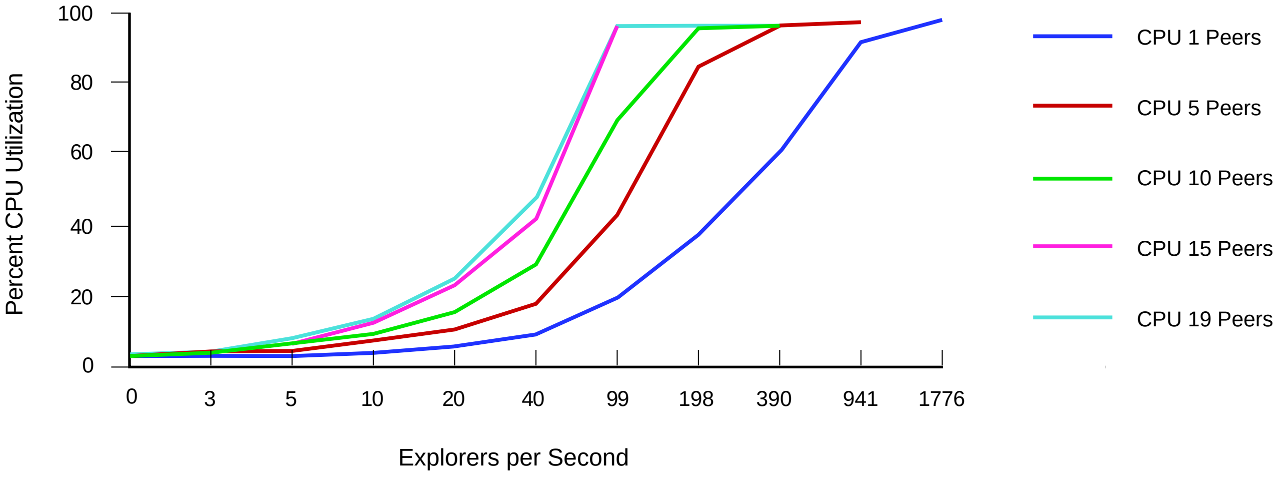
Broadcast Replication/Reduction

Figure 7-3 shows the router utilization based on the number of explorers per second. The practical limit for a Cisco 4700 Series router 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 router 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 tries to find that resource, resulting in at least 50 simultaneous broadcasts (each DLSw+ peer may get multiple explorers but forwards 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 are not accessed by all branches, all the time.

In Cisco IOS Release 11.3, border peers were enhanced to support group caching, which greatly reduces explorers. If you have an earlier version of software, and 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 forwards the explorer everywhere. That is because, prior to Cisco IOS Release 11.3, border peers do not check their cache before forwarding explorers.

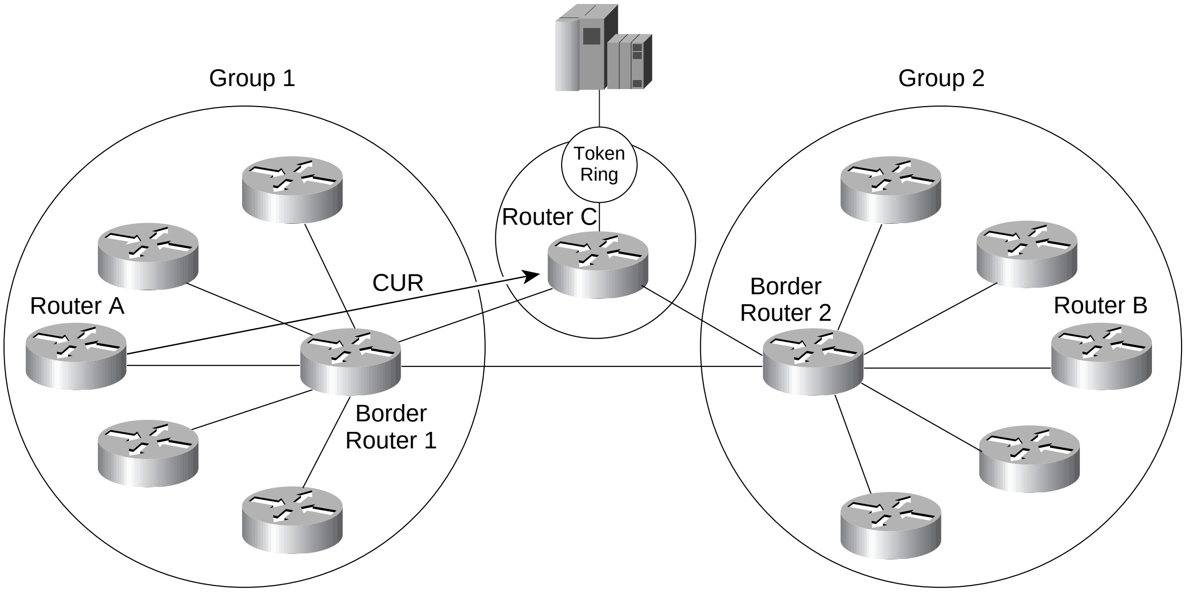
Figure 7-3 CPU Utilization Comparison of a Central Site Router as the Number of Explorers per Second Varies and as the Number of Peering Routers Increases (No Caching Is Assumed, and Each Explorer Received Must Be Replicated to Each Remote Peer)



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 learns that Router C can access the MAC address of the FEP. When Router A receives an explorer for the FEP MAC address, it first checks its cache, where it finds 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 forwards 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 preloads its cache with the MAC address of the FEP and always sends 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



CHAPTER 8

RSRB Migration and Multivendor Interoperability

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

- The reasons you would migrate from RSRB to DLSw+
- The migration implications in terms of management, memory, and performance
- The steps to migrate from RSRB to DLSw+

In addition, this chapter describes interoperability with other RFC 1795 and RFC 2166 implementations, including valid configuration options.

RSRB and DLSw+ Comparison

RSRB, created in 1991, preceded DLSw+ and existed before there were routing standards. RSRB was the original Cisco implementation for transporting LLC2 traffic over an IP network. RSRB addressed a critical need in the market place, thus thousands of RSRB networks were built. DLSw+ has replaced many of these networks, but there are hundreds of RSRB networks still in existence, some with more than 1000 RSRB routers.

The AIW approved the first standard for SNA over IP in 1995. This standard was created in and was later documented in RFC 1795 and RFC 2166. DLSw+ complies with RFC 1795 and RFC 2166 and provides enhancements that allow DLSw+ networks to scale better and provide better availability than either RSRB or standard-only implementations. Most integrated SNA and IP networks that were installed since 1995 have been built using DLSw+.

DLSw+ includes functions that were previously provided in several other Cisco features, including RSRB, SDLC-to-LLC2 conversion (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+?

Cisco has chosen DLSw+ as the strategic solution for SNA transport going forward. RSRB has not been enhanced since 1995 and no enhancements are planned. In addition, at some point in the future the new Cisco IOS Software releases will no longer include RSRB.

DLSw+ provides better functionality, manageability, and control than RSRB. 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 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 rerouting around link failures Local acknowledgment on Token Ring and SDLC	Nondisruptive rerouting around link failures Local acknowledgment on Token Ring and SDLC Local acknowledgment on Ethernet[1] Backup peers[1] Fault tolerant and priority peers[1]
Scalability	Limited broadcast reduction	Broadcast reduction Dynamic peers[1] UDP for UI frames[1] RIF termination[1] Broadcast optimization with peer groups and border peer caching[1]
Flexibility	Media conversion via SDLLC and SR/TLB (PU 2.0 only) SRB dynamics RIF Passthrough Transport options (FST, Direct) Support for end systems on Token Ring, SDLC (with SDLLC), or Ethernet (with SR/TLB) AST[2] FST between unlike media via SDLLC[2] LNM over FST[2]	Media conversion built in (PU 2.0, 2.1 and PU 4) SRB dynamics RIF termination or optional RIF passthrough Transport options (FST, direct) DLSw Lite (LLC2 encapsulation) Support for end systems on Token Ring LANE, Token Ring ISL, and SRB FDDI Capabilities exchange Peer biasing with cost SNA DDR Promiscuous peers Multivendor interoperability

1. Supported by DLSw+ but not by RSRB
2. Supported by RSRB but not by DLSw+

Cisco designed DLSw+ in a modular fashion to maximize stability and to facilitate new feature additions. The circuit concept in DLSw simplifies management. The Cisco implementation protects your investment in the technology and simplifies network integration of acquired companies because it can interoperate with other standard-compliant implementations. Finally, DLSw+ surpasses RSRB as the most commonly employed technique for SNA and client/server integration.

Possible Migration Inhibitors

A few environments will not be able to move from RSRB to DLSw+ at this time. They may be on older software releases and require features that were added to DLSw+ in a recent release of Cisco IOS Software. For example, RIF passthru was added in Cisco IOS Release 12.0 and is required for FEP-to-FEP communication over parallel SRB paths.

There are a few RSRB features not available in DLSw+ or planned for future releases, including the following:

- FST between SDLC and LANs
- LAN Network Manager over FST
- Automatic Spanning Tree (AST), which is 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:

- Does DLSw+ perform as well as RSRB?
- 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. (This comparison assumes that either local acknowledgment is turned on for both or off for both.) However, DLSw+ uses more memory than RSRB. The key reason DLSw+ requires more memory 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 run well with the default memory that comes with the router and software subset. For example, the Cisco 2500 Series router (branch router) in a typical branch environment with 20 to 40 PUs and LUs and the standard memory configuration that comes with any of the IBM images, runs DLSw+ quite well. If you are running RSRB with an older level of the Cisco IOS Software, you may want to verify that 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 that is 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 approximate the memory required by DLSw+ from the formulas provided in Appendix A.

Migration Options

There are several ways to migrate to DLSw+. Which migration option you use depends on how the current RSRB peering structure is set up, whether your RSRB network allows any-to-any communication, and which design you want to use for your DLSw+ network. This section describes the following options:

- Migrating hierarchical networks (where all communication is from remote routers back to one or more central site routers) using
  - Separate routers for RSRB and DLSw+ peering
  - RSRB and DLSw+ concurrently in the same data center peer
- Migrating any-to-any networks from a
  - Fully meshed RSRB network (where every router is peered to every other router) to a fully meshed DLSw+ network
  - Fully meshed RSRB network to DLSw+ peer groups
  - Multihop RSRB network to DLSw+ peer groups

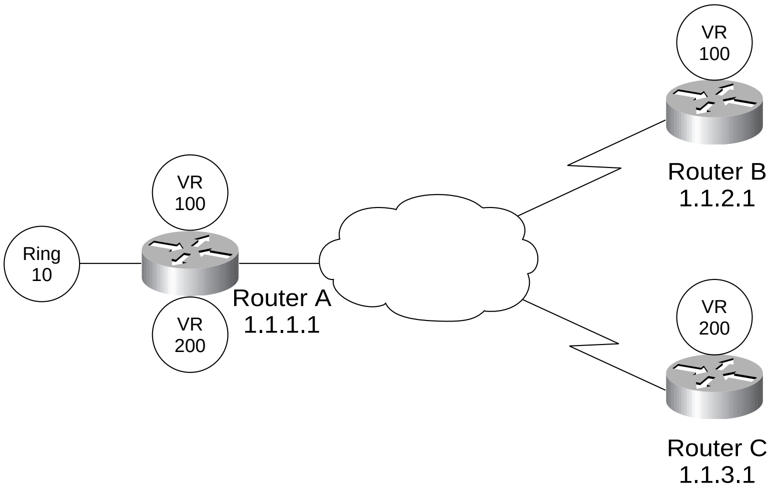
Before you can decide which of these options is best for your environment, it is essential to understand your current RSRB network.

Understanding Your Current RSRB Network Topology

To migrate an existing RSRB network to DLSw+, you must first understand what connectivity your RSRB network provides. Your RSRB network might enable communication that is not obvious from your network definitions.

For example, as shown in Figure 8-1, if Router A has RSRB connections to Router B and Router C, traffic might flow from Router B to Router C even though they are not peers. This situation occurs only if Router A is configured with two virtual rings and a separate physical interface connecting Physical Ring 10 with each virtual ring. Identify whether or not there are multiple ring-group numbers in use within the same part of an RSRB network.

Figure 8-1 Sample RSRB Network with Different Ring Numbers

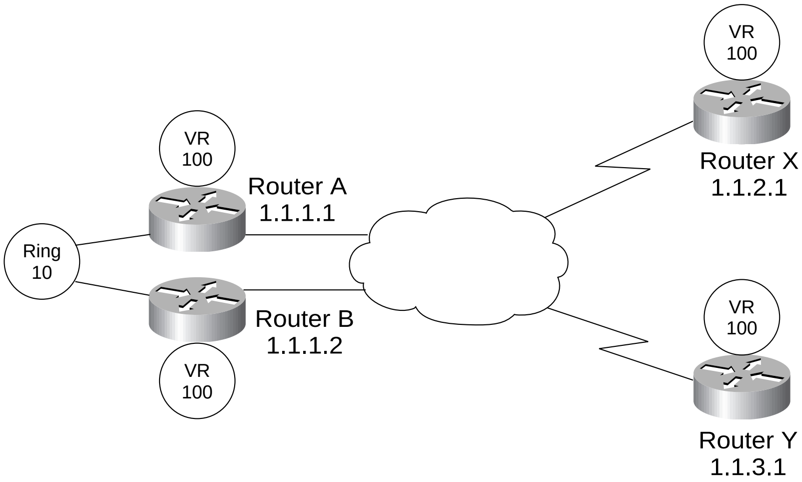


To understand how this router configuration affects the data paths in the network, it is helpful to understand the use of the RIF field in explorer frames used for path discovery. When a device sends a path discovery frame on an SRB media, it usually inserts a RIF indicating that this frame is an explorer frame. When a device propagates an explorer frame through the network, source-route bridges on the network copy the explorer off the ring from which it originated onto one or more other rings that the bridge connects. As the bridge copies the frame, it modifies the RIF, indicating the path that the explorer frame took. The RIF lists every ring and every bridge that this frame has crossed. Hence, by looking at the RIF, it is possible to uniquely identify the entire path a frame has taken through the SRB network.

The bridge checks the existing RIF to ensure that the explorer has not already traversed its ring before it places a new copy of an explorer frame on its ring. This action prevents an explorer from looping around an SRB network until it exceeds the maximum number of permitted bridge "hops" (seven in most SRB implementations).

Many SRB designs make use of this feature by defining all of their virtual ring numbers as the same value. This action ensures that no frame traverses any RSRB hop more than once. In Figure 8-2, Router X peers to Router A and Router Y peers to Router B. If Router A forwards an explorer from Router B onto Ring 10, SRB in Router B does not pick up that explorer. The frame cannot be copied to the virtual ring in Router B because the RIF shows that the explorer has already traversed Ring 100. Using the same virtual ring number for RSRB peers in this manner helps limit the total number of explorers traversing a WAN.

Figure 8-2 RSRB Network with the Same Virtual RIng Numbers



However, if RSRB peers use different virtual ring numbers, an explorer (or, in fact, any frame) might traverse two different RSRB hops. For example, in Figure 8-1, the explorers originating in Router B go through Router A, onto the physical ring attached to Router A, and then back through Router A to Router C. The key point to remember is that data paths through the network are not obvious from a simple examination of the peer definitions. You need to look at ring-group numbers and virtual ring numbers.

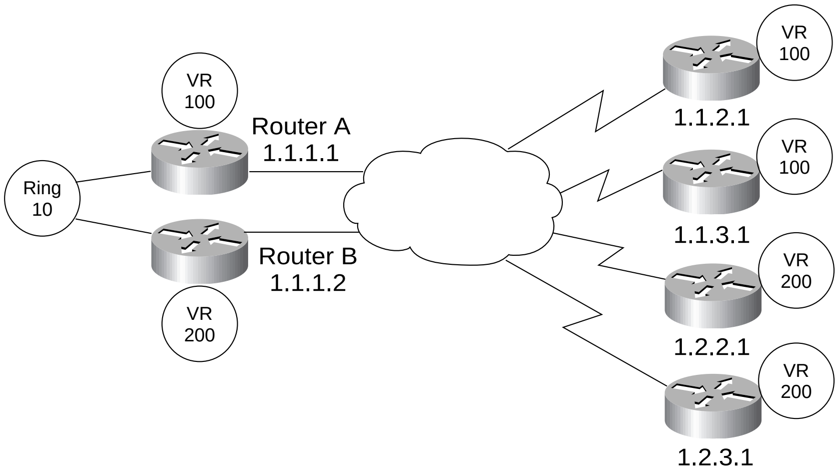
To quickly determine if your network has this characteristic, look for an RSRB router using two different ring-group numbers, as shown in the following configuration:

```
source-bridge ring-group 100
source-bridge remote-peer 100 tcp 1.1.1.1
source-bridge remote-peer 100 tcp 1.1.2.1
source-bridge ring-group 200
source-bridge remote-peer 200 tcp 1.1.1.1
source-bridge remote-peer 200 tcp 1.1.3.1
!
interface TokenRing 0
ip address 1.1.1.1 255.0.0.0
ring-speed 16
source-bridge 10 1 100
source-bridge spanning
!
interface TokenRing 1
no ip address
ring-speed 16
source-bridge 10 1 200
source-bridge spanning
```

Looking again at Figure 8-1 and the preceding configuration, both interfaces TokenRing 0 and TokenRing 1 connect to the same physical ring (Ring 10). This configuration allows two different ring-groups to share data. Data received from peer 1.1.2.1 on ring-group 100 would be put onto ring 10 through interface TokenRing 0. It would then be transferred into ring-group 200 through interface TokenRing 1 and sent to peer 1.1.3.1. Because there is a LAN hop in between (Ring 10) and because two separate ring-groups were used, peers 1.1.2.1 and 1.1.3.1 are able to exchange data even though they do not peer to each other.

A similar situation occurs even if the two ring-groups do not exist in the same router. Assume a situation where Router A and Router B are both connected to the same physical Token Ring via interface TokenRing 0 as shown in Figure 8-3.

Figure 8-3 RSRB with Multiple Routers Connected to the Same Physical Ring



Assume the following configuration on Router A:

```
source-bridge ring-group 100
source-bridge remote-peer 100 tcp 1.1.1.1
source-bridge remote-peer 100 tcp 1.1.2.1
source-bridge remote-peer 100 tcp 1.1.3.1
!
interface TokenRing 0
ip address 1.1.1.1 255.255.255.0
ring-speed 16
source-bridge 10 1 100
source-bridge spanning
```

Assume the following configuration for Router B:

```
source-bridge ring-group 200
source-bridge remote-peer 200 tcp 1.1.1.2
source-bridge remote-peer 200 tcp 1.2.2.1
source-bridge remote-peer 200 tcp 1.2.3.1
!
interface TokenRing 0
ip address 1.1.1.2 255.255.255.0
ring-speed 16
source-bridge 10 1 200
source-bridge spanning
```

Traffic flows from any peer in ring-group 100 to any peer in ring-group 200 through physical Ring 10.

Both of these examples are common in RSRB configurations. Ring 10 in both examples is called an isolation (or de-encapsulation) ring. In both cases, the devices incur additional overhead because data travels through two RSRB hops to get from one remote branch router to another remote branch router. The data is encapsulated and de-encapsulated in TCP/IP twice. In addition, if local acknowledgment is used, then the LLC2 sessions must be terminated an extra time. Often, this configuration is selected to combine two existing RSRB networks because it is an easier solution than reconfiguring an entire RSRB network to match the ring-group number of another.

If either of these types of RSRB configurations exist in a network, then they are excellent candidates for DLSw+ border peers and peer groups. Using DLSw+ peer groups you can maintain the connectivity previously described without the overhead of multiple encapsulation steps. It provides the best of both worlds.

If neither of these conditions exists in your RSRB network, it should be possible to simply migrate by copying the existing RSRB peer definitions to equivalent DLSw+ peer definitions. Although this migration preserves existing data connectivity in the network, it might not result in the optimal network design.

Migrating Steps for a Hierarchical Network

In a hierarchical network, any-to-any connectivity is not required. Typically there are one or a small number of data center routers to which all remote sites must connect. These types of networks are the simplest to migrate. DLSw+ and RSRB can both run in the same router at the same time, but you may prefer to use a new router for the migration. This section describes both options.

Separate Routers for RSRB and DLSw+ Peering

One option for migrating a hierarchical RSRB network to DLSw+ is to put separate DLSw+ routers in parallel with the existing RSRB peering routers at each central site location. In certain situations, this is the only way to go (these situations are discussed later in this paper). It requires extra equipment, but only for the duration of the migration process, at which time the equipment can be redeployed.

To migrate a hierarchical RSRB network to a hierarchical DLSw+ network using separate, parallel routers, do the following:

Step 1. Ensure that your routers are at Cisco IOS Release 10.3(2) or later (see Appendix B to determine which version of Cisco IOS Software you should install to get the features you want).

Step 2. Configure a **dlsw local-peer** command, specifying the **promiscuous** keyword, at the new data center DLSw+ routers. If desired, you can replace the **promiscuous** keyword with static **dlsw remote-peer** command at the end of the migration process.

Step 3. Select one remote site and delete the **source-bridge remote-peer** commands (and any related commands such as SDLLC, SR/TLB, proxy explorer, and NetBIOS name caching that are no longer required).

Step 4. Add the appropriate **dlsw local-peer** command and one or more **dlsw remote-peer** commands (at that same remote site) that point to the central site DLSw+ routers.

Step 5. Visit the RSRB device in the data center and remove the **source-bridge remote-peer** command that referred to the remote router you just modified.

Step 6. Repeat Steps 3 through 5 with the remaining remote sites.

When all of the RSRB remote peers have been migrated to DLSw+, no remote peers will remain connected to the RSRB peering router. This router can now be removed and reused elsewhere in the network.

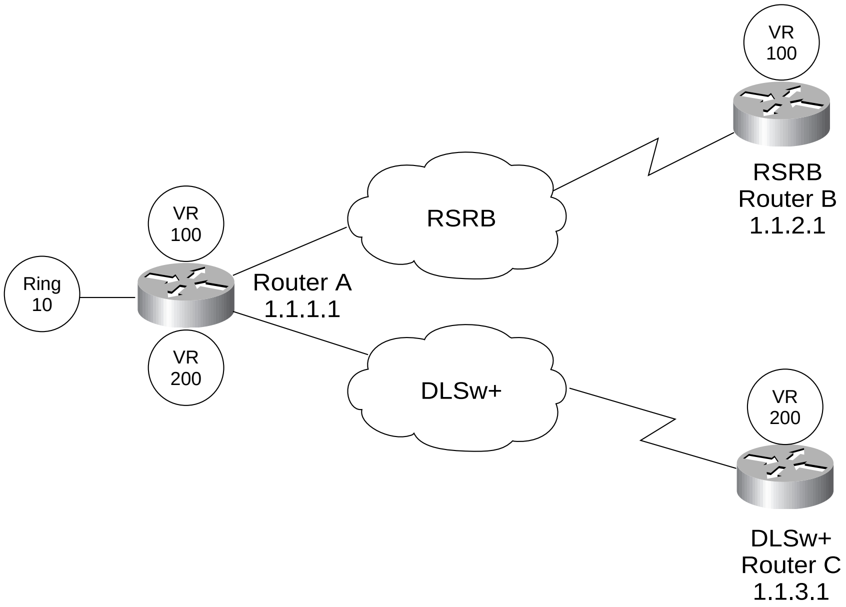
To keep explorers from going into the RSRB network from the DLSw+ network (or vice versa) during the course of the migration, the DLSw+ peers in the data centers should use the same source-bridge ring-group number as the RSRB peers with which they are being placed in parallel. The ring-group used at the remote sites does not matter as much, but it often is best to keep it the same, because this prevents explorer looping if there is an unknown back door data path. By reusing ring-group numbers, the SRB process recognizes packets that have already traversed the WAN via either RSRB or DLSw+ and discards these before they are passed back over the WAN.

RSRB and DLSw+ Concurrently in the Same Data Center Routers

Another option is to add DLSw+ to the existing RSRB peer devices running in the data center. DLSw+ was designed to run concurrently with RSRB in simple connectivity scenarios. However, technology limitations make this unfeasible in some situations. In general, between any pair of routers you should use either DLSw+ or RSRB, but not both, as shown in Figure 8-4. As in the previous example, migrate your RSRB network to DLSw+ one router at a time.

Figure 8-4 shows a sample central site configuration.

Figure 8-4 Central Site Router Configured to Communicate with Both an RSRB Router and a DLSw+ Router



Assume the following configuration on Router A:

```
source-bridge ring-group 100
source-bridge remote-peer 100 tcp 1.1.1.1
source-bridge remote-peer 100 tcp 1.1.2.1
!
dlsw local-peer peer-id 1.1.1.1
dlsw remote-peer 0 tcp 1.1.3.1
!
int tokenring 0
 source-bridge 10 1 100
 source-bridge spanning
```

To run DLSw+ and RSRB concurrently in data center peers, perform the following tasks:

Step 1. Migrate your routers to Cisco IOS Release 10.3(2) or later (see Appendix B to determine which version of Cisco IOS Software you should install to get the features you want).

Step 2. Configure a **dlsw local-peer** command using the same peering address as RSRB at the central site RSRB routers. For ease of migration, use the **promiscuous** keyword for the duration of the migration as discussed in the previous section.

Step 3. Select one remote site and remove the devices **source-bridge remote-peer** commands (and any related commands such as SDLLC, SR/TLB, proxy explorer, and NetBIOS name caching that are no longer required).

Step 4. Add a **dlsw local-peer** command and one or more **dlsw remote-peer** commands (at the same remote site) that point to the central site DLSw+ routers.

Step 5. At the central site RSRB device, remove the **source-bridge remote-peer** command that points to the device you just updated in Step 4.

Step 6. Repeat Steps 3 through 5 with the remaining remote sites. When all remote sites are migrated, remove the local **source-bridge remote-peer** statement (and **source-bridge fst-peername** statement) from the central site router.

Caveats

Using a single router to perform both RSRB and DLSw+ during the migration phase is not always possible or advisable. In some situations, you might end up with loops, and in other situations you might be unable to establish SNA sessions

If both RSRB and DLSw+ remote peers need to access resources locally attached to the peering router via a bridge-group, you should not run DLSw+ and RSRB in the same router. Because of limitations of transparent bridging (that is, the lack of a RIF), you might incur problems if accessing the same bridge-group from both DLSw+ peers and RSRB peers (via the SR/TLB function).

If there are multiple source-bridge ring-groups defined on the RSRB peer device, great care must be taken when adding DLSw+. In particular, do not attempt to put DLSw+ on any RSRB peer that implements a de-encapsulation ring (a ring on which the RSRB peer has two or more interfaces, each bridging traffic to a separate ring-group as shown in Figure 8-1). Because of the way DLSw+ learns resource reachability, this type of scenario will cause problems.

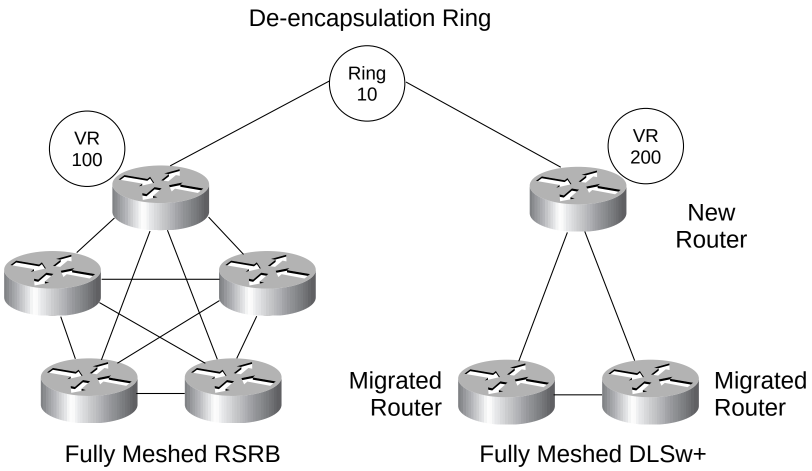
If your RSRB router configuration contains multiple source-bridge ring-groups but does not include a de-encapsulation ring, then DLSw+ and RSRB can both run in the same router as long as you use ring lists to control which ring-groups DLSw+ uses. Ring-lists are required because unlike RSRB, all DLSw+ peers are tied to every ring-group defined in a router.

If the RSRB peer device in the data center is doing reverse SDLLC or reverse QLLC translation (providing upstream connectivity to an SDLC- or QLLC-attached device), the SDLC or QLLC link cannot be shared by RSRB and DLSw+. It is possible to share the router in these cases, but RSRB and DLSw+ must each have its own separate connection to the SNA device in question.

Migrating Any-to-Any Networks

To migrate an any-to-any RSRB network to DLSw+, it is best to use a de-encapsulation ring, which is a physical Token Ring that enables traffic flow between RSRB and DLSw+ networks. Starting with a fully meshed RSRB network, you move routers, one at a time, to either a fully meshed DLSw+ network, as shown in Figure 8-5, or to a peer group network. The two networks support any-to-any connectivity by using the de-encapsulation ring to move data between RSRB peers and DLSw+ peers.

Figure 8-5 Using a De-encapsulation Ring and Multiple Routers to Migrate Any-to-Any Networks



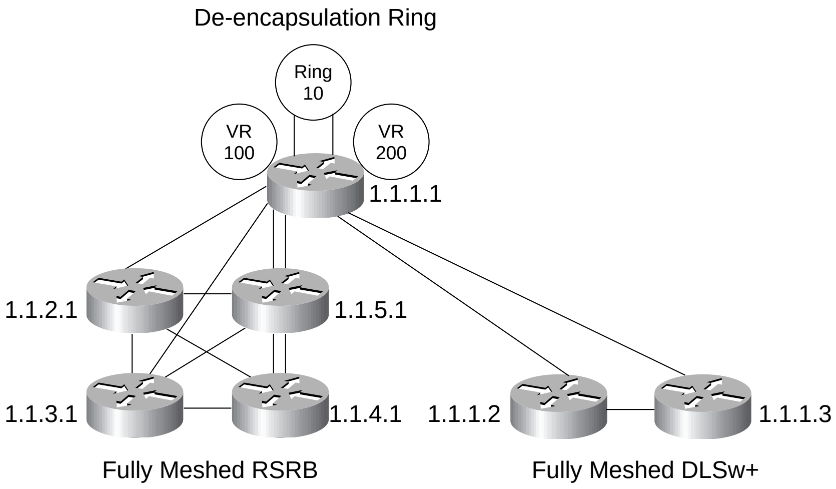
Because of the potential traffic volume on the de-encapsulation ring, this ring should be dedicated to this task. Do not use an existing ring that is handling other traffic. Additionally, during the migration there will be a heavy load on the routers connected to the de-encapsulation ring, especially where local acknowledgment is used. If there is not a lot of delay in the RSRB portion of the network, do not configure RSRB peers for local acknowledgment. This setup will minimize the overhead in the RSRB routers.

When using this type of migration, the DLSw+ peers on the de-encapsulation ring must use a different ring-group number than is used for the RSRB network. Otherwise, traffic will be unable to traverse the de-encapsulation ring from RSRB to DLSw+ or vice versa. As a result of this stipulation and the fact that DLSw+ terminates the RIF, it is important to maintain only one de-encapsulation ring, because explorers would otherwise jump from RSRB to DLSw+ and back again, increasing the overall explorer load on the network.

The de-encapsulation ring will be a single point of failure during the migration. To minimize the impact of this, you can use an intelligent hub that senses failures and wrap-around ports.

In Figure 8-5, a new router was added (attached to the de-encapsulation ring) for simplicity and to ensure that it could handle the load. Alternately, a single router attached to the de-encapsulation ring could be configured for both RSRB and DLSw+. A sample of this is shown in Figure 8-6.

Figure 8-6 Using a De-encapsulation Ring and a Single Router to Migrate Any-to-Any Networks



The following is a sample configuration for the router attached to the isolation ring in Figure 8-6:

```
source-bridge ring-group 100
source-bridge remote-peer 100 tcp 1.1.1.1
source-bridge remote-peer 100 tcp 1.1.2.1
source-bridge remote-peer 100 tcp 1.1.3.1
source-bridge remote-peer 100 tcp 1.1.4.1
source-bridge remote-peer 100 tcp 1.1.5.1

source-bridge ring-group 200
dlsw local-peer peer-id 1.1.1.1
dlsw port-list 1 TokenRing 1
dlsw remote-peer 1 tcp 1.1.1.2
dlsw remote-peer 1 tcp 1.1.1.3
interface TokenRing 0
 source-bridge 10 1 100
 source-bridge spanning
interface TokenRing 1
 source-bridge 10 1 200
 source-bridge spanning
```

Because RSRB remote peers are tied to a specific ring-group (in this case 100), only traffic from interface TokenRing 0 goes to RSRB peers, and traffic from RSRB peers only go out interface TokenRing 0. Because a port list is specified on all the DLSw+ peers, only traffic from interface TokenRing 1 goes to DLSw+ peers, and traffic from DLSw+ peers only go out interface TokenRing 1. If both of these interfaces are connected to the same physical ring, communication between the RSRB and DLSw+ domains is possible (as is desired during an any-to-any network migration).

Fully Meshed RSRB Network to Fully Meshed DLSw+ Network

If your current RSRB routers peer to every other RSRB router (that is, if you have a fully meshed RSRB network), this section describes the easiest way to migrate your network to a fully meshed DLSw+ network. However, you will have a better performing network if you use peer groups instead of a fully meshed DLSw+ network. You might decide to move to that type of design in the future, in which case this example can be viewed simply as the first step, where the second step adds DLSw+ border peers.

First, determine the location of the de-encapsulation ring. Your key goal is to minimize WAN impact. During the migration, all LLC2 sessions that need to traverse from an RSRB site to a DLSw+ site must pass through this de-encapsulation ring. Traffic might have to traverse the WAN into this site and then traverse it again to reach the destination site.

It is possible to use an existing router to attach to this new de-encapsulation ring if the device is being utilized lightly enough and can handle the extra load. Unless a network administrator is confident about the traffic patterns in the network, the ability of an existing router to handle the extra load can be very difficult to determine in advance. Observe the router during the course of the migration to determine when it is getting overloaded and needs some help. Important observations are CPU utilization, buffer utilization (tuning might be useful), and WAN link utilization to this site. Also observe the DLSw+ peers on this de-encapsulation ring, especially CPU utilization because the LLC2 session maintenance is a costly operation.

To migrate a fully meshed RSRB network to a fully meshed DLSw+ network using an existing RSRB router, do the following:

Step 1. Ensure that your routers are at Cisco IOS Release 10.3(2) or later (see Appendix B to determine which version of Cisco IOS Software you should install to get the features you want).

Step 2. Configure a **dlsw local-peer** command, specifying the **promiscuous** keyword, at the router attached to the de-encapsulation ring. If desired, at the end of the migration process the static **dlsw remote-peer** command can replace the **promiscuous** keyword. Also add a **dlsw port-list** command to keep RSRB and DLSw+ traffic separate.

Step 3. Select an RSRB router to be migrated and delete the **source-bridge remote-peer** commands (and any related commands such as SDLLC, SR/TLB, proxy explorer, and NetBIOS name caching that are no longer required).

Step 4. Add the appropriate **dlsw local-peer** command and a **dlsw remote-peer** command (at the same router) that is pointing to the DLSw peer on the de-encapsulation ring. On this **dlsw remote-peer** command, specify the same port-list number specified in Step 2.

Step 5. Add a **dlsw remote-peer** command pointing to every router already converted to DLSw+.

Step 6. Add a **dlsw remote-peer** command pointing to the router you just converted to DLSw+ to every router already converted to DLSw+.

Step 7. Remove the **source-bridge remote-peer** command that referred to the router you just modified from all the RSRB routers, including the RSRB router on the de-encapsulation ring.

Step 8. Repeat Steps 3 through 7 with the remaining RSRB routers.

You do not have to perform Step 7 immediately. RSRB is slightly degraded until it is done because RSRB attempts to connect to a peer that is no longer there. However, it will continue to work for all existing RSRB devices. Because Steps 6 and 7 require visiting the configuration of all devices in the network, you might migrate a batch of routers and do the corresponding updates for the entire batch at once.

To migrate a fully meshed RSRB network to a fully meshed DLSw+ network using a new router for DLSw+, do the following:

Step 1. Ensure that your routers are at Cisco IOS Release 10.3(2) or later (see Appendix B to determine which version of Cisco IOS Software you should install to get the features you want).

Step 2. Attach the new router and your co-located RSRB routers to the de-encapsulation ring. At the new router, configure a **dlsw local-peer** command with the **promiscuous** keyword specified. If desired, at the end of the migration process the **promiscuous** keyword can be replaced with static **dlsw remote-peer** command definitions.

Step 3. Select an RSRB router to be migrated and delete the **source-bridge remote-peer** commands (and any related commands such as SDLLC, SR/TLB, proxy explorer, and NetBIOS name caching that are no longer required).

Step 4. Add the appropriate **dlsw local-peer** command and a **dlsw remote-peer** command (at the same router) pointing to the dlsw peer on the de-encapsulation ring.

Step 5. Add a **dlsw remote-peer** command pointing to every router already converted to DLSw+.

Step 6. Add a **dlsw remote-peer** command pointing to the router you just converted to DLSw+ to every router already converted to DLSw+.

Step 7. Remove the **source-bridge remote-peer** command that referred to the router you just modified from all the RSRB routers, including the RSRB router on the de-encapsulation ring.

Step 8. Repeat Steps 3 through 7 for the remaining RSRB routers.

You do not have to perform Step 7 immediately. RSRB is slightly degraded until it is done because RSRB attempts to connect to a peer that is no longer there. However, it will continue to work for all existing RSRB devices. Because Steps 6 and 7 require visiting the configuration of all devices in the network, you might choose to migrate a batch of routers and do the corresponding updates for the entire batch at once.

At some point in the migration, you might find that either the one RSRB device or the one DLSw+ device on the de-encapsulation ring is insufficient to handle the load required of it. Because the RIF is not terminated in RSRB, it is safe to add additional RSRB peers to the de-encapsulation ring to help divide the load. Because all the RSRB peers within the same cloud share a ring-group number, it is clear that one peer will not read in frames originating from another RSRB peer; furthermore, no RSRB peer will put a frame onto the de-encapsulation ring that has already traversed it. Additional DLSw+ peers can be added to this ring to handle additional load as well; however, more caution must be used than in the RSRB case. DLSw+ permits different ring-group numbers to be used in the same DLSw+ cloud. It is important that all DLSw+ peers sharing this ring use the same ring-group number to avoid explorer looping. Also, because DLSw+ terminates the RIF, it is unable to determine whether an explorer it receives via its peers has traversed the ring already. Because of this situation, it is important that DLSw+ peers not peer directly to each other if they are both directly attached to the de-encapsulation ring.

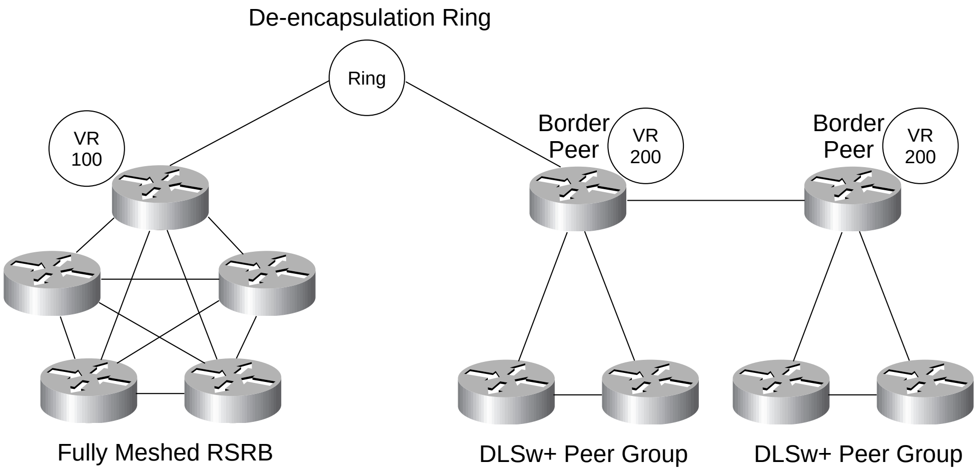
Before adding additional RSRB or DLSw+ peers on the de-encapsulation ring, it might be useful to examine the traffic patterns traversing the ring. It is possible that by identifying a particular RSRB site that is generating a lot of traffic to one of the DLSw+ sites that has already been migrated, the situation can be alleviated by migrating that RSRB site to DLSw+ to reduce this traffic flow. If possible, this setup would have the additional advantage of removing the requirement for multiple encapsulation steps for a large number of sessions, which should improve response time and overall network performance.

Fully Meshed RSRB Network to DLSw+ Peer Groups

Before beginning this migration, refer to the chapter "Designing Meshed Networks" to determine how you want to design the DLSw+ peer groups. After you have determined which routers will be border peers and which routers will belong to each peer group, you can begin the migration. Border peers should be migrated before any member peers in that group. You can migrate one group at a time or you can migrate all border peers first.

Many of the concepts discussed in the previous section still apply here. You still have a de-encapsulation ring as shown in Figure 8-7. Any DLSw+ peer on the de-encapsulation ring is configured as a border peer in its own group (there are no other peers in this peer group.) If additional DLSw+ peers are required to handle the load, then each one will be configured as a border peer in its own group (they will not share the same group number). Border peers on the de-encapsulation ring will not be peered to each other. (This rule is an exception to the general rule that all border peers in a DLSw+ network should be peered to each other.)

Figure 8-7 Migrating Fully Meshed RSRB to DLSw+ Peer Groups



This example uses new routers as DLSw+ border peers for the migration process and migrates one peer group at a time. To migrate a fully meshed RSRB network to a DLSw+ network with peer groups, do the following:

Step 1. Ensure that your routers are at Cisco IOS Release 10.3(2) or later (see Appendix B to determine which version of Cisco IOS Software you should install to get the features you want).

Step 2. Attach the new router and your co-located RSRB routers to the de-encapsulation ring. At the new router, configure a **dlsw local-peer** command with the **promiscuous**, **border**, and **group** keywords specified. If desired, at the end of the migration process the **promiscuous** keyword can be replaced with static **dlsw remote-peer** definitions.

Step 3. Select an RSRB router to be migrated and delete the **source-bridge remote-peer** commands (and any related commands such as SDLLC, SR/TLB, proxy explorer, and NetBIOS name caching that are no longer required).

Step 4. Add the appropriate **dlsw local-peer** command and a **dlsw remote-peer** command (at the same router) to point to the border peer configured in Step 2. The **group** keyword should be specified on the **dlsw remote-peer** command to indicate that this new peer belongs to the same peer group as the border peer to which it is being peered.

Step 5. Remove the **source-bridge remote-peer** command that referred to the router you just modified from all the remaining RSRB routers, including the RSRB router on the de-encapsulation ring.

Step 6. Repeat Steps 3 through 5 for the remaining RSRB routers that will belong to this peer group.

Step 7. Add another border peer. This example assumes it is a new router. This router should not be attached to the de-encapsulation ring. At the new router, configure a **dlsw local-peer** command with the **promiscuous**, **border**, and **group** keywords specified. The group number configured for this new peer should differ from that used for other border peers already configured. If desired, at the end of the migration process the promiscuous keyword can be replaced with static **dlsw remote-peer** command definitions.

Step 8. Configure a **dlsw remote-peer** command pointing to every other border peer already configured.

Step 9. Add a **dlsw remote-peer** command pointing to this new border peer in every border peer already configured,

Step 10. Repeat Steps 3 through 5 for all RSRB peers that will belong to the same peer group as this new border peer.

Step 11. Repeat Steps 7 through 10 for all remaining peer groups and border peers.

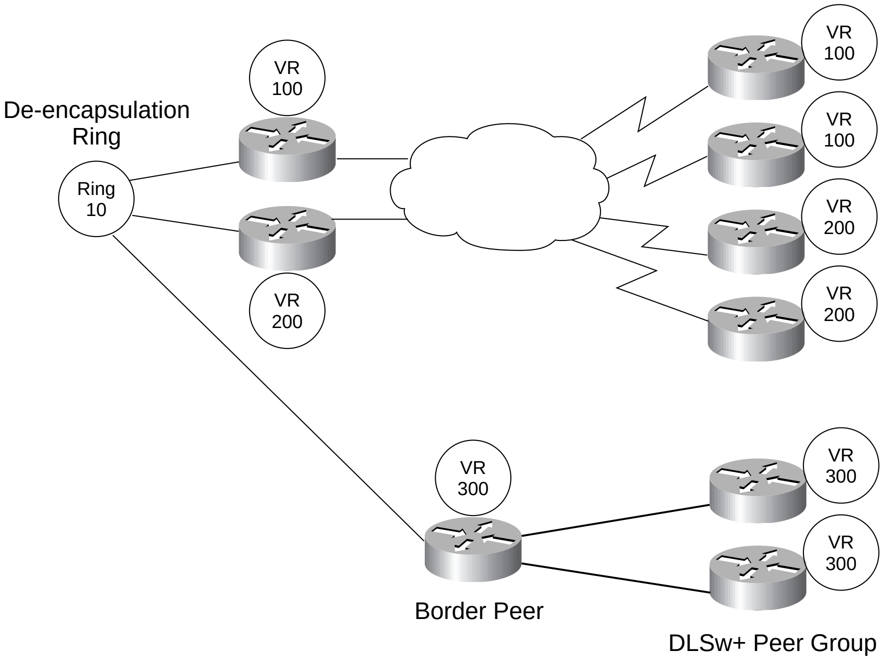
Note that for the majority of new DLSw+ peers (those that are not to be border peers), there are no configuration changes required for any other DLSw+ devices already converted (because the border peer in the group accepts the peer connection promiscuously). For the border peers, the number of previously converted DLSw+ peers that need to be touched is limited to a small subset (only the previously converted border peers).

Multihop RSRB to DLSw+ Peer Groups

For some RSRB environments that required any-to-any communication, full peer meshing was not possible because there were too many peers. For that reason, some large RSRB networks already use the concept of a de-encapsulation ring in order to connect two RSRB clouds using different ring-group numbers, as shown in Figure 8-8. When migrating to DLSw+ from this type of network, the key points to remember are as follows:

- The DLSw+ ring-group number should be different from any of the ring-group numbers being used for RSRB.
- Even if multiple de-encapsulation rings are being used, DLSw+ can connect into only one of them. RSRB supports multiple de-encapsulation rings because it does not terminate the RIF; DLSw+ cannot do this because of RIF termination (unless you are running Cisco IOS Release 12.0 and using RIF passthru).
- Migrate one RSRB ring-group before starting another. Hopefully the RSRB ring-groups were designed so that the majority of traffic flow remained within the same ring-group, with a minority having to traverse ring-groups. By migrating one ring-group before moving onto the next, less stress should be put on the routers attached to the de-encapsulation ring.

Figure 8-8 Migrating from Multihop RSRB to DLSw+ Peer Groups



This example uses new routers as DLSw+ border peers for the migration process and migrates one peer group at a time. To migrate a multihop RSRB network to a DLSw+ network with peer groups, do the following:

Step 1. Ensure that your routers are at Cisco IOS Release 10.3(2) or later (see Appendix B to determine which version of Cisco IOS Software you should install to get the features you want).

Step 2. Attach the new router and your co-located RSRB routers to the de-encapsulation ring. At the new router, configure a **dlsw local-peer** command with the **promiscuous**, **border**, and **group** keywords specified. If desired, at the end of the migration process the **promiscuous** keyword can be replaced with static **dlsw remote-peer** command definitions.

Step 3. Select an RSRB router to be migrated and delete the **source-bridge remote-peer** commands (and any related commands such as SDLLC, SR/TLB, proxy explorer, and NetBIOS name caching that are no longer required).

Step 4. Add the appropriate **dlsw local-peer** command and a **dlsw remote-peer** command (at the same router) that points to the border peer configured in Step 2. The **group** keyword should be specified on the **dlsw remote-peer** command to indicate that this new peer belongs to the same peer group as the border peer to which it is being peered.

Step 5. Remove the **source-bridge remote-peer** command that referred to the router you just modified from all the remaining RSRB routers, including the RSRB router on the de-encapsulation ring.

Step 6. Repeat Steps 3 through 5 for the remaining RSRB routers in the same ring-group.

Step 7. Add another border peer. This example assumes it is a new router. This router should not be attached to the de-encapsulation ring. At the new router, configure a **dlsw local-peer** command with the **promiscuous**, **border**, and **group** keywords specified. The group number configured for this new peer should differ from that used for other border peers already configured. If desired, at the end of the migration process the promiscuous keyword can be replaced with static **dlsw remote-peer** command definitions.

Step 8. Configure a **dlsw remote-peer** command pointing to every other border peer already configured.

Step 9. Add a **dlsw remote-peer** command pointing to this new border peer in every border peer already configured.

Step 10. Repeat Steps 3 through 6 for all RSRB peers that will belong to the same peer group as this new border peer.

Step 11. Repeat Steps 7 through 10 for all remaining peer groups and border peers.

Multivendor Interoperability

You can configure a Cisco DLSw+ router to communicate with a non-Cisco router. However, not all the DLSw+ features will be available. Table 8-2 illustrates the features in DLSw+ that are beyond what the standard offers. Some of the DLSw+ features can be used (with some restrictions) even if the peer at the other end is not a Cisco router.

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 and base-RFC 2166 levels only. Cisco has tested interoperability with several vendors. Contact Cisco to find out the latest status of this interoperability testing.

Table 8-2 Comparison of DLSw+ and Standard DLSw Features

	DLSw Standard Feature	Additional DLsw+ Features
Performance	IP load sharing Circuit-level flow control	Peer[1], port and RIF load sharing Custom and priority queuing Weighted fair queuing and weighted randon early detection ToS/COS mapping RSVP support FST encapsulation

Table 8-2 Comparison of DLSw+ and Standard DLSw Features (Continued)

Availability	Nondisruptive rerouting No data-link control timeouts	Backup peers[1] Fault tolerant peers[1] Load sharing across peers
Scalability	Broadcast reduction Hop count reduction UI/UDP support Multicast	Broadcast optimization with peer groups Border peer caching Ring lists[1]
Flexibility	Media conversion SRB dynamics Capabilities exchange for cache preloading	Peer biasing with cost[1] SNA DDR Diverse data-link control media (QLLC, Reverse SDLLC, Token Ring LANE, Token Ring ISL, SRB over FDDI) Media conversion between SDLC and LLC2 for PU 4-to-PU 4 Dynamic peers DLSw Lite

[1] Can be used with a non-Cisco router

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

- 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 might 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 load balances between two central site non-Cisco routers as long as the Cisco router initiates the CANUREACH exchange; a Cisco router also locally load balances across all interfaces.
- Cisco routers automatically connect to a backup peer upon loss of a primary peer even if the backup peer is a non-Cisco router. 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 automatically terminates the backup peer connection according to the configuration options.
- Cisco routers establish a dynamic peer connection with a remote non-Cisco peer as long as that remote peer accepts either a connection from an unknown peer or support something equivalent to the passive keyword.
- Cisco routers 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 or are specified in the RFC:

dlsw local-peer [**peer-id** *ip-address*] [**lf** *size*] [**keepalive** *seconds*] [**passive**] [**promiscuous**] [**biu-segment**] [**init-pacing-window** *size*] [**max-pacing-window** *size*]

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 as vendor-specific vectors):

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 causes a Cisco router to open four TCP queues, which another vendor's router might not understand or accept. Whether the **dynamic** keyword works 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 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-address**
- **dlsw netbios-name**
- **dlsw icanreach**
- **dlsw load-balance**

CHAPTER 9

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

The following sample shows output from a **show dlsw capabilities** command issued for a local peer:

```
milan#show dlsw capabilities local
DLSw: Capabilities for peer 1.1.1.6(2065)
  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
  UDP Unicast support     : yes
  local-ack configured    : yes
  priority configured     : no
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* | **circuit id**]

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

circuit id—Display the circuit id of the circuit index

The following sample shows output from a **show dlsw circuits** command:

```
milan#show dlsw circuits detail
Index   local addr(lsap)    remote addr(dsap)   state uptime
194 0800.5a9b.b3b2(F0)  800.5ac1.302d(F0)  CONNECTED 00:00:13
        PCEP: 995AA4     UCEP: A52274
        Port: To0/0      peer 172.18.15.166(2065)
        Flow-Control-Tx SQ CW:20, Permitted:28; Rx CW:22, Granted:25 Op:
IWO
        Congestion: LOW(02), Flow Op: Half: 12/5 Reset 1/0
        RIF = 0680.0011.0640
```

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.

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.

The following sample shows output from **show dlsw fastcache** command:

```
milan#show dlsw fastcache
peer               local-mac       remote-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. Before 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

The following sample shows output of this 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 can 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 tells 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* | **udp**]

Syntax Description

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

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

udp—UDP frame forwarding statistics for specified peers

The following sample shows output of the **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 usable, **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 [**group** [*value*]] | **local** | **remote**] | [**mac-address** [*address*]] [**netbios-names** [*name*]]

Syntax Description

group—Displays contents of group reachability cache only

local—Specifies the group number for the reachability check. Only displays group cache entries for the specified group

remote—Displays contents of local reachabilty cache only

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

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

The following sample shows output of the **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. The following sample shows output from a **show interface** command:

```
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. The following sample shows output of a **show llc2** command:

```
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 dlsw transparent cache**
- **show dlsw transparent map**
- **show dlsw transparent neighbor**

- **show frame-relay map**
- **show frame-relay pvc**
- **show lnm station**
- **show interfaces accounting**
- **show ip rsvp request**
- **show ip rsvp reservation**
- **show ip rsvp sender**

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 makes 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 [border-peers [interface *interface* | **ip address** *ip-address*] | **core**
 [flow-control | **messages** | **state** | **xid]** [*circuit-number*] | **local-circuit** *circuit-number* | **peers**
 [interface *interface* **[fast-errors** | **fast-paks]** | **ip address** *ip-address* **[fast-errors** |
 fast-paks | **fst-seq** | **udp]]** | **reachability [error** | **verbose] [sna** | **netbios]]**

Syntax Description

border-peers—Enables debugging output for border peer events.
interface—Specifies a remote peer to debug by a direct interface.
ip address—Specifies a remote peer to debug by its IP address.
core—Enables debugging output for DLSw core events.
flow-control—Enables debugging output for congestion in the WAN or at the remote end station.
messages—Enables debugging output of core messages—specific packets received by DLSw either from one of its peers or from a local medium via the Cisco link services interface.
state—Enables debugging output for state changes on the circuit.
xid—Enables debugging output for the exchange identification-state machine.
circuit-number—Specifies the circuit for which you want core debugging output to reduce the of output
local-circuit—Enables debugging output for circuits performing local conversion. Local conversion occurs when both the input and output data-link connections are on the same local peer and no remote peer exist.
peers—Enables debugging output for peer events.
fast-errors—Debugs errors for fast-switched packets.
fast-paks—Debugs fast-switched packets.
fst-seq—Debug FST sequence numbers on fast switched packets.
udp—Debug UDP packets.
reachability—Enables debugging output for reachability events (explorer traffic). If no options are specified, event-level information is displayed for all protocols.
error | **verbose—**Specifies how much reachability information you want displayed. The **verbose** keyword displays everything, including errors and events. The **error** keyword displays error information only. If no option is specified, event-level information is displayed.
sna | **netbios—**Specifies that reachability information be displayed for only SNA or NetBIOS protocols. If no option is specified, information for all protocols is displayed.

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 cuts 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 [**flow-control** | **messages** | **state** | **xid**]

Syntax Description

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 provides information about congestion in the WAN or at the remote end station. If the WAN or remote station is congested, DLSw+ sends 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 Cisco Link 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 this command. The syntax of the **debug dlsw local-circuit** command is:

debug dlsw 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* [**fast-errors** | **fast-paks**] | **ip address** *ip-address* [**fast-errors** | **fast-paks** | **fst-seq** | **udp**]]

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)
peers—Enables debugging output for peer events
fast-errors—Debugs errors for fast-switched packets
fast-paks—Debugs fast-switched packets
fst-seq—Debug FST sequence numbers on fast switched packets
udp—Debug UDP packets

The following sample shows output from a **debug dlsw peers** command during a normal peer connect sequence, displayed from the router that initiated the 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)
```

The following sample shows output from a **debug dlsw peers** command during a normal peer connect sequence, displayed from the router that received the peer connection request:

```
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)
```

The following sample shows output from a **debug dlsw peers** command during a normal peer disconnect sequence:

```
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 and 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 does not check which protocol a message is related to before printing it.

The following example shows that DLSw+ is receiving TEST frames on the Ethernet interface:

```
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
```

DLSw+ puts 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).

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.

CHAPTER 10

Using CiscoWorks Blue: Maps, SNA View, and Internetwork Status Monitor

This chapter briefly describes some of the enhanced network management tools available for the management of 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 provides an end-to-end view of SNA sessions that traverse the DLSw+ network by correlating SNA PU and LU names with DLSw+ circuits and DLSw+ peers. CiscoWorks Blue Internetwork Status Monitor (ISM) support allows you to manage your router network from the mainframe console.

Note: ISM Version 2.0 works only with IBM's Tivoli NetView for OS/390 (NetView). Earlier releases of ISM also worked with Sterling's SOLVE:Netmaster, which is now Computer Associates' NetworkIT NetMaster. This document discusses the functionality of the latest release, ISM Version 2.0. However, much of the information also applies to the earlier releases.

Using Maps and SNA View

Quite often, the challenge with network management is that there is too much information. Maps and SNA View address this problem by providing information relevant to the problem you are trying to solve in an easy-to-use, graphical interface or a tabular Web interface. Maps and SNA View allow you to correlate DLSw+ circuits and peer connections with PU and LU names. A mainframe component queries VTAM to build a database of SNA PUs and LUs and maintains this database by capturing VTAM messages that indicate state changes. See the "CiscoWorks Blue Maps and SNA View User Guide" for more details. Also see the section "Using the DLSw Application in Maps" within the guide for instructions on how to use the application for DLSw+.

Note: DLSw+ routers must be IP-addressable to be viewed using Maps and you must use either FST or TCP encapsulation to show peers.

CiscoWorks Blue Maps offers both Motif-based and Web-based network management applications.

Managing DLSw+ in Motif-Based Mode

CiscoWorks Blue Maps offers a set of network management applications that use the X Window System and Motif graphical interfaces to display graphical maps of the nodes and links in your network. Each application focuses on a particular protocol: DLSw, RSRB, or APPN. See the "CiscoWorks Blue Maps and SNA View Workstation Installation and Administration Guide" for more details.

Managing DLSw in Web-Based Mode

CiscoWorks Blue Maps offers a set of Web-based client/server applications that let you use Web browsers to display information about DLSw, RSRB and APPN networks. The network information is presented in a tabular format. The Web server runs on your Maps UNIX workstation and collects information from the Cisco routers in the network. You can use a Web browser from any workstation in the network to connect to the Web server to view the network. 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. See the "CiscoWorks Blue Maps and SNA View User Guide" for more details.

Using Internetwork Status Monitor

In Cisco IOS Release 11.0 and later, every Cisco router that shipped with the IBM software feature set also shipped with an SNA service point capability. The service point capability allows a Cisco router to communicate directly to NetView. ISM uses this capability to monitor your Cisco devices from the mainframe.

To use the service point, you must configure your router to use the SNA host function. You must also configure VTAM to recognize the ISM router as a service point 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 ISM function. NetView uses this SSCP-to-PU session to establish an SNA session with the router. ISM uses this SNA session to send commands to the router or receive messages from the router.

If you configure the service point in your routers, your routers send SNA alerts directly to VTAM, greatly improving visibility from your mainframe network management applications. To use ISM in conjunction with DLSw+ requires Cisco IOS Release 11.1(5).

ISM allows you to access the command-line interface of a Cisco router from your NetView console. Using ISM, you can issue any command from your NetView console that you can issue from a Telnet interface to the router. You can configure the router, issue **show** commands, and issue **debug** commands.

ISM is an ideal solution for SNA environments that are just beginning to deploy multiprotocol networks and have NetView expertise but not Simple Network Management Protocol (SNMP) expertise. It minimizes training and equipment costs for network management while providing status monitoring, dynamic alerting, and the gathering of statistical information for trend analysis and capacity planning. The router interface through ISM 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] database).

See the "CiscoWorks Blue ISM User Guide" for more details.

CHAPTER 11

Using DLSw+ with Other Features

This chapter describes how to use DLSw+ in conjunction with other Cisco IOS Software features: SNA Switching Services (SNASw), DSPU, NCIA, and LAN Network Manager. 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 SNASw

DLSw+ TCP encapsulation supports SNASw, Cisco's second-generation APPN platform in Cisco IOS Release 12.1 and higher (DLSw+ FST does not support SNASw). SNASw, available in Cisco IOS Release 12.1 and later, replaces the function provided by the APPN network node (NN) feature of Cisco IOS Software. Cisco is discontinuing the APPN NN feature in Cisco IOS Software beginning with Release 12.1 because of its architectural complexity, scaling limitations, and manageability issues. Older versions of Cisco IOS Software prior to Release 12.1 are reaching their end of engineering support as follows:

- *Release 11.2*—April 16, 2001
- *Release 12.0*—After March 2002

This section discusses why and where SNASw is required, explains the circumstances in which you would run SNASw and DLSw+ together in the same router, and provides an example of how to configure it. SNASw and DLSw+ positioning and interoperability are extensively covered in the *SNASw Design and Implementation Guide* at www.cisco.com/warp/public/cc/pd/ibsw/snasw/tech/snasw_rg.pdf.

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 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 can dynamically reroute around link failures without disrupting SNA sessions.
- APPN Enterprise Extender (EE) can enable SNA transport over native IP between mainframe hosts (using APPN Extended Border Node) or host-to-peripheral SNA devices (using SNASw EE support).

Where and Why to Use SNASw with DLSw+

When the IBM APPN architecture was first defined in the mid-1980s, it supported only LU 6.2 applications. Because most applications were subarea SNA 3270 applications, there was limited migration toward APPN in corporate networks. In VTAM V4R2, 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 SNA application in your network.

SNASw provides DLUR support, which allows SNASw to connect to an upstream DLUS server on a NN server host. This provides dependent PU 2.0 support for peripheral SNA devices.

Where SNASw is implemented depends on the problem you are trying to address. It can be implemented in the data center, distribution layer, or branch.

SNASw in the Data Center

In multihost environments and hosts with multiple logical partitions (LPARs), SNASw allows you to reduce costs and enhance performance by minimizing your dependency on FEPs and NCP software, while migrating to a Cisco CIP/CPA or Catalyst 6500 switch Gigabit Ethernet-attached to an IBM Open Systems Adapter-Express (OSA-Express) on an IBM S/390 or zSeries host.

SNASw and HPR in the data center also allow you to enhance SNA application availability by taking advantage of the capabilities of an IBM Parallel Sysplex (which requires APPN and HPR). This implementation provides the functions required to support necessary SNA routing of client sessions directly to the target application host in addition to providing DLSw+ peer termination points for WAN transport of SNA from remote SNA clients.

By running DLSw+ in the same router as SNASw, you can use the existing DLSw+ network to transport SNA traffic from remote sites over the WAN to the data center and use Cisco's SNASw implementation in the data center to handle SNA routing of client sessions directly to the correct SNA application host, handle DLUR processing for dependent SNA devices, and provide EE support to natively transport the SNA traffic over IP into the IBM S/390 or zSeries host.

SNASw in the Distribution Layer

SNASw can reduce FEP requirements at regional distribution sites while maintaining SNA routing functionality. In multiple data center environments, you can use Cisco routers with SNASw functionality where you have FEPs today to support SNA application routing and preserve COS. Cisco routers cost considerably less than FEPs, are much easier to maintain, support more diverse LAN and WAN media, and can be used to support multiprotocol traffic such as voice over IP (VoIP) and multimedia. By limiting SNASw to the distribution layer and utilizing an existing DLSw+ for remote SNA transport over the WAN, you can minimize cost and avoid scalability issues while still getting the benefit of routing SNA client sessions directly to the target data center application host.

SNASw to the Branch

SNASw can be also deployed to the remote branch. In this case, SNASw can support SNA transport over native IP without any requirement to use DLSw+ for SNA WAN transport. This is accomplished using the SNASw EE feature. Additional details regarding this feature can be found in the *SNASw Design and Implementation Guide* at www.cisco.com/warp/public/cc/pd/ibsw/snasw/tech/snasw_rg.pdf.

VDLC Transport

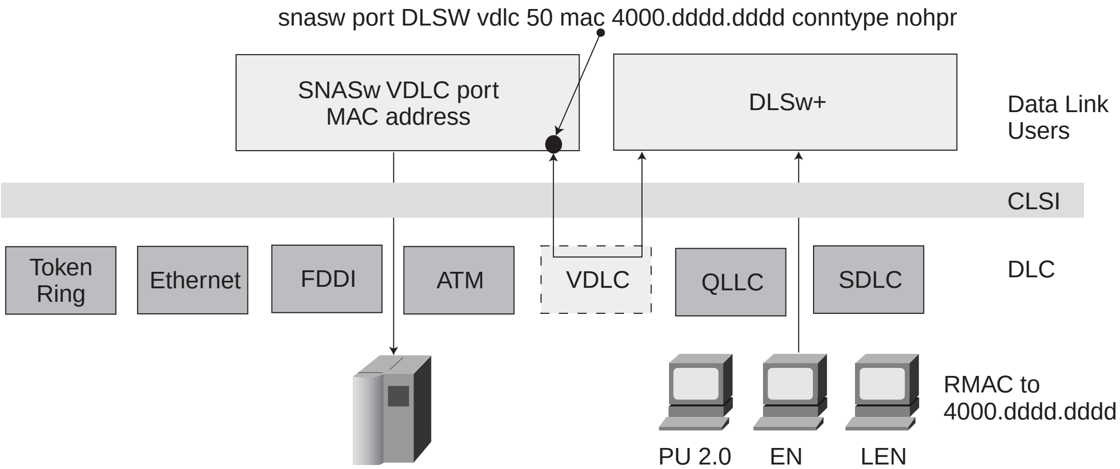
SNASw uses Virtual Data Link Control (VDLC) to connect to DLSw+ transport and local switching technologies. VDLC is used for a number of connectivity options, including:

- Transport over DLSw+ supported media
- DLC local switching support for access to SDLC and QLLC

Using VDLC, SNASw gains full access to DLSw+ SNA transport capabilities, including DLSw+ transport over IP networks, DLSw+ transport over direct interfaces, and DLSw+ support of direct Frame Relay encapsulation (without using IP). SNASw also gains access to devices connecting through SDLC and QLLC (see Figure 11-1).

Note: SDLC and QLLC are transparent to the SNASw code.

Figure 11-1 VDLC Transport



Configuration Details

To configure DLSw+ and SNASw interoperability you need to do the following:

- Configure DLSw+ TCP encapsulation (DLSw+ FST does not interoperate with SNASw because FST does not support VDLC)
- Configure the SNASw control point
- Configure SNASw to transport data over VDLC and define its virtual MAC address
- Configure SNASw links to the upstream NN server host and backup NN server
- Configure SNASw connection network to support dynamic links to all other upstream application hosts

Note: For additional information on SNASw DLSw+ configuration, see the *SNASw Design and Implementation Guide* and the following documentation:

- *Cisco IOS Bridging and IBM Networking Configuration Guide*, Release 12.1, “Configuring SNA Switching Services” (Cisco Documentation CD-ROM or www.cisco.com/univercd/cc/td/doc/product/software/ios121/121cgcr/ibm_c/bcprt2/bcdsnasw.htm)
- *Cisco IOS Bridging and IBM Networking Configuration Guide*, Release 12.1, “SNA Switching Services Commands” (Cisco Documentation CD-ROM or www.cisco.com/univercd/cc/td/doc/product/software/ios121/121cgcr/ibm_r2/br2prt1/br2dsnaw.htm)

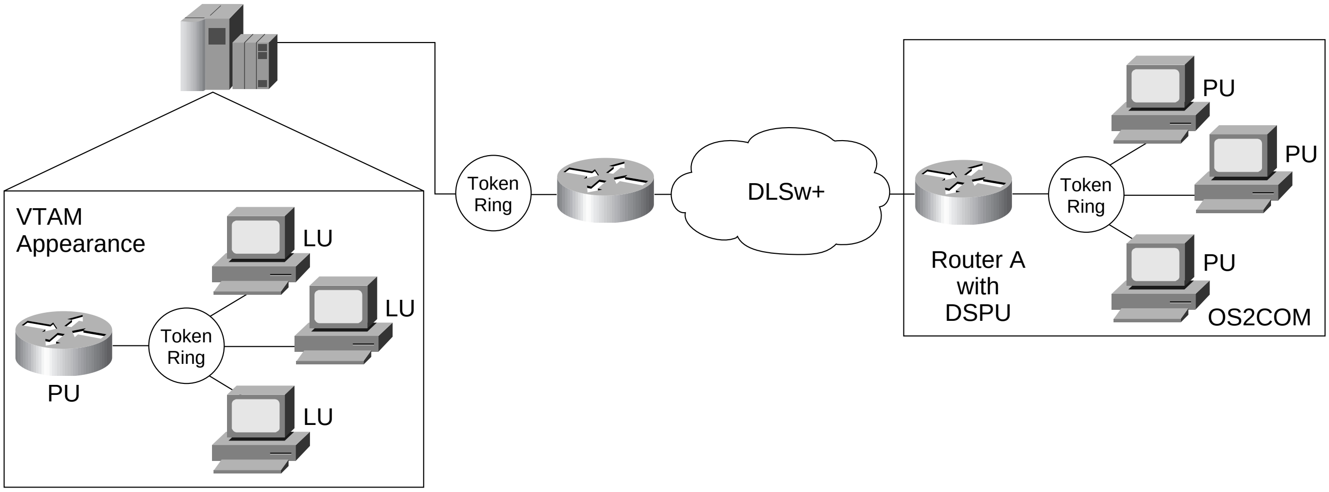
The following sample illustrates SNASw DLSw+ configuration (using DLSw+ local switching):

```
source-bridge ring-group 1072
dlsw local-peer
!

interface Serial0/1
 no ip address
 encapsulation sdlc
 no keepalive
 nrzi-encoding
 clockrate 9600
 sdlc role primary
 sdlc vmac 4000.3174.0000
 sdlc address C5
 sdlc xid C5 01722222
 sdlc partner 4000.4500.00f0 C5
 sdlc line-speed 9600
 sdlc dlsw C5
 !

snasw cpname NETA.R6072
snasw port DLSWPORT vdlc 1072 mac 4000.4500.00f0
snasw port SER0 hpr-ip Serial0/0
snasw link BMVS port SER0 ip-dest 192.168.237.129
```

Using DLSw+ with DSPU

DSPU is an old Cisco IOS IBM feature that historically has been used for consolidating multiple remote SNA 3270 workstations (running 3270 emulation software) or 3270 controllers with minimal SNA LU per PU requirements into a single upstream host (VTAM) PU appearance (see Figure 11-2). DSPU can consolidate devices up to the 255 SNA LU per PU IBM SSCP architectural limitation. In situations where a remote branch office has 20 to 30 (or more) PCs running 3270 emulation software, this could amount to a very large number of SNA PU resources that would need to be configured, enabled, and supported on the enterprise host (CS/390).

Figure 11-2 DSPU Concentrates SNA PUs



If a customer, for example, had 300 remote branch offices, this could equate to as many as 6000 to 9000 PUs in the network that would need to be configured in the host. In this case, DSPU could potentially reduce the CS/390 SNA PU configuration requirement to 300 PUs if each remote branch office had a total SNA LU requirement of 255 LUs per branch.

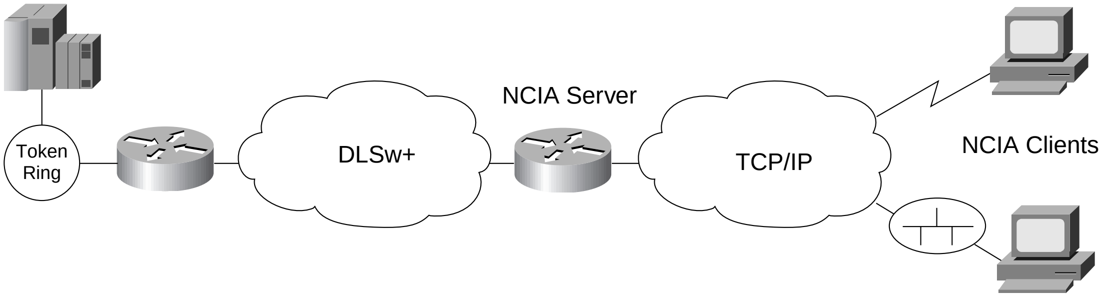
The maximum number of DSPU PUs that can be configured on a single DSPU router is 4096 in the latest versions of the Cisco IOS Software (Release 12.1 and higher).

There are many other Cisco IOS IBM features that can better be utilized to support SNA LU 2.0 green screen and 3270 emulation requirements (CIP/CPA TN3270 Server and SNASw DLUR support).

Using DLSw+ with NCIA

NCIA Server is an old Cisco IOS feature that connects NCIA clients over TCP/IP. The connection from the router running NCIA Server back to the data center host can utilize DLSw+ SNA WAN transport upstream. Figure 11-3 illustrates how NCIA interoperates with DLSw+.

Figure 11-3 NCIA Server Used with DLSw+



The NCIA Phase II architecture (RFC 2114) was originally developed by Cisco several years ago and submitted to the Internet Engineering Task Force (IETF) to address early customer requirements for SNA application access over IP backbone (the IETF specification refers to the architecture as the DLSw Remote Access Protocol [DRAP]).

TN3270 Server (supported on the CIP and CPA) provides similar functions to NCIA. Both allow client emulator software packages access to existing IBM mainframe applications, and both support TCP/IP as the backbone transport for these sessions. However, NCIA differs from TN3270 Server in that with NCIA, the client actually runs a full SNA emulator, as opposed to TN3270 Server where the client needs to run only TN3270 client software.

With NCIA, the client is a full SNA PU and LU. With TN3270, the SNA PU and LUs are located on the TN3270 Server. In both NCIA and TN3270 Server, the native protocol from the client PC is TCP/IP. The primary difference between these implementations is in the location of the SNA PUs and LUs. The NCIA approach is that the full SNA capability built into the SNA 3270 emulator itself is maintained.

The primary advantage of the TN3270 Server approach is that the definition and configuration are eased because each client looks like one or more LUs but does not require the definition of a PU. TN3270 Server is also an accepted industry standard with a very large installed base of devices supporting it (NCIA, on the other hand, is a technology on the way to being phased out).

Cisco DLSw+ does interoperate with NCIA Server Phase I.

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+.

DLSw+ Ethernet Redundancy Feature

This chapter describes how the DLSw+ Ethernet redundancy feature works, how to configure it, sample configurations, and configuration verification tips.

DLSw+ provides redundancy and load balancing when the end systems are connected over any media that support SRB: Token Ring, FDDI, Token Ring LAN Emulation (LANE), and Token Ring InterSwitch Link (ISL). When the end systems are connected to Ethernet, however, DLSw+ has design limitations in providing redundancy and load balancing. See Appendix C, "Ethernet DLSw+ Redundancy," for a complete discussion on network design issues in a DLSw+ environment with Ethernet-attached end systems.

The DLSw+ Ethernet redundancy feature, introduced in Cisco IOS Release 12.0(5)T, provides redundancy and load balancing between multiple DLSw+ peers in an Ethernet environment. The feature also enables DLSw+ to support multiple DLSw+ routers on the same transparent bridged domain that can reach the same MAC address in a switched environment.

DLSw+ Ethernet Redundancy Feature Description

The DLSw+ Ethernet redundancy feature provides redundancy and load balancing between multiple DLSw+ peers in an Ethernet environment. It enables DLSw+ to support parallel paths between two points in an Ethernet environment, ensuring resiliency in the case of a router failure and providing load balancing for traffic load. This feature changes the way DLSW+ devices:

- Establish circuits in a transparent bridged domain
- Populate the SMAC field in explorer packets in a transparent bridged environment
- Support switch devices in an Ethernet environment

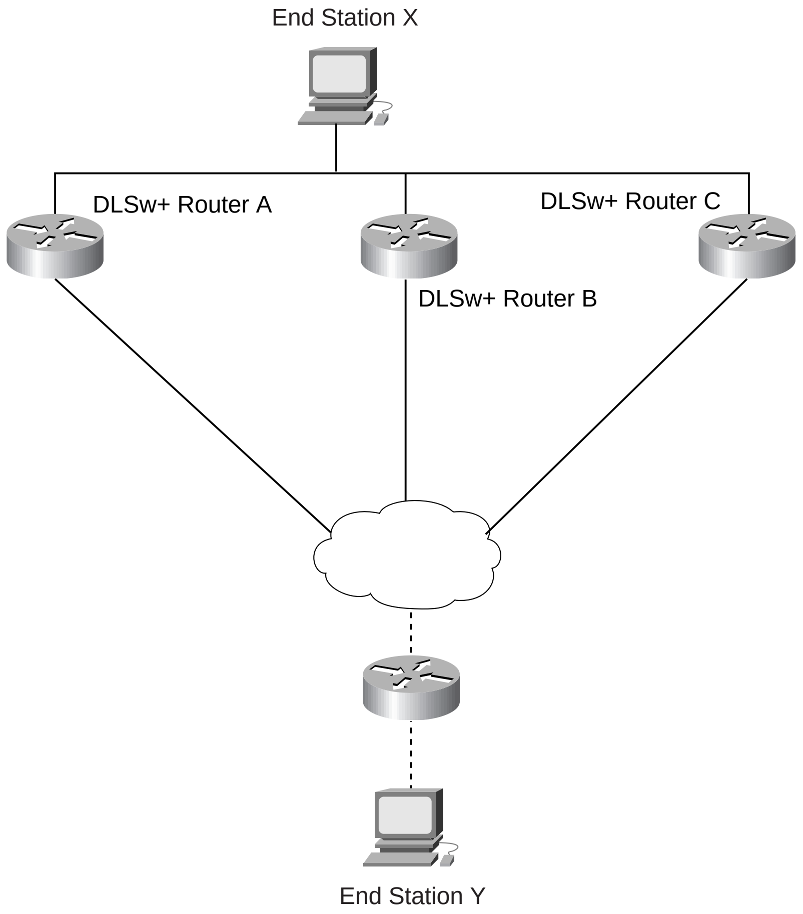
Establish Circuits in a Transparent Bridged Domain

The DLSw+ Ethernet redundancy feature alters the way DLSw+ routers establish circuits so that they recognize when they receive a frame from another router located on its transparent bridged domain. Before this feature, the devices connected to the same transparent bridged domain could not determine whether the frames they received were locally sourced, or whether they originated from another DLSw+ device within the same transparent bridged domain.

With the DLSw+ Ethernet redundancy feature enabled, the DLSw+ devices designate a master router in a transparent bridged domain. All devices on a transparent bridged domain advertise their presence to a multicast MAC address. One of the peers is elected as the master router. This master router maintains a database of all circuits being handled by the DLSw+ devices within its domain. Each device on the transparent bridged maintains an LLC2 session with the master router and asks the master router for permission before starting or accepting a new DLSw+ circuit. Because the master router keeps a database of the circuits being handled, it prevents duplicate circuits from being created for the same SNA session.

In Figure 12-1, DLSw+ Routers A, B, and C are on the same transparent bridged domain. Router B is configured to be the master router.

Figure 12-1 Sample Diagram of DLSw+ with Ethernet Redundancy



In Figure 12-1, the following sequence occurs:

- End Station X sends a SABME to Y (which is seen by all three routers) to begin a NetBIOS session to End Station Y. Assume Routers A, B and C already have reachability information for End Station Y in their remote reachability cache at the time End Station X sends a SABME.
- Router A and Router C indicate they want the circuit by sending IWANTIT frames to Master Router B. IWANTIT frames are sent only in response to frames that start a circuit (SABME, XID); they are not sent in response to an explorer.
- Master Router B can also take the circuit. It waits, however, a designated amount of time before deciding which router gets the circuit in order to receive all qualified recipients. Master Router B bases its decision on the load information that is included in the IWANTIT frame of each router. In this particular case, we will assume that Master Router B decides that Router A should have the circuit and sends Router A an UGOTIT primitive frame.

- Master Router B also denies permission to Router C by sending it a CIRCUIT_TAKEN (CKT_TKN) primitive frame. Master Router B then updates its own cache reachability tables, indicating that the circuit is taken.
- Router A sends a CANUREACH_cs to its remote peer to establish a circuit when it receives permission from the master router.

When the circuit disconnects, Router A notifies Master Router B by sending it a CIRCUIT_GONE (CKT_GONE) primitive. The master router then forwards the CKT_GONE primitive to the other devices on the LAN and removes the circuit from its CKT_TKN database. The only time a master router deletes a record is when it is notified by the device to which it granted the circuit or when it loses its LLC2 session with that peer, for example, if there is a device failure.

Populate the SMAC Field in Explorer Packets in a Transparent Bridged Environment

The DLSw+ Ethernet redundancy feature changes the way a DLSw+ router replaces the SMAC of an explorer packet. Normally, when DLSw+ devices receive a TEST frame, they update their local or remote cache with the SMAC based on whether the packet came from its local LAN or off a WAN. In transparent bridged domains, this situation can create unreliable reachability information. With the DLSw+ Ethernet redundancy feature enabled, the SMAC of an explorer packet sent on the LAN is replaced by the DLSw+ router's own MAC address. When another router on the transparent bridged domain receives the explorer, it recognizes that the SMAC belongs to a DLSw+ router on its own LAN. (The routers in a transparent bridged domain learn of each other's MAC address during the master election process.) Therefore, it does not update its local reachability cache, and it does not forward the explorer over any of its peer connections, thereby enabling more reliable local cache reachability information and decreasing the chance for looping explorers.

Referring to Figure 12-1 again, DLSw+ Routers A, B, and C are on the same transparent bridged domain. The following sequence occurs:

- End Station Y sends an explorer looking for X. The remote peer sends a CANUREACH_ex to A, B and C across the WAN.
- Router C populates its REMOTE reachability cache with Y.
- Router C internally tracks that it is SEARCHING for device X and that when it finds device X, it must inform device Y (from the peer path from which the original CANUREACH_ex came).
- Router C transmits an explorer on the LAN because it does not have any reachability information on X. While doing so, it substitutes Y with its own MAC address in the SMAC field of the TEST frame and then it sends the TEST frame on the LAN.
- Routers A and B see the explorer from the LAN, but because they recognize Router C's MAC address in the SMAC field, they do not update their local reachability cache and do not forward the explorer over any of their DLSw+ peer connections. (Routers A, B and C have learned of each others MAC address during the master election process.)
- End Station X recognizes its MAC address in the TEST poll frame and responds to the SMAC, which is Router C.
- Routers A and B recognize this MAC address and do not act on the frame. The frame reaches Router C, which recognizes the frame as a response to its test poll.
- Router C updates its LOCAL cache with X and remembers that End Station Y was the original device searching for X. It replaces the MAC address with the original SMAC before sending an ICANREACH_ex reply back to the peer that originally sent the CANUREACH_ex.
- Routers A and B similarly respond to the remote peer's CANUREACH_ex by sending a TEST frame and substituting Y with their own MAC addresses.

Configuring the Ethernet Redundancy Feature

To configure the Ethernet redundancy feature, perform the following steps:

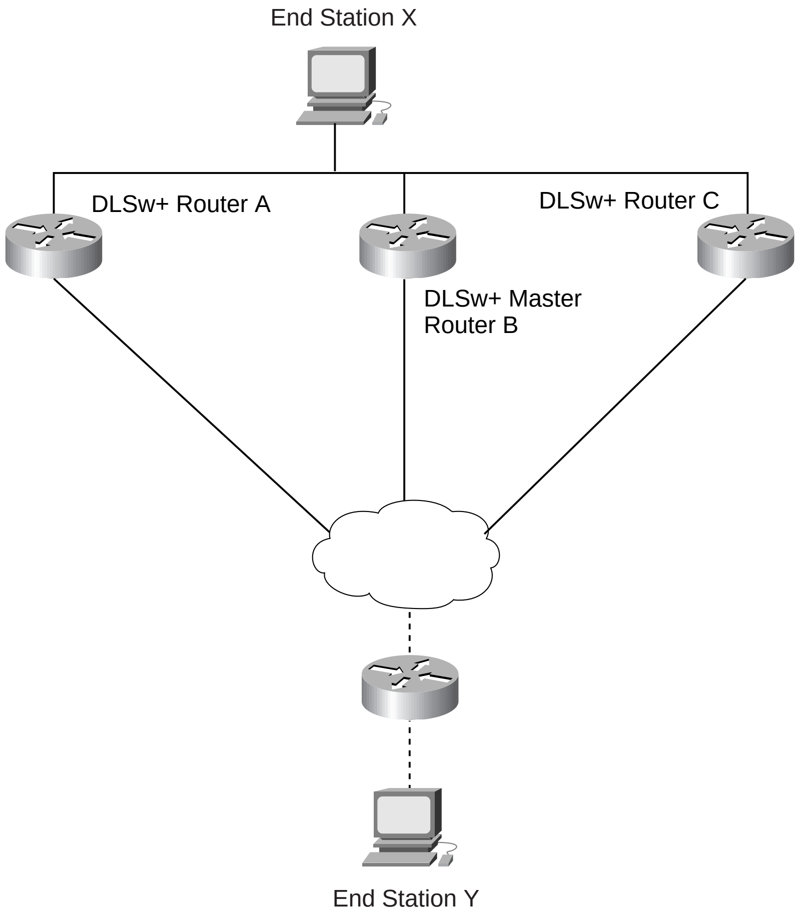
Step 1. Issue the **dlsw transparent redundancy-enable** interface command on the Ethernet interface of all the DLSw+ devices located on the same transparent bridged domain. One of the options with this command is *multicast-mac-address. Configure the multicast-mac-address* option with the same MAC address on all DLSw+ devices configured for Ethernet redundancy. All the DLSw+ devices on the transparent bridged domain advertise their presence to this MAC address. In Figure 12-2, DLSw+ Routers A, B, and C are on the same transparent bridged domain and they all advertise their presence to multicast MAC address 9999.9999.9999.

When the routers that are within a transparent bridged domain learn each other's MAC address, all routers in the transparent bridged domain compete and elect a master based on the router with the lowest MAC address (that is, if it is not already configured based on the **master-priority** value.) If two masters are configured with an equal master priority setting, the router with the lowest MAC address is elected. (See the *Bridging and IBM Networking Command Reference* for command details.)

Step 2. (Optional) Set the timeout value that the master router waits for all requests for a circuit before giving the permission for a router to take a circuit by issuing the **dlsw transparent timers** interface command. You can use the default values or you can create separate timeout values for NetBIOS and SNA sessions.

Figure 12-2 is a sample configuration of DLSw+ with the Ethernet redundancy feature. Routers A, B and C advertise their presence on the Ethernet via their Ethernet interfaces to the multicast MAC address 9999.9999.9999. Because Router B is the master, it keeps a database of all circuits handled within the domain and grants or denies permission for new circuit requests for Router A and Router C. There is no special configuration required for the end stations or for the remote peer. Only the DLSw+ devices on the LAN need the extra configuration. Master Router B waits 1.5 seconds after it receives the first IWANTIT primitive before assigning the new SNA circuit to one of the Ethernet redundancy peers because of the **dlsw transparent timers sna** *1500* command.

Figure 12-2 Sample Diagram of DLSw+ with Ethernet Redundancy



```
Router A
dlsw local-peer peer id 10.2.24.2
dlsw remote-peer 0 tcp 10.2.17.1
int e1
 ip address 150.150.2.1 255.255.255.0
 dlsw transparent redundancy-enable 9999.9999.9999

Router B
dlsw local-peer peer-id 10.2.24.3
dlsw remote-peer 0 tcp 10.2.17.1
int e1
 ip address 150.150.2.3 255.255.255.0
 dlsw transparent redundancy-enable 9999.9999.9999 master-priority 1
 dlsw transparent timers sna 1500

Router C
dlsw local-peer peer-id 10.2.24.4
dlsw remote-peer 0 tcp 10.2.17.1
int e1
 ip address 150.150.2.4 255.255.255.0
 dlsw transparent redundancy-enable 9999.9999.9999

Router D
dlsw local-peer peer-id 10.2.17.1 promiscuous
```

Verifying the Ethernet Redundancy Feature

There are several **show** commands that enable the user to verify the configuration for Ethernet redundancy. Verify that the master router is configured correctly by issuing the **show dlsw transparent neighbor** command on the appropriate routers. The following sample shows output from the **show dlsw transparent neighbor** command:

```
routerB#show dlsw transparent neighbor
Interface E0
0006.e278.6c0e  SELF                     Master
0009.fa50.0b1c  Rcvd Master-Accepted     VALID
```

The output shows that Router B is the master router whose MAC address is 0006.e278.6c0e. The other router, with a MAC address of 0009.fa50.0b1c, is a slave router on the common domain. The master router received a packet from the slave and notes the router is VALID

Verify that the cache of the routers is populating correctly by issuing the **show dlsw reachability** command on the Ethernet redundancy routers. The following sample shows output from the **show dlsw reachability** command:

```
DLSw Local MAC address reachability cache list
Mac Addr        status     Loc.    port                 rif
0004.f557.c156  FOUND      LOCAL   Ethernet1       --no rif--
DLSw Remote MAC address reachability cache list
Mac Addr        status     Loc.    peer
0004.f557.c164  FOUND      REMOTE  10.2.17.1(2065) max-lf(1500)
DLSw Local NetBIOS Name reachability cache list
NetBIOS Name    status     Loc.    port                 rif
stationx        FOUND      LOCAL   Ethernet1       --no rif--
DLSw Remote NetBIOS Name reachability cache list
NetBIOS Name    status     Loc.    peer
stationy        FOUND      REMOTE  10.2.17.1(2065) max-lf(17800)
```

The output shows that End Station X (0004.f557.c156) is reachable directly through interface Ethernet 1. Verify that the master router has the correct circuits in its cache by issuing the **show dlsw transparent cache** command. The following sample shows output from the **show dlsw transparent cache** command issued on Master Router B:

```
routerB#show dlsw transparent cache
Interface Ethernet0/1
 Circuit Cache
local addr(lsap)    remote addr(dsap)  state         Owner
0000.3028.92b6(08)  0007.0db1.238c(08) POSITIVE       SELF
0000.3028.92b6(08)  0008.dec3.609e(12) NEGATIVE       0009.fa50.0b1c
Total number of circuits in the Cache:2
```

The output shows that there are two circuits in the transparent bridging domain. The first circuit listed is "owned" by Router B, and all traffic over the circuit flows through Router B. The second circuit listed has been granted to the redundant peer with the MAC address 0009.fa50.0b1c, and all traffic over that circuit flows through the corresponding router. The "state" column indicates to which peer the circuit belongs.

Verify the number of circuits being handled by each of those peers by issuing the **show dlsw peer** command on the Ethernet redundancy routers. The following sample shows output from the **show dlsw peer** command:

```
router A#show dlsw peer
Peers:               state     pkts_rx   pkts_tx  type  drops ckts
TCP   uptime
 TCP 10.2.17.1       CONNECT       4936     61068  conf      0    0    0
16:17:13
Total number of connected peers:1
Total number of connections:    1

router B#show dlsw peer
Peers:               state     pkts_rx   pkts_tx  type  drops ckts
TCP   uptime
 TCP 10.2.17.1       CONNECT     748975  15817022  conf      0    1    0
16:15:47
Total number of connected peers:1
Total number of connections:    1

router C#show dlsw peer
Peers:               state     pkts_rx   pkts_tx  type  drops ckts
TCP   uptime
 TCP 10.2.17.1       CONNECT       7387    104617  conf      0    1    5
16:17:04
Total number of connected peers:1
Total number of connections:    1

router D#show dlsw peer
Peers:               state     pkts_rx   pkts_tx  type  drops ckts
TCP   uptime
 TCP 10.2.24.2       CONNECT      61068      4936  prom      0    0    0
16:17:14
 TCP 10.2.24.3       CONNECT   15817022    748975  prom      0    1    0
16:15:47
 TCP 10.2.24.4       CONNECT     104617      7387  prom      0    1    0
16:17:04
Total number of connected peers:3
Total number of connections:    3
```

The output shows that redundant peers A, B, and C all are connected to remote Router D. Currently there are two circuits, one between Router B and Router D, and the other between Router C and Router D.

Configuration Considerations with Ethernet Redundancy

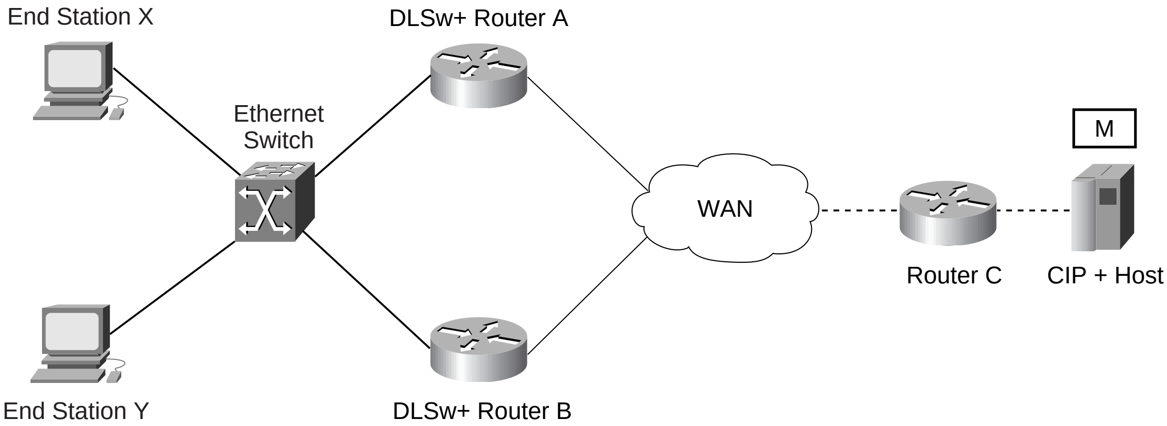
Because of issues with the propagation of UI frames, NetBIOS browsing is not supported in this release. Users must know the NetBIOS name of the server to which they wish to connect.

Do not configure both the global **dlsw bridge-group** command (in order to bridge to another LAN interface) and the interface **dlsw transparent-redundancy** command on the same device. You can, however, configure the interface **bridge-group** command on non-Ethernet redundancy interfaces. This type of configuration means that no bridging occurs between the two groups. Also, do not configure transparent bridging on an Ethernet interface that is configured for Ethernet redundancy. You can, however, configure two separate interfaces (on separate Ethernet segments) on the same router being able to reach the same MAC address with either the Ethernet redundancy feature or by using the **dlsw bridge-group** command (using separate bridge groups).

Support Switch Devices in an Ethernet Environment

Ethernet redundancy with switch devices requires further changes because of the way in which switches handle and direct traffic. Switches direct traffic by observing a frame's SMAC and by observing from which port the frame arrives. They forward all traffic to a particular address from a specific port rather than flooding all of its ports. In a normal Ethernet environment, this method is sufficient because there can only be one unique path to any MAC address. However, this method does not work in an environment where there are multiple DLSW+ routers on the same transparent bridged domain that can reach the same MAC address (see Figure 12-3).

Figure 12-3 DLSw+ with Ethernet Redundancy in a Switched Environment

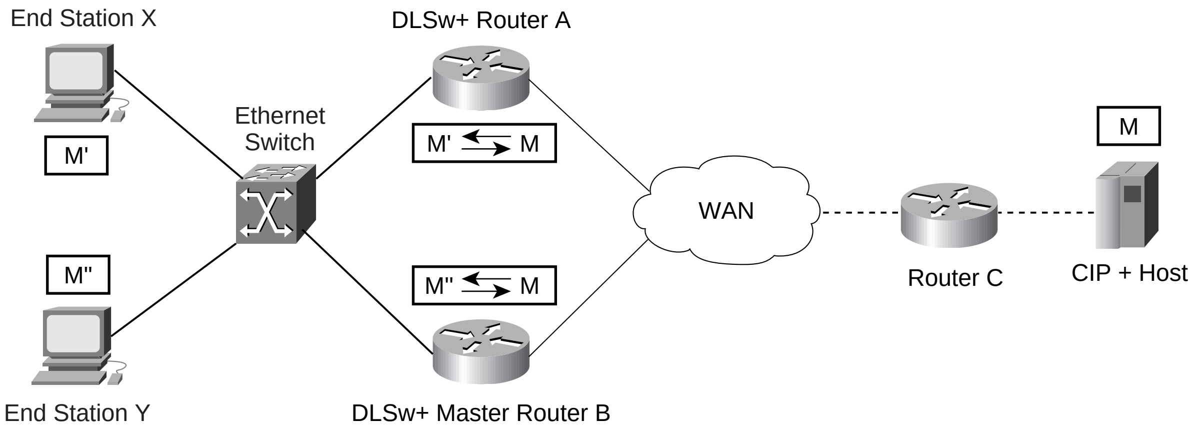


Because Routers A and B are hooked to different ports on the Ethernet switch, the switch sees traffic from one SMAC coming into multiple ports. The Ethernet switch thinks the MAC address of the Host appears on two different places on a LAN. This design breaks the Ethernet rule of having only one path to any MAC address. It gives the appearance of a bridging loop that the Spanning-Tree Protocol did not resolve. Because SNA is connection oriented, the session is eventually torn down.

The DLSw+ Ethernet redundancy feature solves this problem with MAC address mapping. MAC address mapping ensures that a particular SMAC is seen by the switch on only one port at a time. Furthermore, the routers monitor each other's MAC address mapping so that they adequately serve as each other's back up in the case of a router failure.

In Figure 12-4, Router A is configured to map MAC address M' (M prime) to MAC address M and Router B is configured to map MAC address M" (M double prime) to MAC address M. End Station X is configured to use M' as its SNA DMAC and End Station Y is configured to use M" as its SNA DMAC.

Figure 12-4   DLSw+ Network Showing MAC Address Mapping



The following sequence occurs:

- End Station X sends out a TEST poll searching for its DMAC M'.
- The switch floods the request to all its ports because it is a new circuit and the switch has not heard of M'.
- The switch notes the port through which End Station X can be reached.
- Router A sees the TEST poll and recognizes that it is mapping M' to MAC address M. It replies to the switch with a TEST final.
- The switch populates its cache with M' and, because it knows where End Station X is now, the frame is directed out a single port rather than flooded to all its ports.
- The end station sends an XID because it is ready to start an SNA session.
- The Ethernet switch directs the frame out the port to which Router A is attached because it has seen a packet with the SMAC of M'.
- Router A asks Router B permission to take the circuit since Router B is the master.
- Router A receives permission to take the circuit because it is doing MAC address mapping for M' to M. Router A sends a CANUREACH_cs to begin the process of creating a circuit.
- Router A does MAC address translation by replacing M' in the DMAC field with M, the actual MAC address of the mainframe resource. From this point forward, any frames directed from Router A toward the WAN are referred to as M and any frames being directed from Router A toward the LAN are referred to as M'. Some level of load balancing is achieved if half of the end stations are configured to use DMAC M' and the other half are configured to use M".

In the case of a router failure, the other router detects the failure and seamlessly takes over the failed router's mapping responsibilities. In Figure 12-4, if Router A fails, the switch thinks it can still reach MAC address M' out the port that is connected to failed Router A. Router B takes over the mapping responsibilities for Router A by sending a TEST frame with SMAC M' and a multicast DMAC to the switch. The switch notes the SMAC M' and assumes the resource moved and updates its CAM table appropriately. Now End Station X tries to reestablish its connection to the mainframe by sending out an XID poll to its configured DMAC M'. The switch knows to direct this frame out the port to which Router B is attached because of the TEST frame Router B sent earlier. Router B assumes the mapping responsibilities of Router A by mapping M' to M and continues its own mapping responsibilities of mapping M" to M.

When Router A recovers, Master Router B realizes that Router A should be mapping M' to M. Both routers cannot map M' to M simultaneously because the switch cannot handle multiple ports with reachability to the same MAC address. Master Router B, therefore, stops mapping M' to M and the existing sessions are taken down and recovered through Router A.

Configuring Ethernet Redundancy in a Switched Environment

To configure Ethernet redundancy in a switched environment, you need to:

Step 1. Configure the **dlsw transparent redundancy-enable** interface command on those DLSw+ devices located on the same transparent bridged domain. Configure the multicast MAC address to which all the devices will advertise their presence. It is at this point that you would also elect the master router.

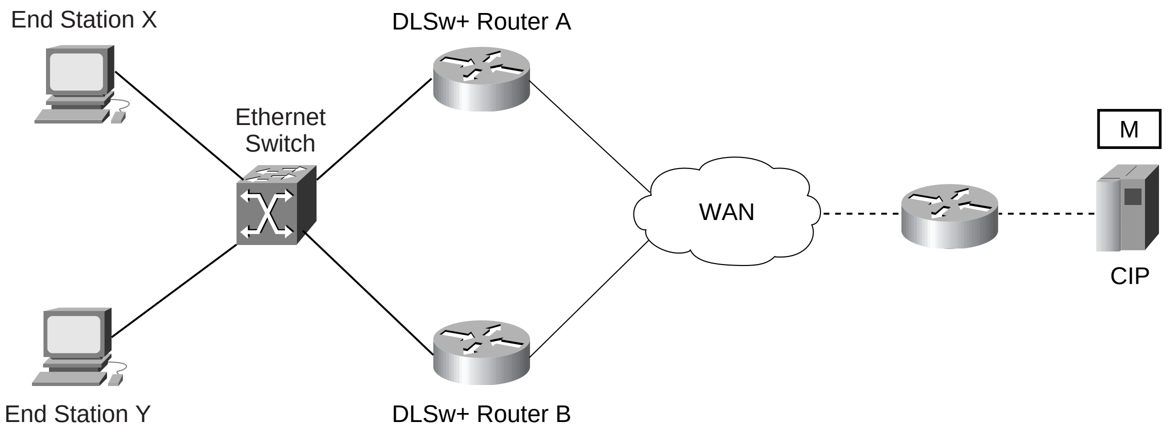
Step 2. (Optional) Configure the **dlsw transparent timers** interface command to set the amount of time that the master router waits for all requests for a circuit before giving permission to a router to accept a circuit. You can use the default values or you can create separate timeout values for NetBIOS and SNA sessions.

Step 3. Issue the **dlsw transparent switch-support** global configuration command on the routers connected to the Ethernet switch.

Step 4. Issue the **dlsw transparent map** interface command on the Ethernet interfaces that are connected to the Ethernet switch to enable MAC address mapping. The user creates the **local mac** address which maps to the actual MAC address that is specified in the **remote mac** option. It is recommended that you configure a backup to the DLSw+ device that will resume the mapping responsibilities if the DLSw+ device fails.

Figure 12-5 is a sample configuration of the DLSw+ Ethernet redundancy feature in a switched environment. The Ethernet switch sees the device with MAC address 4000.0010.0001 one port at a time because Router A and Router B have mapped different MAC addresses to it. This configuration is known as MAC address mapping. Router A is configured so that noncanonical MAC address 4000.0001.0000 (canonical MAC address 0200.0080.0000 as configured in the router) maps to the actual device with noncanonical MAC address 4000.0010.0001 (canonical MAC address 0200.0008.0080). Router B is configured so that noncanonical MAC address 4000.0201.0001 (canonical MAC address 0200.4080.0080) maps to the actual device with noncanonical MAC address 4000.0010.0001 (canonical MAC address 0200.0008.0080). Router A and B backup one another. Router A is configured as the master with a default priority of 100. Master Router B waits 1.5 seconds after it receives the first IWANTIT primitive before assigning the new SNA circuit to one of the Ethernet redundancy peers because of the **dlsw transparent timers sna 1500** command.

Figure 12-5 DLSw+ with Ethernet Redundancy in a Switched Environment



```
Router A
dlsw local peer peer-id 10.2.17.1
dlsw remote-peer 0 tcp 10.3.2.1
dlsw transparent switch-support
int e 0
 mac-address 4000.0000.0001
 ip address 150.150.2.1 255.255.255.0
 dlsw transparent redundancy-enable 9999.9999.9999 master-priority 1
 dlsw transparent map local-mac 0200.0080.0000 remote-mac 0200.0008.0080  neighbor 4000.0000.0011
 dlsw transparent timers sna 1500

Router B
dlsw local peer peer-id 10.2.17.2 promiscuous
dlsw transport switch-support
int e 1
 mac-address 4000.0000.0011
 ip address 150.150.3.1 255.255.255.0
 dlsw transparent redundancy-enable 9999.9999.9999
 dlsw transparent local-mac 0200.4080.0080 remote-mac 0200.0008.0080  neighbor 4000.0000.0001
```

Verifying the Ethernet Redundancy Feature in a Switched Environment

Use the **show dlsw transparent map** command (in addition to the other commands listed in the "Verifying the Ethernet Redundancy feature" section of this chapter) to verify the configuration in a switched environment.

1. Verify that the created MAC address to which all the Ethernet redundancy routers are mapped is configured correctly by issuing the **show dlsw transparent map** command on all the routers configured for Ethernet redundancy.The command should be issued on all the routers configured for the Ethernet redundancy feature to ensure that the local MAC addresses match. By viewing the output you can also verify that a router is configured to backup another router's MAC address mapping functions.

The output from Router A and Router B shows the created MAC addresses are `4000.0001.0000` and `4000.0201.0001`:

```
router A#show dlsw transparent map
Interface Vlan200
      LOCAL Mac          REMOTE MAC       BACKUP
      ---------          ----------       ------
    4000.0001.0000     4000.0010.0001   0200.0000.0088     STATIC
    4000.0201.0001     4000.0010.0001   0200.0000.0088     DYNAMIC(Passive)

router B#show dlsw transparent map
Interface Vlan200
      LOCAL Mac          REMOTE MAC        BACKUP
      ---------          ----------        ------
    4000.0201.0001    4000.0010.0001     0200.0000.0080     STATIC
    4000.0001.0000    4000.0010.0001     0200.0000.0080     DYNAMIC(Passive)
```

Configuration Considerations with Ethernet Redundancy in a Switched Environment

DLSw+ local switching is not supported between two Ethernet redundancy interfaces, or between an Ethernet redundancy interface and any other LAN-type media (Token Ring, ISL, LANE, or FDDI).

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 2500 Series routers 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 4500 or 4700 Series routers 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 4x00 Series routers 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 Releases 10.3 and 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 Releases 10.3 and 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 + (TCP queue size x buffer size)]
+
number of concurrent LLC2 connections x [LLC2 maximum window size x MTU]

The default TCP window size is 20 K and the default TCP queue size is 100. The buffer size is the size of the buffer that will fit the LAN interface 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 accounts for non-DLSw+ traffic.

APPENDIX B

DLSw+ Support Matrix

Tables B-1 to B-5 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.

TCP encapsulation works with all the types of media combinations listed in Table B-1. Direct Encapsulation does not work with any of the media combinations except over:

- Serial HDLC in Cisco IOS Release 10.3 and above
- Frame Relay (with and without local acknowledgement) in Cisco IOS Release 11.0 and above
- RSP support in Cisco IOS Release 11.1 and above

Table B-1 refers to the media in which the DLSw+ router is directly connected.

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

Media	FST (Requires Release 11.1 for RSP Support)
Serial HDLC	Release 10.3
Frame	Release 10.3(13), 11.0(9), and 11.1(4)
ATM	Release 11.1(5)
ATM/1490	Release 11.1(5)
Token Ring LANE	No
Ethernet LANE	No
FDDI	Release 10.3(12), 11.0(9), and 11.1(4)
Token Ring	Release 10.3(12), 11.0(9), and 11.1(4)
Ethernet	Release 10.3
SMDS	Release 11.0(12) and 11.1(7)[1]
X.25	No
PPP	Release 11.2(2)

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

Table B-2 Media Conversion Options

Media/PU Type	TCP/IP	TCP/IP with RIF Passthru	FST[1]	Direct / Passthru[1]	DLSw Lite (Direct/Lack)
Token Ring-to-Token Ring	Release 10.3[2]	Release 12.0	Release 10.3[2]	Release 10.3[2]	Release 11.0[2]
Ethernet-to-Ethernet	Release 10.3	No	Release 12.0	Release 12.0	Release 11.0
SDLC-to-SDLC: **PU 4/5-to-PU 2.x/1** **PU 2.1-to-PU 2.1**	Release 10.3	No	No	No	Release 11.0
Multidrop SDLC 2.1	Release 11.0	No	No	No	Release 11.0
Token Ring-to-Ethernet	Release 10.3	No	Release 12.0	Release 12.0	Release 11.0
Token Ring-to-SDLC PU 4/5-to-PU 4/5	Release 12.0	No	No	No	Release 12.0
Token Ring-to-SDLC **PU 4/5-to-PU 2.X** **PU 2.1-to-PU 2.1**	Release 10.3	No	No	No	Release 11.0
Token Ring-to-QLLC[3]	Release 11.0	No	No	No	Release 11.0
Ethernet -to-QLLC[3]	Release 11.0	No	No	No	Release 11.0
Ethernet-to-SDLC PU4/5-to-PU4/5	Release 12.0	No	No	No	Release 12.0
Ethernet-to-SDLC **PU 4/5-to-PU 2.X** **PU 2.1-to-PU 2.1**	Release 10.3	No	No	No	Release 11.0
SDLC-to-QLLC	Release 11.0	No	No	No	Release 11.0
SRB/FDDI-to-Token Ring/Ethernet/FDDI (Cisco 7x00 only)	Release 11.2	No	No	No	Release 11.2
TB FDDI	Release 11.1	No	No	No	Release 11.2
CIP/CSNA	Release 11.0	Release 12.0	Release 11.0	Release 11.0	Release 11.0

1. Requires Release 11.1 for RSP support
2. TCP Passthru peer type is required for PU 4-to-PU 4 connectivity if more than one data path is available
3. 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	TCP/IP with RIF Passthru	FST[1]	Direct/ Passthru[1]	DLSw Lite (Direct/Lack)
Dynamic Peers[2]	Release 11.1	No	No	No	No
SNA DDR	Release 11.1	No	No	No	No
DLSw Version 2 Support (RFC 2166)	Release 11.3	No	No	No	No
DLSw+ Group Cache	Release 11.3	No	Release 11.3	No	No
Peer Group Clusters	Release 12.0(3)T	No	Release 12.0(3)T	No	No
Border Peers	Release 10.3	No	Release 10.3	No	No
On-Demand Peers[3]	Release 10.3	No	Release 10.3	No	No
Backup Peers	Release 10.3 / 11.3 enhanced[4]	No	Release 10.3 / 11.3 enhanced[4]	Release 11.3	Release 11.3
Enhanced Load Balancing	Release 12.0(3)T	No	No	No	Release 12.0(3)T
Circuit History	Release 12.0(3)T	No	No	No	Release 12.0(3)T

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

Features	TCP/IP	TCP/IP with RIF Passthru	FST[1]	Direct/ Passthru[1]	DLSw Lite (Direct/Lack)
Ethernet Redundancy	Release 12.0(5)T	No	No	No	No
NetBIOS DDR	Release 11.3	No	No	No	No
UDP Unicast	Release 11.3	No	No	No	No
LNM, DSPU, or NSP over DLSw+ (both running in same router)	Release 11.1(5)	No	No	No	Release 11.1(5)
APPN over DLSw+ (both running in same router)	Release 11.2	No	No	No	Release 11.2
Maps Support (MIB)	Release 11.1(5)	No	Release 11.1(5)	No	No
SNA View PU Correlation	Release 11.1(5)	No	No	No	Release 11.1(5), name to MAC/SAP pair only
80D5 Encapsulation (global only)	Release 11.0	No	No	No	Release 11.0
Switching of IP/IPX	Release 11.3	No	No	No	Release 11.3
ToS/COS Mapping	Release 11.3	No	No	No	No
RSVP Reservations	Release 12.0(3)T	No	No	No	No
HPR Support	Release 11.3	Release 12.0	Release 11.3	Release 11.3	Release 11.3

1. Requires Release 11.1 for RSP support
2. Configured remote peers that are only connected when required
3. Remote peers that are dynamically learned via border peers and only connected when required
4. Allows TCP and FST peers to backup each other

Table B-4 Features Independent of Encapsulation Type

Features	Release Level Required
Fault Tolerance across Multiple Active Peers	Release 10.3
Cost	Release 10.3
Promiscuous and Passive Peers[1], [2]	Release 10.3
Load Balancing across Multiple RIFs per Interface	Release 11.3(5)

1. Peering to routers that are not preconfigured
2. Configured remote peers for which this local peer does not initiate a peer connection

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

Media Type	Token Ring	Ethernet	FDDI	CIP LANE	LANE	Token Ring LANE	ISL	Token Ring ISL	SDLC	QLLC
Token Ring	No	No	No	No	No	No	No	No	Release 11.1	Release 11.1
Ethernet	No	No	No	No	No	No	No	No	Release 11.1	Release 11.1
FDDI	No	No	No	No	No	No	No	No	Release 11.1	Release 11.1
CIP LAN	No	No	No	No	No	No	No	No	Release 11.1	Release 11.1
LANE	No	No	No	No	No	No	No	No	Release 11.2	Release 11.2

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

Media Type	Token Ring	Ethernet	FDDI	CIP LANE	LANE	Token Ring LANE	ISL	Token Ring ISL	SDLC	QLLC
Token Ring LANE	No	No	No	No	No	No	No	No	Release 11.3	Release 11.3
ISL	No	No	No	No	No	No	No	No	Release 11.2	Release 11.2
Token Ring ISL	No	No	No	No	No	No	No	No	Release 12.0	Release 12.0
SDLC	Release 11.1	Release 11.1	Release 11.1	Release 11.1	Release 11.2	Release 11.3	Release 11.2	Release 12.0	Release 11.1	Release 11.1
QLLC	Release 11.1	Release 11.1	Release 11.1	Release 11.1	Release 11.2	Release 11.3	Release 11.2	Release 12.0	Release 11.1	Release 11.1

Ethernet DLSw+ Redundancy

This appendix discusses network design issues in a DLSw+ environment with Ethernet-attached end systems. It first reviews SRB. Then it describes how DLSw+ provides redundancy and load balancing when end systems are connected over any media that supports SRB—that is, Token Ring, FDDI, Token Ring LANE, and Token Ring ISL. Finally, it explains the limitations in providing redundancy in Ethernet environments and recommends design techniques for addressing these limitations.

SRB Redundancy and Load Balancing

SRB allows multiple concurrently active paths in a bridged network. To prevent loops, SRB packets are sent along a route that is specified in the RIF in each packet. The RIF is placed in the packet by the source end system (hence the term source routing). The RIF lists each ring number and bridge along the path.[1]

The source end system discovers the route by use of an SRB explorer. When an LLC2 connection is first established (or for connectionless traffic, when the cache has expired), the source end system sends a TEST or XID frame in an SRB explorer.[2] Every source-route bridge that sees the frame copies the packet to each of its locally attached rings and concurrently updates the RIF to include its bridge number and the next ring number.

If a bridge notices that the next ring matches a ring already in the RIF, the frame is not copied to that ring. This prevents loops in the network. When the explorer reaches the destination, the destination flips a bit to indicate that the RIF should be read in the opposite direction, and it returns the frame to the source. If there are multiple paths through the network, the end system receives multiple responses. The source end system typically selects the first response back (presumed to be the best path at that moment) and uses its RIF for the connection or the cache. In SNA, each SNA PU has its own LLC2 connection, so each PU can potentially find a new path through an SRB network. This enables rudimentary load balancing of SNA traffic across a source-route bridged network.

SRB also allows duplicate, concurrently active MAC addresses in the network, a feature that has made SRB a key component in many SNA network designs. A common technique used in SNA network design is to give multiple FEPs or CIPs the same MAC address. Using this technique, if one of the channel gateways fails, sessions are dropped but are automatically reconnected over the alternate channel gateway. In addition, multiple gateways can be used concurrently to more effectively utilize resources and minimize the impact of any single

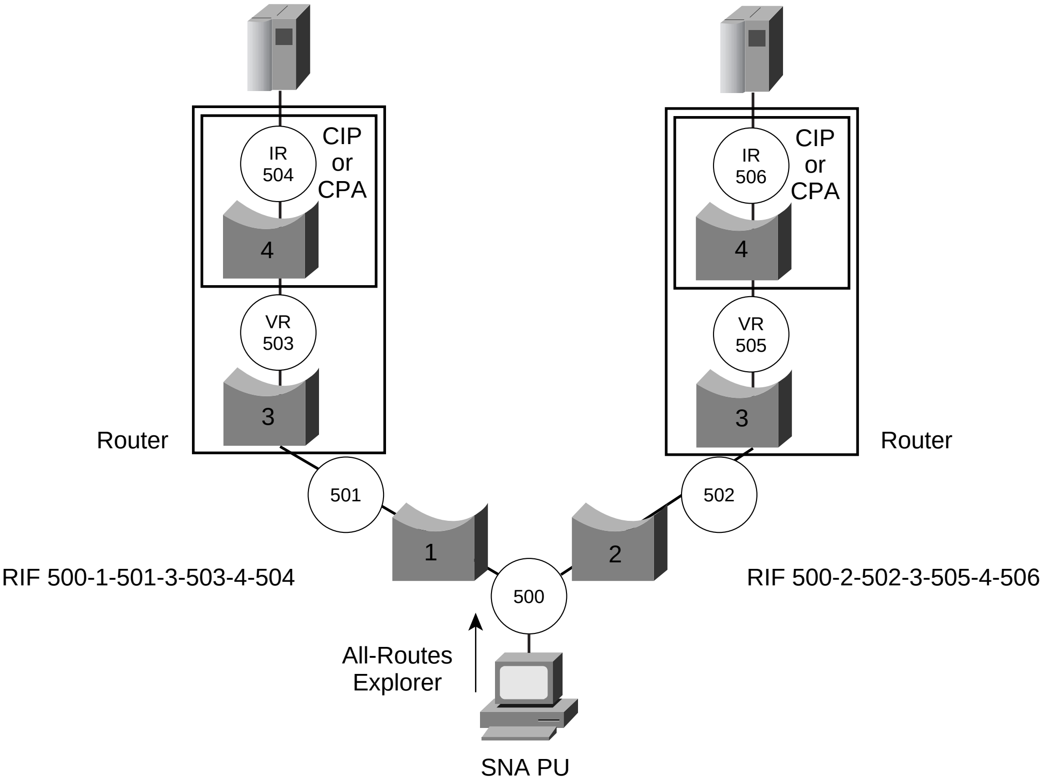
[1] Each Token Ring in an SRB network must be assigned a unique Token Ring number, and each bridge connecting the same pair of rings must have a unique bridge number.

[2] There are two types of explorers: all-routes explorers and spanning-tree explorers. Source-route bridges copy all-routes explorers onto every possible ring. Source-route bridges copy spanning-tree explorers to only one ring, relying on the SRB spanning-tree algorithm to determine which ring to use. For simplicity, this paper describes only all-routes explorers and describes the typical operation of SNA end systems.

failure. Load balancing in SRB environments is not very deterministic (basically, the first adapter to respond to a TEST frame is the adapter that is used for the duration of the PU connection), but it helps in periods of high traffic where one adapter may get congested. Assigning duplicate MAC addresses to both a FEP and a CIP provides a safe means to migrate from a FEP to a CIP, allowing one to back up the other automatically in the case of a failure.

Figure C-1 shows an example of an SRB network. In this figure, there are two CIPs with the same MAC address. When the SNA PU sends out a TEST frame in an SRB explorer, it gets two responses, each with a different RIF. It uses the first response it receives for the LLC2 connection.

Figure C-1 SRB-Enabled Load Balancing and Redundancy



DLSw+ Redundancy and Load Balancing

DLSw+ enables and improves upon the capabilities of SRB by using a more deterministic method for load balancing, or alternatively, provides customizable selection of the preferred path. DLSw+ also eliminates session outages in the event of certain failures or error conditions (using IP to reroute around link failures and TCP to retransmit dropped frames or frames in error). Finally, DLSw+ extends the load balancing capability to end systems that connect over serial media or Ethernet (with some limitations).

With DLSw+, remote branch routers can peer to multiple central site routers. If multiple central site routers can reach a given MAC address, there are two ways to control which central site router is used. These methods, or modes, are known as fault tolerant and load balance.

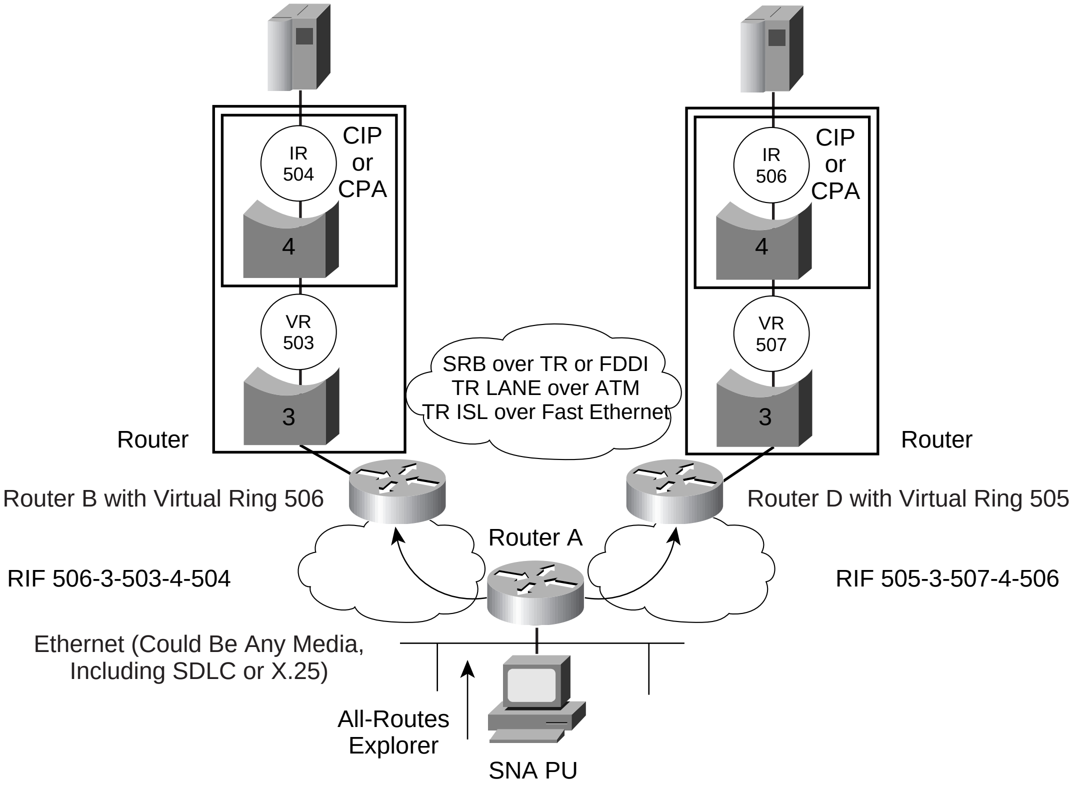
The default is fault tolerant. In this mode, remote routers peer to two or more central site routers. During circuit establishment, the remote router sends a CANUREACH frame to each of its central site peers. The first router to respond is used for the circuit. Alternatively, you can configure the DLSw+ cost parameter to control which router is the preferred peer and which ones are alternates. In either case, if the preferred peer is not available, an alternate one is automatically selected.

If **dlsw load-balance** is specified in the remote routers, the load balance mode is used. DLSw+ sets up each new circuit with a different DLSw+ peer, alternating through the list of capable peers in either a round-robin or enhanced load balancing fashion. (See the "Advanced Features" chapter for details.) Each SNA PU uses a unique circuit.

In addition to allowing remote routers to load balance across central site routers, DLSw+ also allows central site routers to load balance across alternate RIFs or Token Ring ports. Simply specify **dlsw load-balance** in the central site routers. Although round robin does not provide perfect load balancing, it is vastly better than traditional SRB.

DLSw+ allows branch end systems to attach to DLSw+ over Ethernet (or serial protocols) and still benefit from duplicate Token Ring adapters at a central site. Figure C-2 shows a network where the remote end system is attached to the DLSw+ router over Ethernet, but the upstream SNA device is attached over an SRB-capable medium. Note that the central site DLSw+ routers could attach upstream over either Token Ring, FDDI, Asynchronous Transfer Mode (ATM), or even Fast Ethernet. As long as SRB is supported on the medium (for example, by using Token Ring LANE on ATM or Token Ring ISL on Fast Ethernet), load balancing across duplicate RIFs or SRB ports is supported.

Figure C-2 Load Balancing Using DLSw+



Transparent Bridging Redundancy

SRB is not supported on Ethernet. Instead, transparent bridging is used. As the name implies, transparent bridges are not visible to the end systems. For that matter, they are not visible to each other. When a frame arrives at a transparent bridge, the bridge has no way to determine where the frame has been. Hence, the only way to prevent loops in this environment is to not have any loops. At any given point in time, there can be only one path between any two MAC addresses on an Ethernet LAN. In addition, duplicate, concurrently active MAC addresses are not supported. These two characteristics of transparent bridging prevent load balancing, but redundancy is still possible. The IEEE Spanning-Tree Protocol allows multiple paths between two points in a transparent bridge environment by ensuring that only one path is active at any given point in time.

A spanning tree is a subset of a transparently bridged network in which exactly one path exists between any pair of nodes. The Spanning-Tree Protocol defines a means for transparent bridges to communicate with each other and determine which ones will be forwarding and which ones will be listening. The forwarding bridges forward frames. The forwarding bridges and Ethernet LANs comprise the spanning tree. The listening bridges simply listen to determine if they need to take over for a forwarding bridge.

The DLSw standard does not support Ethernet IEEE spanning tree.

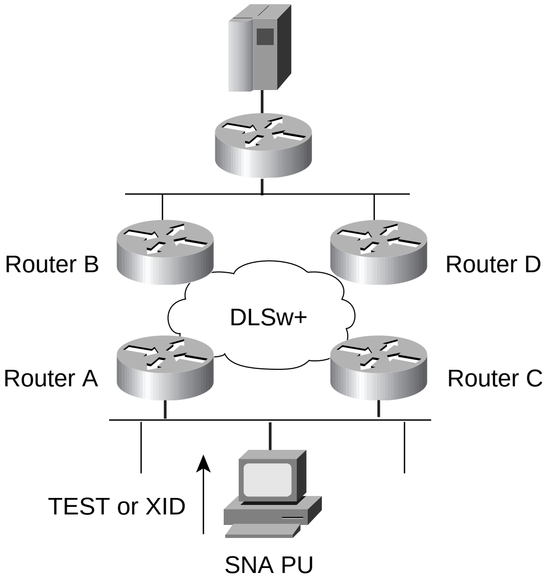
DLSw+ Redundancy in an Ethernet Environment

As long as the Ethernet end system that initiates a connection has only one way to get to a DLSw+ network, DLSw+ can provide load balancing and redundancy at the central site, as illustrated in Figure C-2. This configuration addresses the requirements of most environments, because most environments only have a single router at a branch. However, if an end system has multiple paths into a DLSw+ network, loops are possible.

Ethernet-to-Ethernet

Figure C-3 illustrates an invalid configuration with multiple active paths between two Ethernet LANs. In this example, we will assume Router A peers to Router B and Router C peers to Router D. Because all four DLSw+ routers are “forwarding” bridges (from the perspective of the Spanning-Tree Protocol), a packet destined for an unknown MAC address would loop endlessly (if these were standard DLSw+ routers without the plus features).

Figure C-3   Invalid DLSw+ Ethernet Configuration



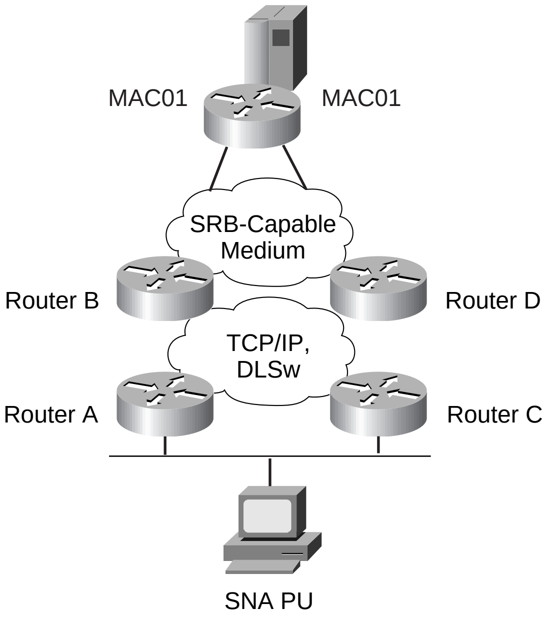
DLSw+ has a number of features designed to minimize unnecessary broadcasts. For example, in the situation described in Figure C-3, DLSw+ prevents endless looping with its explorer firewalling feature. If DLSw+ notes that it is already searching for a particular resource, it blocks subsequent explorers for that resource. Another DLSw+ feature is local resource learning. When DLSw+ determines that a resource is local, it blocks remote searches for that resource. This feature, which is usually quite useful, can create problems in an environment such as the one illustrated in Figure C-3. When the SNA PU issues a TEST frame, it is picked up by both Router A and Router C. For simplicity, we will ignore Router C for now. Router A forwards a CANUREACH frame to Router B. Router B drops the TEST frame on the Ethernet LAN where it is seen not only by the channel gateway, but also by Router D. Router D records the SNA PU as a local resource, even though it is in fact remote. When Router B gets a positive TEST response from the CIP, it forwards an ICANREACH frame to Router A. Router A drops a positive TEST response on the LAN. Router C sees that TEST response from the channel gateway (a CIP in this example) and inappropriately learns that the CIP is local (because the source address of the CIP is picked up in a TEST frame on the local LAN port). Likewise, if Router C gets a positive TEST response, it drops it on the LAN and Router A marks the CIP as local. As long as these entries are in the caches, neither DLSw+ peer forwards CANUREACH frames across the WAN and hence new circuits cannot be established.

Another problem with this configuration is that duplicate circuits can be created for the same SNA session. When the SNA PU sends a TEST frame, both DLSw+ routers see the frame and forward it in a CANUREACH frame to their respective peers. Both Router B and Router D copy the TEST frame onto the Ethernet LAN at the central site, specifying the source address of the SNA PU. The CIP sees two frames and responds to both of them. Both Router B and Router D see the first response and send an ICANREACH frame back to its respective peer and establish a circuit. (Both ignore the second response.) The next frame from the SNA PU is an XID. It is picked up by both Router A and Router C, sent to both Router B and Router D, and then forwarded to the CIP. The CIP, upon receiving two LLC2 packets with the same sequence number, terminates the LLC2 connection and, hence, the SNA connection is never established.

Using DLSw+ to Connect Ethernet LANs to an SRB-Capable Medium

If one end of the connection supports SRB, the situation is improved, but you can still have problems. In Figure C-4, assume that Router B peers to Router A and Router D peers to Router B. Both sets of DLSw+ peers see the traffic between the client workstation and the CIP. Neither pair of DLSw+ peers is aware of the other, and both pairs set up a circuit to transport the traffic between the client and the CIP. Duplicate packets arrive on the SNA session, resulting in the session being terminated. Also, if the virtual rings in Router B and Router D are not the same, packets dropped on the SRB-capable medium by Router B are picked up by Router D and forwarded back across the DLSw+ network causing a loop. With proper network design and some advanced DLSw+ features, however, you can avoid these problems.

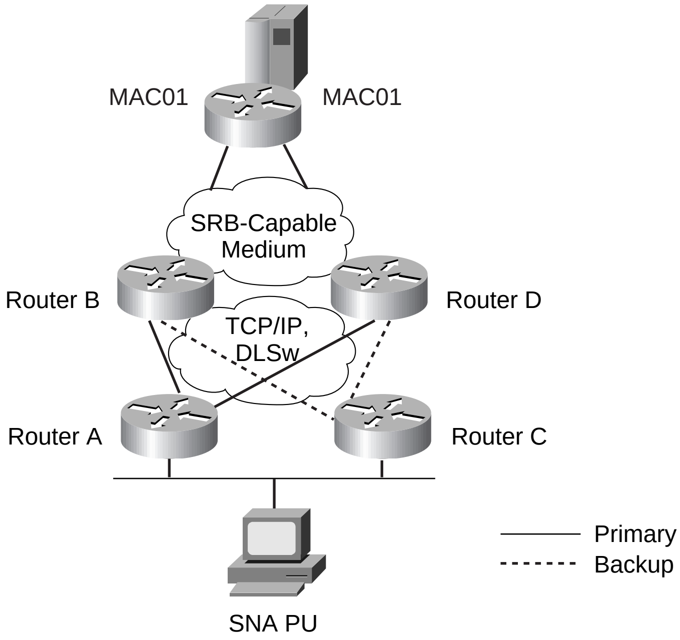
Figure C-4 Ethernet-to-SRB Configuration



To prevent these issues, you can use the network design shown in Figure C-5. Either TCP or FST[3] encapsulation can be used to transport the SNA traffic across an IP network. Router A and Router C are passive (meaning that they do not initialize their peer connections) and promiscuous (meaning that they establish a peer connection with a remote DLSw+ router that was not predefined). Neither router defines any remote peers. Router B and Router D define remote peers and initialize the DLSw+ peer connections. Both Router B and Router D are configured with the same virtual ring number to prevent loops. In both of the central site routers, Router A is configured as a peer, and Router C is configured as a backup peer to Router A, as shown in Figure C-5. The **linger** parameter on the backup peer statement is set to 0. With this configuration, as long as Router A is available, all DLSw+ circuits are between Router A and either Router B or Router D.

[3] FST encapsulation with media conversion requires Cisco IOS Release 11.3 or higher.

Figure C-5 Designing SR/TLB for Ethernet Redundancy



To load balance, specify **dlsw load-balance**. By configuring load balancing in the remote routers (Router A and Router C), they evenly distribute circuits across the central site routers, Router B and Router D. By configuring load balancing in Router B and Router D, they load balance across duplicate RIFs[4] or ports. This also allows you to evenly distribute traffic across the pair of CIPs or FEPs, which appear to SRB to simply be two RIFs to the same adapter or MAC address. By load balancing traffic across a pair of central site routers and a pair of CIPs, any single failure disrupts only half of the network. Hence, recovery is faster.

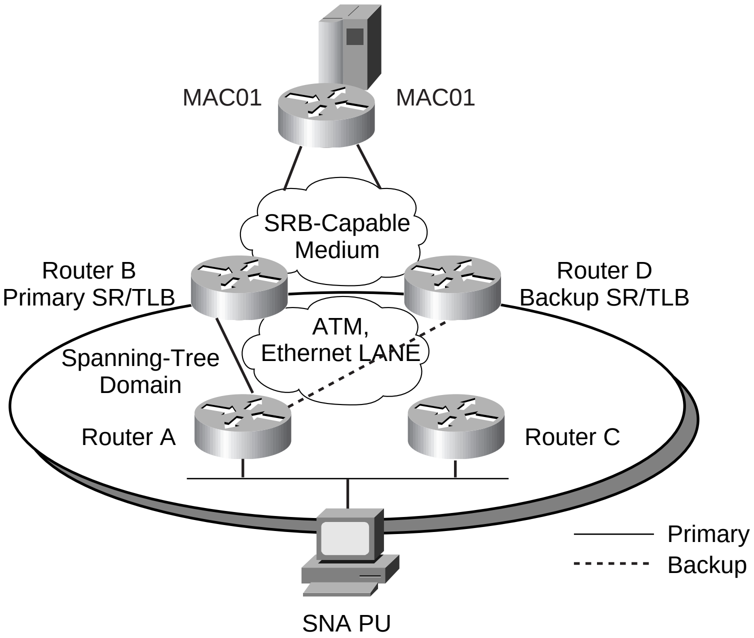
Router A is a single point of failure (that is, sessions are disrupted if you lose Router A). However, if Router A fails, Router C dynamically takes over. If Router A then becomes operational, it is imperative that Router C be disconnected immediately (hence, linger = 0) to prevent looping. Unfortunately, this means SNA sessions are disrupted again when the fallback occurs. All recovery is dynamic, requiring no intervention.

Using ST/TLB to Connect Ethernet LANs to an SRB-Capable Medium

An alternative means to connect Token Ring to Ethernet is SR/TLB. In general, bridging is never a better alternative than DLSw+ across a WAN, but in an ATM environment (for example, across a metropolitan area network), SR/TLB merits investigation. Some ATM environments might use Ethernet LANE to transfer SNA across the ATM backbone and use SR/TLB for media conversion, as illustrated in Figure C-6. This design assumes that both SR/TLB routers are at the central site, the channel-attached router has two CIPs, and each SR/TLB router is connected to both of the CIPs in the central site router.

[4] Load balancing across duplicate RIFs requires Cisco IOS Release 11.3(4.1) or higher

Figure C-6 Designing SR/TLB for Ethernet Redundancy



At the central site, only one translational bridge can be active at a time. Manual intervention is required to cause Router D to take over for Router B. At the remote site, only Router A or Router C is forwarding at any one time, relying on spanning tree to determine which is forwarding and which is listening. No manual intervention is required for Router C to take over for Router A.

SR/TLB allows duplicate concurrently active MACs. When a unicast frame is bridged from the transparent bridge domain to the source-route bridged domain, SR/TLB first checks to see if it has an entry in its RIF cache matching the DMAC address. If it does, SR/TLB uses that RIF. If it does not, SR/TLB sends a spanning tree explorer, which finds both of the CIPs. Each CIP responds to the spanning-tree explorer with a directed response. The SR/TLB router caches the first response it receives. All sessions use the RIF in that response and, hence, the same CIP.

If one of the CIPs fails, all sessions using that CIP are disrupted. The RIF cache times out, another explorer is sent out, and the alternative CIP is found. All subsequent sessions are established over that CIP.

Table C-1 compares the trade-offs from each of these solutions.

Table C-1 DLSw+ and SR/TLB Feature Comparison

Feature	DLSw+	SR/TLB
Dynamic Recovery from Failure of Central Site DLSw+ or SR/TLB Router	Yes	No, manual intervention required
Dynamic Recovery from Failure of a CIP/ Channel	Yes	Yes (if each SR/TLB router is connected to both CIPs as described previously)
Fallback	Dynamic and immediate but disruptive	Manual intervention required; disruptive, but can be done when convenient
Load Balancing across Central Site Routers	Yes	No
Load Balancing across Duplicate MACs	Yes	No

Table C-1 DLSw+ and SR/TLB Feature Comparison (Continued)

Feature	DLSw+	SR/TLB
Performance	Approximate; assumes TCP encapsulation; FST encapsulation is faster. Cisco 7500-RSP4: 3600 pps at 70% CPU Cisco 7200: 300–3000 pps at 70% CPU Cisco 7200: 150–1500 pps at 70% CPU Cisco 4700: 1400 pps at 70% CPU[1]	• Fast switched SR/TLB • Cisco 7500-RSP2: 14,000 pps (100% CPU utilization), assuming Token Ring to 10-Mbps Ethernet; LANE impact not factored in
Future	Redundancy to be supported in a future release	No enhancements planned

[1] 1400 pps, when sending data from one mainframe to another, can represent significant throughput. DLSw+ is not impacted by packet size, and mainframe-to-mainframe traffic can use packets up to the 1500-byte limit imposed by Ethernet. Hence, this could be 2.1 MB of traffic, depending on packet size.

Remote Ethernet Switches

If switches are installed at the remote site, there are additional design considerations. Figure C-7 shows an example of a network with remote Ethernet switches. The best design in this case is the design recommended in Figure C-5.

Figure C-7 Ethernet Switches at Remote Site

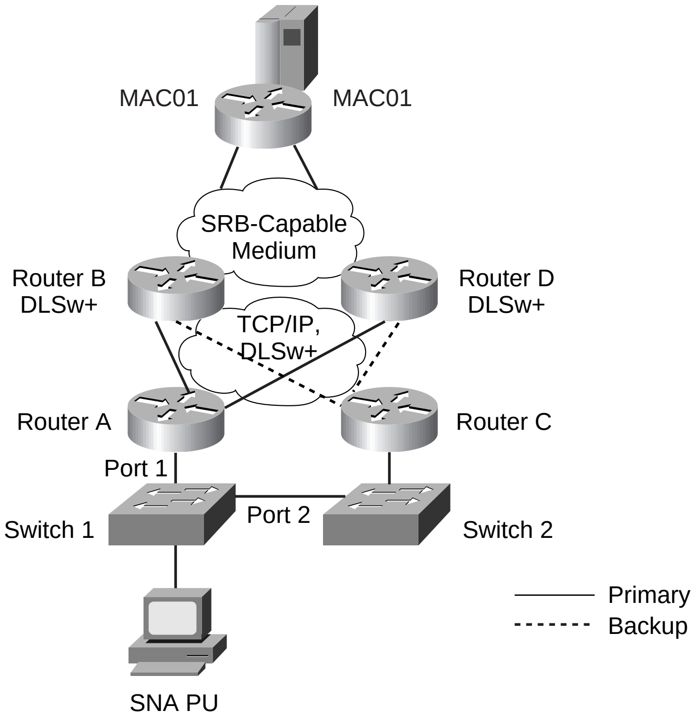


Figure C-7 shows the same configuration as Figure C-5 except that in this diagram, two Ethernet switches have been added. During normal operation, Router B and Router D each establish a peer connection with Router A. When the SNA PU sends the first TEST frame, Switch 1 sends it over both Port 1 and Port 2. Router C has no active peer connections and, hence, merely updates its local reachability table and discards the TEST frame. Router A forwards the TEST frame to Router B and Router D and establishes a circuit with one of them. As part of the circuit establishment, Router A sends a positive response to the SNA PU. When Switch 1 sees the response, it updates its forwarding table (also known as content addressable memory [CAM]) to indicate that the MAC address of the CIP or IBM 3745 (for simplicity, referred to as MAC01) is reachable through Port 1.

If Router A fails, Switch 1 clears its entry for MAC01. (This scenario assumes that the entire router fails.) The next time the SNA PU sends a TEST frame, it is again sent out both ports, but this time, Router C establishes the circuit (let's assume that it uses Router B) and sends the TEST response. Hence, Switch 1 updates its CAM showing that MAC01 is reachable via Port 2. Because Router A is the primary peer and Router C is only the backup peer, Router B continually tries to reestablish a peer connection with Router A. Because **linger** was not set to 0, if Router A succeeds, it immediately terminates its connection to Router C (and terminates any SNA sessions using that peer connection).

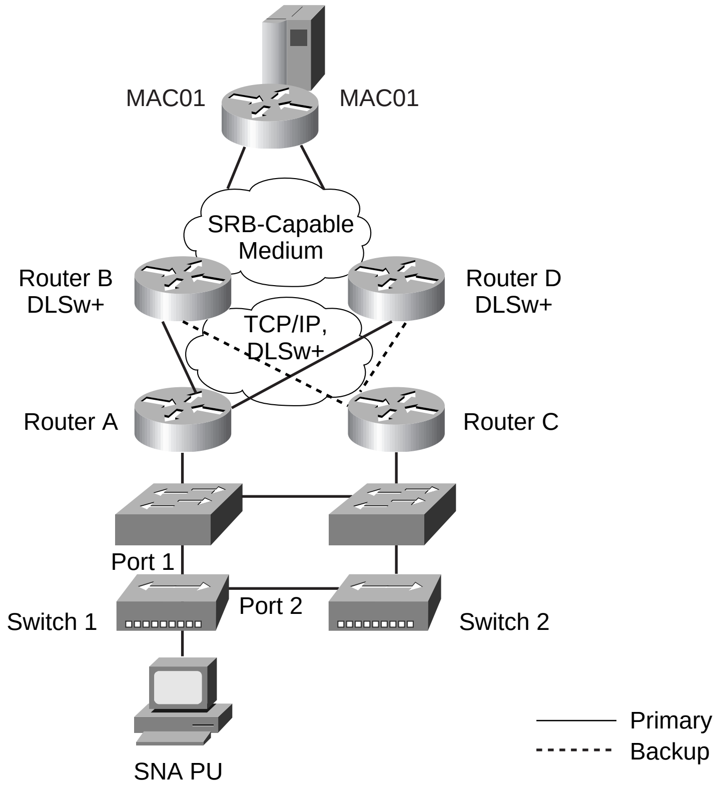
Because the SNA session is terminated, the SNA PU sends out another TEST frame. Unfortunately, it only goes to Router C because the CAM in Switch 1 still points to Port 2. Router C is active, but the DLSw+ peer connection is not active. Hence, until the CAM is cleared, Router A cannot see the TEST frame and the circuit cannot be established.

To minimize the impact of this problem, you can set the CAM aging timers lower. However, note that when the CAM timer for an entry expires, frames destined for that MAC address are flooded to all the ports on a switch, so there is a trade-off that must be considered. Another alternative is shown in the next example.

Remote Ethernet Switches with Hubs

Figure C-8 is almost the same diagram as Figure C-7, with the addition of two hubs. The hubs allow Switch 1 to reach both DLSw+ routers through either Port 1 or Port 2. Because the hubs introduce a loop in the network, the switches use the Ethernet Spanning-Tree Protocol to ensure that only one of these ports is active at a time. Regardless of which port is active, both Router A and Router C are reachable. Hence, the CAM problem described in the previous scenario is eliminated. As soon as Router A resumes its peer connection to Router B, Router A sees the TEST frames from the SNA PU and establishes the circuit.

Figure C-8 Remote Ethernet Switches Connected by One or More Hubs



Conclusion

Providing true router redundancy in an Ethernet environment is much more complex and difficult than in an equivalent Token Ring (or to be more precise, SRB) environment. This is because SRB allows multiple active paths and uses the RIF to prevent loops. Transparent bridging, used in Ethernet environments, relies on spanning tree to prevent loops and does not allow duplicate active paths. With careful design, redundancy is possible, although perhaps not optimal.

If you are running Cisco IOS Release 12.0(5)T and later, you can configure the DLSw+ Ethernet redundancy feature. This feature provides redundancy and load balancing when end systems are connected over Ethernet. See Chapter 12 "DLSw+ Ethernet Redundancy Feature" for more details.

DLSw Version 2 (RFC 2166) Questions and Answers

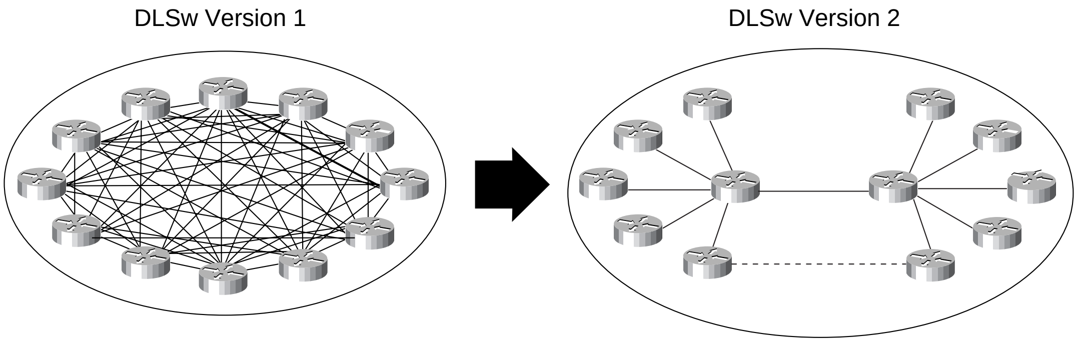
Q. What are the key functions in the DLSw Version 2 (RFC 2166) standard?

A. DLSw Version 2 added enhancements over the DLSw Version 1 (RFC 1795) standard to address scalability for fully meshed networks (networks requiring branch-to-branch connectivity). In the Version 1 standard, full mesh connectivity required all peers to have configured peer connections to every other peer. Every broadcast search to locate destination SNA or NetBIOS resources resulted in an explorer frame being transmitted to every other peer. In a fully meshed network of 50 sites, a single branch router required 49 peer connections. Hence, each explorer had to be replicated 49 times and sent over the same link 49 times!

Several enhancements in the Version 2 standard address this problem, as illustrated in Figure D-1. These enhancements are:

- A multicast technique to ensure that each broadcast results in only a single explorer over every link
- DLSw peer-on-demand, which allows explorers to be forwarded to routers without requiring remote peer connections to be configured in advance
- UDP unicast, which eliminates TCP retransmission of explorers and UI frames that may occur during periods of congestion, ensuring that steady-state data traffic has as much bandwidth as possible

Figure D-1 Comparison of DLSw Version 1 and Version 2



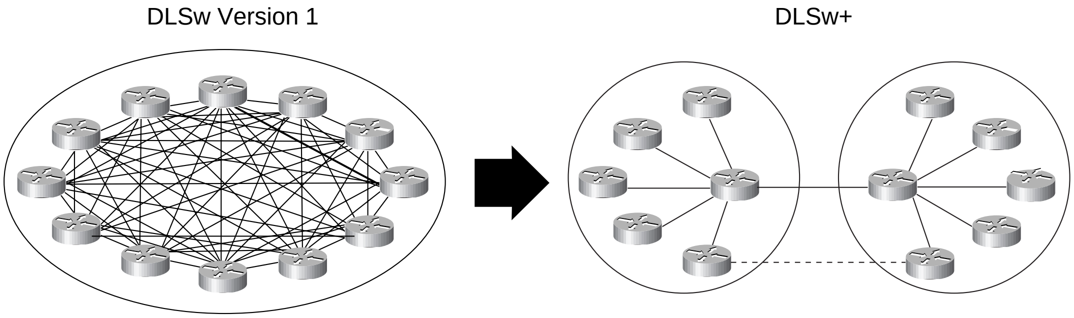
Q. The DLSw Version 2 enhancements sound a great deal like some of the plus features of Cisco DLSw+. What are the differences?

A. The Version 2 standard addresses the same problems addressed by some of the plus features (in fact, the AIW scalability work group that resulted in DLSw Version 2 began after Cisco presented the DLSw plus features to the AIW committee).

DLSw+ border peer functionality, shown in Figure D-2, addresses these problems in the following ways:

- DLSw+ uses peer groups and border peers to ensure that each broadcast results in only a single explorer over every link. The DLSw+ border peer function does not require running an IP multicast protocol as is the case with Version 2.
- DLSw+ allows explorers to be forwarded to routers without requiring a peer connection in advance. Upon finding a resource off a particular peer, DLSw+ supports the peer-on-demand feature, which allows two peers to dynamically establish peer connections without any prior configuration.

Figure D-2 Comparison of DLSw Version 1 and DLSw+



Q. In which Cisco IOS release did Cisco implement support for the Version 2 standard?

A. DLSw Version 2 is supported in Cisco IOS Release 11.3 and higher.

Q. Does Cisco DLSw+ support UDP unicast?

A. Cisco supports UDP unicast in DLSw+ and DLSw Version 2.

Q. Does Cisco DLSw+ support IP multicast?

A. The IP multicast function defined in the DLSw+ Version 2 standard (and supported by Cisco Version 2 support) provides functionality similar to the Cisco border peer and peer-on-demand functionality. However, multicast cannot be used in conjunction with DLSw+ border peer caching. The DLSw multicast feature should only be enabled in a Cisco router if Version 2 interoperability between Cisco and non-Cisco DLSw routers is required (and the non-Cisco routers support DLSw Version 2).

Q. Can a DLSw+ router interoperate with another router that supports the DLSw Version 2 standard?

A. Cisco support for the DLSw Version 2 standard is fully RFC 2166 compliant as specified by AIW. The only purpose for Cisco DLSw Version 2 support is to allow Cisco routers to interoperate with other non-Cisco routers that are fully compliant with the DLSw RFC standards only (1795 and 2166). Cisco DLSw+ should always be used when there are Cisco routers on both ends of a DLSw peer connection.

Q. What are the advantages of using DLSw+ now that an enhanced DLSw RFC-compliant standard is available that addresses scalability issues?
A. DLSw+ has offered a number of features that were additions to the base Version 1 RFC standard for a considerable period of time now. These features included not only border peers and peer-on-demand capabilities, but also load balancing, backup peers, dynamic peers, SNA DDR, and support for CiscoWorks Blue Maps and SNA View.

Border peer caching minimizes broadcasts by having border peers cache entries for NetBIOS servers and SNA resources. In this way, when a resource has been found on behalf of one branch router, all subsequent branch requests for that resource are forwarded directly to the correct router, greatly reducing broadcast traffic for frequently accessed resources. By contrast, the Version 2 standard does not include this feature and instead forwards every single explorer to all sites every time a different branch router needs to find a resource. The Version 2 standard uses a multicast technique to ensure that each search results in a single explorer request over *every* link. The DLSw+ enhancement takes advantage of the border peer cache to ensure that only the first search request results in an explorer over every link. Each subsequent search results in a single explorer request over a *single* link.

DLSw+ also supports UDP unicast, a feature adopted by Cisco DLSw+ from the Version 2 standard that allows non-data traffic to be transported in unicast UDP frames instead of TCP frames. During periods of congestion, explorer and UI frames would selectively be dropped rather than data frames (DLSw Version 1 retransmitted all frame types including explorer and UI frames, aggravating congestion situations). UDP unicast minimized the impact of explorers and UI frames on mission-critical data traffic.

SNA COS to IP ToS mapping with DLSw+ can prioritize (using various Cisco queuing algorithms such as custom and weighted fair queuing) between SNA and NetBIOS, or within SNA by LOCADDR or by MAC or SAP pair (known as SAP prioritization) when transporting SNA traffic over a DLSw+ backbone. In the case of routers running Cisco SNASw and DLSw+ combined, APPN COS transmission priority is mapped to IP precedence bits automatically without requiring any router configuration.

SDLC to LLC2 for PU 4/5 allows PU 4/5 devices to connect over DLSw+ even if one is connected over SDLC. This is key in SNA Network Interconnect (SNI) environments that are being migrated to a CIP at one end but still require a FEP at the other end for SNI (PU 4/5 to PU 5 connection).

DLSw+ backup peer support provides the capability for one DLSw+ peer to be configured as a backup for another peer. When the primary peer becomes unavailable, the backup automatically takes over (after three failed keepalive attempts). Fallback to the primary can also be enabled to occur automatically (configuration commands provide control over when fallback occurs).

MIB enhancements enable the plus features of DLSw+ to be managed using the CiscoWorks Blue products, Maps and SNA View. In addition, new traps included in DLSw+ alert network management stations of peer or circuit failures.

Q. Why is the multicast feature of the DLSw Version 2 standard not turned on by default in Cisco routers?
A. The Version 2 standard still has a scalability limit that Cisco addresses optimally with DLSw+ border peer caching. With the DLSw Version 2 multicast technique, if 200 branch routers need to access the same server or SNA device, there will be 200 multicasts (IP multicast explorers). DLSw+ border peers support caching, so that only the first request from the first branch router is broadcast. Subsequent searches from different routers are unicast to the correct router. Furthermore, most SNA customers do not implement IP multicast, because the current scalability technique (DLSw+ border peers) does not require multicast to eliminate unnecessary broadcast duplication.

DLSw+ TCP Performance

Overview

Cisco Systems performed a series of tests to determine the percentage of the CPU utilized on various Cisco router platforms as a function of data frames transported between two Data-Link Switching Plus (DLSw+) TCP peers. This data can help customers make a ballpark comparison between the processing capabilities of different router model types and a determination of how many data frames per second (fps) a particular router platform can support.

The routers tested were configured for very little other than DLSw+ (for instance, loopback interfaces were not used because that would have required the use of a routing protocol). Obviously in any real network environment there will be additional overhead for other configured features and protocols that must be factored in.

In addition, the results represent the load on the router in a very static controlled lab environment. Different frame sizes can be packaged by TCP differently, and other factors such as explorer load can also affect router CPU utilization (even for the same frame rate, router performance could be better or worse).

Sizing for determining the appropriate Cisco router platform is almost completely dependent on the amount of traffic and packet rate. From a historical perspective with Cisco DLSw+, this is predicated on how many packets per second of Logical Link Control, type 2 (LLC2) traffic a router has to forward. Increasing the number of SNA physical units (PUs) or DLSw+ peers and having no LLC2 traffic has very little impact on router CPU utilization, However, when sending frames across DLSw+ peers, router CPU utilization rises proportionally to the number of frames per second being sent inbound and outbound.

The most important information required to make an accurate router sizing determination is the peak rate of transactions per second (tps) and the transaction profile (number of bytes inbound and outbound). If the transaction rate is combined with the SNA response unit (RU) size information, then the rate of frames per second and the number of frames per transaction can be calculated. For example, if through analysis of the transaction profile it is determined that the transaction size is 2000 bytes outbound and 1000 bytes inbound with an RU size of 1024 bytes, then basically you have two frames out and one frame in for a total of three frames.

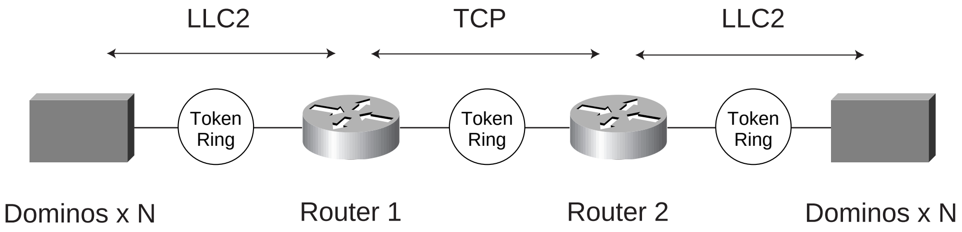
Lastly, although CPU utilization on most of the router platforms tested was driven to 70 percent router CPU utilization (and higher), note that it is recommended to design DLSw+ routers to utilize no more 50 percent of the CPU. This is especially true if there are multiple active peers configured on the DLSw+ routers for availability and redundancy. If redundancy is important, determine an acceptable fault-domain, double it, and peer that number of sites to a pair of redundant peers (half on each) that back each other up. Use peer load balancing between the redundant peers if load balancing between DLSw+ peers is also desired. For example, if 250 is determined to be the acceptable fault-domain, then put 250 peers on each redundant Cisco DLSw+ router so that effectively they back each other up. If one fails, you then have 500 sites on the remaining DLSw+ peer router.

For more information about designing DLSw+ networks, consult the *DLSw+ Design and Implementation Guide* at www.cisco.com/warp/public/cc/pd/ibsw/ibdlsw/prodlit/toc_rg.htm.

The Test Environment

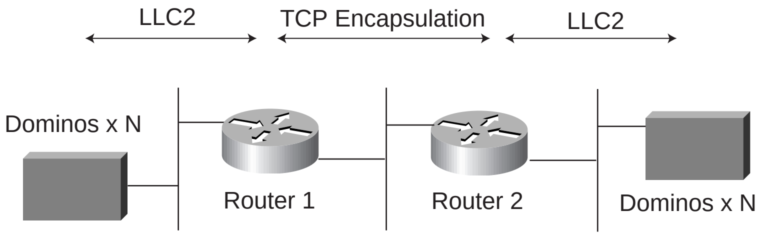
Figure 1 illustrates the basic Token Ring test environment that was used. Router 1 was the unit under test and Router 2 was either a Route Switch Processor 4 (RSP4) in the case of the Network Processing Engine (NPE)-150 and all smaller routers, or an RSP8 with the remaining routers. Wavetek Wandel Goltermann Domino was the test tool used to generate SNA traffic. We discovered during our testing that SNA sessions would not remain stable on the Domino when the data rates exceeded 700 fps on a Token Ring LAN. As a workaround to this problem, one Domino was attached to two Token Ring interfaces (400 fps per each interface) when it was required to drive data rates in excess of 700 fps.

Figure E-1 Token Ring: Branch Routers



The Ethernet topology (shown in Figure 2) was the same except both the LAN and WAN were replaced with Ethernet. There was no problem with running all the Ethernet traffic over one interface; however, at high data rates, we discovered that sessions could not be sustained using 10 MB Ethernet. For the NPE-300 and RSP8, 10 MB Ethernet was replaced with 100 MB Ethernet and these problems were eliminated. The CPU utilization was measured on both 10 MB and 100 MB Ethernet for frame rates up to 2000 fps to ensure that results were consistent between them.

Figure E-2 Ethernet: Branch Routers



Because the Cisco 1600 and 1700 Series routers do not have multiple LAN interfaces, a V.35 serial cable was used for the WAN connection. Data was also taken with a serial WAN on the Cisco 2600 and 3600 Series routers to tie the Cisco 1600 and 1700 data to the rest. For the small routers, Ethernet hubs and Token Ring media access units (MAUs) were used on the LAN segments. However, because of the problem with Token Ring and to simplify the physical setup of the test bed, the MAUs were replaced with Catalyst® switches. A Catalyst 5500 was used for the Token Ring LAN network and a Catalyst 3900 was used for the Token Ring WAN network. A Catalyst 2900XL was used for the Ethernet WAN, and two Catalyst 2900XLs trunked together were used for the LAN interface. This was only to simplify the physical setup. All Dominos were attached to one Catalyst 2900 in one lab, and all routers were attached the second Catalyst 2900 in the other lab. Then only one wire needed to be run between the two labs.

For the Route Switch Module (RSM), Multilayer Switch Feature Card (MSFC), and Route Switch Feature Card (RSFC), the Catalyst 2900 was trunked not to another Catalyst 2900, but to its respective switch. Because of the problem with Token Ring, separate VLANs were used with each Domino on the Catalyst 3900. However, in the Ethernet environment one VLAN was used for all the Dominos on the host side, and a separate VLAN was used for all the Dominos on the terminal side.

Nothing was configured on the routers except what was essential to take measurements. Therefore, the configurations for all the routers were the same. Each router was configured with local and remote peer statements. For Ethernet, a bridge group was configured on the LAN interface. For Token Ring, a separate ring group was configured for each LAN interface. Additionally, the load interval on all interfaces was set to 30 seconds, and a clock rate of 800,000 bits per second (bps) was used on the serial interfaces. No tuning was done on the routers; all parameters (LLC2, TCP, DLSw+, and so on) were set to the default.

Version 2.4 of the Domino Analyzer program and Version 1.2 of the SNAGEN application, which runs under the analyzer, were used.

The following LLC2 parameters were used on the Domino:

- T1 timer: 2s
- Ti timer: 30s
- Retries: 8 times
- Test frame: 10s
- Activation delta: 50ms

The host side Domino sent 128-byte data frames, and a 128-byte definite response frame was sent by the terminal side Domino. The host side Domino was always on the RSP4 and RSP8 side, and the terminal side Domino was connected to the unit undergoing testing. A different host address was assigned to each host Domino. The Dominos exchanged fixed (PU 2.0) exchange identification (XID) packets. PUs and logical units (LUs) were combined to generate data in the following way. For all data rates less than 800 fps, one LU per PU was used. For data rates that were multiples of 800 fps, four LUs per PU were used. For intermediate data rates, four LUs per PU were used up to the multiple of 800, and one LU per PU was used for the remaining. For example, the following frame rates consist of the following PUs per LUs. Remember that for each LU there is one command frame and one response frame sent per second.

- 24 fps: 12 PUs, one LU per PU
- 50 fps: 25 PUs, one LU per PU
- 100 fps: 50 PUs, one LU per PU
- 800 fps: 100 PUs, four LUs per PU
- 900 fps: 100 PUs, four LUs per PU; 50 PUs, one LU per PU
- 1600 fps: 200 PUs, four LUs per PU
- 1700 fps: 300 PUs, four LUs per PU; 50 PUs, one LU per PU

Initially the CPU utilization data was taken from the router manually by executing the **sh proc cpu** command. For the branch routers the CPU load stabilized quickly. Checks were done to ensure that the five-minute average converged to the one-minute average. After doing this we used the one-minute average after seven minutes, unless there was a large discrepancy between the five-second and one-minute averages. In those cases we waited for the one-minute and five-minute averages to converge.

For the larger routers, however, there was more variation in the CPU load. For these, a script recorded all the statistics, along with the interface and TCP statistics approximately every 20 seconds (we then performed manual averaging of the data). The interface statistics for the WAN and LAN were used to confirm that DLSw+ router was processing the correct amount of data.

Results

Figure 3 compares the number of Token Ring data frames per second processed with the corresponding router CPU utilization for the Cisco 3640, 4700, Catalyst 5000 RSM, RSP2, and NPE-150 hardware platforms. Figure 4 compares the number of Token Ring data frames per second processed with the corresponding router CPU utilization for the Cisco 4700, NPE-150, NPE-200, NPE-300, RSP4, and RSP8 hardware platforms.

Figure E-3 DLSw+ TCP Performance (Token Ring): Platforms Set A

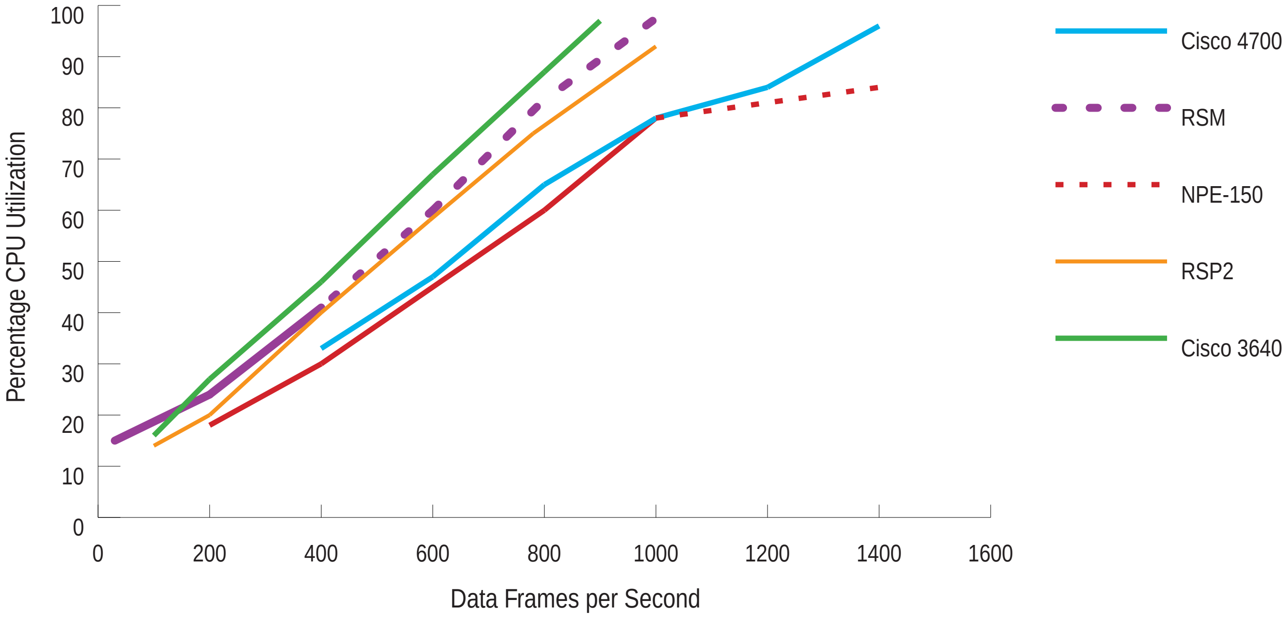
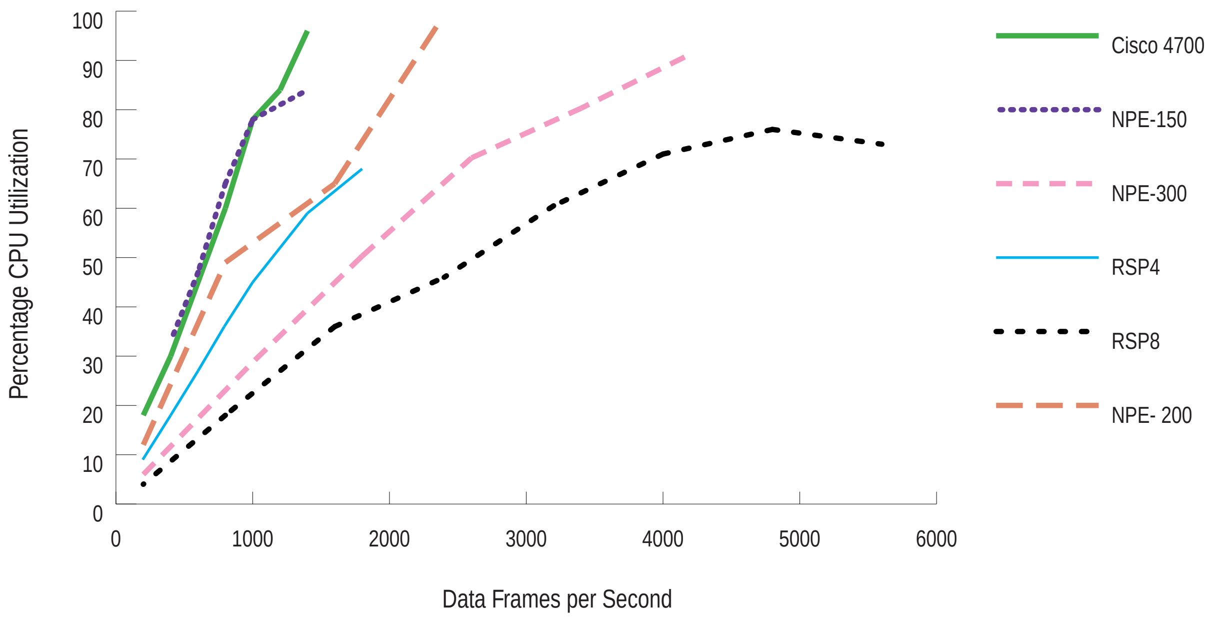


Figure E-4 DLSw+ TCP Performance (Token Ring): Platforms Set B



Note: When the RSP4 was tested, there were not enough Dominos available to fully utilize the router. Therefore, a CPU utilization result of less than 100 percent does not indicate that the router is incapable of handling more traffic.

There was a decrease in the CPU load when using switches instead of hubs, because frames arrived faster via the switch and TCP was able to encapsulate more data frames in one TCP packet. However, this phenomenon was not observed for data rates less than 1000 fps, so there was no inconsistency between the data taken over hubs and that taken over switches.

Figure 5 compares the number of Ethernet data frames per second processed with the corresponding router CPU utilization for various low-end Cisco hardware platforms (Cisco 1600 with serial WAN, 2600, 2600 with serial WAN, 3640, 3640 with serial WAN, RSP2, RSM, and Cisco 1720 with serial WAN). Figure 6 compares the number of Ethernet data frames per second processed with the corresponding router CPU utilization for a second set of low-end Cisco hardware platforms (RSM, RSP2, Cisco 4700, RSFC, MSFC, NPE-200, RSP4, NSFC2, NPE-300, and RSP8).

Figure E-5 DLSw+ TCP Performance (Ethernet): Platforms Set A

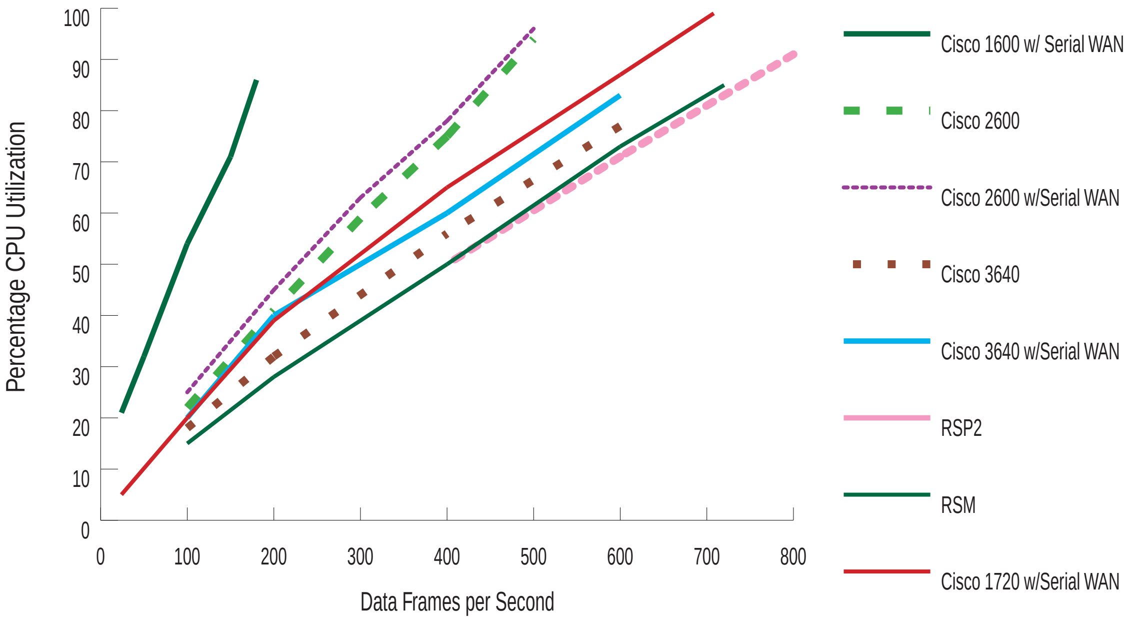
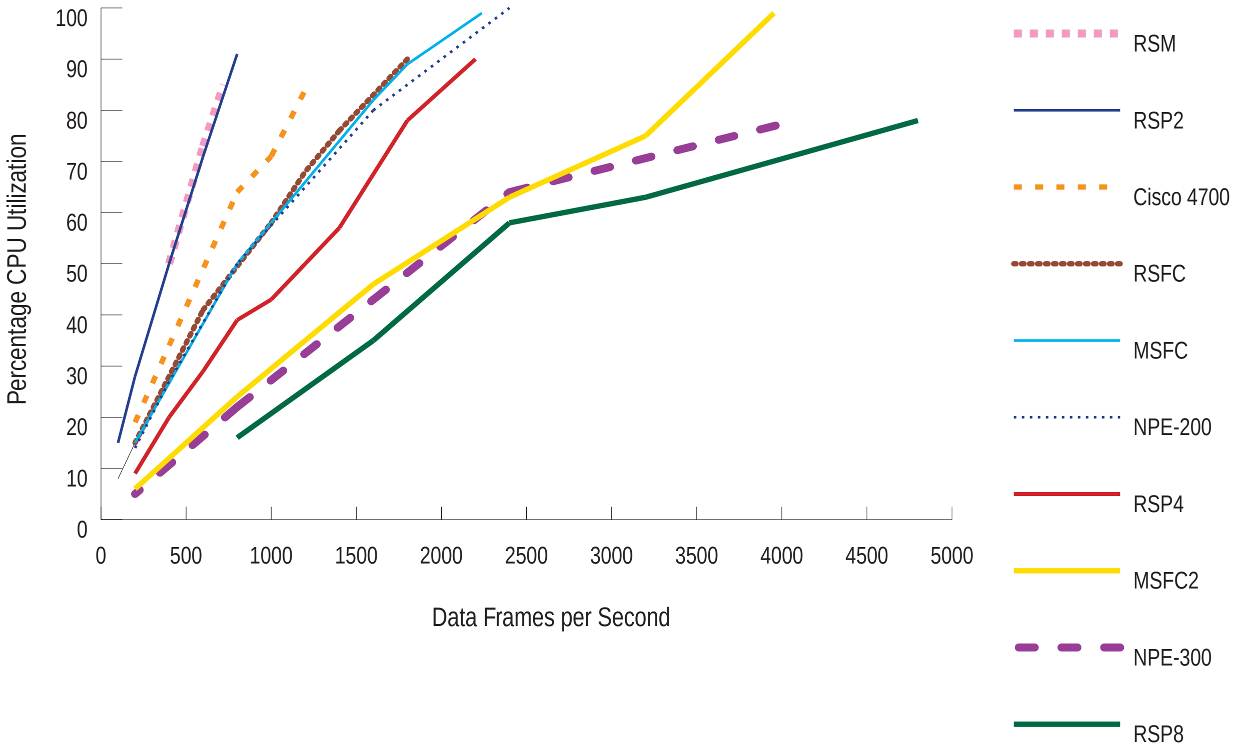


Figure E-6 DLSw+ TCP Performance (Ethernet): Platforms Set B



Version Information

For the Cisco 1720 modular access router, Cisco IOS Release 12.1(3.3) was used; for all other routers Release 12.1(3.1) was used. Because there was no official image for the MSFC, a special image built off the Release 12.1(3.1) code was used.