

# Chapter 1

## LLC2 and SDLC Link-Level Support

---

# 1

This chapter provides an overview of Cisco's implementation of the two IBM data-link level protocols:

- The LLC2 protocol used in LAN media connectivity, including Token Rings
- The SDLC serial link protocol used in WAN media IBM connectivity, including serial links

LLC2 and SDLC are never used alone, but provide a data-link level support for other, higher level router features.

The following features in software Release 9.0 use LLC2:

- Local Acknowledgment
- IBM LAN Network Manager support
- SDLLC™ SDLC/LLC2 Media Translation
- ISO Connection-Mode Network Services (CMNS)

The feature in software Release 9.0 that uses SDLC is SDLC/LLC2 Media Translation, which is documented in the chapter, "SDLLC: SDLC to LLC2 Media Translation."

---

**Note:** SDLC, in the sense described here, is not and must not be configured for the SDLC Serial Tunnel, which is described in the "Configuring Serial Tunneling in SDLC and HDLC Environments" chapter. The section on SDLC in this chapter describes situations where the Cisco router acts as an SDLC *station*. With Serial Tunneling, the Cisco router does not appear as a station, but merely *passes through* the frames between two other SDLC-speaking stations.

---

Information on fine tuning and customizing these protocols for use in the Cisco router is also found in this chapter. Most often, you will not need to customize these parameters, as the default system values for the parameters described in this chapter provide good results for most purposes. You will want to refer to this chapter when performance or functionality may be suffering, and a modification of these link-level parameters offers some hope in the solution of your current network problem. Also, if your network exhibits highly unusual characteristics in its very design, such as unusually large delay parameters, you will want to consult this chapter for the effects such designs can have on the operation of SDLC or LLC2.

---

## Overview and Comparison of SDLC and LLC2

LLC2 and SDLC are both data-link level, reliable protocols. Link-level protocols operate below the network level; that is, network protocols such as ISO CMNS and SNA run on top of these layers.

They are reliable because two stations communicating with LLC2 or SDLC will do the following:

- Open and close sessions with each other before and after sending data
- Acknowledge the receipt of any frames the other station sends
- Control the flow of data between them by limiting the number of frames that may be sent by one before the other acknowledges any of them
- Recover from errors and inform the other station of this

LLC2 and SDLC protocols differ primarily in two ways: the *role* each station plays in relationship to the other, and how the stations are *addressed*.

### Role of LLC2 and SDLC Stations

The following describes the different roles of LLC2 and SDLC stations:

- LLC2 stations communicate in an *asynchronous balanced mode (ABM)*. LLC2-speaking stations are considered *peers*, and may send data to other LLC2 stations at any time. This ability to send data at any time is called being *asynchronous*. *Balanced mode* means that each station can send to others without permission.
- SDLC stations communicate in a *normal response mode (NRM)*. SDLC-speaking stations assign one station to be the *primary* station and the other to be the *secondary* station. Only the primary may start and stop sessions, and the secondary may send data to the primary only in response to being *polled* by the primary. This requirement that only one of the two stations may talk at a given time is called *normal* communications. The ability for the secondary station to talk only in response to the primary station's polls is called *response mode*.

### Addressing of LLC2 and SDLC Stations

The following describes how LLC2 and SDLC stations are addressed:

- All frames used in LLC2 communication have both a source and destination address. LLC2-speaking stations usually communicate over a local area network (LAN) on which many stations communicate. As such, each station needs its own address. Because the LLC2 protocol runs at the link layer, the MAC (interface) address is typically used.

- SDLC frames have only a single address. SDLC-speaking stations communicate on serial links. Typically, there is one primary and one secondary station. If there were only two stations on any serial link, then no address would be needed. Any frame received would be, of course, meant for the station receiving it. However, modem or line sharing devices (MSDs or LSDs) are sometimes used to allow one primary station to communicate to many secondary stations. As there is only one primary, it needs no address, but rather each secondary station needs one, as each must distinguish frames meant for them versus frames meant for the others. Therefore, all SDLC frames specify a single address: the address of the secondary station.

---

## *Cisco's Implementation of LLC2*

The Cisco router supports LLC2 connections over the following IEEE interfaces:

- MCI Ethernet
- MEC Ethernet
- Token Ring
- IGS Ethernet
- FDDI

LLC2 connections are used in support of the IBM LAN Network Manager, LLC2 Local Acknowledgment, SDLLC SDLC/LLC2 Media Translation, and CMNS.

All of the LLC2 commands are optional and may be changed when you want to fine-tune LLC2 performance.

## *Configuring LLC2*

LLC2 is configured by interface. The following shows the LLC2 interface subcommand syntax for all LLC2 commands:

**llc2** *keyword value*

The following list includes some instances in which you may change LLC2 parameters:

- To control the maximum number of I-frames received before being acknowledged
- To control the maximum amount of time to allow incoming I-frames to stay unacknowledged
- To control the frequency of polls
- To control the response time to poll frames

Table 1-1 shows the LLC2 parameters that you may set. The table shows, for each value, the minimum and maximum allowed values, as well as the default. Following this table are descriptions of each parameter and some typical situations when you may wish to change parameter values.

*Table 1-1* LLC2 Parameters

<b>Command</b>	<b>Argument</b>	<b>Minimum</b>	<b>Maximum</b>	<b>Default</b>
<b>llc2 ack-max</b>	<i>packet-count</i>	1	255	3
<b>llc2 ack-delay-time</b>	<i>msec</i>	1	60000	3200
<b>llc2 idle-time</b>	<i>msec</i>	1	60000	10000
<b>llc2 local-window</b>	<i>packet-count</i>	1	127	7
<b>llc2 n2</b>	<i>retry-count</i>	1	255	8
<b>llc2 t1-time</b>	<i>msec</i>	1	60000	1000
<b>llc2 tbusy-time</b>	<i>msec</i>	1	60000	9600
<b>llc2 trej-time</b>	<i>msec</i>	1	60000	3200
<b>llc2 tpf-time</b>	<i>msec</i>	1	60000	1000
<b>llc2 xid-neg-val-time</b>	<i>msec</i>	1	60000	0
<b>llc2 xid-retry-time</b>	<i>msec</i>	1	60000	60000

---

**Note:** All timers are specified in milliseconds.

---

The **llc2 ack-max** command controls the maximum number of information frames (I-frames) received by the Cisco router before it must send an acknowledgment to these frames to their sender.

**llc2 ack-max** *packet-count*

As stated in the overview, an LLC2-speaking station may send only a predetermined number of frames before it must wait for an acknowledgment from the receiver. If the receiver waits until receiving a large number of frames before acknowledging any of them, it reduces overhead on the network, as an acknowledgment for five frames can specify that all five have been received, as opposed to sending a separate acknowledgment for each single frame. Therefore, the number of acknowledgments traveling on the network can be reduced. To reduce network overhead, make this parameter as large as possible. However, some LLC2-speaking stations expect this to be a low number. For example, some NetBIOS-speaking stations expect an acknowledgment to every frame. Therefore, for these stations, this number is best set to 1. Experiment with this parameter to determine the best configuration.

The **llc2 ack-delay-time** controls the maximum amount of time the Cisco router allows incoming I-frames to stay unacknowledged.

**llc2 ack-delay-time** *msec*

Each LLC2 station, upon receiving an information frame, starts an *alarm clock*, or timer. If this alarm clock goes off before an acknowledgment for the frame has been sent, an acknowledgment will be sent for the frame, even if the **llc2 ack-max** number of received frames has not been reached. For example, assume you have the following configuration:

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

At time 0, for example, two information frames are received. The ack-max 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 in 800 milliseconds, an acknowledgment will be sent anyway, as the ack-delay timer alarm will have gone off. At this point, as all frames are acknowledged, the counter for the ack-max purposes will be reset to zero.

Experiment with this parameter to determine the configuration that balances undesirable acknowledgment network overhead and quick response time (by receipt of timely acknowledgments) for stations sending large amounts of information frames.

The **llc2 idle-time** command controls the frequency of polls during periods of idle traffic.

#### **llc2 idle-time** *msec*

Periodically, when nothing is going on in an LLC2 session (no information frames are being transmitted), LLC2 stations are likely to send an “are you there” frame to each other, to which a similar frame is sent back letting each other know that they are, in fact, still there. This parameter controls the amount of idle time that must go by before one of these “are you there” frames are sent. These “are you there” frames are actually called *Receiver Readys*.

Set this number to a low enough value to ensure a timely discovery by the Cisco router to discover if an LLC2 station to which it is talking has died. However, you do not want to set this too low, as you will create a network overhead with too many “are you there” frames.

The **llc2 local-window** command controls the maximum number of information frames sent by the router before it waits for an acknowledgment to these frames.

#### **llc2 local-window** *packet-count*

Set this number to the maximum value that can be supported by the stations to which the Cisco router communicates. Setting this value too large can cause frames to be lost, as the receiving station may not be able to receive all of them.

The **llc2 n2** command controls the amount of times the Cisco router retries various operations, including the sending of an unacknowledged frame, or retries the polling of a remote busy station.

#### **llc2 n2** *retry-count*

This parameter controls the *retry-count* limit. An LLC2 station must have some limit to the number of times it will resend a frame when the receiver of that frame has not acknowledged it for any reason. After the router is told a remote station is busy, it will poll again based on the *retry-count* until it gives up. When this retry count is exceeded, the LLC2 station terminates its session with the other station.

Very often, there is little need to change the retry count limit, and its change is not recommended.

The **llc2 t1-time** command controls how long the Cisco router waits for an acknowledgment to transmitted I-frames.

**llc2 t1-time msec**

When an LLC2 station sends a frame, it waits for an acknowledgment from the receiver saying that this frame has been received. Of course, the sending station cannot wait forever for a response. When the frame is sent, a timer is started. This timer is called the *T1 timer* and is controlled by this parameter. If this timer reaches its limit before the acknowledgment is received, the Cisco router tries again and resends the frame.

The amount of times the router will retry sending a frame before disconnecting the session is controlled by the **llc n2** parameter.

---

**Note:** The **llc2 t1-time** parameter is extremely important, and is probably the one you most want to change. Ensure that this parameter can account for the roundtrip time between the Cisco router and its LLC2-speaking stations under heavy network loading conditions. You want to avoid causing unnecessary retransmissions to be sent at any cost; they will only add to the network load.

---

The **llc2 tbusy-time** command controls the amount of time the Cisco router allows the other LLC2 station to stay in a busy state before this router attempts to poll the remote station again, hoping for a clear state.

**llc2 tbusy-time msec**

LLC2 stations have the ability to tell each other that they are temporarily busy, so the other station should not attempt to send any new information frames. The frames sent to indicate this are called *Receiver Not Ready (RNR) frames*.

Usually, there is little reason to change this parameter. You would only do so to increase the value if you discover that you have LLC2-speaking stations that have unusually long busy periods before they clear their busy state indicating that they are ready to receive more I-frames. Therefore, you would not want to time these stations out.

The amount of times the router will retry polling the remote station hoping for a clear state before disconnecting the session is controlled by the **llc n2** parameter.

The **llc2 trej-time** command controls the amount of time the Cisco router waits for a resend of a rejected frame before resending the reject (REJ) command to the remote.

**llc2 trej-time msec**

When an LLC2 station sends an information frame, a sequence number (called the N(S) value or *Number of Sent frame*) 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.

Of course, the LLC2 station that sends the REJ does not want to wait forever to receive the correct frame. Therefore, upon sending a reject, it starts a *reject timer*. If the frames are not received before this timer expires, the reject is resent.

The amount of times the router retries sending the REJ frame without getting the rejected I-frame sent to it before disconnecting the session is controlled by the **llc n2** parameter.

The **llc2 tpf-time** command controls the amount of time the router waits for a *final* response to a *poll* frame that it sent before the router resends the original *poll* frame.

#### **llc2 tpf-time msec**

LLC2 was designed from earlier serial link protocols, including SDLC. Therefore, when sending a command that must receive a response, or finishing the transmission of data when there is no more to send, a *poll* bit is sent in the frame. This is the receiving station's clue that the sender is expecting some response from it, be it an acknowledgment of information frames, or an acknowledgment of more administrative things, such as the starting and stopping of the session. Once a sender gives out the *poll* bit, it may not 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. Obviously, if the receiver is poorly designed or buggy, it may never give that *final* bit back to the sender. Therefore, the sender could be stuck, waiting for a reply that will never come. To avoid this problem, when a poll-bit-set frame is sent, a *transmit-poll-frame (TPF)* timer is started. If this timer expires, the Cisco router assumes that it may send another frame with a *poll* bit.

Usually, you will not want to change this value. If you do, it should be somewhat larger than the **t1-time**, as frames with *poll* bits can often take some time longer for a station to process once received. Increase this timer if you find that you have such slow LLC2 stations.

The amount of times the router retries sending the *poll* frame to the remote station hoping for a response before disconnecting the session is controlled by the **llc n2** parameter.

The **llc2 xid-neg-val-time** command controls the frequency of XID transmissions by the Cisco router.

#### **llc2 xid-neg-val-time msec**

LLC2-speaking stations can communicate *exchange of identification (XID)* frames to each other. These frames identify the stations at a higher level than the MAC address, and also can contain information about the configuration of the station. These frames are typically sent only during setup and configuration periods when it is deemed that sending them is useful. The greatest frequency at which this information is transferred is controlled by this timer.

---

**Note:** Do not change the **llc2 xid-neg-val-time** parameter unless requested by Cisco.

---

The **llc2 xid-retry-time** command controls how long the router waits for a reply to the XID frames it sends to remote stations.

**llc2 xid-retry-time msec**

Along with the periodic normal retransmittal of XID frames controlled by the **xid-neg-val-time**, the amount of time that the Cisco router will wait for an XID from the remote LLC2 station in response to its XID must be controlled. This xid-retry timer controls this value. It should be set to a somewhat larger value than the T1 time, and should be set in situations where the roundtrip delay through the network is unexpectedly large.

## Monitoring LLC2

The following EXEC command, with example output, gives the state of LLC2 connections within a Cisco router:

**show llc2**

The **show llc2** command shows the LLC2 connections active in the router. The following is an example of this output:

```
TokenRing0 DTE=1000.5A59.04F9,40002224444 SAP=04/04, State=NORMAL
V(S)=5, V(R)=5, Last N(R)=5, Local window=7, Remote Window=127
ack-max=3, n2=8, Next timer in 7768
xid-retry timer 0/60000 ack timer 0/1000
p timer 0/1000 idle timer 7768/10000
rej timer 0/3200 busy timer 0/9600
ack-delay timer 0/3200
CMNS Connections to:
Address 1000.5A59.04F9 via Ethernet2
Protocol is up
Interface type X25-DCE RESTARTS 0/1
Timers: T10 1 T11 1 T12 1 T13 1
```

Table 1-2 describes the **show llc2** command fields.

**Table 1-2** Show LLC2 Field Descriptions

Field	Description
Interface	Interface on which the session is established.
DTE	Address of the station to which the Cisco router is talking on this session. (The router's address is the MAC address of the interface on which the connection is established, except when Local Acknowledgment or SDLLC is used, in which case the address used by the router is shown as in this example, following the DTE address and separated by a comma.)
SAP	The other station's and the router's (remote/local) Service Access Point for this connection. The SAP is analogous to a "port number" on the router and allows for multiple sessions between the same two stations.



Field	Description
State	The current state of the LLC2 session which are any of:
ADM	Asynchronous Disconnect Mode—A connection is not established, and either end can begin one.
SETUP	Request to begin a connection has been sent to the remote station, and this station is waiting for a response to that request.
RESET	A previously open connection has been reset because of some error by this station, and this station is waiting for a response to that reset command.
D_CONN	This station has requested a normal, expected, end of communications with the remote, and is waiting for a response to that disconnect request.
ERROR	This station has detected an error in communications and has told the other station of this. This station is waiting for a reply to its posting of this error.
NORMAL	Connection between the two sides is fully established, and normal communication is occurring.
BUSY	Normal communication state exists, except busy conditions on this station make it such that this station cannot receive information frames from the other station at this time.
REJECT	Out-of-sequence frame has been detected on this station, and this station has requested that the other resend this information.
AWAIT	Normal communication exists, but this station has had a timer expire, and is trying to recover from it (usually by resending the frame that started the timer).
AWAIT_BUSY	A combination of the AWAIT and BUSY states.
AWAIT_REJ	A combination of the AWAIT and REJECT states.
V(S)	Sequence number of the next information frame this station will send.
V(R)	The sequence number of the next information frame this station expects to receive from the other station.
Last N (R)	Last sequence number of this station's transmitted frames acknowledged by the remote station.
Local Window	How many frames this station may send before requiring an acknowledgment from the remote station.
Remote Window	How many frames this station can accept from the remote.
ack-max, n2	Value of these parameters, as given in the previous configuration section.
Next timer in <nnn>	How many milliseconds before the next timer, for any reason, goes off.
Timer Values	A series of timer values in the form of next-time/time-between, where "next-time" is the next time, in milliseconds, that the timer will wake, and "time-between" is the time, in milliseconds, between each timer wakeup. A "next-time" of zero indicates that the given timer is not enabled, and will never wake.

Field	Description
CMNS parameters:	
Address	MAC address of remote station.
Protocol State	UP indicates the LLC2 and X.25 protocols are in a state where incoming and outgoing Call Requests can be made on this LLC2 connection.
Interface type	One of: X25-DCE, X25-DTE, or X25-DXE (both DTE and DCE).
RESTARTS	Restarts sent/received on this LLC2 connection.
Timers	T10, T11, T12, T13 (or T20, T21, T22, T23 for DTE); these are Request packet timers. These are similar in function to X.25 parameters of the same name.

## Debugging LLC2

The following commands describe how to retrieve debugging information for LLC2. The output of these commands is detailed and contains internal program information. It can only be adequately understood when used in conjunction with Cisco staff. All commands are disabled with the corresponding **undebug** command.

### **debug llc2-events**

The **debug llc2-events** command turns on LLC2 event debugging. Actions that cause the current LLC2 state to change are displayed.

### **debug llc2-packets**

The **debug llc2-packets** command turns on LLC2 packets debugging. All LLC2 frames input or output to and from the router are displayed.

### **debug llc2-test**

The **debug llc2-test** command turns on internal LLC2 code debugging.

---

**Note:** LLC2 debugging can generate large amounts of console output that can adversely affect the operation of LLC2 on the router, as packets can be delayed long enough to cause retransmissions. Only use LLC2 debugging when suggested by Cisco Systems staff.

---

---

## Cisco's Implementation of SDLC

This section describes the commands that apply to the SDLC implementation. They are the counterpoint to the LLC2 commands specified in the LLC2 section of this chapter.

SDLC is used in Cisco's implementation of SDLLC, the media translator between LLC2 and SDLC.

Cisco's implementation of SDLC supports *multipoint*, using a modem or line sharing device (MSD or LSD), in those configurations where the device speaks a full duplex protocol to the Cisco router.

Currently, the Cisco router can only act as the primary end of the SDLC session. As a result, in SDLLC applications, the Token Ring station must always act as the primary station, and the SDLC station must always act as the secondary station.

## Configuring SDLC

All of the commands in this section are given to the serial interface that supports the SDLLC traffic.

The **encapsulation sdlc-primary** command identifies any interface that requires an SDLC configuration. This command must always be given, no matter what functionality is using the SDLC support. The specification of **sdlc-primary** emphasizes that the Cisco router is always the primary station when behaving as an SDLC station.

### **encapsulation sdlc-primary**

For example, if you are running SDLLC on interface serial 0, begin the configuration with:

```
interface serial 0
encapsulation sdlc-primary
```

After the **encapsulation sdlc-primary** command, assign the set of secondary stations attached to the serial link using a sequence of **sdlc address** commands:

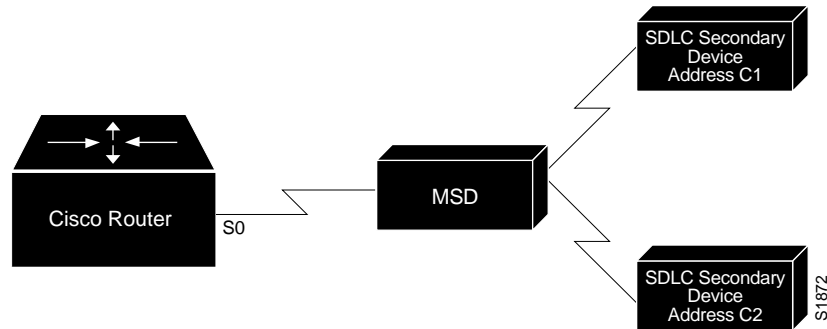
### **sdlc address hexbyte** **no sdlc address**

The addresses are given in hexadecimal (base 16), and are given one per line. For example, to configure interface serial 0 to have two SDLC secondary stations attached to it through a modem sharing device (MSD), with addresses C1 and C2, begin your SDLC configuration of this interface with the following:

```
interface serial 0
encapsulation sdlc-primary
sdlc address c1
sdlc address c2
```

The network for this configuration is shown in Figure 1-1:

**Figure 1-1** Two SDLC Secondary Stations Attached to a Single Serial Interface Through an MSD.



SDLC is configured per interface. The following shows the SDLC command syntax for all SDLC interface subcommands:

**sdlc** *keyword value*

---

**Note:** All of these commands are optional and modify the behavior of SDLC on the link.

---

Table 1-3 shows the SDLC parameters. Each description is found following the table.

**Table 1-3** SDLC Parameters

Command	Argument	Minimum	Maximum	Default
<b>sdlc frmr-disable</b>		enable/disable		disabled
<b>sdlc t1</b>	<i>timeout</i>	1	64000	3000
<b>sdlc n1</b>	<i>bit-count</i>	1	12000	12000
<b>sdlc n2</b>	<i>retry-count</i>	1	255	20
<b>sdlc k</b>	<i>window-size</i>	1	7	7
<b>sdlc holdq</b>	<i>address queue-size</i>	1	none	12
<b>sdlc poll-pause-timer</b>	<i>msec</i>	100	10000	100
<b>sdlc fair-poll-timer</b>	<i>msec</i>	100	10000	500
<b>sdlc poll-limit-value</b>	<i>count</i>	1	10	1

The **[no] sdlc frmr-disable** command is the only subcommand that has no value assigned to it.

---

**Note:** Do not change the default values for the SDLC commands, unless situations warrant their change, or unless Cisco Systems staff suggest a change. All timers are given in milliseconds.

---

### *Controlling the SDLC Protocol Behavior*

Specify the **frmr-disable** command to indicate that secondary stations off this link do not support *Frame Rejects* (FRMRs). The command syntax is as follows:

```
sdlc frmr-disable  
no sdlc frmr-disable
```

FRMRs are *error indications* that can be sent to an SDLC station indicating that a protocol error has occurred. Not all SDLC stations support FRMRs. If this command is enabled, thereby disabling FRMRs, the Cisco router simply drops the line when it receives an error by sending a disconnect request to the remote station.

The default value for this command is to send FRMRs. This default behavior can be retrieved by specifying the **no** version of this command.

### *Controlling the SDLC Timers and Retry Counts*

The **sdlc t1 timeout** command controls the amount of time the Cisco router waits for a reply to a frame or sequence of frames.

```
sdlc t1 timeout
```

When an SDLC station sends a frame, it waits for an acknowledgment from the receiver saying that this frame has been received. Of course, the sending station cannot wait forever for a response. When the frame is sent, a timer is started. For reasons of being consistent with the original specification of SDLC, this timer is called the *T1 timer* and is controlled by this parameter. If this timer reaches its limit before the acknowledgment is received, the Cisco router will try again and resend the frame.

The number of times that a retry of a frame is tried is controlled with the **sdlc n2** command.

This parameter is extremely important, and is probably the one you most want to change. Ensure that this parameter can account for the roundtrip time between the Cisco router and its SDLC speaking stations under heavy network loading conditions. You want to avoid causing unnecessary retransmissions to be sent at any cost, as they will only add to the network load.

The **sdlc n2** *retry-count* command controls the number of times the Cisco router attempts to retry an operation that has timed out.

**sdlc n2** *retry-count*

The **sdlc n2** parameter controls the *retry-count* limit. When the router as an SDLC station has to resend a frame because the receiver of that frame has not acknowledged it for any reason, the SDLC station must have some limit to the number of times it will try again. When this retry count is exceeded, the SDLC station will terminate its session with the other station.

Very often, there is little need to change this parameter, and its change is not recommended.

### *Controlling the SDLC Frame and Window Sizes*

The **sdlc n1** *bit-count* command controls the maximum size of an incoming frame, in bits, on this link.

**sdlc n1** *bit-count*

Frames received that exceed the specified size, in bits, are rejected, and an error condition will ensue.

The **sdlc k** *window-size* command controls the local send window size of the Cisco router.

**sdlc k** *window-size*

When the Cisco router is communicating with SDLC, it must have a parameter that controls the maximum number of information frames it will receive before the other end must stop sending and wait for an acknowledgment. This allows for the better creation of buffer pools and other memory objects. The **k** parameter controls this *window* of acceptable frames.

### *Affecting the Output Buffering Ability*

The following command controls the output buffering ability.

The **sdlc holdq** *address queue-size* command controls the maximum number of outstanding packets that may be held for pending transmission to a remote SDLC station. This is specified on a per-address basis.

**sdlc holdq** *address queue-size*

Consider the example of the SDLLC media translator. This feature allows an LLC2-speaking SNA station on a Token Ring to communicate to an SDLC-speaking SNA station on a serial link. However, the frame sizes and window sizes on Token Rings are often much larger than those acceptable for serial links. The fact that serial links are often much slower than Token Rings often makes this problem worse. Therefore, temporary backlogs can exist in periods of high data transfer from the Token Ring station to the serial station. The router needs to create a holding place for these backlogged frames as they are awaiting transmission on the serial link. This command is specified for each SDLC address, and thereby, each SDLC secondary station on the serial link.

Therefore, if you have an SDLC station off serial 0 of address C1, and wanted to change its output hold queue length to 30 frames (of any size), you would give the following commands:

```
interface serial 0
encapsulation sdhc-primary
sdhc address c1
sdhc holdq c1 30
```

Use this command to increase the hold queue size from the default maximum when you find that there exist temporary loading conditions which exceed the default hold queue length of the router.

### *Affecting Polling of the Secondary Stations*

The following three parameters affect the behavior of polls in the system:

**[no] sdhc poll-pause-timer** *msec*

**[no] sdhc fair-poll-timer** *msec*

**[no] sdhc poll-limit-value** *count*

It is assumed that you are familiar with how polls operate on SDLC links before you attempt to change these values. A brief summary follows.

SDLC secondary stations, particularly when an MSD/LSD is used, are typically attached to the Cisco router out each of its SDLC-speaking serial stations. These stations are true SDLC secondaries, and expect the Cisco router to perform its duty as an SDLC primary station, polling them fairly and in a timely manner. As secondary stations, these stations cannot send data to the primary (Cisco router) until they are polled for that data. As they are sharing a single serial link, only one station may be the actively polled station at any one time.

To meet the *timely* requirement, the Cisco router must show intelligence in accepting streams of input, or in giving streams of output, to a single station, even if these streams require that more than one packet or window size of packets must be transferred. The goal of this intelligence is to keep on giving output to, or polling for, input from the same station until a full transaction of data has completely transferred in one direction.

However, meeting this requirement impacts the *fairness* to the other stations. The Cisco router, as the primary, cannot show undue favoritism to a suddenly noisy secondary, and avoid its responsibility of polling the other stations in their turn. Also, the router does not want to flood the serial link with polls for otherwise idle stations. The following three parameters help control this behavior. For all three parameters, retain the defaults unless a problem is evidenced, and adjust them as little as possible until the desired behavior is obtained. Fairness is also obtained by, in an idle state, polling stations in a round-robin manner.

The **sdhc poll-pause-timer** *msec* command controls how long the router pauses between sending each series of poll frames.

**sdhc poll-pause-timer** *msec*

**no sdhc poll-pause-timer**

As is typical for the primary station of an SDLC connection, polls are periodically generated by the Cisco router to each of the secondary stations to solicit their input. After a full cycle of polls to all stations off a single serial interface are sent and replied to, and there is no output to give to any secondary station from the primary station, the router will pause and let the line remain idle for the amount of this timer before beginning to poll the stations again.

As the secondaries cannot transmit data (such as a user hitting an Enter key) until they are polled, increasing this timer can increase response time to the users. However, making this parameter too small can flood the serial link with unneeded polls, and require the secondary stations to spend wasted CPU time processing them.

The default value is 100 msec. This default value can be retrieved by specifying the **no** version of this command.

The **sdhc fair-poll-timer msec** command controls the maximum time that output may be sent to any secondary before polling for input must begin.

**sdhc fair-poll-timer msec**  
**no sdhc fair-poll-timer**

As is typical for the primary station of an SDLC connection, output from the primary to any of the secondary stations is given a higher priority than receiving input from any of them. The **fair-poll-timer** indicates the maximum time that can elapse before output from the primary is no longer given that priority and the next poll of the next secondary station is attempted.

Increasing this parameter allows screen updates to appear smoother, but has the side effect of possibly increasing, for example, the time it takes for users hitting an Enter key to have their data sent to their station.

The default value is 500 msec. This default value can be retrieved by specifying the **no** version of this command.

The **sdhc poll-limit-value count** command controls how many times a single secondary station may be polled for input before the next station in the poll list must be polled.

**sdhc poll-limit-value count**  
**no sdhc poll-limit-value**

As is typical for the primary station of an SDLC connection, if a secondary station sends its full possible window of input to the primary Cisco router, the router will immediately repoll this same secondary for more data, in an attempt to capture the complete transaction at one time. The **poll-limit-value** indicates how many times this can happen before the next station in the poll loop must be polled.

Increasing the value allows for smoother transaction processing, but can delay the polling of other stations or the giving of output to other stations.

The default value is 1. This default value can be retrieved by specifying the **no** version of this command.



## Monitoring SDLC

The following command displays the SDLC information for a given interface:

```
show interfaces type unit
```

For example, assume that interface Serial 0 is supporting the SDLLC function, and as such is configured as an SDLC primary interface. The following output would result from the **show interfaces** command:

```
Serial 0 is up, line protocol is up
Hardware is MCI Serial
MTU 1500 bytes, BW 1544 Kbit, DLY 20000 usec, rely 255/255, load 1/255
Encapsulation SDLC-PRIMARY, loopback not set
  Timers (msec): poll pause 100 fair poll 500. Poll limit 1
    [T1 3000, N1 12016, N2 20, K 7] timer: 56608 Last polled device: none
    SDLLC [ma: 0000.0C01.14--, ring: 7 bridge: 1, target ring: 10
      largest token ring frame 2052]
SDLC addr C1 state is CONNECT
  VS 6, VR 3, RCNT 0, Remote VR 6, Current retransmit count 0
  Hold queue: 0/12 IFRAMES 77/22 RNRs 0/0 SNRMs 1/0 DISCs 0/0
  Poll: clear, Poll count: 0, chain: p: C1 n: C1
  SDLLC [largest SDLC frame: 265, XID: disabled]
Last input 00:00:02, output 00:00:01, output hang never
Output queue 0/40, 0 drops; input queue 0/75, 0 drops
Five minute input rate 517 bits/sec, 30 packets/sec
Five minute output rate 672 bits/sec, 20 packets/sec
  357 packets input, 28382 bytes, 0 no buffer
  Received 0 broadcasts, 0 runts, 0 giants
  0 input errors, 0 CRC, 0 frame, 0 overrun, 0 ignored, 0 abort
  926 packets output, 77274 bytes, 0 underruns
  0 output errors, 0 collisions, 0 interface resets, 0 restarts
  2 carrier transitions
```

Table 1-4 shows the fields relevant to all SDLC connections on this interface.

**Table 1-4** Show Interfaces Fields and Descriptions

Parameter	Description
Timers (msec)	
poll pause, fair poll, poll limit	Current values of these timers, as described in the configuration section, for this interface.
T1, N1, N2, K	Values for these parameters, as described in the configuration section, for this interface.

Table 1-5 shows other data given for each SDLC secondary configured to be attached to this interface.

**Table 1-5** SDLC Secondary Descriptions

<b>SDLC Secondary</b>	<b>Description</b>
addr	Address of this secondary.
State	Current state of this connection which are any of:
DISCONNECT	No communication is being attempted to this secondary.
CONNECT	A normal connect state exists between this router and this secondary.
DISSCENT	This router has sent a disconnect request to this secondary and is awaiting its response.
SNRMSENT	This router has sent a connect request (SNRM) to this secondary and is awaiting its response.
THEMBUSY	This secondary has told this router that it is temporarily unable to receive any more information frames.
USBUSY	This router has told this secondary that it is temporarily unable to receive any more information frames.
BOTHBUSY	Both sides have told each other that they are temporarily unable to receive any more information frames.
ERROR	This router has detected an error, and is waiting for a response from the secondary acknowledging this.
VS	Sequence number of the next information frame this station sends.
VR	Sequence number of the next information frame from this secondary that this station expects to receive.
Remote VR	Last frame transmitted by this station that has been acknowledged by the other station.
Current retransmit count:	How many times the current I-frame or sequence of I-frames has been retransmitted.
Hold Queue	Number of frames in hold queue/Maximum size of hold queue.
IFRAMEs, RNRs, SNRMs, DISCs	Sent/received count for these frames.
Poll	“Set” if this router has a poll outstanding to the secondary; “clear” if it does not.
Poll Count	How many polls, in a row, have been given to this secondary at this time.
Chain	Shows the previous (p) and next (n) secondary address on this interface in the <i>round robin loop</i> of polled devices.

## Debugging SDLC

### **debug sdlc**

This command displays SDLC frames received and sent by any Cisco router serial interface involved in supporting SDLC end station functions. The debugging output may be disabled with the **undebug** command.

---

## LLC2 Interface Subcommand Summary

The following is an alphabetical list of the LLC2 interface subcommands.

### **llc2 ack-delay-time** *msec*

Controls the maximum amount of time the Cisco router allows incoming I-frames to stay unacknowledged. Each LLC2 station, upon receiving an information frame, starts an *alarm clock*, or timer. If this alarm clock goes off before an acknowledgment for the frame has been sent, an acknowledgment will be sent for the frame, even if the **llc2 ack-max** number of received frames has not been reached.

### **llc2 ack-max** *packet-count*

Controls the maximum number of information frames (I-frames) received by the Cisco router before it must send an acknowledgment to these frames to their sender. Experiment with this parameter to determine the best configuration.

### **llc2 idle-time** *msec*

Controls the frequency of polls during periods of idle traffic.

### **llc2 local-window** *packet-count*

Controls the maximum number of information frames sent by the router before it waits for an acknowledgment to these frames.

### **llc2 n2** *retry-count*

Controls the amount of times the Cisco router retries various operations, including the sending of an unacknowledged frame, or retries the polling of a remote busy station.

**llc2 t1-time** *msec*

Controls how long the Cisco router waits for an acknowledgment to transmitted I-frames. The **llc2 t1-time** parameter is extremely important, and is probably the one you most want to change.

**llc2 tbusy-time** *msec*

Controls the amount of time the Cisco router allows the other LLC2 station to stay in a busy state before this router attempts to poll the remote station again, hoping for a clear state.

**llc2 tpf-time** *msec*

Controls the amount of time the router waits for a final response to a poll frame that it sent before the router resends the original poll frame.

**llc2 trej-time** *msec*

Controls the amount of time the Cisco router waits for a resend of a rejected frame before resending the reject (REJ) command to the remote.

**llc2 xid-neg-val-time** *msec*

Controls the frequency of XID transmissions by the Cisco router. Do not change the **llc2 xid-neg-val-time** parameter unless requested by Cisco.

**llc2 xid-retry-time** *msec*

Controls how long the router waits for a reply to the XID frames it sends to remote stations.

---

## *SDLC Interface Subcommand Summary*

The following is an alphabetical list of the SDLC interface subcommands.

**encapsulation sdlc-primary**

Identifies any interface that requires an SDLC configuration. This command must always be given, no matter what functionality is using the SDLC support. The specification of **sdlc-primary** emphasizes that the Cisco router is always the primary station when behaving as an SDLC station.

**[no] sdhc address** *hexbyte*

Assigns the set of secondary stations attached to the serial link using a sequence of **sdhc address** commands.

**[no] sdhc fair-poll-timer** *msec*

Controls the maximum time that output may be sent to any secondary before polling for input must begin. The **fair-poll-timer** indicates the maximum time that can elapse before output from the primary is no longer given that priority and the next poll of the next secondary station is attempted. The default value is 500 msec. This default value can be retrieved by specifying the **no** version of this command.

**[no] sdhc frmr-disable**

Indicates that secondary stations off this link do not support *Frame Rejects* (FRMRs). If this command is enabled, thereby disabling FRMRs, the Cisco router simply drops the line when it receives an error by sending a disconnect request to the remote station. The default value for this command is to send FRMRs. This default behavior can be retrieved by specifying the **no** version of this command.

**sdhc holdq** *address queue-size*

Controls the maximum number of outstanding packets that may be held for pending transmission to a remote SDLC station. This is specified on a per-address basis.

This command is specified for each SDLC address, and thereby, each SDLC secondary station on the serial link. Use this command to increase the hold queue size from the default maximum when you find that there exist temporary loading conditions which exceed the default hold queue length of the router.

**sdhc k** *window-size*

Controls the local send window size of the Cisco router. The **k** parameter controls this *window* of acceptable frames.

**sdhc n1** *bit-count*

Controls the maximum size of an incoming frame, in bits, on this link. Frames received that exceed the specified size, in bits, are rejected, and an error condition will ensue.

**sdhc n2** *retry-count*

Controls the number of times the Cisco router attempts to retry an operation that has timed out. When this retry count is exceeded, the SDLC station will terminate its session with the other station.

**[no] sdhc poll-limit-value** *count*

Controls how many times a single secondary station may be polled for input before the next station in the poll list must be polled. The **poll-limit-value** indicates how many times this can happen before the next station in the poll loop must be polled. The default value is 1. This default value can be retrieved by specifying the **no** version of this command.

**[no] sdhc poll-pause-timer** *msec*

Controls how long the router pauses between sending each series of poll frames. The default value is 100 msec. This default value can be retrieved by specifying the **no** version of this command.

**sdhc t1** *timeout*

Controls the amount of time the Cisco router waits for a reply to a frame or sequence of frames.