Configuring LLC2 and SDLC Parameters

You do not need to configure Logical Link Control, type 2 (LLC2) Protocol because it is already enabled on Token Ring interfaces. This chapter describes how to modify the default settings of LLC2 parameters as needed.

To support the Synchronous Data Link Control (SDLC) protocol, you must configure the router to act as a primary or secondary SDLC station. You also can change default settings on any SDLC parameters. Configuration examples for both LLC2 and SDLC are given at the end of the chapter.

For a complete description of the LLC2 and SDLC commands mentioned in this chapter, refer to the "LLC2 and SDLC Commands" chapter in the *Cisco IOS Bridging and IBM Networking Command Reference* (Volume 1 of 2). To locate documentation of other commands that appear in this chapter, use the command reference master index or search online.

This chapter contains the following sections:

- LLC2 Configuration Task List, page 9
- Monitoring and Maintaining LLC2 Stations, page 13
- SDLC Configuration Task List, page 14
- Monitoring and Maintaining SDLC Stations, page 20
- LLC2 and SDLC Configuration Examples, page 21

To identify the hardware platform or software image information associated with a feature, use the Feature Navigator on Cisco.com to search for information about the feature or refer to the software release notes for a specific release.

Technology Overview

The LLC2 and SDLC protocols provide data link layer support for higher-layer network protocols and features such as SDLC Logical Link Control (SDLLC) and RSRB with local acknowledgment. The features that are affected by LLC2 parameter settings are listed in the "The Cisco Implementation of LLC2" section on page 2. The features that require SDLC configuration and use SDLC parameters are listed in the "The Cisco Implementation of SDLC" section on page 2.

LLC2 and SDLC package data in frames. LLC2 and SDLC stations require acknowledgments from receiving stations after a set amount of frames have been sent before sending further data. The tasks described in this chapter modify default settings regarding the control field of the data frames. By

Americas Headquarters:
Cisco Systems, Inc., 170 West Tasman Drive, San Jose, CA 95134-1706 USA

modifying the control field parameters, you can determine the number of acknowledgments sent for frames received and the level of polling used to determine available stations. In this manner, you can set the amount of resources used for frame checking and optimize the network load.

SDLC is used as the primary SNA link-layer protocol for WAN links. SDLC defines two types of network nodes: primary and secondary. Primary nodes poll secondary nodes in a predetermined order. Secondary nodes then send any outgoing data. When configured as primary and secondary nodes, our routers are established as SDLC stations.

The Cisco Implementation of LLC2

The Cisco LLC2 implementation supports the following features:

- Local acknowledgment for Remote Source-Route Bridging (RSRB)

  This feature is used in our implementation of RSRB as described in the chapter “Configuring Source-Route Bridging.”

  Because LANs are now connected through RSRB and WAN backbones, the delays that occur are longer than LLC2 allows for bidirectional communication between hosts. Our local acknowledgment feature addresses the problem of delays, resending data, and loss of user sessions.

- IBM LNM support

  Routers using 4- or 16-Mbps Token Ring interfaces configured for Source-Route Bridging (SRB) support Lan Network Manager (LNM) and provide all IBM bridge program functions. With LNM, a router appears as an IBM source-route bridge, and can manage or monitor any connected Token Ring interface.

  LNM support is described in the chapter “Configuring Source-Route Bridging.”

- SDLLC media translation

  The SDLLC feature provides media translation between the serial lines running SDLC and Token Rings running LLC2. SDLLC consolidates the IBM SNA networks running SDLC into a LAN-based, multiprotocol, multimedia backbone network.

  SDLLC is described in the chapter “Configuring IBM Network Media Translation.”

- ISO Connection-Mode Network Service (CMNS)

  Cisco’s CMNS implementation runs X.25 packets over LLC2 so that X.25 can be extended to Ethernet, Fiber Distributed Data Interface (FDDI), and Token Ring media.

The Cisco Implementation of SDLC

The Cisco SDLC implementation supports the following features:

- Frame Relay Access Support (FRAS)

  With FRAS, a router functions as a Frame Relay Access Device (FRAD) for SDLC, Token Ring, and Ethernet-attached devices over a Frame Relay Boundary Network Node (BNN) link.

  Frame Relay access support is described in the chapter “Configuring SNA Frame Relay Access Support.”

- SDLLC media translation

  The SDLLC feature provides media translation between the serial lines running SDLC and Token Rings running LLC2. SDLLC consolidates the IBM SNA networks running SDLC into a LAN-based, multiprotocol, multimedia backbone network.

  SDLLC is described in the chapter "Configuring IBM Network Media Translation."

- SDLC local acknowledgment

  SDLC local acknowledgment is used with SDLC STUN. TCP/IP must be enabled. With local acknowledgment, STUN SDLC connections can be terminated locally at the router, eliminating the need for acknowledgments to be sent across a WAN.

  SDLC local acknowledgment is described in the section "Establish the Frame Encapsulation Method" in the chapter "Configuring STUN and BSTUN."

IBM Network Media Translation

The Cisco IOS software includes the following media translation features that enable network communications across heterogeneous media:

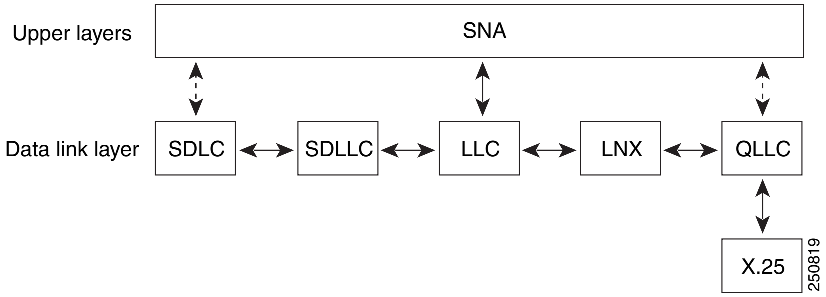
- SDLLC media translation enables a device on a Token Ring to communicate with a device on a serial link.
- QLLC conversion enables an IBM device to communicate with an X.25 network without having to install the X.25 software on local IBM equipment.

SDLLC is Cisco's proprietary software feature that enables a device on a Token Ring to communicate with a device on a serial link by translating between LLC2 and SDLC at the link layer.

SNA uses SDLC and LLC2 as link layer protocols to provide a reliable connection. The translation function between these industry-standard protocols takes place in the proprietary Cisco software.

Figure 1 illustrates how SDLLC provides data link layer support for SNA communication.

Figure 1 SNA Data Link Layer Support



SDLLC Media Translation Features

The SDLLC feature allows a PU 4, PU 2.1, or PU 2 to communicate with a PU 2 SDLC device as follows:

- SDLLC with direct connection—A 37x5 FEP on a Token Ring and the 3x74 cluster controller connected to a serial line are each connected to an interface on the same router configured with SDLLC.

- SDLLC with RSRB—A 37x5 FEP on a Token Ring and a 3x74 cluster controller connected to a serial line are connected to different routers. Only the device to which the 3x74 is connected is configured with SDLLC. The routers communicate via RSRB using direct encapsulation, RSRB over an FST connection, or RSRB over a TCP connection.
- SDLLC with RSRB and local acknowledgment—A 37x5 FEP on a Token Ring and a 3x74 cluster controller connected to a serial line are connected to different routers. Only the device to which the 3x74 is connected is configured with SDLLC. The routers communicate via RSRB over a TCP connection that has local acknowledgment enabled.

In all these topologies, each IBM end node (the FEP and cluster controller) has no indication that its counterpart is connected to a different medium running a different protocol. The 37x5 FEP responds as if the 3x74 cluster controller were communicating over a Token Ring, whereas the 3x74 responds as though the 37x5 FEP were communicating over a serial line. That is, the SDLLC software makes translation between the two media transparent to the end nodes.

Virtual Token Ring Concept

Central to Cisco's SDLLC feature is the concept of a virtual Token Ring device residing on a virtual Token Ring. Because the Token Ring device expects the node with which it is communicating also to be on a Token Ring, each SDLLC device on a serial line must be assigned an SDLLC virtual Token Ring address (SDLLC VTRA). Like real Token Ring addresses, SDLLC VTRAs must be unique across the network.

In addition to the SDLLC VTRA, an SDLLC virtual ring number must be assigned to each SDLLC device on a serial line. (The SDLLC virtual ring number differs from the virtual ring group numbers that are used to configure RSRB and multiport bridging.)

As part of its virtual telecommunications access method (VTAM) configuration, the IBM node on the Token Ring has knowledge of the SDLLC VTRA of the serial device with which it communicates. The SDLC VTRA and the SDLLC virtual ring number are a part of the SDLLC configuration for the router's serial interface. When the Token Ring host sends out explorer packets with the SDLLC VTRA as the destination address in the MAC headers, the router configured with that SDLLC VTRA intercepts the frame, fills in the SDLLC virtual ring number address and the bridge number in the RIF, then sends the response back to the Token Ring host. A route is then established between the Token Ring host and the router. After the Cisco IOS software performs the appropriate frame conversion, the system uses this route to forward frames to the serial device.

Resolving Differences in LLC2 and SDLC Frame Size

IBM nodes on Token Ring media normally use frame sizes greater than 1 KB, whereas the IBM nodes on serial lines normally limit frame sizes to 265 or 521 bytes. To reduce traffic on backbone networks and provide better performance, Token Ring nodes should send frames that are as large as possible. As part of the SDLLC configuration on the serial interface, the largest frame size the two media can support should be selected. The Cisco IOS software can fragment the frames it receives from the Token Ring device before forwarding them to the SDLC device, but it does not assemble the frames it receives from the serial device before forwarding them to the Token Ring device.

Maintaining a Dynamic RIF Cache

SDLLC maintains a dynamic RIF cache and caches the entire RIF; that is, the RIF from the source station to destination station. The cached entry is based on the best path at the time the session begins. SDLLC uses the RIF cache to maintain the LLC2 session between the router and the host FEP. SDLLC does not age these RIF entries. Instead, SDLLC places an entry in the RIF cache for a session when the session begins and flushes the cache when the session terminates. You cannot flush these RIFs because if you flush the RIF entries randomly, the Cisco IOS software cannot maintain the LLC2 session to the host FEP.

Other Considerations

The following are additional facts regarding SDLC and SDLLC:

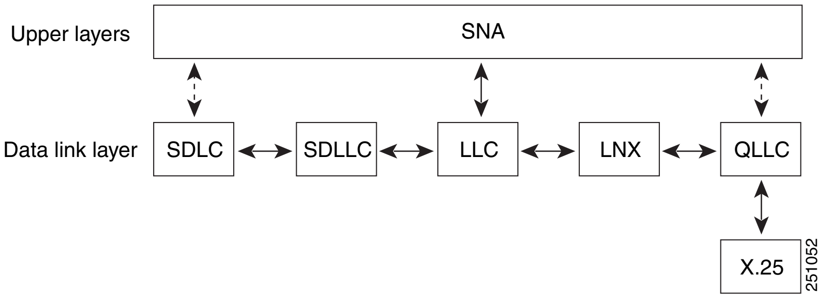
- As part of Cisco's SDLC implementation, only modulus 8 Normal Response Mode (NRM) sessions are maintained for the SDLC session.
- SDLC sessions are always locally acknowledged. LLC2 sessions can be optionally configured for local acknowledgment.
- SDLLC does not apply to SNA subarea networks, such as 37x5 FEP-to 37x5 FEP communication.
- Parameters such as the maximum number of information frames (I-frames) outstanding before acknowledgment, frequency of polls, and response time to poll frames can be modified per interface. If local acknowledgment is not enabled, these parameters are modified on the SDLC interface. If local acknowledgment is enabled, these parameters are modified on the Token Ring interface.
- Local acknowledgment only applies when the remote peer is defined for RSRB using IP encapsulation over a TCP connection. If no local acknowledgment is used, the remote peer can be defined for RSRB using direct encapsulation, RSRB using IP encapsulation over an Fast- Sequenced Transport (FST) connection, or RSRB using IP encapsulation over a TCP connection.

QLLC Conversion

Qualified Logical Link Control (QLLC) is a data link protocol defined by IBM that allows Systems Network Architecture (SNA) data to be transported across X.25 networks. (Although IBM has defined other protocols for transporting SNA traffic over an X.25 network, QLLC is the most widely used.)

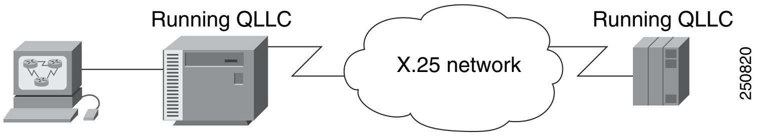
Figure 2 illustrates how QLLC conversion provides data link layer support for SNA communication.

Figure 2 *SNA Data Link Layer Support*



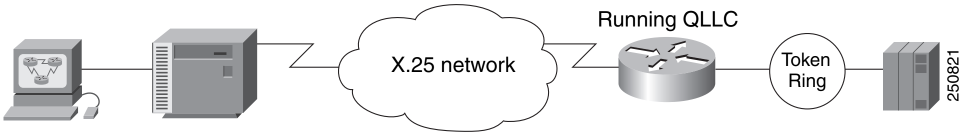
As shown in Figure 3, any devices in the SNA communication path that use X.25, whether end systems or intermediate systems, require a QLLC implementation.

Figure 3 *SNA Devices Running QLLC*



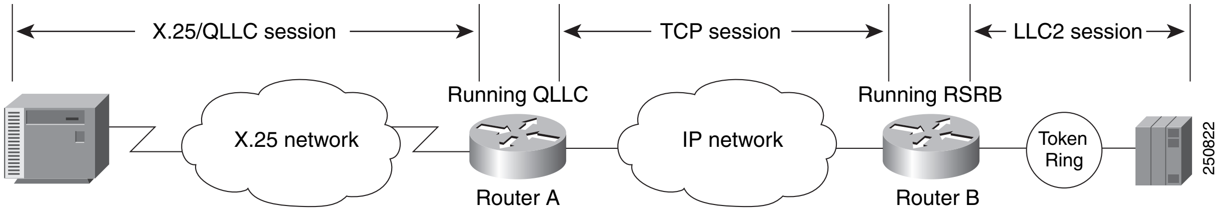
As shown in Figure 4, the QLLC conversion feature eliminates the need to install the X.25 software on local IBM equipment. A device attached locally to a Token Ring network can communicate through a router running the QLLC Conversion feature with a remote device attached to an X.25 network using QLLC. Typically, the locally attached device is an FEP, an AS 400, or a PS/2, and the remote device is a terminal controller or a PS/2. In this case, only the remote device needs an X.25 interface and the FEP can communicate with the terminal controller as if it were directly attached via a Token Ring network.

Figure 4 *Router Running QLLC Conversion Feature*



More elaborate configurations are possible. The router that implements QLLC conversion need not be on the same Token Ring network as the FEP. As shown in Figure 5, QLLC/LLC2 conversion is possible even when an intermediate IP WAN exists between the router connected to the X.25 network and the router connected to the Token Ring.

Figure 5 QLLC Conversion Running on a Router with an Intermediate IP Network

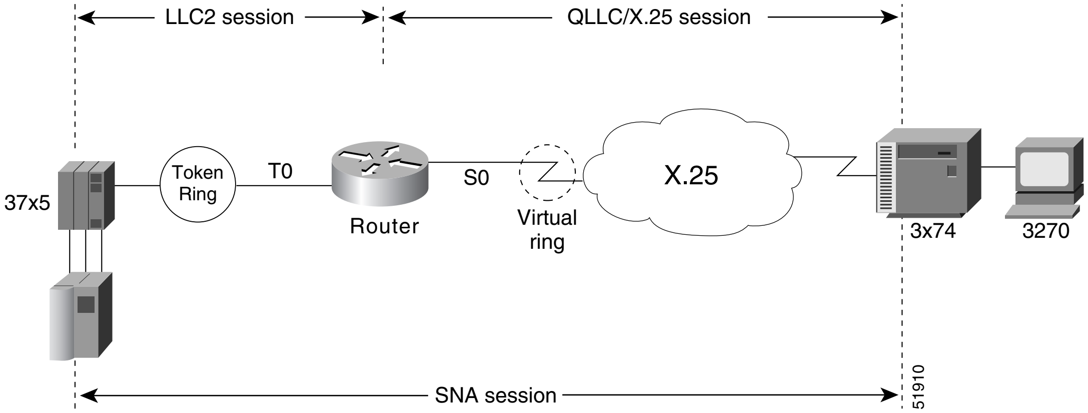


The Cisco Implementation of QLLC Conversion

SNA uses QLLC and X.25 as link layer protocols to provide a reliable connection. QLLC itself processes QLLC control packets. In a Token Ring environment, SNA uses LLC to provide a reliable connection. The LAN-to-X.25 (LNX) software provides a QLLC conversion function to translate between LLC and QLLC.

Figure 6 shows the simplest QLLC conversion topology: a single Token Ring device (for example, a 37x5 FEP) communicates with a single remote X.25 device (in this case a 3x74 cluster controller). In this example, a router connects the Token Ring network to the X.25 network.

Figure 6 QLLC Conversion Between a Single 37x5 and a Single 3x74

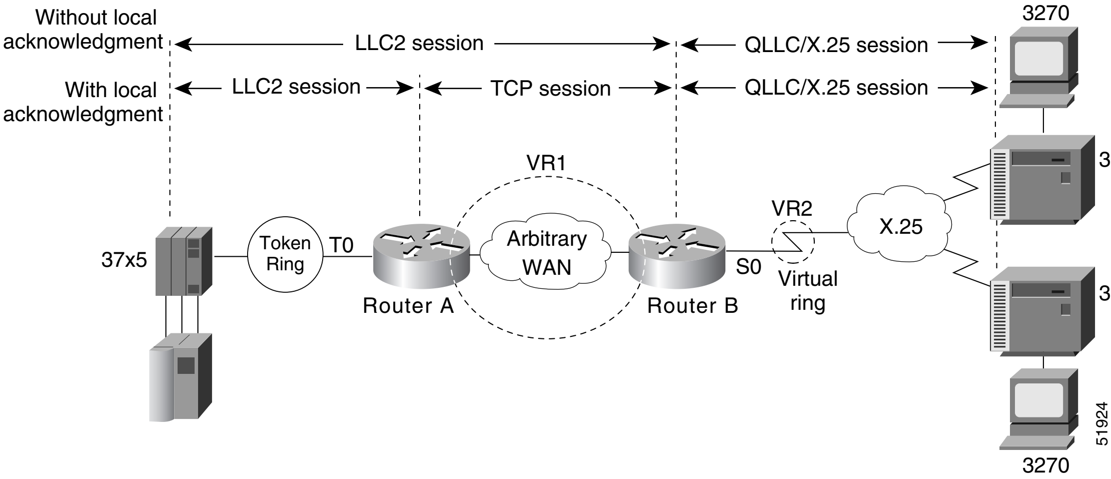


In Figure 6, each IBM end node has no indication that its counterpart is connected to a different medium running a different protocol. The 37x5 FEP responds as if the 3x74 cluster controller were communicating over a Token Ring, whereas the 3x74 responds as though the 37x5 FEP were communicating over an X.25 network. This is accomplished by configuring the router's X.25 interface as a virtual Token Ring, so that the X.25 virtual circuit appears to the Token Ring device (and to the router itself) as if it were a Token Ring to which the remote X.25 device is attached.

Also in this figure, the LLC2 connection extends from the 37x5 FEP across the Token Ring network to the router. The QLLC/X.25 session extends from the router across the X.25 network to the 3x74 cluster controller. Only the SNA session extends across the Token Ring and X.25 networks to provide an end-to-end connection from the 37x5 FEP to the 3x74 cluster controller.

As Figure 7 shows, a router need not directly connect the two IBM end nodes; instead, some type of backbone WAN can connect them. Here, RSRB transports packets between Router A and Router B, while Router B performs all conversion between the LLC2 and X.25 protocols. Only the router attached to the serial line (Router B) needs to be configured for QLLC conversion. Both Router A and Router B are configured for normal RSRB.

Figure 7 QLLC Conversion Between a Single 37x5 and Multiple 3x74s Across an Arbitrary WAN



How communication sessions are established over the communication link varies depending on whether or not LLC2 local acknowledgment has been configured on Router A’s Token Ring interface. In both cases, the SNA session extends end-to-end and the QLLC/X.25 session extends from Router B to the 3x74 cluster controller. If LLC2 local acknowledgment has not been configured, the LLC2 session extends from the 37x5 FEP across the Token Ring network and the arbitrary WAN to Router B. In contrast, when LLC2 local acknowledgment has been configured, the LLC2 session extends from the 37x5 FEP Router A, where it is locally terminated. A TCP session is then used across the arbitrary WAN to Router B.

Comparing QLLC Conversion to SDLLC

Although the procedures you use to configure QLLC are similar to those used to configure SDLLC, there are structural and philosophical differences between the point-to-point links that SDLC uses and the multiplexed virtual circuits that X.25 uses.

The most significant structural difference between QLLC conversion and SDLLC is the addressing. To allow a device to use LLC2 to transfer data, both SDLLC and QLLC provide virtual MAC addresses. In SDLLC, the actual MAC address is built by combining the defined virtual MAC (whose last byte is 0x00) with the secondary address used on the SDLC link; in this way, SDLLC supports multidrop. In QLLC conversion, multidrop is meaningless, so the virtual MAC address represents just one session and is defined as part of the X.25 configuration. Because one physical X.25 interface can support many simultaneous connections for many different remote devices, you only need one physical link to the X.25 network. The different connections on different virtual circuits all use the same physical link.

The most significant difference between QLLC conversion and SDLLC is the fact that a typical SDLC/SDLLC operation uses a leased line. In SDLC, dial-up connections are possible, but the maximum data rate is limited. In QLLC, both switched virtual circuits (SVCs) and permanent virtual circuits (PVCs) are available, but the favored use is SVC. While the router maintains a permanent connection to the X.25 network, a remote device can use each SVC for some bounded period of time and then relinquish it for use by another device. Using a PVC is very much like using a leased line.

Table 1 shows how the QLLC commands correspond to the SDLLC commands.

Table 1 QLLC and SDLLC Command Comparison

QLLC Command	Analogous SDLLC Command
qllc largest-packet	**sdllc ring-largest-frame**, **sdllc sdlc-largest-frame**
qllc partner	**sdllc partner**
qllc sap	**sdllc sap**
qllc srb, **x25 map qllc**, **x25 pvc qllc**	**sdllc traddr**
qllc xid	**sdllc xid**
source-bridge qllc-local-ack	**source-bridge sdllc-local-ack**

Other Implementation Considerations

Consider the following when implementing QLLC conversion:

- To use the QLLC conversion feature, a router must have a physical link to an X.25 public data network (PDN). It must also have an SRB/RSRB path to an IBM Front-End Processor (FEP). This link could be a Token Ring or Ethernet interface, or even FDDI, if RSRB is being used.
- QLLC conversion can run on any router with at least one serial interface configured for X.25 communication and at least one other interface configured for SRB or RSRB.
- QLLC conversion security depends upon access control in SRB/RSRB and X.25 and upon exchange identification (XID) validation.

You can configure DLSw+ for QLLC connectivity, which enables 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.
- Remote X.25-attached SNA devices can access an FEP or an AS/400 over a Token Ring or over SDLC.

For information on configuring DLSw+ for QLLC conversion, refer to the "Configuring DLSw+" chapter.

You can configure DSPUs for QLLC. For more information on this configuration, refer to the "Configuring DSPU and SNA Service Point Support" chapter.

LLC2 Configuration Task List

Because LLC2 is already enabled on a Token Ring, you do not need to enable it on the router. However, you can enhance LLC2 performance by completing the following tasks:

- Controlling Transmission of I-Frames, page 10
- Establishing the Polling Level, page 12
- Setting Up XID Transmissions, page 13

See the "LLC2 and SDLC Configuration Examples" section on page 21 for examples.

Controlling Transmission of I-Frames

Control the number of information frames (I-frames) and acknowledgments sent on the LLC2 network by completing the tasks described in the following sections:

- Setting the Maximum Number of I-Frames Received Before Sending an Acknowledgment, page 10
- Setting the Maximum Delay for Acknowledgments, page 10
- Setting the Maximum Number of I-Frames Sent Before Requiring Acknowledgment, page 10
- Setting the Number of Retries Allowed, page 11
- Setting the Time for Resending I-Frames, page 11
- Setting the Time for Resending Rejected Frames, page 11

Setting the Maximum Number of I-Frames Received Before Sending an Acknowledgment

You can reduce overhead on the network by increasing the maximum number of frames the Cisco IOS software can receive at once before it must send the sender an acknowledgment. To do so, use the following command in interface configuration mode:

Command	Purpose
`Router(config-if)# llc2 ack-max packet-count`	Sets maximum number of I-frames the router can receive before it sends an acknowledgment.

Setting the Maximum Delay for Acknowledgments

You can ensure timely receipt of acknowledgments so that sending data is not delayed. Even if the maximum amount of frames has not been reached, you can set a timer forcing the router to send an acknowledgment and reset the maximum amount counter to 0.

To set the maximum delay time, use the following command in interface configuration mode:

Command	Purpose
`Router(config-if)# llc2 ack-delay-time milliseconds`	Sets the I-frame acknowledgment time.

Setting the Maximum Number of I-Frames Sent Before Requiring Acknowledgment

You can set the maximum number of I-frames that the router sends to an LLC2 station before the software requires an acknowledgment from the receiving end. A higher value reduces overhead on the network. Ensure that the receiving LLC2 station can handle the number of frames set by this value.

To set this value, use the following command in interface configuration mode:

Command	Purpose
`Router(config-if)# llc2 local-window packet-count`	Sets the maximum number of I-frames the router sends before it requires an acknowledgment.

Setting the Number of Retries Allowed

You can set the number of times the router will re-send a frame when the receiving station does not acknowledge the frame. Once this value is reached, the session is dropped. This value also is used to determine how often the software will retry polling a busy station. Use this command in conjunction with the **llc2 t1-time** command described in the "Setting the Time for Resending I-Frames" section on page 11. Using them together ensures that the sending of frames is monitored at a reasonable level, while limiting the number of unsuccessful repeated tries.

To set the number of retries, use the following command in interface configuration mode:

Command	Purpose
`Router(config-if)# llc2 n2 retry-count`	Establishes the number of times the router will re-send unacknowledged frames or try polling a busy station.

Setting the Time for Resending I-Frames

You can set the amount of time the router waits before resending unacknowledged I-frames. This interval is called the *T1 time*. Use this command in conjunction with setting the number of retries and setting the transit poll-frame timer. Using these commands in conjunction with each other provides a balance of network monitoring and performance.

To set the T1 time, use the following command in interface configuration mode:

Command	Purpose
`Router(config-if)# llc2 t1-time milliseconds`	Controls how long the router waits for an acknowledgment of transmitted I-frames.

Note Ensure that you allow enough time for the round trip between the router and its LLC2-speaking stations. Under heavy network loading conditions, resending I-frames every 3000 ms is appropriate.

Setting the Time for Resending Rejected Frames

You can set the amount of time that the router will wait for an expected frame before sending a reject command (REJ). Typically, when an LLC2 station sends an I-frame, a sequence number is included in the frame. The LLC2 station that receives these frames will expect to receive them in order. If it does not, it can reject a frame and indicate which frame it is expecting to receive instead. If the correct frame is not sent to the software before the reject timer expires, the software sends a REJ to the remote station and disconnects the LLC2 session.

To set the reject timer, use the following command in interface configuration mode:

Command	Purpose
`Router(config-if)# llc2 trej-time milliseconds`	Sets the time the Cisco IOS software waits for a resend of a rejected frame before sending a reject command to the remote station.

Establishing the Polling Level

You can control the amount of polling that occurs on the LLC2 network by completing the tasks described in the following sections:

- Setting the Polling Frequency, page 12
- Setting the Polling Interval, page 12
- Setting the Transmit-Poll-Frame Timer, page 12

Setting the Polling Frequency

You can set the optimum interval of time after which the router sends Receiver Ready messages or frames that tell other LLC2 stations that the router is available. These polls occur during periods of idle time on the network.

To set polling frequency, use the following command in interface configuration mode:

Command	Purpose
`Router(config-if)# llc2 idle-time milliseconds`	Controls the polling frequency during idle traffic.

Setting the Polling Interval

The amount of time the router waits until repolling a busy station can also be set. Use this command in conjunction with setting the number of retries. Typically, you do not need to use this command unless an LLC2 station has unusually long busy periods before clearing the busy state. In this case, you should increase the value so that the station does not time out.

To set the polling interval, use the following command in interface configuration mode:

Command	Purpose
`Router(config-if)# llc2 tbusy-time milliseconds`	Sets the amount of time the router will wait before repolling a busy station.

Setting the Transmit-Poll-Frame Timer

When the router sends a command that must receive a response, a poll bit is sent in the frame. When the software sends the poll bit, it cannot send any other frame with the poll bit set until the receiver replies to that poll frame with a frame containing a final bit set. When the timer expires, the software assumes that it can send another frame with a poll bit.

Set the transmit-poll-frame timer to reduce problems with receiving stations that are faulty and cannot send the frame with the final bit set by using the following command in interface configuration mode:

Command	Purpose
`Router(config-if)# llc2 tpf-time milliseconds`	Sets the amount of time the router waits for a final response to a poll frame before the resending it.

This value should be larger than the T1 time. The T1 time determines how long the software waits for receipt of an acknowledgment before sending the next set of frames. See the "Setting the Time for Resending I-Frames" section on page 11 for more information.

Setting Up XID Transmissions

You can control the number of frames used for identification on the LLC2 network by completing the tasks described in the following sections:

- Setting the Frequency of XID Transmissions, page 13
- Setting the Time for XID Retries, page 13

Setting the Frequency of XID Transmissions

XID frames identify LLC2 stations at a higher level than the MAC address and contain information about the configuration of the stations.You can set how often the router sends an XID frame by using the following command in interface configuration mode:

Command	Purpose
Router(config-if)# **llc2 xid-neg-val-time** *milliseconds*	Sets the frequency of XID transmissions.

⚠ **Caution** Do not change the value unless requested by your technical support representative.

Setting the Time for XID Retries

You can set the amount of time the router waits for a reply to the XID frames it sends to remote stations. The value should be larger than the T1 time, which indicates how long the software waits for an acknowledgment before dropping the session.

To set the time for XID retries, use the following command in interface configuration mode:

Command	Purpose
Router(config-if)# **llc2 xid-retry-time** *milliseconds*	Sets how long the router waits for a reply to the XID frames it sends to remote stations.

Monitoring and Maintaining LLC2 Stations

You can display the configuration of LLC2 stations to determine which LLC2 parameters need adjustment. Use the following command in privileged EXEC mode:

Command	Purpose
Router# **show llc2**	Displays the configuration of LLC2 stations.

SDLC Configuration Task List

The SDLC tasks described in this section configure the router as an SDLC station. (This is in contrast to a router configured for SDLC Transport, where the device is not an SDLC station, but passes SDLC frames between two SDLC stations across a mixed-media, multiprotocol environment.) The first task is required; you accomplish it with the appropriate set of commands for your network needs. The remaining tasks are optional: you can perform them as necessary to enhance SDLC performance.

- Enabling the Router as a Primary or a Secondary SDLC Station, page 14
- Enabling SDLC Two-Way Simultaneous Mode, page 16
- Determining the Use of Frame Rejects, page 17
- Setting SDLC Timer and Retry Counts, page 17
- Setting SDLC Frame and Window Sizes, page 18
- Controlling the Buffer Size, page 18
- Controlling Polling of Secondary Stations, page 18
- Configuring an SDLC Interface for Half-Duplex Mode, page 19
- Specifying the XID Value, page 20
- Specifying the SAPs, page 20
- Setting the Largest SDLC I-Frame Size, page 20

See the "LLC2 and SDLC Configuration Examples" section on page 21 for examples.

Enabling the Router as a Primary or a Secondary SDLC Station

SDLC defines two types of network nodes: primary and secondary. Primary nodes poll secondary nodes in a predetermined order. Secondaries then send if they have outgoing data. When configured as primary and secondary nodes, our devices are established as SDLC stations.

Depending on your particular network needs, perform the tasks in one of the following sections to enable the router as an SDLC station:

- Establishing an SDLC Station for Frame Relay Access Support, page 14
- Establishing an SDLC Station for DLSw+ Support, page 15
- Establishing an SDLC Station for SDLLC Media Translation, page 16

Establishing an SDLC Station for Frame Relay Access Support

You can establish the router to be any of the following:

- Primary SDLC station
- Secondary SDLC station
- Either primary or secondary, depending on the role of the end stations or on XID negotiations
- Primary Node Type 2.1 (NT2.1) node

To establish devices as SDLC stations when you plan to configure Frame Relay access support, use the following commands in interface configuration mode:

	Command	Purpose
Step 1	Router(config-if)# **encapsulation sdlc**[1]	Sets the encapsulation type of the serial interface to SDLC.
Step 2	Router(config-if)# **sdlc role** {**none** | **primary** | **secondary** | **prim-xid-poll**}	Establishes the role of the interface.

[1] For information on the **nrzi-encoding** interface configuration command, refer to the *Cisco IOS Configuration Fundamentals Configuration Guide*.

If the interface does not play a role, the router can be either primary or secondary, depending on the end stations. The SDLC end station must be configured as negotiable or primary NT2.1. When the end stations are configured as physical unit (PU) type 2, you can set the role of the interface to primary or secondary. When the end station is configured as secondary NT2.1, you must set the role of the interface to poll the primary XID.

Note Currently, Frame Relay access support does not support the secondary role.

Establishing an SDLC Station for DLSw+ Support

To establish devices as SDLC stations when you plan to configure our DLSw+ feature, use the following commands in interface configuration mode:

	Command	Purpose
Step 1	Router(config-if)# **encapsulation sdlc**	Sets the encapsulation type of the serial interface to SDLC.
Step 2	Router(config-if)# **sdlc role** {**none** | **primary** | **secondary** | **prim-xid-poll**}	Establishes the role of the interface.
Step 3	Router(config-if)# **sdlc vmac** *mac-address*	Configures a MAC address for the serial interface.
Step 4	Router(config-if)# **sdlc partner** *mac-address sdlc-address* {**inbound** | **outbound**}	Specifies the destination address with which an LLC session is established for the SDLC station.
Step 5	Router(config-if)# **sdlc dlsw** {*sdlc-address* | **default** | **partner** *mac-address* [**inbound** | **outbound**]}	Attaches SDLC addresses to DLSw+.

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. Use the **sdlc role prim-xid-poll** command if all devices are type PU 2.1.

For additional DLSw+ configuration commands, refer to the "Configuring DLSw+" chapter in this publication.

Establishing an SDLC Station for SDLLC Media Translation

To establish devices as SDLC stations when you plan to configure our SDLLC media translation feature, use the commands in the order listed in the following table. One serial interface can have two or more secondary stations attached to it through a modem sharing device. Each secondary station address must be assigned to the primary station. You must use the following commands in interface configuration mode for the serial interface:

	Command	Purpose
Step 1	`Router(config-if)# encapsulation sdlc-primary`	Establishes a router as the primary SDLC station on the serial line.
Step 2	`Router(config-if)# encapsulation sdlc-secondary`	Establishes other routers as secondary SDLC stations.
Step 3	`Router(config-if)# sdlc address hexbyte [echo]`	Assigns secondary stations to a primary station.

Use the **show interfaces** command to list the configuration of the SDLC serial lines. Use the **no sdlc address** command to remove a secondary address assignment. Addresses are hexadecimal (base 16).

Enabling SDLC Two-Way Simultaneous Mode

SDLC two-way simultaneous mode allows SDLC link stations to a full-duplex serial line efficiently. With a two-way simultaneous mode, the primary link station can send data to a secondary link station while there is an outstanding poll.

For a primary link station, SDLC two-way simultaneous mode operates in either a multidrop link environment or point-to-point link environment.

In a multidrop link environment, a two-way simultaneous primary station is able to poll a secondary station, receive data from the station, and send data (I-frames) to other secondary stations by using the **sdlc simultaneous half-datamode** command.

In a point-to-point link environment, a two-way simultaneous primary station can send data (I-frames) to a secondary station, although there is an outstanding poll, as long as the window limit is not reached by using the **sdlc simultaneous full-datamode** command.

For a secondary link station, the SDLC two-way simultaneous mode operates only in a point-to-point link environment and allows data (I-frames) to be received after a poll frame has already been received by using the **sdlc simultaneous full-datamode** command.

To enable a two-way simultaneous mode, use one of the following commands in interface configuration mode, as needed:

Command	Purpose
Router(config-if)# **sdlc simultaneous full-datamode**	Enables the primary station in a point-to-point link environment to send data to and receive data from the polled secondary station.
or	Enables the secondary station in a point-to-point link environment to receive data from the primary station after it has already been polled.
Router(config-if)# **sdlc simultaneous half-datamode**	Enables the primary station in a multidrop link environment to send data to other secondary link stations while receiving data from the polled secondary link station.

Determining the Use of Frame Rejects

You can specify that a secondary station does not send frame reject messages, or reject commands indicating frame errors. If you do so, the router drops an SDLC connection if the system receives an error from the secondary station.

To determine handling of frame rejects, use the following command in interface configuration mode:

Command	Purpose
Router(config-if)# **sdlc frmr-disable**	Specifies that this secondary station does not support frame rejects.

To specify that the secondary station does support frame rejects, use the **no sdlc frmr-disable** command.

Setting SDLC Timer and Retry Counts

When an SDLC station sends a frame, it waits for an acknowledgment from the receiver indicating that this frame has been received. You can modify the time the router allows for an acknowledgment before resending the frame. You can also determine the number of times that a software re-sends a frame before terminating the SDLC session. By controlling these values, you can reduce network overhead while continuing to check sending of frames.

Use the SNRM timer only if you want to have a unique timeout period to wait for a reply to a SNRM. To specify a SNRM timer that is different from the T1 response time, set the SDLC SNRM timer using the **sdlc snrm-timer** command in interface configuration mode:

Command	Purpose
Router(config-if)# **sdlc t1** *milliseconds*	Controls the amount of time the Cisco IOS software waits for a reply. Default value is 3000 ms.
Router(config-if)# **sdlc n2** *retry-count*	Determines the number of times that the Cisco IOS software resends a frame before terminating the SDLC session.
Router(config-if)# **sdlc snrm-timer** *number*	Specifies a SNRM timer that is different from the T1 response time.

Setting SDLC Frame and Window Sizes

You can set the maximum size of an incoming frame and set the maximum number of I-frames (or window size) the router will receive before sending an acknowledgment to the sender. By using higher values, you can reduce network overhead.

To set SDLC frame and window sizes, use one of the following commands in interface configuration mode, as needed:

Command	Purpose
`Router(config-if)# sdlc n1 bit-count`	Sets the maximum size of an incoming frame.
`Router(config-if)# sdlc k window-size`	Sets the local window size of the router.
`Router(config-if)# sdlc poll-limit-value count`	Controls how many times a single secondary station can be polled for input before the next station must be polled.
`Router(config-if)# sdlc address hexbyte [echo] [ack-mode] [xid-poll] [switched] [seconly] [xid-passthru] [passive] [K num]`	Specifies the address used on the SDLC line, and any other unique options on how the address is treated. **Note** The **ack-mode** option supports applications that require local termination of an SDLC connection with address ff. This option is available only if the hexbyte parameter is configured with a value of ff. You should use this option only if you use the SDLC address ff as a regular (not a broadcast) address.

Controlling the Buffer Size

You can control the buffer size on the router. The buffer holds data that is waiting to be sent to a remote SDLC station. This command is particularly useful in the case of the SDLLC media translator, which allows an LLC2-speaking SNA station on a Token Ring to communicate with an SDLC-speaking SNA station on a serial link. The frame sizes and window sizes on Token Rings are often much larger than those acceptable for serial links, and serial links are often slower than Token Rings.

To control backlogs that can occur during periods of high data transfer from the Token Ring to the serial line, use the following command in interface configuration mode on a per-address basis:

Command	Purpose
`Router(config-if)# sdlc holdq address queue-size`	Sets the maximum number of packets held in queue before transmitting.

Controlling Polling of Secondary Stations

You can control the intervals at which the router polls secondary stations, the length of time a primary station can send data to a secondary station, and how often the software polls one secondary station before moving on to the next station.

Keep the following points in mind when using these commands:

- Secondary stations cannot send data until they are polled by a primary station. Increasing the poll-pause timer increases the response time of the secondary stations. Decreasing the timer can flood the serial link with unneeded polls, requiring secondary stations to spend wasted CPU time processing them.
- Increasing the value of the poll limit allows for smoother transactions between a primary station and a single secondary station, but can delay polling of other secondary stations.

To control polling of secondary stations, use one of the following commands in interface configuration mode, as needed:

Command	Purpose
`Router(config-if)# sdlc poll-pause-timer milliseconds`	Controls how long the Cisco IOS software pauses between sending each poll frame to secondary stations on a single serial interface.
`Router(config-if)# sdlc poll-limit-value count`	Controls how many times a single secondary station can be polled for input before the next station must be polled.

To retrieve default polling values for these operations, use the **no** forms of these commands.

Configuring an SDLC Interface for Half-Duplex Mode

By default, SDLC interfaces operate in full-duplex mode. To configure an SDLC interface for half-duplex mode, use the following command in interface configuration mode:

Command	Purpose
`Router(config-if)# half-duplex`	Configures an SDLC interface for half-duplex mode.

On an interface that is in half-duplex mode and that has been configured for DCE, you can adjust the delay between the detection of a Request To Send (RTS) signal and the assertion of the Clear To Send (CTS) signal. To do so, use the following command in interface configuration mode:

Command	Purpose
`Router(config-if)# half-duplex timer cts-delay value`	Delays the assertion of a CTS.

On an interface that is in half-duplex mode and that has been configured for DTE, you can adjust the time the interface waits for the DCE to assert CTS before dropping an RTS. To do so, use the following command in interface configuration mode:

Command	Purpose
`Router(config-if)# half-duplex timer rts-timeout value`	Adjusts the amount of time before interface drops an RTS.

Specifying the XID Value

The exchange of identification (XID) value you define on the router must match that of the IDBLK and IDNUM system generation parameters defined in VTAM on the Token Ring host to which the SDLC device will be communicating. To specify the XID value, use the following command in interface configuration mode:

Command	Purpose
`Router(config-if)#` **`sdlc xid`** *`address xid`*	Specifies the XID value to be associated with the SDLC station.

Specifying the SAPs

SAPs are used by the CMCC adapter to establish communication with VTAM on the mainframe and to identify Logical Link Control (LLC) sessions on a CMCC's internal adapter. To configure SAPs in SDLC, use the following command in interface configuration mode:

Command	Purpose
`Router(config-if)#` **`sdlc saps`** *`address ssap dsap`*	Configures SDLC-to-LLC sessions with respect to the SSAP and DSAP on the LLC.

Setting the Largest SDLC I-Frame Size

Generally, the router and the SDLC device with which it communicates should support the same maximum SDLC I-frame size. The larger this value, the more efficient the line usage, thus increasing performance.

After the SDLC device has been configured to send the largest possible I-frame, you must configure the router to support the same maximum I-frame size. The default is 265 bytes. The maximum value the software can support must be less than the value of the LLC2 largest frame value defined when setting the largest LLC2 I-frame size.

To set the largest SDLC I-frame size, use the following command in interface configuration mode:

Command	Purpose
`Router(config-if)#` **`sdlc sdlc-largest-frame`** *`address size`*	Sets the largest I-frame size that can be sent or received by the designated SDLC station.

Monitoring and Maintaining SDLC Stations

To monitor the configuration of SDLC stations to determine which SDLC parameters need adjustment, use the following command in privileged EXEC mode:

Command	Purpose
`Router#` **`show interfaces serial`**	Displays SDLC station configuration information.

You determine the status of end stations by sending an SDLC test frame to a physical unit via its SDLC address and router interface. You can either send out the default information string or a predefined one. You can send a preset number of test frames a continuous stream that can later be halted. The **sdlc test serial** command pre-check for correct interface and SDLC address of the end station. You can view the results of the test frames after the frames have been sent or a SDLC test frame stop has been executed.

To send an SDLC test frame, use the following command in privileged EXEC mode:

Command	Purpose
`Router# sdlc test serial number address [iterations	continuous	stop	string string]`	Sends an SDLC test frame.

Note Only a device configured as primary is allowed to send test frames.

LLC2 and SDLC Configuration Examples

The following sections provide LLC2 and SDLC configuration examples:

- LLC2 Configuration Example, page 21
- SDLC Two-Way Simultaneous Mode Configuration Example, page 22
- SDLC Encapsulation for Frame Relay Access Support Configuration Examples, page 22
- SDLC Configuration for DLSw+ Example, page 23
- Half-Duplex Configuration Example, page 23
- SDLC-to-LLC2 FID4 Frame Conversion Examples, page 23

LLC2 Configuration Example

You can configure the number of LLC2 frames received before an acknowledgment. For this example, assume that at time 0, two I-frames are received. The maximum amount of three has not been reached, so no acknowledgment for these frames is sent. If a third frame, which would force the router to send an acknowledgment, is not received within 800 ms, an acknowledgment is sent anyway, because the delay timer alarm is activated.

```
interface tokenring 0
 llc2 ack-max 3
 llc2 ack-delay-time 800
```

At this point, because all frames are acknowledged, the counter for the maximum amount of I-frames will be reset to zero.

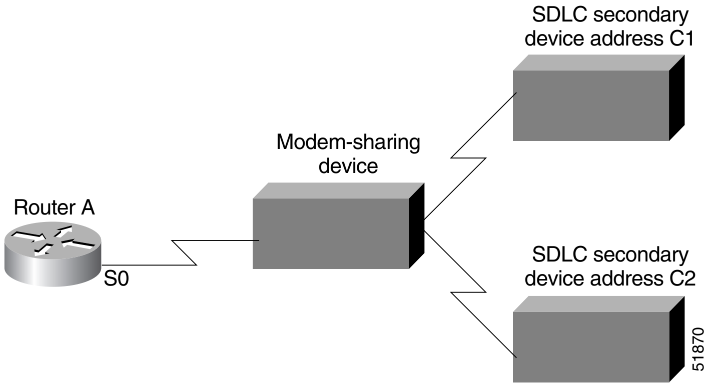
SDLC Two-Way Simultaneous Mode Configuration Example

The following configuration defines serial interface 0 as the primary SDLC station with two SDLC secondary stations, C1 and C2, attached to it through a modem-sharing device. Two-way simultaneous mode is enabled.

```
interface serial 0
 encapsulation sdlc-primary
 sdlc address c1
 sdlc address c2
 sdlc simultaneous half-datamode
```

The network for this configuration is shown in Figure 8.

Figure 8 Two SDLC Secondary Stations Attached to a Single Serial Interface Through a Modem-Sharing Device


SDLC secondary device address C1
Modem-sharing device
Router A
S0
SDLC secondary device address C2
51870

SDLC Encapsulation for Frame Relay Access Support Configuration Examples

The following examples describe possible SDLC encapsulation configurations if you plan to configure Frame Relay access support.

The following configuration is appropriate if the SDLC station is a negotiable or primary Node Type 2.1 station:

```
interface serial 2/6
 no ip address
 encapsulation sdlc
 clockrate 9600
 fras map sdlc C1 serial 2/0 frame-relay 32 4 4
 sdlc address C1
```

The following configuration is appropriate if the SDLC station is a secondary Node Type 2.1 station:

```
interface serial 2/6
 no ip address
 encapsulation sdlc
 clockrate 9600
 fras map sdlc C1 serial 2/0 frame-relay 32 4 4
 sdlc role prim-xid-poll
 sdlc address C1
```

The following configuration is appropriate if the SDLC station is a secondary PU 2 station:

```
interface serial 2/6
 no ip address
 encapsulation sdlc
 clockrate 9600
 fras map sdlc C1 serial 2/0 frame-relay 32 4 4
 sdlc role primary
 sdlc address C1
 sdlc xid C1 01700001
```

SDLC Configuration for DLSw+ Example

The following example describes the SDLC configuration with DLSw+ support implemented. In this example, 4000.3745.001 is the MAC address of the host. The router serves as the primary station, while the remote secondary stations, C1, C2, and C3, are reserved for DLSw+ and cannot be used by any other data-link user. The SNRM timer is configured with a value of 2500 ms.

If the **k** parameter is not specified on the **sdlc address** command, the value will be the setting of the **sdlc k** parameter, which is specified as 1; thus C1 and C2 will use **k** value of 1, but the C3 station will have more bandwidth because it has a specified **k** value of 7.

```
interface serial 0
encapsulation sdlc
sdlc role primary
sdlc vmac 4000.3174.0000
sdlc k 1
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 addr c3 k 7
sdlc xid c3 01754321
sdlc partner 4000.3745.0001 c3
sdlc snrm-timer 2500
sdlc dlsw c1 c2 c3
```

Note If the **no** form of this command is specified, the value of the t1 timer will be used for the SNRM timer.

Half-Duplex Configuration Example

In the following example, an SDLC interface has been configured for half-duplex mode:

```
encapsulation sdlc-primary
 half-duplex
```

SDLC-to-LLC2 FID4 Frame Conversion Examples

The following sample configurations demonstrate SDLC-to-LLC2 conversions for FID4 frames. When you implement these conversion, keep the following considerations in mind:

- If NCP is the primary, the first PU 4 line uses SDLC address 0x01, the second uses 0x02, and so on.

- The SDLC address is used to modify the last byte of the SDLC virtual MAC address (**sdlc vmac**). This modified value is coded in the XCA subarea major node.
- Specify the **echo** option in the **sdlc address** command. With the **echo** option specified, the primary polls with an address in the range 01 to 7E, and the secondary replies with the first bit set to 1. For example, if the primary polls with 04 (0000 0100), the secondary replies with 84 (1000 0100).
- Set **mtu** slightly larger than the maximum packet size used by NCP. Set **sdlc N1** equal to (**mtu** + 2) * 8, which is **mtu**, plus 2 bytes for the SDLC header, times 8 (because N1 is coded in bits, not bytes).
- If the router is providing a clock for the FEP, specify a **clockrate**.
- If the SDLC line has **NRZI=YES**, specify **nrzi-encoding**.
- Ensure that the SDLC- attached FEP is the SDLC primary device, using one of the following methods:
  - Ensure that the SDLC FEP has a higher subarea than the Token Ring-attached FEP (or Token Ring-attached host).
  - Do not configure a secondary SDLCST entry on the GROUP statement for the SDLC line:

    ```
    SDLCPRIM SDLCST GROUP=xxxx
    SDLCSEC  SDLCST GROUP=yyyy

    GROUP SDLCST=(SDLCPRIM,,)
    NAME1 LINE ADDR=nnn
    NAME2 PU PUTYPE=4
    ```

- The SDLC connection requires modulo 8. Ensure that the SDLC group/line and the SDLCST groups are configured with **modulo = 8** and **maxout = 7**.

DLSW Remote Peer Connection Configuration Example

The following sample configurations are for a DLSW remote peer connection using two routers. Two different sample configurations are given for the remote DLSW peer:

- Connected to a CIP-attached router
- Connected to a Token Ring-attached subarea, such as NTRI FEP

Configuration for SDLC-Attached Router

The following configuration statements are for the SDLC-attached router:

```
dlsw local-peer peer-id 10.2.2.2
dlsw remote-peer 0 tcp 10.1.1.1
interface Serial1
 description sdlc configuration PU4/PU4
 mtu 6000
 no ip address
 encapsulation sdlc
 no keepalive
 nrzi-encoding
 clockrate 9600
 sdlc vmac 4000.3745.0000
 sdlc N1 48016
 sdlc address 04 echo
 sdlc partner 4000.1111.0020 04
 sdlc dlsw 4
```

Configuration for Remote DLSW Peer Connected to a CIP-Attached Router

The following configuration statements are for a remote DLSW peer connected to a CIP-attached router:

```
source-bridge ring-group 1111
dlsw local-peer peer-id 10.1.1.1
dlsw remote-peer 0 tcp 10.2.2.2
interface Channel5/0
 csna 0100 20
interface Channel5/2
 lan TokenRing 0
  source-bridge 1 1 1111
  adapter 0 4000.1111.0020
```

Configuration for Remote DLSW Peer Connected to a Token Ring-Attached Subarea

The following configuration statements are for a remote DLSW peer connected to a Token Ring-attached subarea, such as NTRI FEP:

```
source-bridge ring-group 1111
dlsw local-peer peer-id 10.1.1.1
dlsw remote-peer 0 tcp 10.2.2.2
interface token ring 6/0
 ring-speed 16
 source-bridge 2 1 1111
```

DLSW Local-Switching Connection Configuration Example

The following sample configurations are for a DLSW local-switching connection, using one router. Two different sample configurations are given:

- Connection to a CIP-attached router
- Connection to a Token Ring-attached subarea, such as NTRI FEP

Configuration for a Connection to a CIP-Attached Router

The following configuration statements are for a connection to a CIP-attached router:

```
source-bridge ring-group 1111
dlsw local-peer
interface Serial1/0
 description sdlc configuration PU4/PU4
 mtu 6000
 no ip address
 encapsulation sdlc
 no keepalive
 nrzi-encoding
 clockrate 9600
 sdlc vmac 4000.3745.0000
 sdlc N1 48016
 sdlc address 04 echo
 sdlc partner 4000.1111.0020 04
 sdlc dlsw 4
interface Channel5/0
 csna 0100 20
interface Channel5/2
 lan TokenRing 0
  source-bridge 1 1 1111
  adapter 0 4000.1111.0020
```

Configuration for a Connection to a Token Ring-Attached Subarea

The following configuration statements are for a connection to a Token Ring-attached subarea, such as NTRI FEP:

```
source-bridge ring-group 1111
dlsw local-peer
interface Serial1/0
 description sdlc configuration PU4/PU4
 mtu 6000
 no ip address
 encapsulation sdlc
 no keepalive
 nrzi-encoding
 clockrate 9600
 sdlc vmac 4000.3745.0000
 sdlc N1 48016
 sdlc address 04 echo
 sdlc partner 4000.1111.0020 04
 sdlc dlsw 4
interface token ring 6/0
 ring-speed 16
 source-bridge 2 1 1111
```

SDLC FEP Configuration

The following configuration statements are for the SDLC FEP:

```
00084 *******************************************************************"
00085 SDLCPRIM SDLCST GROUP=INNPRIM,         SDLC STATEMENTS FOR INN         *
00086                MAXOUT=7,                                               *
00087                MODE=PRIMARY,                                           *
00088                PASSLIM=254,                                            *
00089                RETRIES=(5,2,5),                                        *
00090                SERVLIM=4
00091 SDLCSEC  SDLCST GROUP=INNSEC,          SDLC STATEMENTS FOR INN         *
00092                MAXOUT=7,                                               *
00093                MODE=SECONDARY,                                         *
00094                PASSLIM=254,                                            *
00095                RETRIES=(5,2,5)
00286 *******************************************************************"
00287 *                                                                 *"
00288 *           GROUP MACROS FOR INN CONNECTIONS                      *"
00289 *                                                                 *"
00290 *******************************************************************"
00291 GRPINN   GROUP ACTIVTO=60,             SEC WAIT FOR PRIM               *
00292                ANS=CONT,                                               *
00293                CLOCKNG=EXT,                                            *
00294                DATRATE=HIGH,                                           *
00295                DIAL=NO,                                                *
00296                DUPLEX=FULL,                                            *
00297                IRETRY=NO,                                              *
00298                ISTATUS=ACTIVE,                                         *
00299                LNCTL=SDLC,                                             *
00300                MAXOUT=7,                                               *
00301                MAXPU=1,                                                *
00302                MONLINK=YES,                                            *
00303                NEWSYNC=NO,                                             *
00304                NRZI=NO,                                                *
00305                PASSLIM=254,                                            *
00306                PAUSE=0.2,                                              *
00307                REPLYTO=1,                                              *
00308                RETRIES=(3,1,3),                                        *
```

```
00309                SDLCST=(SDLCPRIM,SDLCSEC),                              *
00310                SERVLIM=255,                                            *
00311                TGN=2,                                                  *
00312                TRANSFR=27,                                             *
00313                TYPE=NCP
00314 *"
00315 ERNLN012 LINE ADDRESS=012,ISTATUS=ACTIVE
00316 ERNPU012 PU PUTYPE=4
00317 *"
```

Token Ring FEP Subarea Configuration

The following configuration statements are for the Token Ring FEP subarea:

```
********************************************************                06260099
* SDLCST STATEMENT FOR SDLC CONNECTED NCP-NCP LINKS    *                06270099
********************************************************                06280099
N46DPRIS SDLCST GROUP=N46DPRIG,        *                               X06290099
               MAXOUT=7,               * FRAMES RECIEVED BEFORE RESPONSX06300099
               MODE=PRIMARY,           * PRIMARY MODE                  X06310099
               PASSLIM=254,            * MAXIMUM # OF PIUS SENT TO PU  X06320099
               RETRIES=(3,2,30),       * RETRIES                       X06330099
               SERVLIM=4               * REGULAR / SPECIAL SCANS        06340099
N46DSECS SDLCST GROUP=N46DSECG,                                        X06350099
               MAXOUT=7,                                               X06360099
               MODE=SECONDARY,                                         X06370099
               PASSLIM=254,                                            X06380099
               RETRIES=3                                                06390099
*********************************************************************** 46680099
*        TOKEN RING PHYSICAL DEFINTIONS                               * 46690099
*********************************************************************** 46700099
N46DPTR1 GROUP ECLTYPE=(PHYSICAL,SUBAREA),                             X46710099
               NPACOLL=YES                                              46720099
N46LYA   LINE  ADDRESS=(1088,FULL),      TIC ADDRESS                   X46730099
               ISTATUS=ACTIVE,                                         X46743099
               OWNER=H53,                                              X46750099
               PORTADD=1,                                              X46760099
               MAXTSL=1108,                                            X46770099
               RCVBUFC=4095,          MAX FROM RING TO NCP             X46780099
               LOCADD=400000001C46    3745 ADDRESS ON RING              46790099
N46PYA   PU    ANS=CONT                                                 46800099
N46UYA   LU    ISTATUS=INACTIVE     DUMMY LU                            46810099
*              STATOPT=OMIT                                             46820099
*********************************************************************** 46829999
*        TOKEN RING LOGICAL DEFINITIONS - SUBAREA LINKS               * 46830099
*********************************************************************** 46830199
N46DLTR1 GROUP ECLTYPE=(LOGICAL,SUBAREA),  * LOGICAL SUBAREA GROUP *   X46830299
               ISTATUS=INACTIVE,                                       X46830399
               NPACOLL=YES,                                            X46830499
               OWNER=H53,                                              X46830599
               PHYSRSC=N46PYA                                           46830699
N46LXA47 LINE  SDLCST=(N46DPRIS,N46DSECS),ISTATUS=ACTIVE                46830799
N46PXA47 PU    ADDR=04400037450004                                      46830999
```

VTAM XCA Subarea Major Node

The following configuration statements are for the VTAM XCA subarea major node:

```
00001          VBUILD TYPE=XCA
00002 SUBAPRT  PORT  ADAPNO=0,                                               *
00003                CUADDR=100,                                             *
00004                MEDIUM=RING,                                            *
```

```
00005                SAPADDR=4,                                              *
00006                TIMER=30
00007 SUBAGRP  GROUP DIAL=NO
00008 SUBALN   LINE  USER=SNA
00009 SUBAPU   PU    MACADDR=4000374500004,                                  *
00010                PUTYPE=4,                                               *
00011                SAPADDR=4,                                              *
00012                SUBAREA=63,                                             *
00013                TGN=2
```

Cisco and the Cisco Logo are trademarks of Cisco Systems, Inc. and/or its affiliates in the U.S. and other countries. A listing of Cisco's trademarks can be found at www.cisco.com/go/trademarks. Third party trademarks mentioned are the property of their respective owners. The use of the word partner does not imply a partnership relationship between Cisco and any other company. (1005R)

Any Internet Protocol (IP) addresses used in this document are not intended to be actual addresses. Any examples, command display output, and figures included in the document are shown for illustrative purposes only. Any use of actual IP addresses in illustrative content is unintentional and coincidental.

© 2008 Cisco Systems, Inc. All rights reserved.