CHAPTER 22

Configuring STUN and SDLC Local Acknowledgment

Our serial tunneling (STUN) feature allows Synchronous Data Link Control (SDLC) or High-level Data-Link Control (HDLC) devices to connect to one another through a multiprotocol internetwork rather than through a direct serial link. STUN encapsulates SDLC frames in either the Transmission Control Protocol/Internet Protocol (TCP/IP) or the HDLC protocol. STUN provides a straight pass-through of all SDLC traffic (including control frames, such as *Receiver Ready*) end-to-end between Systems Network Architecture (SNA) devices.

Our *SDLC Transport* or *SDLC Local Acknowledgment* feature provides local termination of the SDLC session, so that control frames no longer travel the WAN backbone networks. This means that end nodes do not time out, and there is no loss of sessions.You can configure your network with STUN, or with STUN *and* SDLC Local Acknowledgment.

To enable SDLC Local Acknowledgment, routers first must be enabled for STUN and configured to appear on the network as primary or secondary SDLC nodes. TCP/IP encapsulation must be enabled. Our SDLC Transport feature also provides priority queuing for TCP-encapsulated frames.

Note If you have Software Release 9.0 or earlier, you can only enable STUN. In lieu of Local Acknowledgment, the proxy-polling feature is provided.The functions provided by proxy polling have been enhanced and superseded by SDLC Local Acknowledgment. Use of proxy polling is NOT recommended.

For a complete description of the commands mentioned in this chapter, refer to the "STUN and SDLC Local Acknowledgement Commands" chapter of the *Router Products Command Reference* publication. For historical background and a technical overview of STUN and SDLC Local Acknowledgment, see the *Internetworking Technology Overview* publication.

Configuration Task List

To configure STUN, or STUN *and* SDLC Local Acknowledgment, complete the following tasks. The first and second tasks are required; the others are optional:

- Set Up a STUN Network
- Establish the SDLC Frame Encapsulation Method
- Set up Traffic Priorities
- Configure Proxy Polling

  Proxy polling is used for Software Release 9.0 or earlier only. Proxy polling has been superseded by SDLC Local Acknowledgment and is NOT recommended.
- Monitor STUN Network Activity

This chapter describes how to perform these tasks. At the end of the chapter are configuration examples.

Set Up a STUN Network

To set up the STUN network, complete the following tasks:

- Enable STUN on a global basis
- Configure the STUN protocol groups
- Enable STUN interfaces and place in a STUN group

Enable STUN on a Global Basis

Perform the following task in global configuration mode to enable STUN:

Task	Command
Enable the STUN for a particular IP address.	**stun peer-name** *ip-address*

STUN can only be enabled on full-duplex lines with Ready To Send (RTS) and Clear To Send (CTS) set to high. If your SDLC network uses a multipoint configuration (one primary and two or more secondary stations), ensure that Carrier Detect (CD) and Receive Line Signal Detect (RLSD) are set to low on the primary station.

When configuring redundant links, ensure that the STUN peer names you choose on each router are the IP addresses of interfaces that are not part of the normal path between the two routers. See "STUN Configuration Examples" later in this chapter.

Configure the STUN Protocol Groups

Each STUN interface must be placed in a group that defines the ISO 3309-compliant framed protocol running on that link. Packets will only travel between STUN interfaces that are in the same protocol group. You can create the following protocol groups:

- SDLC (if you choose the SDLC protocol, you can also configure SDLC transmission groups)
- Non-SDLC (for example, if your network is running HDLC)

You can also create your own protocol type.

SDLC Protocol

You can define SDLC protocol groups to associate interfaces with the SDLC protocol. To define an SDLC protocol group, perform the following task in global configuration mode:

Task	Command
Define an SDLC protocol group and assign a group number.	**stun protocol-group** *group-number* **sdlc**

If you specify the keyword **sdlc** in the **stun protocol-group** command string, you cannot specify the **stun route all** command on that interface.

For an example of how to configure an SDLC protocol group, see "Example of Configuring Serial Link Address Prioritization using STUN TCP/IP Encapsulation" later in this chapter.

Note You must specify the SDLC protocol if you want to use the Local Acknowledgment feature or if you need to set up proxy polling (Software Release 9.0 or earlier).

SDLC Transmission Groups

You can set up an SNA transmission group. A transmission group is a set of groups providing parallel links to the same pair of IBM establishment controllers. This provides redundancy of paths so that if one or more links go down, an alternate path can be used. To define an SDLC transmission group, perform the following task in global configuration mode:

Task	Command
Define an SDLC protocol group, assign a group number, and create an SNA transmission group.	**stun protocol-group** *group-number* **sdlc sdlc-tg**

All STUN connections in a transmission group must connect to the same IP address and use the SDLC Local Acknowledgment feature.

For an example of how to configure an transmission group, see "Example of Configuring Transmission Groups" later in this chapter.

Non-SDLC Protocols

You can define non-SDLC protocol groups to associate interfaces with a non-SDLC protocol. For example, your networks may be set to support HDLC instead of SDLC. If you choose to create your own protocol, you can establish a non-SDLC protocol group by performing the following task in global configuration mode:

Task	Command
Define a non-SDLC protocol group and assign a group number.	**stun protocol-group** *group-number* **basic**

Create Your Own Protocol

If you create your protocol, first define a non-SDLC protocol group as described in "Non-SDLC Protocols." To create your protocol, you can select one of the following address formats and lengths: decimal base 10 (4 bytes), hexadecimal base 16 (8 bytes), or octal base 8 (4 bytes).

Your serial protocol must meet the following criteria:

- Full-duplex is supported (RTS and CTS must be set to high).
- It uses standard HDLC checksums and frames.
- Addresses are contained in constant location (offset) within the frame.
- Addresses are located on a byte boundary.

To create your own protocol, perform the following task in global configuration mode:

Task	Command
Create your own protocol.	**stun schema** *name* **offset** *constant-offset* **length** *address-length* **format** *format-keyword*

Enable STUN Interfaces and Place in STUN Group

You can must enable STUN on serial interfaces and place these interfaces in the protocol groups you have defined.To remove an interface from a group, use the **no stun group** command. To enable STUN on an interface and to place the interface in a STUN group, perform the following tasks in interface configuration mode:

Task	Command
Enable STUN function on a serial interface.	**encapsulation stun**
Place the interface in a previously defined STUN group.	**stun group** *group-number*

Once a given serial link is configured for the STUN function, it is no longer a shared multiprotocol link. All traffic that arrives on the link will be transported to the corresponding peer as determined by the current STUN configuration.

Establish the SDLC Frame Encapsulation Method

To allow SDLC frames to travel across a multimedia, multiprotocol network, you must encapsulate them using one of the following methods:

- TCP encapsulation with SDLC Local Acknowledgment and priority queuing
- TCP encapsulation without Local Acknowledgment
- HDLC encapsulation

Configure TCP Encapsulation with SDLC Local Acknowledgment and Priority Queuing

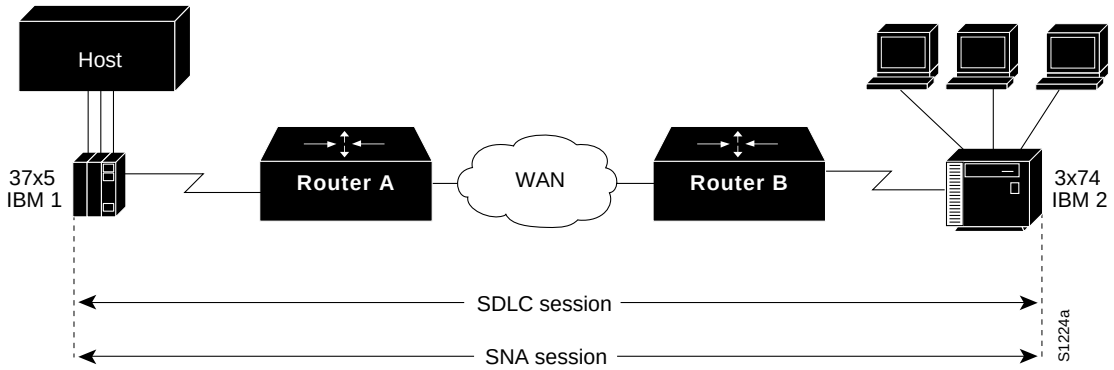
You can configure SDLC Local Acknowledgment using TCP encapsulation. When you configure SDLC Local Acknowledgment, you also have the option of enabling support for priority queuing.

Note To enable SDLC Local Acknowledgment, you must have used the **stun protocol-group** command using the **sdlc** option to create an SDLC protocol group.

SDLC sessions require that end nodes send acknowledgments upon receipt of data frames before allowing further data to be transmitted. SDLC Local Acknowledgment provides local termination of the SDLC session, so that control frames no longer travel the WAN backbone networks. This means that end nodes do not time out, and a loss of sessions does not occur.

Figure 1-1 illustrates an SDLC session. IBM 1, using a serial link, can communicate with IBM 2 on a different serial link separated by a wide area backbone network. Frames are transported between Router A and Router B using STUN. However, the SDLC session between IBM 1 and IBM 2 is still end-to-end. Every frame generated by IBM 1 traverses the backbone network to IBM 2, which, upon receipt of the frame, acknowledges it.

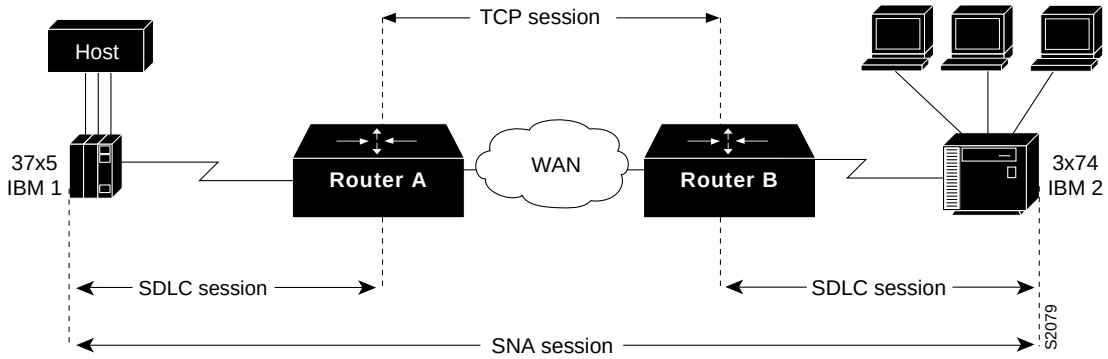
Figure 1-1 SDLC Session without Local Acknowledgment



With SDLC Local Acknowledgment, the SDLC session between the two IBM end nodes is not end-to-end but instead terminates at the two local routers, as shown in Figure 1-2. The SDLC session with IBM 1 ends at Router A, and the SDLC session with IBM 2 ends at Router B. Both Router A and Router B execute the full SDLC protocol as part of SDLC Local Acknowledgment. Router A

acknowledges frames received from IBM 1. The node IBM 1 treats the acknowledgments it receives as if they are from IBM 2. Similarly, Router B acknowledges frames received from IBM 2. The node IBM 2 treats the acknowledgments it receives as if they are from IBM 1.

Figure 1-2 SDLC Session with Local Acknowledgment



Assign the Router an SDLC Primary or Secondary Role

To establish Local Acknowledgment, the router must play the role of an SDLC primary or secondary node. Primary nodes poll secondary nodes in a predetermined order. Secondaries then transmit if they have outgoing data.

In the IBM environment, a Front End Processor (FEP) is the primary station and cluster controllers are secondary stations. If the router is connected to an cluster controller, it should appear as a FEP and must therefore be assigned the role of a primary SDLC node. If the router is connected to a FEP, it should appear as a cluster controller and must therefore be assigned the role of a secondary SDLC node.

To assign the router a primary or secondary role, perform one of the following tasks in interface configuration mode:

Task	Command
Assign the STUN-enabled router an SDLC primary role.	**stun sdlc-role primary**
Assign the STUN-enabled router an SDLC secondary role.	**stun sdlc-role secondary**

Use the **no** form of these commands to remove SDLC role assignments.

Enable the SDLC Local Acknowledgment Feature

To enable SDLC Local Acknowledgment, complete the following task in global configuration mode:

Task	Command
Establish SDLC Local Acknowledgment using TCP encapsulation.	**stun route address** *address-number* **tcp** *ip-address* [**local-ack**] [**priority**]

You can use the **priority** keyword (to set up the four levels of priorities to be used for TCP encapsulated frames) at the same time you enable Local Acknowledgment. The **priority** keyword is described in the following section. Use the **no** form of this command to disable SDLC Local Acknowledgment. For an example of how to enable Local Acknowledgment, see "Example of Configuring Serial Link Address Prioritization using STUN TCP/IP Encapsulation" later in this chapter.

Establish Priority Queuing Levels

With SDLC Local Acknowledgment enabled, you can establish priority levels used in priority queuing for serial interfaces. The priority levels are as follows:

- Low
- Medium
- Normal
- High

To set the priority queuing level, perform the following task in interface configuration mode:

Task	Command
Establish the four levels of priorities to be used in priority queuing.	**stun route address** *address-number* **tcp** *ip-address* [l**ocal-ack**] **priority**

Use the **no** form of this command to disable priority settings. For an example of how to establish priority queuing levels, see "Example of Configuring Serial Link Address Prioritization using STUN TCP/IP Encapsulation" later in this chapter.

Configure TCP Encapsulation without Local Acknowledgment

If you do not want to use SDLC Local Acknowledgment and only need to forward all SDLC frames encapsulated in TCP, complete the following task in interface configuration mode:

Task	Command
Forward all TCP traffic for this IP address.	**stun route all tcp** *ip-address*

Use the **no** form of this command to disable forwarding of all TCP traffic.

Configure HDLC Encapsulation

You can encapsulate SDLC frames using the HDLC protocol. The outgoing serial link still can be used for other kinds of traffic (the frame is not TCP encapsulated). To configure HDLC encapsulation, perform one of the following tasks in interface configuration mode:

Task	Command
Forward all HDLC traffic of the identified interface number.	**stun route all interface serial** *interface-number*
Forward all HDLC traffic on a direct STUN link.	**stun route all interface serial** *interface-number* [**direct**]
Forward HDLC traffic of the identified address.	**stun route address** *address-number* **interface serial** *interface-number*
Forward HDLC traffic of the identified address across a direct STUN link.	**stun route address** *address-number* **interface serial** *interface-number* [**direct**]

Use the **no** forms of these commands to disable HDLC encapsulation.

Set up Traffic Priorities

You can use the following methods to determine the order in which traffic should be handled on the network:

- Assign queuing priorities
- Prioritize STUN traffic over all other traffic

Assign Queuing Priorities

In addition to COS, you can assign queuing priorities by one of the following:

- Serial interface address
- Logical Unit (LU) address

Prioritize by Serial Interface Address

You can prioritize traffic on a per serial interface address basis. You may want to do this so that traffic between one source-destination pair will always be sent before traffic between another source-destination pair.

Note You must first enable Local Acknowledgment and priority levels.

To prioritize traffic, perform the following tasks in global configuration mode:

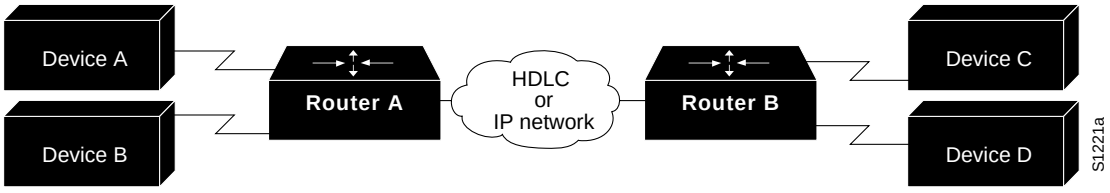
Task	Command
Assign a queuing priority to the address of the STUN serial interface.	**priority-list** *list* **stun** *queue-keyword* **address** *group-number address-number*
Assign a queuing priority to TCP port.	**priority-list** *list* **ip** *queue-keyword* **tcp** *tcp port number*

You must also perform the following task in interface configuration mode:

Task	Command
Assign a priority list to a priority group.	**priority-group** *list*

Figure 1-3 illustrates serial link address prioritization. Device A communicates with Device C, and Device B communicates with Device D. With the serial link address prioritization, you can choose to give A-C a higher priority over B-D across the serial tunnel.

Figure 1-3 Serial Link Address Prioritization



To disable priorities, use the **no** forms of these commands.

For an example of how to prioritize traffic, see "Example of Configuring Serial Link Address Prioritization using STUN TCP/IP Encapsulation" later in this chapter.

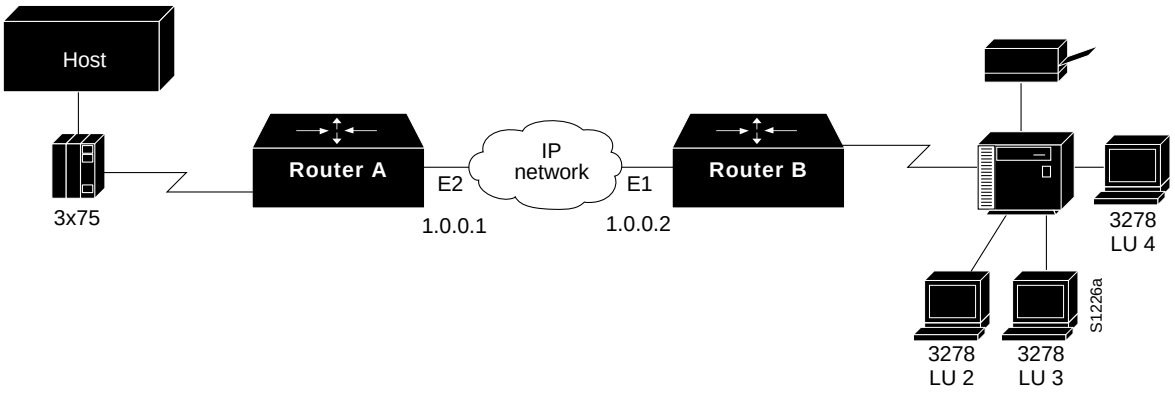
Prioritize by Logical Unit Address

SNA local logical unit (LU) address prioritization is specific to IBM SNA connectivity and is used to prioritize SNA traffic on either Serial Tunnel (STUN) or Remote Source-Route Bridging (RSRB). To set the queuing priority by LU address, perform the following task in interface configuration mode:

Task	Command
Assign a queuing priority based upon logical unit addresses.	**locaddr-priority-list** *list address-number queue-keyword*

In Figure 1-4, Logical Unit Address prioritizing can be set so that particular LUs receive data in preference to others or so that LUs have priority over the printer, for example.

Figure 1-4 SNA Logical Unit (LU) Address Prioritization



To disable this priority, use the **no** form of this command.

For an example of how to prioritize traffic, see "Example of Configuring Serial Link Address Prioritization using STUN TCP/IP Encapsulation" later in this chapter.

Prioritize STUN Traffic over All Other Traffic

You can prioritize STUN traffic to be routed first before all other traffic on the network. To give STUN traffic this priority, perform the following task in global configuration mode:

Task	Command
Prioritize STUN traffic in your network over that of other protocols.	**priority-list** *list* **stun** *high* **address** *group-number address-number*

To disable this priority, use the **no** form of this command.

For an example of how to prioritize STUN traffic over all other traffic, see "Example of Configuring Serial Link Address Prioritization using STUN TCP/IP Encapsulation" later in this chapter.

Configure Proxy Polling

In normal communication between an SDLC primary node and its secondary node, the secondary node is only allowed to send data to the primary node in response to a poll from the primary node. The proxy polling feature alleviates the load across the network by allowing our routers to act as proxies for the primary and secondary nodes, thus keeping polling traffic off of the shared links.

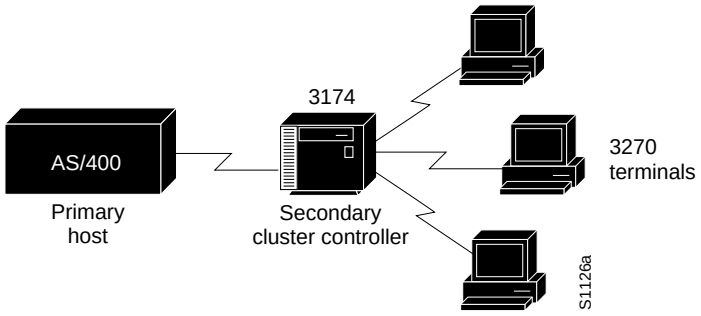
Note The proxy polling feature is provided for compatibility with Software Release 9.0 or earlier.The functions provided by proxy polling have been enhanced and superseded by the SDLC Transport with TCP/IP encapsulation and Local Acknowledgment features. Use of proxy polling is not recommended.

Configuring proxy polling consists of two tasks:

- Enable proxy polling
- Set the proxy polling intervals

In Figure 1-5, an AS/400 host is attached to a 3174 controller that handles transfers from several attached 3270 terminals.

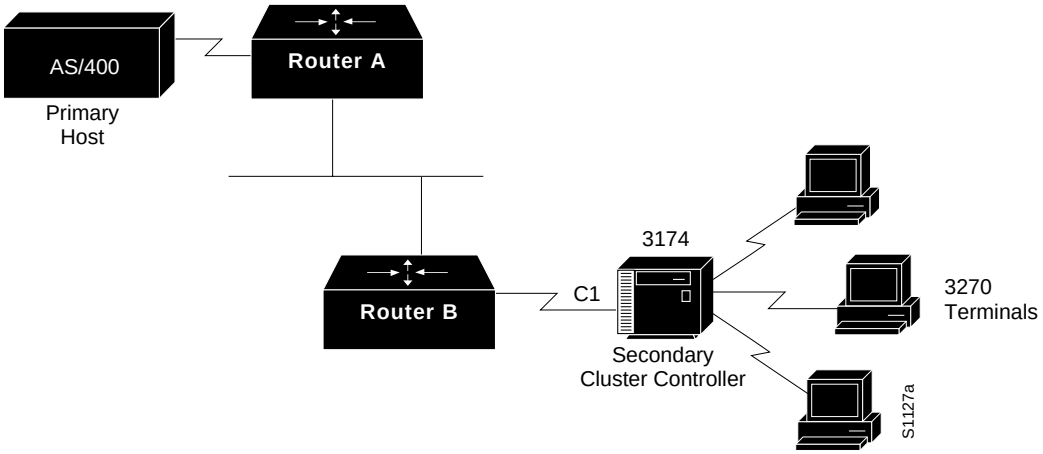
Figure 1-5 IBM SDLC Configuration without Proxy Polling



The 3270-style terminals attached to the 3174 controller cannot initiate a data transfer because the 3174 is the secondary node on the link. The primary host ensures a reasonable response time for its secondary nodes by sending out polls at a rate that is often more than 20 times per second. With two devices sharing a single, dedicated serial line, as in the preceding example, this poses no problem, because the link would be idle without the polls.

In Figure 1-6, if proxy polling is not enabled, the frequent polls and their replies constantly travel between the two routers across the shared Ethernet. Such constant traffic can create bottlenecks and loads that may not be appropriately handled.With proxy polling enabled, Router A can reply to the AS/400 poll requests as a proxy for the secondary node, thereby keeping the polls and requests off of the shared Ethernet. Similarly, Router B can as a proxy for the primary node and periodically send polls to the secondary 3174 device, thereby keeping its replies off of the shared cable. Only significant information is passed across the shared Ethernet.

Figure 1-6 IBM SDLC Configuration with Proxy Polling



Enable Proxy Polling

If you have Software Release 9.0 or earlier, you may want to enable proxy polling to alleviate the load across the network. You can enable proxy polling with or without specifying the router as a primary or secondary SDLC node. It is not necessary for you to specify a primary or secondary SDLC node in cases where such connections are negotiable. If you do not specify the SDLC role, proxying is disabled until session start-up. Until this time, all polls travel through the network.

To enable proxy polling, perform one of the following tasks in interface configuration mode:

Task	Command
Enable proxy polling specifying either a primary or a secondary node.	**stun proxy-poll address** *address* **modulus** *modulus* **{primary | secondary}**
Enable proxy polling without specifying a primary or secondary node.	**stun proxy-poll address** *address* **discovery**

To disable proxy polling, use the **no stun proxy-poll address** command.

For an example of how to enable proxy polling, see "Example of Configuring Proxy Polling for STUN" later in this chapter.

Set the Proxy Polling Intervals

You can set the proxy poll intervals to maximum network performance while ensuring that polling is monitored. You also can set the intervals after which the router will pass through a poll from the primary SDLC device through the network to the secondary device. This action causes the secondary device's reply to also traverse the entire network. This periodic pass-through provides an insurance mechanism that makes sure the primary SDLC device maintains an accurate status of the secondary SDLC device.

You can perform one or both of the following tasks to set one or both of the proxy polling intervals. To set the passthrough interval, perform the following task in global configuration mode:

Task	Command
Set the interval after which the router on the primary side of an SDLC connection passes a poll from the primary SDLC device through the network to the secondary SDLC device.	**stun primary-pass-through** *seconds*

To set the poll interval, perform the following task in interface configuration mode:

Task	Command
Set the interval at which the primary node sends a proxy poll to the secondary node.	**stun poll-interval** *milliseconds*

To return to default polling values, use the **no** forms of these commands.

Monitor STUN Network Activity

You can list statistics regarding STUN interfaces, protocol groups, number of packets sent and received, proxy states, and more. To get activity information, perform one or more of the following tasks in EXEC mode:

Task	Command
List the status display fields for STUN interfaces.	**show stun**
Examine the proxy states on an address basis.	**show stun sdlc**

STUN Configuration Examples

This section provides the following example configurations that you can use as a guide to configuring your STUN environment:

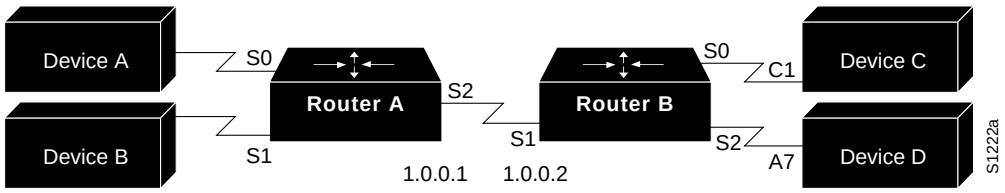
- Example of configuring STUN priorities (page 22-14)
- Example of configuring serial link address prioritization using STUN TCP/IP encapsulation (page 22-15)
- Example of configuring STUN multipoint implementation using a line-sharing device (page 22-18)
- Example of configuring Local Acknowledgment on a per-STUN-peer basis (page 22-20)
- Example of configuring Local Acknowledgment for STUN packets (page 22-20)
- Configuring LOCADDR priority groups—simple example (page 22-21)

Example of Configuring STUN Priorities

The following configurations set the priority of STUN hosts A, B, C, and D.

Assume that the link between Router A and Router B in Figure 1-7 is a serial tunnel that uses the simple serial transport mechanism.

Figure 1-7 STUN Simple Serial Transport



Device A communicates with Device C (SDLC address C1) with a high priority. Device B communicates with Device D (SDLC address A7) with a normal priority. The router configurations follow.

Configuration for Router A

```
stun peer-name 1.0.0.1
stun protocol-group 1 sdlc
stun protocol-group 2 sdlc
!
interface serial 0
no ip address
encapsulation stun
stun group 1
stun route address C1 interface serial 2
!
interface serial 1
no ip address
encapsulation stun
stun group 2
stun route address A7 interface serial 2
!
interface serial 2
ip address 1.0.0.1 255.0.0.0
priority-group 1
!
priority-list 1 stun high address 1 C1
priority-list 1 stun low address 2 A7
!
```

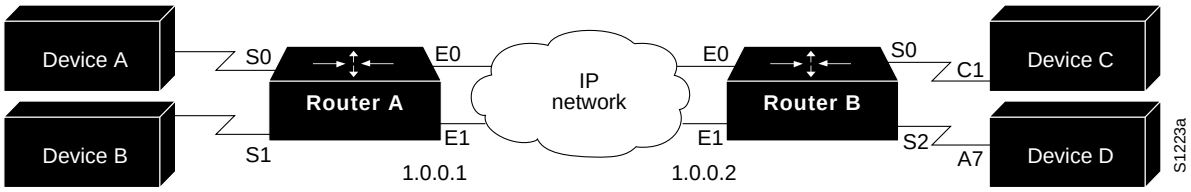
Configuration for Router B

```
stun peer-name 1.0.0.2
stun protocol-group 1 sdlc
stun protocol-group 2 sdlc
!
interface serial 0
no ip address
encapsulation stun
stun group 1
stun route address C1 interface serial 1
!
interface serial 1
ip address 1.0.0.2 255.0.0.0
priority-group 1
!
interface serial 2
no ip address
encapsulation stun
stun group 2
stun route address A7 interface serial 1
!
priority-list 1 stun high address 1 C1
priority-list 1 stun low address 2 A7
!
```

Example of Configuring Serial Link Address Prioritization Using STUN TCP/IP Encapsulation

Assume the that link between Router A and Router B is a serial tunnel that uses the TCP/IP encapsulation as shown in Figure 1-8. Device A communicates with Device C (SDLC address C1) with a high priority. Device B communicates with Device D (SDLC address A7) with a normal priority.

Figure 1-8 STUN TCP/IP Encapsulation



The configuration of each router follows.

Configuration for Router A

```
stun peer-name 1.0.0.1
stun protocol-group 1 sdlc
stun protocol-group 2 sdlc
!
interface serial 0
no ip address
encapsulation stun
stun group 1
stun route address C1 tcp 1.0.0.2 local-ack priority
priority-group 1
!
interface serial 1
no ip address
encapsulation stun
stun group 2
stun route address A7 tcp 1.0.0.2 local-ack priority
priority-group 2
!
interface ethernet 0
ip address 1.0.0.1 255.0.0.0
!
interface ethernet 1
ip address 1.0.0.3 255.0.0.0
!
priority-list 1 protocol ip high tcp 1994
priority-list 1 protocol ip medium tcp 1990
priority-list 1 protocol ip normal tcp 1991
priority-list 1 protocol ip low tcp 1992
priority-list 1 stun high address 1 C1
!
priority-list 2 protocol ip high tcp 1994
priority-list 2 protocol ip medium tcp 1990
priority-list 2 protocol ip normal tcp 1991
priority-list 2 protocol ip low tcp 1992
priority-list 2 stun normal address 2 A7
!
hostname routerA
router igrp
network 1.0.0.0
```

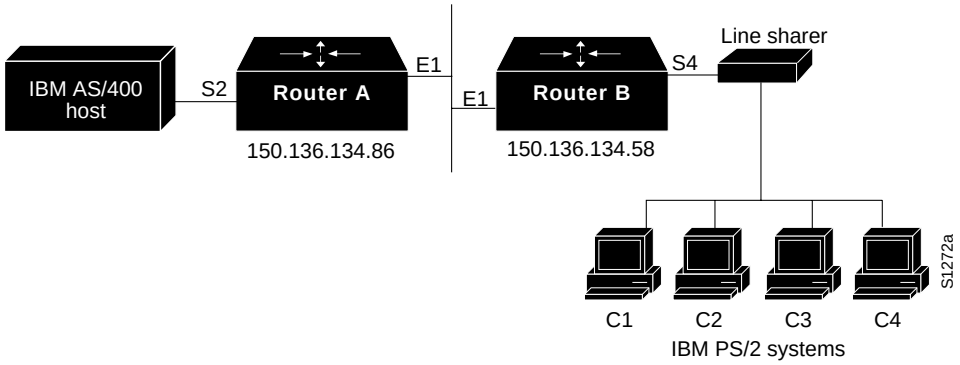
Configuration for Router B

```
stun peer-name 1.0.0.2
stun protocol-group 1 sdlc
stun protocol-group 2 sdlc
!
interface serial 0
no ip address
encapsulation stun
stun group 1
stun route address C1 tcp 1.0.0.1 local-ack priority
priority-group 1
!
interface serial 2
no ip address
encapsulation stun
stun group 2
stun route address A7 tcp 1.0.0.1 local-ack priority
priority-group 2
!
interface ethernet 0
ip address 1.0.0.2 255.0.0.0
!
interface ethernet 1
ip address 1.0.0.4 255.0.0.0
!
priority-list 1 protocol ip high tcp 1994
priority-list 1 protocol ip medium tcp 1990
priority-list 1 protocol ip normal tcp 1991
priority-list 1 protocol ip low tcp 1992
priority-list 1 stun high address 1 C1
!
priority-list 2 protocol ip high tcp 1994
priority-list 2 protocol ip medium tcp 1990
priority-list 2 protocol ip normal tcp 1991
priority-list 2 protocol ip low tcp 1992
priority-list 2 stun normal address 2 A7
!
hostname routerB
router igrp 109
network 1.0.0.0
```

Example of Configuring STUN Multipoint Implementation Using a Line-Sharing Device

In Figure 1-9, four separate PS/2 computers are connected to a line-sharing device off of Router B. Each PS/2 computer has four sessions open on an AS/400 device attached to Router A. Router B functions as the primary station, while Router A functions as the secondary station. Both routers locally acknowledge packets from the IBM PS/2 systems.

Figure 1-9 STUN Communication Involving a Line-Sharing Device



Configuration for Router A

```
! enter the address of the stun peer
stun peer-name 150.136.134.86
! specify that group 4 uses the SDLC protocol
stun protocol-group 4 sdlc

interface ethernet 1
! enter the IP address for the Ethernet interface
ip address 150.136.134.86 255.255.255.0
!
! description of IBM AS/400 link
interface serial 2
! description of IBM AS/400 link; disable the IP address on a serial interface
no ip address
! enable STUN encapsulation on this interface
encapsulation stun
! apply previously defined stun group 4 to serial interface 2
stun group 4
! establish this router as a secondary station
stun sdlc-role secondary
! wait up to 63000 msec for a poll from the primary before timing out
sdlc poll-wait-timeout 63000
! list addresses of secondary stations (PS/2 systems) attached to link
sdlc address C1
sdlc address C2
sdlc address C3
sdlc address C4
! use tcp encapsulation to send frames to SDLC stations C1, C2, C3, or
! C4 and locally terminate sessions with these stations
stun route address C1 tcp 150.136.134.58 local-ack
stun route address C2 tcp 150.136.134.58 local-ack
stun route address C3 tcp 150.136.134.58 local-ack
stun route address C4 tcp 150.136.134.58 local-ack
```

Configuration for Router B

```
! enter the address of the stun peer
stun peer-name 150.136.134.58
! this router is part of SDLC group 4
stun protocol-group 4 sdlc
!
interface ethernet 1
! enter the IP address for the Ethernet interface
ip address 150.136.134.58 255.255.255.0
!
! description of PS/2 link
interface serial 4
! disable the IP address on a serial interface
no ip address
! enable STUN encapsulation on this interface
encapsulation stun
! apply previously defined stun group 4 to serial interface 2
stun group 4
! establish this router as a primary station
stun sdlc-role primary
! send output to a secondary station for up to 2000 milliseconds
! through interface serial 4 before polling for input must begin
sdlc fair-poll-timer 1500
! wait 2000 milliseconds for a reply to a frame before resending it
sdlc t1 2000
! resend a frame up to four times if not acknowledged
sdlc n2 4
! list addresses of secondary stations (PS/2 systems) attached to link
sdlc address C1
sdlc address C2
sdlc address C3
sdlc address C4
! use tcp encapsulation to send frames to SDLC stations C1, C2, C3, or
! C4 and locally terminate sessions with these stations
stun route address C3 tcp 150.136.134.86 local-ack
stun route address C1 tcp 150.136.134.86 local-ack
stun route address C4 tcp 150.136.134.86 local-ack
stun route address C2 tcp 150.136.134.86 local-ack
! set the clockrate on this interface to 9600 bits per second
clockrate 9600
```

Example of Configuring Local Acknowledgment on a Per-STUN-Peer Basis

The following example shows a sample configuration for a pair of routers performing SDLC Local Acknowledgment:

Configuration for Router A

```
!
stun peer-name 150.136.64.92
stun protocol-group 1 sdlc
!
interface Serial 0
no ip address
encapsulation stun
stun group 1
stun sdlc-role secondary
sdlc address C1
stun route address C1 tcp 150.136.64.93 local-ack
clockrate 19200
!
```

Configuration for Router B

```
!
stun peer-name 150.136.64.93
stun protocol-group 1 sdlc
!
interface Serial 0
no ip address
encapsulation stun
stun group 1
stun sdlc-role primary
sdlc address C1
stun route address C1 tcp 150.136.64.92 local-ack
clockrate 19200
!
```

Example of Configuring Local Acknowledgment for STUN Packets

This example shows the configuration for a router set up to provide local acknowledgment for STUN packets:

```
! sample global command
stun peer-name 150.136.64.92
! sample protocol-group command
stun protocol-group 1 sdlc
!
interface serial 0
no ip address
encapsulation stun
stun group 1
stun sdlc-role secondary
sdlc address C1
! provide local acknowledgment for SDLC packets destined for a
! STUN peer at address C1
stun route address C1 tcp 150.136.64.93 local-ack
clockrate 19200
!
```

Configuring LOCADDR Priority Groups—Simple Example

This example shows how to establish queuing priorities on a STUN interface based on an LU address:

```
! sample stun peer-name global command
stun peer-name 131.108.254.6
! sample protocol-group command for reference
stun protocol-group 1 sdlc
!
interface serial 0
! disable the ip address for interface serial 0
no ip address
! enable the interface for STUN
encapsulation stun
! sample stun group command
stun group 2
! sample stun route command
stun route address 10 tcp 131.108.254.8 local-ack priority
!
! assign priority group 1 to the input side of interface serial 0
locaddr-priority 1
priority-group 1
interface Ethernet 0
! give locaddr-priority-list 1 a high priority for LU 02
locaddr-priority-list 1 02 high
! give locaddr-priority-list 1 a low priority for LU 05
locaddr-priority-list 1 05 low
```

Example of Configuring LOCADDR Priority Groups for STUN

The following configuration example shows how to assign a priority group to an input interface.

Configuration for Router A

```
stun peer-name 1.0.0.1
stun protocol-group 1 sdlc
!
interface serial 0
no ip address
encapsulation stun
stun group 1
stun route address C1 tcp 1.0.0.2 local-ack priority
clockrate 19200
locaddr-priority 1
priority-group 1
!
interface Ethernet 0
ip address 1.0.0.1 255.255.255.0
!
locaddr-priority-list 1 02 high
locaddr-priority-list 1 03 high
locaddr-priority-list 1 04 medium
locaddr-priority-list 1 05 low
!
!
priority-list 1 protocol ip high tcp 1994
priority-list 1 protocol ip medium tcp 1990
priority-list 1 protocol ip normal tcp 1991
priority-list 1 protocol ip low tcp 1992
```

Configuration for Router B

```
stun peer-name 1.0.0.2
stun protocol-group 1 sdlc
!
interface serial 0
no ip address
encapsulation stun
stun group 1
stun route address C1 tcp 1.0.0.1 local-ack priority
clockrate 19200
locaddr-priority 1
priority-group 1
!
interface Ethernet 0
ip address 1.0.0.2 255.255.255.0
!
locaddr-priority-list 1 02 high
locaddr-priority-list 1 03 high
locaddr-priority-list 1 04 medium
locaddr-priority-list 1 05 low
!
priority-list 1 protocol ip high tcp 1994
priority-list 1 protocol ip medium tcp 1990
priority-list 1 protocol ip normal tcp 1991
priority-list 1 protocol ip low tcp 1992
```

Example of Configuring Transmission Groups

The following example shows sample configurations in two routers performing Transmission Group (TG) support (for a two-link TG):

Configuration for Router A

```
! sample stun peer-name global command
stun peer-name 131.108.254.6
! this router is part of protocol group 3 and an sdlc transmission group (TG)
stun protocol-group 3 sdlc-tg
!
interface serial 0
! sample ip address command
no ip address
! sample encapsulation stun subcommand
encapsulation stun
! place interface serial0 in previously defined STUN group 3
stun group 3
! establish the sdlc address as A7
sdlc address A7
! provide stun route subcommand
stun route address A7 tcp 131.108.254.7 local-ack
!
! must enter the next line with local acknowledgment to include
! the router in a transmission group
stun route address A7 tcp 131.108.254.7 local ack
```

Configuration for Router B

```
! sample stun peer-name global command
stun peer-name 131.108.254.7
! this router is part of protocol group 3 and an sdlc transmission group (TG)
stun protocol-group 3 sdlc-tg
!
interface serial 3
! sample ip address subcommand
no ip address
! sample encapsulation stun subcommand
encapsulation stun
! place interface serial0 in previously defined STUN group 3
stun group 3
! establish the sdlc address as A7
sdlc address A7
! provide stun route subcommand
stun route address A7 tcp 150.136.64.92 local-ack
!
! must enter the next line with local acknowledgment to include
! the router in a transmission group
stun route address A7 tcp 150.136.64.92.2 local ack
```

Example of Configuring Proxy Polling for STUN

The following example enables proxy polling for a secondary device at address C1 on interface serial 0 running with modulus 8.

```
stun peer-name 1.0.0.1
stun protocol-group 1 sdlc
!
interface Serial 0
no ip address
encapsulation stun
stun group 3
stun sdlc-role secondary
stun proxy-poll address C1 modulus 8 secondary
sdlc address C1
stun route address C1 tcp 150.136.64.93 local-ack
clockrate 19200
!
```