

Configuring Data-Link Switching Plus

This chapter describes how to configure data-link switching plus (DLSw+), Cisco's implementation of the DLSw standard for Systems Network Architecture (SNA) and NetBIOS devices. For a complete description of the DLSw+ commands mentioned in this chapter, refer to the "DLSw+ Configuration Commands" chapter of the *Bridging and IBM Networking Command Reference*. To locate documentation of other commands that appear in this chapter, use the command reference master index or search online.

DLSw+ Configuration Task List

DLSw+ supports local or remote media conversion between LANs and Synchronous Data Link Control (SDLC) or Qualified Logical Link Control (QLLC). For clarity, the configuration task list below describes configuration in a Token Ring environment. The only differences for SDLC and Ethernet are the specific commands needed to configure those media, plus a media-specific command to associate the interface with DLSw+.

To configure DLSw+, complete the tasks in the following sections:

- Define a Source-Bridge Ring Group for DLSw+
- Define a DLSw+ Local Peer for the Router
- Define a DLSw+ Ring List or Port List
- Define a DLSw+ Bridge Group List
- Define DLSw+ Remote Peers
- Enable DLSw+ over Frame Relay
- Enable DLSw+ on a Token Ring or FDDI Interface
- Enable DLSw+ on an Ethernet Interface
- Enable DLSw+ on an SDLC Interface
- Enable DLSw+ over QLLC
- Enable NetBIOS DDR

- Enable NetBIOS DDR
- Tune the DLSw+ Configuration
- Monitor and Maintain the DLSw+ Network

See the end of this chapter for “DLSw+ Configuration Examples.” Media-specific configuration examples for Ethernet and SDLC are also provided. For details of SDLC commands in the sample SDLC configuration, refer to the “LLC2 and SDLC Commands” chapter of the *Bridging and IBM Networking Command Reference*.

Define a Source-Bridge Ring Group for DLSw+

The source-bridge ring can be shared between DLSw+ and Source Route Bridging /Remote Source Route Bridging (SRB/RSRB). In DLSw+, the source-bridge ring group specifies the virtual ring that will appear to be the last ring in the RIF. Because RIFs are terminated at the router, there is no correlation between the ring-group number specified in DLSw+ peers. The numbers can be the same for management simplicity, but they do not have to be. To define a source-bridge ring group for DLSw+, use the following command in global configuration mode:

Command	Purpose
source-bridge ring-group <i>ring-group</i> [<i>virtual-mac-address</i>]	Define a ring group.

Refer to “DLSw+ Using TCP Encapsulation and LLC2 Local Acknowledgment—Basic Configuration Example” for a sample configuration file.

Define a DLSw+ Local Peer for the Router

Defining a DLSw+ local peer for a router enables DLSw+. You specify all local DLSw+ parameters as part of the local peer definition. To define a local peer, use the following command in global configuration mode:

Command	Purpose
dlsw local-peer [peer-id <i>ip-address</i>] [group <i>group</i>] [border] [cost <i>cost</i>] [if <i>size</i>] [keepalive <i>seconds</i>] [passive] [promiscuous] [init-pacing-window <i>size</i>] [max-pacing-window <i>size</i>][biu-segment]	Define the DLSw+ local peer.

Refer to “DLSw+ Using TCP Encapsulation and LLC2 Local Acknowledgment—Basic Configuration Example” for a sample configuration.

Define a DLSw+ Ring List or Port List

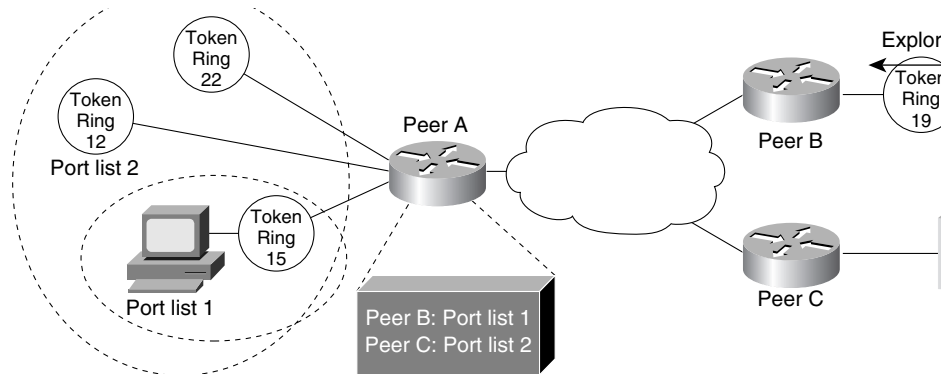
DLSw+ ring lists map traffic on a local interface to remote peers. You can create a ring list of local ring numbers and apply the list to remote peer definitions. Traffic received from a remote peer is only forwarded to the rings specified in the ring list. Traffic received from a local interface is only forwarded to peers if the input ring number appears in the ring list applied to the remote peer definition. The definition of a ring list is optional. If you want all peers and all rings to receive all traffic, you do not have to define a ring list. Simply specify 0 for the list number in the remote peer statement.

To define a ring list, use the following command in global configuration mode:

Command	Purpose
<code>dlsw ring-list list-number rings ring-number</code>	Define a ring list.

DLSw+ port lists map traffic on a local interface (either Token Ring or serial) to remote peers. Port lists do not work with Ethernet interfaces, or any other interface types connected to DLSw+ by means of a bridge group. You can create a port list of local ports and apply the list to remote peer definitions. Traffic received from a remote peer is only forwarded to peers if the input port number appears in the port list applied to the remote peer definition. The port list command provides a single command to specify both serial and Token Ring interfaces. Figure 98 shows how port lists are used to map traffic.

Figure 98 Mapping Traffic Using Port Lists



The definition of a port list is optional. If you want all peers and all interfaces to receive all traffic, you do not have to define a port list. Simply specify 0 for the list number in the remote peer statement.

To define a port list, use the following command in global configuration mode:

Command	Purpose
<code>dlsw port-list list-number type number</code>	Define a port list.

Note Either the ring list or the port list command can be used to associate rings with a given ring list. The ring list command is easier to type in if you have a large number of rings to define.

Define a DLSw+ Bridge Group List

DLSw+ bridge group lists map traffic on the local Ethernet bridge group interface to remote peers. You can create a bridge group list and apply the list to remote peer definitions. Traffic received from a remote peer is only forwarded to the bridge group specified in the bridge group list. Traffic received from a local interface is only forwarded to peers if the input bridge group number appears in the bridge group list applied to the remote peer definition. The definition of a bridge group list is

optional. Because each remote peer has a single list number associated with it, if you want traffic to go to a bridge group and to either a ring list or port list, you should specify the same list number in each definition

To define a bridge-group list, use the following command in global configuration mode:.

Command	Purpose
dlsw bgroup-list <i>list-number</i> bgroups <i>number</i>	Define a ring list.

Define DLSw+ Remote Peers

You can define three types of encapsulation on a remote peer by performing the tasks in the following sections:

- Configure TCP Encapsulation
- Configure FST Encapsulation
- Configure Direct Encapsulation

The configuration commands for each of the encapsulation methods share some optional parameters that permit support for specific features, such as backup peers. There are some command options for Transmission Control Protocol (TCP) encapsulation support features that are not available with other encapsulation types: SNA dial-on-demand routing (DDR) and configured dynamic peers. The configuration combinations for each of the encapsulation methods and features are described in greater detail following the command tables.

Configure TCP Encapsulation

To configure TCP encapsulation on a remote peer, use the following command in global configuration mode:

Command	Purpose
dlsw remote-peer <i>list-number</i> tcp <i>ip-address</i> [backup-peer <i>ip-address</i> frame-relay interface serial <i>number</i> <i>dldci-number</i> interface <i>name</i>] [bytes-netbios-out <i>bytes-list-name</i>] [cost <i>cost</i>] [dest-mac <i>mac-address</i>] [dmac-output-list <i>access-list-number</i>] [dynamic] [host-netbios-out <i>host-list-name</i>] [keepalive <i>seconds</i>] [lf <i>size</i>] [linger <i>minutes</i>] [lsap-output-list <i>list</i>] [no-llc <i>minutes</i>] [priority] [tcp-queue-max <i>size</i>] [timeout <i>seconds</i>]	Define a TCP encapsulation remote peer.

Table 7 is a list of valid port numbers used for TCP connections when the **priority** keyword is used with the **dlsw remote-peer** command:

Table 7 Valid Port Numbers for TCP Connections

Priority	Port
High	2065
Medium	1981
Normal	1982
Low	1983

Backup Peers

The **backup-peer** option is common to all encapsulation types (direct, FST, and TCP) on a remote peer and specifies that this remote peer is a backup peer for the router with the specified IP-address, Frame Relay Data-Link Control Identifier (DLCI) number, or interface name. When the primary peer fails, all circuits over this peer are disconnected and the user can start a new session via their backup peer. Prior to Cisco IOS Release 11.2(6)F, you could configure backup peers only for primary FST and TCP.

Also, when you specify the **backup-peer** option in a **dls w remote-peer tcp** command, the backup peer is activated only when the primary peer becomes unreachable. Once the primary peer is reactivated, all new sessions use the primary peer and the backup peer remains active only as long as there are LLC2 connections using it. You can use the **linger** option to specify a period (in minutes) that the backup peer remains connected after the connection to the primary peer is reestablished. When the linger period expires, the backup peer connection is taken down.

Note Because the expiration of the linger period may cause active LLC2 sessions to be terminated, you should not use the **linger** option unless you want active LLC2 sessions over the backup peer to be terminated. If the linger option is not specified the backup peer will remain active as long as circuits remain. It will not, however, pass explorers and will not create any new circuits while the primary is up.

SNA DDR

In TCP encapsulation, you can use the **keepalive** and **timeout** options to run DLSw+ over a switched line and have the Cisco IOS software take the switched line down dynamically when it is not in use. Utilizing these options gives the IP Routing table more time to converge when a network problem hinders a remote peer connection. In small networks with good IP convergence time and ISDN lines that start quickly, it is not as necessary to use the **keepalive** option. To use this feature, you must set the **keepalive** value to zero, and you may need to use a lower value for the **timeout** option than the default, which is 90 seconds.

Configured Dynamic Peers

In TCP encapsulation, the **dynamic** option and its suboptions **no-llc** and **inactivity** allow you to specify and control the activation of dynamic peers, which are configured peers that are activated only when required. Dynamic peer connections are established only when there is DLSw+ data to send. The dynamic peer connections are taken down when the last LLC2 connection using them terminates and the time period specified in the **no-llc** option expires. You can also use the **inactivity** option to take down dynamic peers when the circuits using them are inactive for a specified number of minutes.

Note Because the **inactivity** option may cause active LLC2 sessions to be terminated, you should not use this option unless you want active LLC2 sessions to be terminated.

The **dest-mac** and **dmac-output-list** options allow you to specify filter lists as part of the **dls w remote-peer** command to control access to remote peers. For static peers in direct, FST, or TCP encapsulation, these filters control which explorers are sent to remote peers. For dynamic peers in TCP encapsulation, these filters also control the activation of the dynamic peer. For example, you can specify at a branch office that a remote peer is activated only when there is an explorer frame destined for the Media Access Control (MAC) address of an Front-End Processor (FEP).

The **dest-mac** option permits the connection to be established only when there is an explorer frame destined for the specified MAC address. The **dmac-output-list** option permits the connection to be established only when the explorer frame passes the specified access list. To permit access to a single MAC address, use the **dest-mac** option, because it is a configuration “short-cut” compared to the **dmac-output-list** option.

Refer to “DLSw+ Using TCP Encapsulation and LLC2 Local Acknowledgment—Basic Configuration Example” for a sample configuration.

Local Acknowledgment

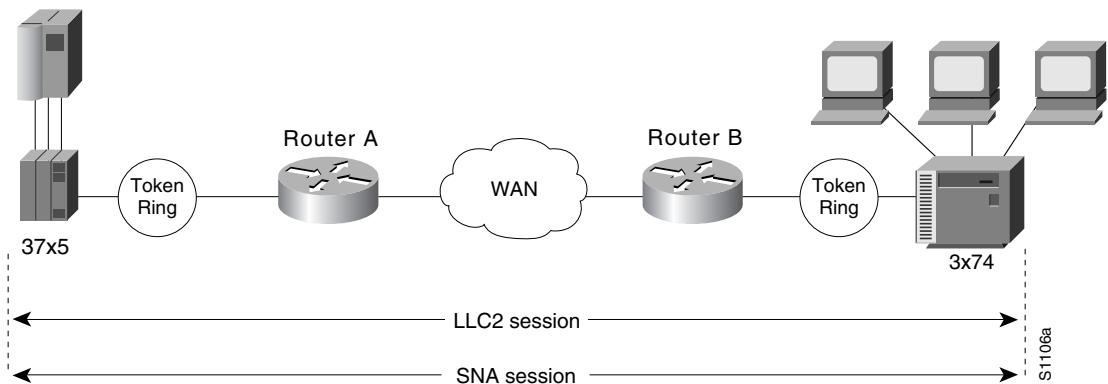
When you have LANs separated by wide geographic distances, and you want to avoid multiple retransmissions or loss of user sessions that can occur with time delays, encapsulate the source-route bridged traffic inside IP datagrams passed over a TCP connection between two routers with local acknowledgment enabled.

Logical Link Control, type 2 (LLC2) is an ISO standard data-link level protocol used in Token Ring networks. LLC2 was designed to provide reliable transmission of data across LAN media and to cause minimal or at least predictable time delays. However, RSRB and WAN backbones created LANs that are separated by wide, geographic distances—spanning countries and continents. As a result, LANs have time delays that are longer than LLC2 allows for bidirectional communication between hosts. Local acknowledgment addresses the problem of unpredictable time delays, multiple retransmissions, and loss of user sessions.

In a typical LLC2 session, when one host sends a frame to another host, the sending host expects the receiving host to respond positively or negatively in a predefined period of time commonly called the *T1 time*. If the sending host does not receive an acknowledgment of the frame it sent within the T1 time, it retries a few times (normally 8 to 10). If there is still no response, the sending host drops the session.

Figure 99 illustrates an LLC2 session in which a 37x5 on a LAN segment communicates with a 3x74 on a different LAN segment separated via a wide-area backbone network. Frames are transported between Router A and Router B by means of DLSw+. However, the LLC2 session between the 37x5 and the 3x74 is still end-to-end; that is, every frame generated by the 37x5 traverses the backbone network to the 3x74, and the 3x74, on receipt of the frame, acknowledges it.

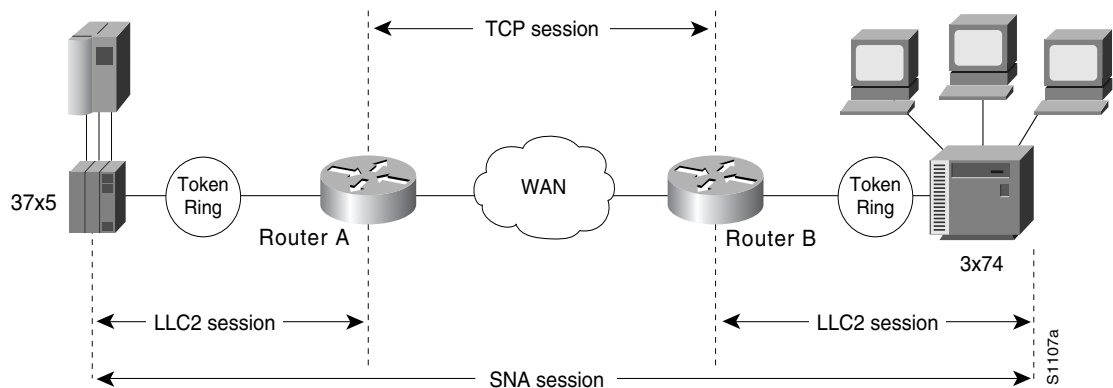
Figure 99 LLC2 Session without Local Acknowledgment



On backbone networks consisting of slow serial links, the T1 timer on end hosts could expire before the frames reach the remote hosts, causing the end host to retransmit. Retransmission results in duplicate frames reaching the remote host at the same time as the first frame reaches the remote host. Such frame duplication breaks the LLC2 protocol, resulting in the loss of sessions between the two IBM machines.

One way to solve this time delay is to increase the timeout value on the end nodes to account for the maximum transit time between the two end machines. However, in networks consisting of hundreds or even thousands of nodes, every machine would need to be reconfigured with new values. With local acknowledgment for LLC2 enabled, the LLC2 session between the two end nodes would not be end-to-end, but instead, would terminate at two local routers. Figure 100 shows the LLC2 session with the 37x5 ending at Router A and the LLC2 session with the 3x74 ending at Router B. Both Router A and Router B execute the full LLC2 protocol as part of local acknowledgment for LLC2.

Figure 100 LLC2 Session with Local Acknowledgment



With local acknowledgment for LLC2 enabled in both routers, Router A acknowledges frames received from the 37x5. The 37x5 still operates as if the acknowledgments it receives are from the 3x74. Router A looks like the 3x74 to the 37x5. Similarly, Router B acknowledges frames received from the 3x74. The 3x74 operates as if the acknowledgments it receives are from the 37x5. Router B looks like the 3x74 to 37x5. Because the frames do not have to travel the WAN backbone networks to be acknowledged, but are locally acknowledged by routers, the end machines do not time out, resulting in no loss of sessions.

Enabling local acknowledgment for LLC2 has the following advantages:

- Local acknowledgment for LLC2 solves the T1 timer problem without having to change any configuration on the end nodes. The end nodes are unaware that the sessions are locally acknowledged. In networks consisting of hundreds or even thousands of machines, this is a definite advantage. All the frames acknowledged by the Cisco IOS software appear to the end hosts to be coming from the remote IBM machine. In fact, by looking at a trace from a protocol analyzer, one cannot say whether a frame was acknowledged by the local router or by a remote IBM machine. The MAC addresses and the RIFs generated by the Cisco IOS software are identical to those generated by the remote IBM machine. The only way to find out whether a session is locally acknowledged is to use either a **show local-ack** command or a **show source-bridge** command on the router.
- All the supervisory (RR, RNR, REJ) frames that are locally acknowledged go no farther than the router. Without local acknowledgment for LLC2, *every* frame traverses the backbone. With local acknowledgment, only data (I-frames) traverse the backbone, resulting in less traffic on the

backbone network. For installations in which customers pay for the amount of traffic passing through the backbone, this could be a definite cost-saving measure. A simple protocol exists between the two *peers* to bring up or down a TCP session.

Notes on Using LLC2 Local Acknowledgment

LLC2 local acknowledgment is enabled with TCP and DLSw+ Lite remote peers.

If the LLC2 session between the local host and the router terminates in either router, the other will be informed to terminate its connection to its local host.

If the TCP queue length of the connection between the two routers reaches the high-water mark, the routers sends Receiver-Not-Ready (RNR) messages to the local hosts until the queue limit is reduced to below this limit. It is possible, however, to prevent the RNR messages from being sent by using the **dlsw llc2 nornr** command.

The configuration of the LLC2 parameters for the local Token Ring interfaces can affect overall performance. Refer to the chapter “Configuring LLC2 and SDLC Parameters” in this manual for more details about fine-tuning your network through the LLC2 parameters.

The routers at each end of the LLC2 session execute the full LLC2 protocol, which could result in some overhead. The decision to use local acknowledgment for LLC2 should be based on the speed of the backbone network in relation to the Token Ring speed. For LAN segments separated by slow-speed serial links (for example, 56 kbps), the T1 timer problem could occur more frequently. In such cases, it might be wise to turn on local acknowledgment for LLC2. For LAN segments separated by a T1, backbone delays will be minimal; in such cases, FST or direct should be considered. Speed mismatch between the LAN segments and the backbone network is one criterion to help you decide to use local acknowledgment for LLC2.

There are some situations (such as the receiving host failing between the time the sending host sends data and the time the receiving host receives it), in which the sending host would determine, *at the LLC2 layer*, that data was received when it actually was not. This error occurs because the router acknowledges that it received data from the sending host before it determines that the receiving host can actually receive the data. But because both NetBIOS and SNA have error recovery in situations where an end device goes down, these higher-level protocols will resend any missing or lost data. Because these transaction request/confirmation protocols exist above LLC2, they are not affected by tight timers, as is LLC2. They also are transparent to local acknowledgment.

If you are using NetBIOS applications, note that there are two NetBIOS timers—one at the link level and one at the next higher level. Local acknowledgment for LLC2 is designed to solve link timeouts only. If you are experiencing NetBIOS session timeouts, you have two options:

- Experiment with increasing your NetBIOS timers and decreasing your maximum NetBIOS frame size.
- Avoid using NetBIOS applications on slow serial lines.

Configure FST Encapsulation

To configure (FST) encapsulation on a remote peer, use the following command in global configuration mode:

Command	Purpose
dlsw remote-peer <i>list-number</i> fst <i>ip-address</i> [backup-peer [<i>ip-address</i> frame-relay interface serial <i>number</i> <i>dcli-number</i> interface name]] [bytes-netbios-out <i>bytes-list-name</i>] [cost <i>cost</i>] [dest-mac <i>mac-address</i>] [dmac-output-list <i>access-list-number</i>] [host-netbios-out <i>host-list-name</i>] [keepalive <i>seconds</i>] [lf <i>size</i>] [linger <i>minutes</i>] [lsap-output-list <i>list</i>]	Define an FST encapsulation remote peer.

Configure Direct Encapsulation

Direct encapsulation is supported over High-Level Data Link Control (HDLC) and Frame Relay. Direct encapsulation over Frame Relay comes in two forms: DLSw Lite (LLC2 encapsulation) and Passthru. To configure direct encapsulation, use the following commands in global configuration mode.

Step	Command	Purpose
1	dlsw remote-peer <i>list-number</i> frame-relay interface serial <i>number</i> <i>dcli-number</i> [backup-peer [<i>ip-address</i> frame-relay interface serial <i>number</i> <i>dcli-number</i> interface name]] [bytes-netbios-out <i>bytes-list-name</i>] [cost <i>cost</i>] [dest-mac <i>mac-address</i>] [dmac-output-list <i>access-list-number</i>] [host-netbios-out <i>host-list-name</i>] [keepalive <i>seconds</i>] [lf <i>size</i>] [linger <i>minutes</i>] [lsap-output-list <i>list</i>] pass-thru	Define a direct encapsulation in Frame Relay for the remote peer.
2	dlsw remote-peer <i>list-number</i> interface serial <i>number</i> [backup-peer [<i>ip-address</i> frame-relay interface serial <i>number</i> <i>dcli-number</i> interface name]] [bytes-netbios-out <i>bytes-list-name</i>] [cost <i>cost</i>] [dest-mac <i>mac-address</i>] [dmac-output-list <i>access-list-number</i>] [host-netbios-out <i>host-list-name</i>] [keepalive <i>seconds</i>] [lf <i>size</i>] [linger <i>minutes</i>] [lsap-output-list <i>list</i>] pass-thru	Define a direct encapsulation in HDLC for the remote peer.

Note To distinguish between DLSw Lite and Passthru, use the **pass-thru** option in the **dlsw remote peer** command

Enable DLSw+ over Frame Relay

You can configure the Cisco IOS software for direct encapsulation of DLSw+ in Frame Relay according to RFC 1490. DLSw+ direct encapsulation over RFC 1490 supports either pass-through or local acknowledgment (data link control termination). DLSw+ supports transport for both SNA and NetBIOS data. SNA physical unit (PU) 2.0 and PU 2.1 are supported.

The configuration requires a minimum number of Permanent Virtual Circuits (PVCs) (since multiple protocols can share a single PVC). A minimum number of PVCs simplifies configuration because multiple PUs can share a PVC without requiring configuration of multiple Service Access Points (SAPs).

The **pass-thru** option adds the smallest amount of link overhead and requires minimal processing cycles in the attaching routers when compared to TCP/IP encapsulation.

Enable DLSw+ on a Token Ring or FDDI Interface

The local acknowledgment option prevents data link control timeouts during periods of WAN congestion. It also minimizes WAN traffic by keeping data link control acknowledgments and keepalives off the WAN.

No controller or FEP upgrades are required to take advantage of this feature (if already Token Ring attached), reducing costs and simplifying migration to Frame Relay.

To enable DLSw+ over Frame Relay with passthru, use the following commands, beginning in global configuration mode:

Step	Command	Purpose
1	interface serial <i>number</i>	Specify the serial port.
2	encapsulation frame-relay	Enable Frame Relay encapsulation.
3	frame-relay map dlsw <i>dlsi-number</i>	Define the mapping between DLSw+ and the DLCI.
4	dlsw remote-peer <i>list-number</i> frame-relay interface <i>serial number dlsi-number</i> pass-thru	Define direct encapsulation in Frame Relay.

Note The DLSw+ remote peer statement should specify **pass-thru**.

To enable DLSw+ over Frame Relay with local acknowledgment (also known as DLSw Lite), use the following commands, beginning in global configuration mode:

Step	Command	Purpose
1	interface serial <i>number</i>	Specify the serial port.
2	encapsulation frame-relay	Enable Frame Relay encapsulation.
3	frame-relay map llc2 <i>dlsi-number</i>	Define the mapping between DLSw+ and the DLCI.
4	fdlsw remote-peer <i>list-number</i> frame-relay interface <i>serial number dlsi-number</i>	Define direct encapsulation in Frame Relay.

Enable DLSw+ on a Token Ring or FDDI Interface

To enable DLSw+ on a Token Ring or FDDI interface, use the following command in interface configuration mode:

Command	Purpose
source-bridge <i>local-ring bridge-number ring-group</i>	Enable DLSw+ on a Token Ring or FDDI interface.

By default, DLSw+ receives traffic from all defined source-bridge ring-groups in the router. By configuring SRB on the interface, it permits DLSw+ to communicate on that interface.

Note The *ring-group* number specified in the **source-bridge** command must be the number of a defined source-bridge ring-group or DLSw+ will not see this interface.

Enable DLSw+ on an Ethernet Interface

To enable DLSw+ on certain Ethernet interfaces, use the following command in global configuration mode:

Command	Purpose
dlsw bridge-group <i>group-number</i>	Enable DLSw+ on an Ethernet interface.
bridge group <i>bridge group</i>	Number of the bridge group to which the interface belongs.

In interface mode, the interfaces must also be put into the bridge-group.

Note By default, the Cisco IOS software translates Token Ring LLC2 to Ethernet 802.3 LLC2. To configure the router to translate Token Ring LLC2 frames into Ethernet 0x80d5 format frames, refer to the section “Enable Token Ring LLC2-to-Ethernet Conversion” in the “Configuring Source-Route Bridging” chapter of the *Bridging and IBM Networking Command Reference*.

Enable DLSw+ on an SDLC Interface

To establish devices as SDLC stations, use the following commands in interface configuration mode:

Step	Command	Purpose
1	encapsulation sdlc	Set the encapsulation type of the serial interface to SDLC.
2	sdlc role { none primary secondary prim-xid-poll }	Establish the role of the interface.
3	sdlc vmac <i>mac-address</i> ¹	Configure a MAC address for the serial interface.
4	sdlc address <i>hexbyte</i> [echo]	Assign a set of secondary stations attached to the serial link.
5	sdlc partner <i>mac-address</i> <i>sdlc-address</i>	Specify the destination address with which an LLC session is established for the SDLC station.

1 The last byte of the MAC address must be 00.

To enable DLSw+ on an SDLC interface, use the following command in interface configuration mode:

Command	Purpose
sdlc dlsw { <i>sdlc-address</i> default partner <i>mac-address</i> [inbound outbound] }	Enable DLSw+ on an SDLC interface.

Use the **default** option if you have more than 10 SDLC devices to attach to the DLSw+ network. To configure an SDLC multidrop line downstream, you configure the SDLC role as either **primary** or **prim-xid-poll**. SDLC role **primary** specifies that any PU without the **xid-poll** parameter in the **sdlc address** command is a PU 2.0 device. SDLC role **prim-xid-poll** specifies that every PU is type 2.1. We recommend that you specify **sdlc role primary** if all SDLC devices are type PU 2.0 or a mix of PU 2.0 and PU 2.1. Specify **sdlc role prim-xid-poll** if all devices are type PU 2.1

To configure DLSw+ to support LLC2-to-SDLC conversion for PU 4 or PU 5 devices, specify the **echo** option in the **sdlc address** command. A PU 4-to-PU 4 configuration requires that **none** be specified in the **sdlc role** command.

Refer to the sections “DLSw+ with SDLC Multidrop Support Configuration Examples” and “DLSw+ with LLC2-to-SDLC Conversion between PU 4-to-PU 4 Communication” for sample configurations.

Enable DLSw+ over QLLC

You can configure DLSw+ for QLLC connectivity, which enables both of the following scenarios:

- Remote LAN-attached devices (physical units) or SDLC-attached devices can access an FEP or an AS/400 over an X.25 network.

Our QLLC support allows remote X.25-attached SNA devices to access an FEP without requiring X.25 NCP Packet Switching Interface (NPSI) in the FEP. This may eliminate the requirement for NPSI (if GATE and DATE are not required), thereby eliminating the recurring license cost. In addition, because the QLLC attached devices appear to be Token Ring-attached to the Network Control Program (NCP), they require no preconfiguration in the FEP. Remote X.25-attached SNA devices can also connect to an AS/400 over Token Ring using this support.

- Remote X.25-attached SNA devices can access an FEP or an AS/400 over a Token Ring or over SDLC.

For environments just beginning to migrate to LANs, our QLLC support allows deployment of LANs in remote sites while maintaining access to the FEP over existing NPSI links. Remote LAN-attached devices (physical units) or SDLC-attached devices can access a FEP over an X.25 network without requiring X.25 hardware or software in the LAN-attached devices. The Cisco IOS software supports direct attachment to the FEP over X.25 without the need for routers at the data center for SNA traffic.

To enable QLLC connectivity for DLSw+, use the following commands in interface configuration mode:

Step	Command	Purpose
1	encapsulation x.25	Specify an interface as an X.25 device.
2	x25 address subaddress	Activate X.25 subaddresses.
3	x25 map qlc x121-addr [x25-map-options]	Associate a virtual MAC address with the X.121 address of the remote X.25 device.
4	qlc dlsw {subaddress subaddress pvc pvc-low [pvc-high]} [vmac vmacaddr [poolsize]] [partner partner-macaddr] [sap ssap dsap] [xid xidstring] [npsi-poll]	Enable DLSw+ over QLLC.

Enable NetBIOS DDR

DLSw+ can filter NetBIOS Session Alive packets from the WAN. NetBIOS periodically sends Session Alive packets as LLC2 I-frames. These packets do not require a response and are superfluous to the function of proper data flow. Furthermore, these packets keep dial-on-demand interfaces up and this up time causes unwanted per-packet charges in DDR networks. By filtering these NetBIOS Session Alive packets, you reduce traffic on the WAN as well as some costs that are associated with DDR.

To enable NetBIOS DDR, use the following command in global configuration mode:

Command	Purpose
<code>dlsw netbios keepalive-filter</code>	Enable NetBIOS DDR.

Tune the DLSw+ Configuration

To modify an existing configuration parameter, perform one or more of the tasks in the following sections:

- Configure DLSw+ Timers
- Configure Maximum Entries in Group Cache
- Configure Peer-on-Demand Defaults
- Configure Promiscuous Peer Defaults
- Configure Static Resources Capabilities Exchange
- Configure Static Paths
- Configure Duplicate Path Handling
- Disable Border Peer Caching
- Disable UDP Unicast
- Enable RIF Passthru

Configure DLSw+ Timers

To configure DLSw+ timers (including group cache entries), use the following command in global configuration mode:

Command	Purpose
<code>dlsw timer { 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 } time</code>	Configure DLSw+ timers.

Configure Maximum Entries in Group Cache

To define the maximum entries in a group cache, use the following command in global configuration mode:

Command	Purpose
dlsw group-cache max-entries <i>number</i>	Define the maximum entries in a group cache.

Configure Peer-on-Demand Defaults

To configure peer-on-demand defaults, use the following command in global configuration mode:

Command	Purpose
dlsw peer-on-demand-defaults [<i>fst</i>] [bytes-netbios-out <i>bytes-list-name</i>] [cost <i>cost</i>] [dest-mac <i>destination mac-address</i>] [dmac-output-list <i>access-list-number</i>] [host-netbios-out <i>host-list-name</i>] [inactivity <i>minutes</i>] [keepalive <i>seconds</i>] [If <i>size</i>] [lsap-output-list <i>list</i>] [port-list <i>port-list-number</i>] [priority] [tcp-queue-max]	Configure peer-on-demand defaults.

Configure Promiscuous Peer Defaults

To configure promiscuous peer defaults, use the following command in global configuration mode:

Command	Purpose
dlsw prom-peer-defaults [bytes-netbios-out <i>bytes-list-name</i>] [cost <i>cost</i>] [dest-mac <i>destination-mac-address</i>] [dmac-output-list <i>access-list-number</i>] [host-netbios-out <i>host-list-name</i>] [keepalive <i>seconds</i>] [If <i>size</i>] [lsap-output-list <i>list</i>] [tcp-queue-max <i>size</i>]	Configure promiscuous peer defaults.

Configure Static Resources Capabilities Exchange

To reduce explorer traffic destined for this peer, the peer can send other peers a list of resources for which it has information (**icanreach**) or does not have information (**icannotreach**). This information is exchanged as part of a capabilities exchange. To configure static resources that will be exchanged as part of a capabilities exchange, use one of the following commands in global configuration mode:

Command	Purpose
dlsw icannotreach saps <i>sap [sap...]</i>	Configure a resource not locally reachable by the router.
or	or
dlsw icanreach { mac-exclusive netbios-exclusive mac-address <i>mac-addr</i> [mask <i>mask</i>] netbios-name <i>name</i> saps }	Configure a resource locally reachable by the router.

Configure Static Paths

To configure static paths to minimize explorer traffic originating in this peer, use one of the following commands in global configuration mode:

Command	Purpose
dls mac-addr <i>mac-addr</i> { ring <i>ring number</i> remote-peer { interface serial number ip-address ip-address } rif rif string group group }	Configure the location or path of a static MAC address.
or	or
dls netbios-name <i>netbios-name</i> { ring <i>ring number</i> remote-peer { interface serial number ip-address ip-address } rif rif string group group }	Configure a static NetBIOS name.

Configure Duplicate Path Handling

To configure duplicate path handling, use the following command in global configuration mode:

Command	Purpose
dls duplicate-path-bias [load-balance]	Configure duplicate path handling.

Disable Border Peer Caching

To disable the border peer caching feature, use the following command in global configuration mode:

Command	Purpose
dls group-cache disable	Disable the border peer caching feature.

Disable UDP Unicast

DLSw Version 2 uses User Datagram Protocol (UDP) Unicast in response to an IP multicast. When address resolution packets (CANREACH_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.

UDP Unicast uses UDP source port 0. However, some firewall products treat packets that use UDP source port 0 as security violations, discarding the packets and preventing DLSw connections. To avoid this situation, use one of the following procedures:

- Configure the firewall to allow UDP packets to use UDP source port 0.
- Use the **dls udp-disable** command to disable UDP Unicast and send address resolution packets in the existing TCP session.

To disable UDP Unicast, use the following command in global configuration mode:

Command	Purpose
dls udp-disable	Disable UDP Unicast.

Enable RIF Passthru

By default, DLSw+ terminates the Routing Information Field (RIF) for Token Ring, terminates the LLC for all media types, and forwards data only across a WAN with DLSw+ and TCP/IP headers. The RIF is a field in source-route bridged frames that indicates the SRB path the frame should take when traversing a Token Ring network. In the case of an explorer packet, the RIF is a field in the source-route bridged frame that indicates the SRB path that the SRB explorer has traversed so far. The RIF is limited to seven hop counts by the IBM standards. Because DLSw+ terminates the RIF at the virtual ring, the network's scalability increases because the hop count of the packet starts over, and the packet can traverse seven additional hops. Also, RIF termination simplifies network design because ring numbers no longer have to be unique throughout an entire enterprise.

Some environments do not function properly if the RIF is terminated. For that reason, DLSw+ now supports the RIF Passthru feature, in which the entire source-route bridged path appears in the RIF.

The RIF Passthru feature enables two key functions when DLSw+ is used between FEPs (PU 4s). First, RIF Passthru is required to allow multiple active paths between FEPs. Second, RIF Passthru is required to remotely load an NCP (in other words, set initialization mode/request initialization mode support). See the **dlsw remote-peer tcp** command for further usage guidelines.

To enable RIF Passthru, use the following command in global configuration mode:

Command	Purpose
dlsw remote-peer tcp [rif-passthru <i>virtual-ring-number</i>]	Enable RIF Passthru.

Monitor and Maintain the DLSw+ Network

To monitor and maintain activity on the DLSw+ network, use one or more of the following commands in privileged EXEC mode:

Command	Purpose
clear dlsw circuit	Close all the DLSw+ circuits ¹
clear dlsw reachability	Remove all entries from the DLSw+ reachability cache.
clear dlsw statistics	Reset to zero the number of frames that have been processed in the local, remote, and group caches.
show dlsw capabilities interface <i>type number</i>	Display capabilities of a direct-encapsulated remote peer.
show dlsw capabilities ip-address <i>ip-address</i>	Display capabilities of a TCP/FST remote peer.
show dlsw capabilities local	Display capabilities of the local peer.
show dlsw circuits	Display DLSw+ circuit information.
show dlsw fastcache	Display the fast cache for FST and direct-encapsulated peers.
show dlsw peers	Display DLSw+ peer information.
show dlsw reachability	Display DLSw+ reachability information.
dlsw disable	Disable and re-enable DLSw+ without altering the configuration.

Command	Purpose
<code>show dlsw statistics [border-peers]</code>	Display the number of frames that have been processed in the local, remote, and group caches.

- 1 Issuing the `clear dlsw circuits` command will cause the loss of any associated LLC2 sessions.

DLSw+ Configuration Examples

The following sections provide DLSw+ configuration examples:

- DLSw+ Using TCP Encapsulation and LLC2 Local Acknowledgment—Basic Configuration Example
- DLSw+ Using TCP Encapsulation with Local Acknowledgment—Peer Group Configuration Example 1
- DLSw+ Using TCP Encapsulation with Local Acknowledgment—Peer Group Configuration Example 2
- DLSw+ with SDLC Multidrop Support Configuration Examples
- DLSw+ with LLC2-to-SDLC Conversion between PU 4-to-PU 4 Communication
- DLSw+ Translation between Ethernet and Token Ring Configuration Example
- DLSw+ Translation between FDDI and Token Ring Configuration Example
- DLSw+ Translation between SDLC and Token Ring Media Example
- DLSw+ over Frame Relay Configuration Example
- DLSw+ over QLLC Configuration Examples
- DLSw+ with RIF Passthru Configuration Example

DLSw+ Using TCP Encapsulation and LLC2 Local Acknowledgment—Basic Configuration Example

This sample configuration requires the following tasks, which are described in earlier sections of this document:

- Define a Source-Bridge Ring Group for DLSw+
- Define a DLSw+ Local Peer for the Router
- Define DLSw+ Remote Peers
- Enable DLSw+ on the Appropriate LAN Interface

Figure 101 illustrates a DLSw+ configuration with local acknowledgment.

Figure 101 DLSw+ with Local Acknowledgment—Simple Configuration



Router A

```

source-bridge ring-group 10
!
dlsw local-peer peer-id 10.2.25.1
dlsw remote-peer 0 tcp 10.2.5.2
  interface loopback 0
  ip address 10.2.25.1 255.255.255.0
.
.
.
interface tokenring 0
  no ip address
  ring-speed 16
  source-bridge active 25 1 10
  source-bridge spanning
    
```

Router B

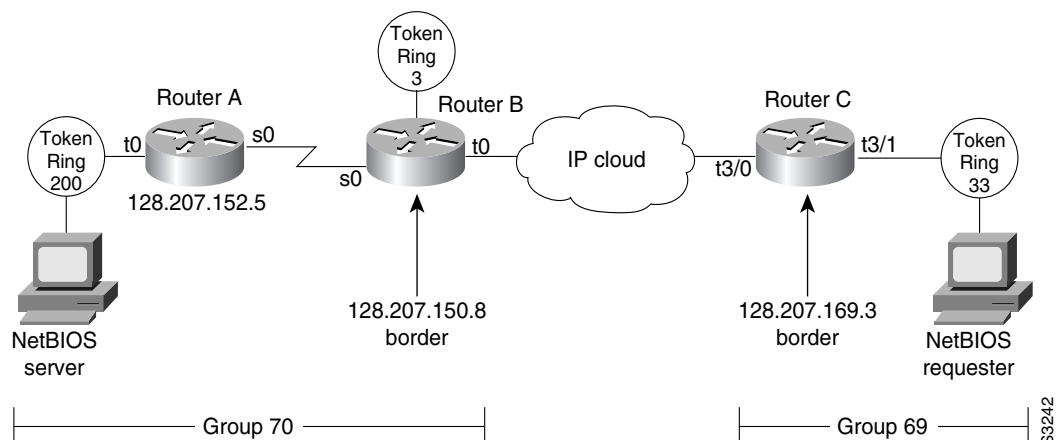
```

source-bridge ring-group 12
dlsw local-peer peer-id 10.2.5.2
dlsw remote-peer 0 tcp 10.2.25.1
  interface loopback 0
  ip address 10.2.5.2 255.255.255.0
.
.
.
interface tokenring 0
  no ip address
  ring-speed 16
  source-bridge active 5 1 12
  source-bridge spanning
    
```

DLSw+ Using TCP Encapsulation with Local Acknowledgment—Peer Group Configuration Example 1

Figure 102 illustrates border peers with TCP encapsulation, showing circuits to each other. Router A is configured to operate in promiscuous mode, and border peers Routers B and C forward broadcasts. This configuration reduces processing requirements in Router A (the access router) and still supports any-to-any networks.

Figure 102 DLSw+ with Peer Groups Specified (Example 1)



Router A

```
hostname RouterA
!
source-bridge ring group 31
dlsw local-peer peer-id 128.207.152.5 group 70 promiscuous
dlsw remote peer 0 tcp 128.207.150.8
!
interface serial 0
 ip unnumbered tokenring
  clockrate 56000
!
interface tokenring 0
 ip address 128.207.152.5 255.255.255.0
 ring-speed 16
 source-bridge 200 13 31
 source-bridge spanning
!
.
.
.
router igrp 777
network 128.207.0.0
```

Router B

```
hostname RouterB
!
.
.
.
source-bridge ring-group 31
dlsw local-peer peer-id 128.207.150.8 group 70 border promiscuous
dlsw remote-peer 0 tcp 128.207.169.3
!
.
.
.
interface serial 0
 ip unnumbered tokenring 0
  bandwidth 56
!
.
.
.
interface tokenring 0
 ip address 128.207.150.8 255.255.255.0
 ring-speed 16
 source-bridge 3 14 31
 source-bridge spanning
!
router igrp 777
network 128.207.0.0
```

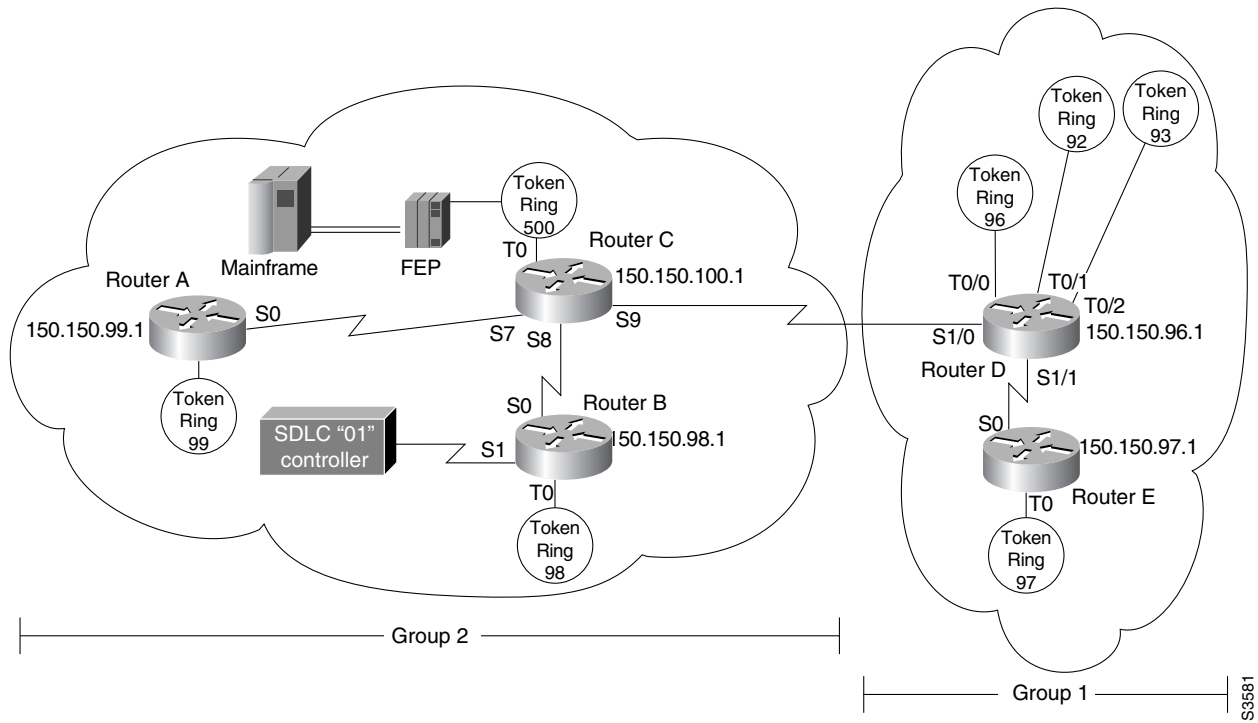
Router C

```
hostname RouterC
!
.
.
.
source-bridge ring-group 69
dlsw local-peer peer-id 128.207.169.3 group 69 border promiscuous
dlsw remote-peer 0 tcp 128.207.150.8
!
.
.
.
interface tokenring 3/0
description fixed to flashnet
ip address 128.207.2.152 255.255.255.0
ring-speed 16
multiring all
!
interface tokenring 3/1
ip address 128.207.169.3 255.255.255.0
ring-speed 16
source-bridge 33 2 69
source-bridge spanning
!
.
.
.
router igrp 777
network 128.207.0.0
```

DLSw+ Using TCP Encapsulation with Local Acknowledgment—Peer Group Configuration Example 2

Figure 103 illustrates a peer group configuration that allows any-to-any connection except for Router B. Router B has no connectivity to anything except router C because the **promiscuous** keyword is omitted.

Figure 103 DLSw+ with Peer Groups Specified (Example 2)



Router A

```

hostname Router A
!
.
.
.
source-bridge ring-group 2000
dlsw local-peer peer-id 150.150.99.1 group 2 promiscuous
dlsw remote-peer 0 tcp 150.150.100.1
!
interface loopback 0
 ip address 150.150.99.1 255.255.255.192
!
!
.
.
.
interface tokenring 0
 no ip address
 ring-speed 16
 source-bridge 99 1 2000
 source-bridge spanning
!
.
.
.
router eigrp 202
 network 150.150.0.0

```

Router B

```

hostname RouterB
!
.
.
.
source-bridge ring-group 2000
dlsw local-peer peer-id 150.150.98.1 group 2
dlsw remote-peer 0 tcp 150.150.100.1
!
interface loopback 0
 ip address 150.150.98.1 255.255.255.192
!
.
.
.
interface serial 1
 no ip address
 encapsulation sdhc
 no keepalive
 clockrate 9600
 sdhc role primary
 sdhc vmac 4000.8888.0100
 sdhc address 01
 sdhc xid 01 05d20006
 sdhc partner 4000.1020.1000 01
 sdhc dlsw 1
!
interface tokenring 0
 no ip address
 ring-speed 16
 source-bridge 98 1 2000
 source-bridge spanning
!
.
.
.
router eigrp 202
 network 150.150.0.0

```

Router C

```

hostname RouterC
!
.
.
.
source-bridge ring-group 2000
dlsw local-peer peer-id 150.150.100.1 group 2 border promiscuous
dlsw remote-peer 0 tcp 150.150.96.1
!
interface loopback 0
 ip address 150.150.100.1 255.255.255.192
!
.
.
.
!
interface tokenring 0
 no ip address
 ring-speed 16
 source-bridge 500 1 2000
!

```

```
router eigrp 202
network 150.150.0.0
```

Router D

```
hostname RouterD
!
.
.
.
source-bridge ring-group 2000
dlsw local-peer peer-id 150.150.96.1 group 1 border promiscuous
dlsw remote-peer 0 tcp 150.150.100.1
!
interface loopback 0
ip address 150.150.96.1 255.255.255.192
!
.
.
!
.
.
.
interface tokenring 0/0
no ip address
ring-speed 16
source-bridge 96 1 2000
source-bridge spanning
!
interface tokenring 0/1
no ip address
ring-speed 16
source-bridge 92 1 2000
source-bridge spanning
!
.
.
.
router eigrp 202
network 150.150.0.0
```

Router E

```
hostname RouterE
!
.
.
.
source-bridge ring-group 2000
dlsw local-peer peer-id 150.150.97.1 group 1 promiscuous
dlsw remote-peer 0 tcp 150.150.96.1
!
interface loopback 0
ip address 150.150.97.1 255.255.255.192
!
!
.
.
.
interface tokenring 0
no ip address
ring-speed 16
source-bridge 97 1 2000
```

```

source-bridge spanning
!
.
.
router eigrp 202
network 150.150.0.0

```

DLSw+ with SDLC Multidrop Support Configuration Examples

In the following example, all devices are type PU 2.0,

```

interface serial 2
mtu 4400
no ip address
encapsulation sdhc
no keepalive
clockrate 19200
sdhc role primary
sdhc vmac 4000.1234.5600
sdhc address C1
sdhc xid C1 05DCCCC1
sdhc partner 4001.3745.1088 C1
sdhc address C2
sdhc xid C2 05DCCCC2
sdhc partner 4001.3745.1088 C2
sdhc dlsw C1 C2

```

The following example shows mixed PU 2.0 (device using address C1) and PU 2.1 (device using address C2) devices:

```

interface serial 2
mtu 4400
no ip address
encapsulation sdhc
no keepalive
clockrate 19200
sdhc role primary
sdhc vmac 4000.1234.5600
sdhc address C1
sdhc xid C1 05DCCCC1
sdhc partner 4001.3745.1088 C1
sdhc address C2 xid-poll
sdhc partner 4001.3745.1088 C2
sdhc dlsw C1 C2

```

In the following example, all devices are type PU 2.1 (Method 1):

```

interface serial 2
mtu 4400
no ip address
encapsulation sdhc
no keepalive
clockrate 19200
sdhc role primary
sdhc vmac 4000.1234.5600
sdhc address C1 xid-poll
sdhc partner 4001.3745.1088 C1
sdhc address C2 xid-poll
sdhc partner 4001.3745.1088 C2
sdhc dlsw C1 C2

```

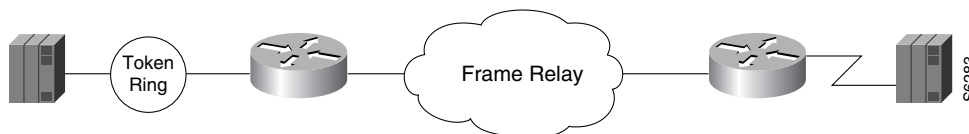
In the following example, all devices are type PU 2.1 (Method 2):

```
interface serial 2
  mtu 4400
  no ip address
  encapsulation sdlc
  no keepalive
  clockrate 19200
  sdlc role prim-xid-poll
  sdlc vmac 4000.1234.5600
  sdlc address C1
  sdlc partner 4001.3745.1088 C1
  sdlc address C2
  sdlc xid C2 05DCCCC2
  sdlc partner 4001.3745.1088 C2
  sdlc dlsw C1 C2
```

DLSw+ with LLC2-to-SDLC Conversion between PU 4-to-PU 4 Communication

The following example is a sample configuration for LLC2-to-SDLC conversion for PU 4-to-PU 4 communication as shown in Figure 104:

Figure 104 LLC2-to-SDLC Conversion for PU 4-to-PU 4 Communication



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 llc2 46
  frame-relay map llc2 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 llc 4000.1060.1000 4 4 Serial0 frame-relay 45 4 4
```

Router B

```

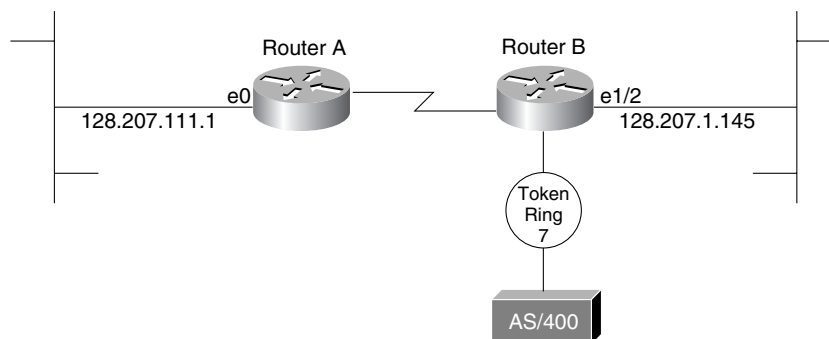
interface serial 0
  mtu 4096
  ip address 10.4.21.3 255.255.255.0
  encapsulation frame-relay IETF
  keepalive 12
  no fair-queue
  frame-relay map llc2 53
  frame-relay map llc2 54
  frame-relay map llc2 56
  frame-relay map ip 10.4.21.1 53 broadcast
  frame-relay map ip 10.4.21.2 54 broadcast
  frame-relay map ip 10.4.21.4 56 broadcast
  frame-relay lmi-type ansi

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

DLSw+ Translation between Ethernet and Token Ring Configuration Example

DLSw+ also supports Ethernet media. The configuration is similar to other DLSw+ configurations, except for configuring for a specific media. The following example shows Ethernet media (see Figure 105).

Figure 105 DLSw+ Translation between Ethernet and Token Ring



Router A

```
hostname RouterA
!
.
.
.
dlsw local-peer peer-id 128.207.111.1
dlsw remote-peer 0 tcp 128.207.1.145 lf 1500
dlsw bridge-group 5
!
interface Ethernet 0
 ip address 128.207.111.1 255.255.255.0
 bridge-group 5
!
.
.
bridge 5 protocol ieee
!
.
```

Router B

```
hostname RouterB
!
.
.
.
source-bridge transparent 500 1000 1 5
dlsw local-peer peer-id 128.207.1.145
dlsw remote-peer 0 tcp 128.207.111.1 lf 1500
dlsw bridge-group 5
.
.
.
interface ethernet 1/2
 ip address 128.207.1.145 255.255.255.0
 bridge-group 5
.
.
.
interface tokenring 2/0
 no ip address
 ring-speed 16
 source-bridge 7 1 500
 source-bridge spanning
!
.
.
.
bridge 5 protocol ieee
```

Because DLSw+ does not do local translation between different LAN types, Router B must be configured for SR/TLB by issuing the **source-bridge transparent** command. Also, note that the bridge groups are configured on the ethernet interfaces.

DLSw+ Translation between FDDI and Token Ring Configuration Example

DLSw+ also supports FDDI media. , in this case FDDI. The configuration is similar to other DLSw+ configurations except for configuring for a specific media type. The following example shows FDDI media (see Figure 106).

Figure 106 DLSw+ Translation between FDDI and Token Ring



In the following configuration, an FDDI ring on Router A is connected to a Token Ring on Router B across a DLSw+ link.

Router A

```
source-bridge ring-group 10
dlsw local-peer peer-id 132.11.11.2
dlsw remote-peer 0 tcp 132.11.11.3
interface fddi 0
no ip address
source-bridge active 26 1 10
source-bridge spanning
```

Router B

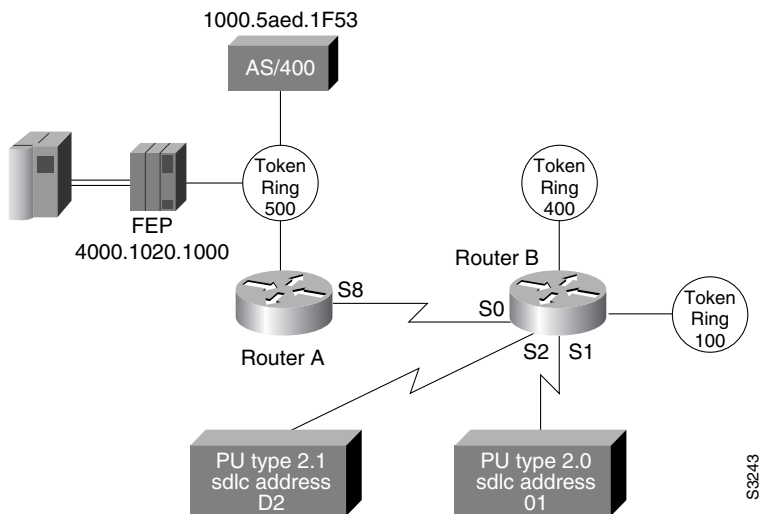
```
source-bridge ring-group 10
dlsw local peer peer-id 132.11.11.3
dlsw remote-peer 0 tcp 132.11.11.2
interface tokenring 0
no ip address
source-bridge active 25 1 10
source-bridge spanning
```

DLSw+ Translation between SDLC and Token Ring Media Example

DLSw+ provides media conversion between local or remote LANs and SDLC. For additional information about configuring SDLC parameters, refer to the chapter “Configuring LLC2 and SDLC Parameters.”

Figure 107 illustrates DLSw+ with SDLC encapsulation. For this example, 4000.1020.1000 is the MAC address of the FEP host (PU 4.0). The MAC address of the AS/400 host is 1000.5aed.1f53, which is defined as Node Type 2.1. Router B serves as the primary station for the remote secondary station 01. Router B can serve as either primary station or secondary station to remote station D2.

Figure 107 DLSw+ Translation between SDLC and Token Ring Media



Router A

```

hostname RouterA
!
.
.
.
source-bridge ring-group 2000
dlsw local-peer peer-id 150.150.10.2
dlsw remote-peer 0 tcp 150.150.10.1
!
.
.
.
interface serial 8
ip address 150.150.10.2 255.255.255.192
clockrate 56000
!
.
.
.
interface tokenring 0
no ip address
ring-speed 16
source-bridge 500 1 2000
source-bridge spanning
!
.
.
.
router eigrp 202
network 150.150.0.0
    
```

Router B

```

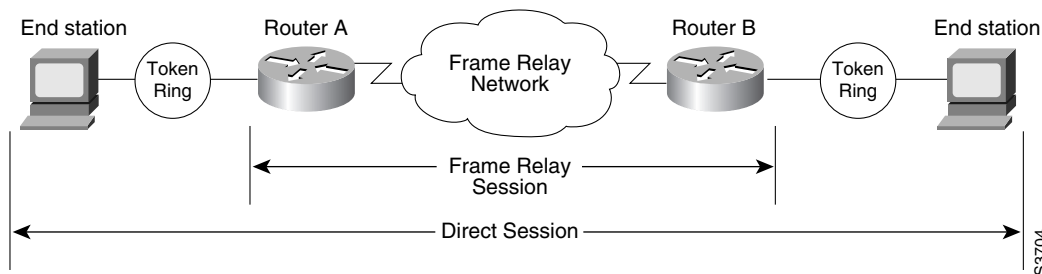
hostname RouterB
!
.
.
.
source-bridge ring-group 2000
dlsw local-peer peer-id 150.150.10.1
dlsw remote-peer 0 tcp 150.150.10.2
!
.
.
.
interface serial 0
 ip address 150.150.10.1 255.255.255.192
!
interface serial 1
 description PU2 with SDLC station role set to secondary
 no ip address
 encapsulation sdslc
 no keepalive
 clockrate 9600
 sdslc role primary
 sdslc vmac 4000.9999.0100
 sdslc address 01
 sdslc xid 01 05d20006
 sdslc partner 4000.1020.1000 01
 sdslc dlsw 1
!
interface serial 2
 description Node Type 2.1 with SDLC station role set to negotiable or primary
 encapsulation sdslc
 sdslc role none
 sdslc vmac 1234.3174.0000
 sdslc address d2
 sdslc partner 1000.5aed.1f53 d2
 sdslc dlsw d2
!
interface tokenring 0
 no ip address
 ring-speed 16
 source-bridge 100 1 2000
 source-bridge spanning
!
interface tokenring 1
 no ip address
 ring-speed 16
 source-bridge 400 1 2000
 source-bridge spanning
!
router eigrp 202
 network 150.150.0.0

```

DLSw+ over Frame Relay Configuration Example

Frame Relay support extends the DLSw+ capabilities to include Frame Relay in direct mode. Frame Relay support includes permanent virtual circuit capability. DLSw+ runs over Frame Relay with or without local acknowledgement. It supports the Token Ring-to-Token Ring connections similar to Fast-Sequenced Transport and other direct data link controls. Figure 108 illustrates a DLSw+ configuration over Frame Relay with RIF passthrough.

Figure 108 DLSw+ over Frame Relay



The following configuration examples are based on Figure 108. The Token Rings in the illustration are in Ring 2.

Router A

```

source-bridge ring-group 100
dlsw local-peer
dlsw remote-peer 0 frame-relay interface serial 0 30 passthru
!
interface tokenring 0
ring-speed 16
source-bridge active 1 1 100
!
interface serial 0
mtu 3000
no ip address
encapsulation frame-relay
frame-relay lmi-type ansi
frame-relay map dlsw 30
    
```

Router B

```

source-bridge ring-group 100
dlsw local-peer
dlsw remote-peer 0 frame-relay interface serial 0 30 passthru
!
interface tokenring 0
ring-speed 16
source-bridge active 2 1 100
!
interface serial 0
mtu 3000
no ip address
encapsulation frame-relay
frame-relay lmi-type ansi
frame-relay map dlsw 30
    
```

DLSw+ over QLLC Configuration Examples

The following three examples describe QLLC support for DLSw+.

Example 1

In this configuration, DLSw+ is 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 cluster controller is given a virtual MAC address of 1000.0000.0001. This virtual MAC address is mapped to the X.121 address of the 3174 (31104150101) in the X.25 attached router.

```
interface serial 0
  encapsulation x25
  x25 address 3110212011
  x25 map qllc 1000.0000.0001 31104150101
  qllc dlsw partner 4000.1611.1234
```

Example 2

In this configuration, a single IBM 3174 cluster controller 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 151102.

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, and a source SAP of 08 when communicating with the AS/400.

```
interface serial 0
  encapsulation x25
  x25 address 31102
  x25 map qllc 1000.0000.0001 33204
  qllc dlsw subaddress 150101 partner 4000.1161.1234
  qllc dlsw subaddress 150102 partner 4000.2034.5678 sap 04 08
```

Example 3

In this example, two different X.25 resources want to communicate over X.25 to the same FEP.

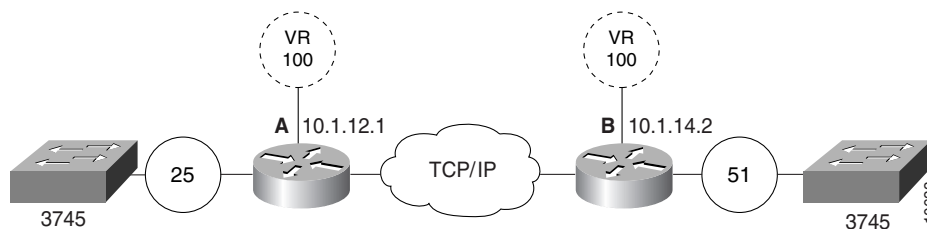
In the router attached to the X.25 network, every X.25 connection request for X.121 address 31102150101 is directed to DLSw+. The first SVC to be established will be mapped to virtual MAC address 1000.0000.0001. The second SVC to be established will be mapped to virtual MAC address 1000.0000.0002.

```
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 4000.1611.1234
```

DLSw+ with RIF Passthru Configuration Example

Figure 109 is a sample configuration for DLSw+ using the RIF Passthru feature.

Figure 109 Network Configuration with 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
  source-bridge spanning
```

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
```