CHAPTER 23

Configuring STUN

Cisco's serial tunnel (STUN) implementation allows Synchronous Data Link Control (SDLC) devices and High-Level Data Link Control (HDLC) devices to connect to one another through a multiprotocol internetwork.

This chapter describes the STUN features and lists the tasks you must perform to configure a STUN network in either passthrough or local acknowledgment mode. For a complete description of the commands mentioned in this chapter, refer to the "STUN Commands" chapter in the *Router Products Command Reference* publication.

Note The use of Software Release 9.0 or earlier is discouraged. If you have Software Release 9.0 or earlier, you can enable STUN in passthrough mode only. In lieu of local acknowledgment, the proxy-polling feature is available. However, the functions provided by proxy polling have been enhanced and superseded by the STUN local acknowledgment feature and the use of proxy polling is no longer supported.

Cisco's Implementation of Serial Tunneling

Our STUN implementation provides the following features:

- Encapsulates SDLC frames in either the Transmission Control Protocol/Internet Protocol (TCP/IP) or the HDLC protocol.
- Allows two devices using SDLC- or HDLC-compliant protocols that are normally connected by a direct serial link to be connected through one or more Cisco routers, reducing leased-line costs.

  When you replace direct serial links with routers, serial frames can be propagated over arbitrary media and topologies to another router with a STUN link to an appropriate end point. The intervening network is not restricted to STUN traffic, but rather, is multiprotocol. For example, instead of running parallel backbones for DECnet and SNA/SDLC traffic, this traffic now can be integrated into an enterprise backbone network.
- Allows networks with IBM mainframes and communications controllers to share data using Cisco routers and existing network links. As an SDLC function, STUN fully supports the IBM Systems Network Architecture (SNA), and allows IBM SDLC frames to be transmitted across the network media and and/or shared serial links. Figure 23-1 illustrates a typical network configuration with and without STUN.

- Encapsulates SDLC frame traffic packets and routes them over any of the supported network media—serial, Fiber Distributed Data Interface (FDDI), Ethernet, and Token Ring, X.25, Switched Multimegabit Data Service (SMDS), and T1/T3—using TCP/IP encapsulation. Because TCP/IP encapsulation is used, you can use any of the Cisco routing protocols to route the packets.
- Copies frames to destinations based on address. STUN in passthrough mode does not modify the frames in any way or participate in SDLC windowing or retransmission; these functions are left to the communicating hosts. However, STUN in local acknowledgment mode does participate in SDLC windowing and retransmission through local termination of the SDLC session.
- Ensures reliable data transmission across serial media having minimal or predictable time delays. With the advent of STUN and wide-area network (WAN) backbones, serial links now can be separated by wide, geographic distances spanning countries and continents. As a result, these serial links have time delays that are longer than SDLC allows for bidirectional communication between hosts. The STUN local acknowledgment feature addresses the problems of unpredictable time delays, multiple retransmissions, or loss of sessions.
- Provides for configuration of redundant links to provide transport paths in the event part of the network goes down.

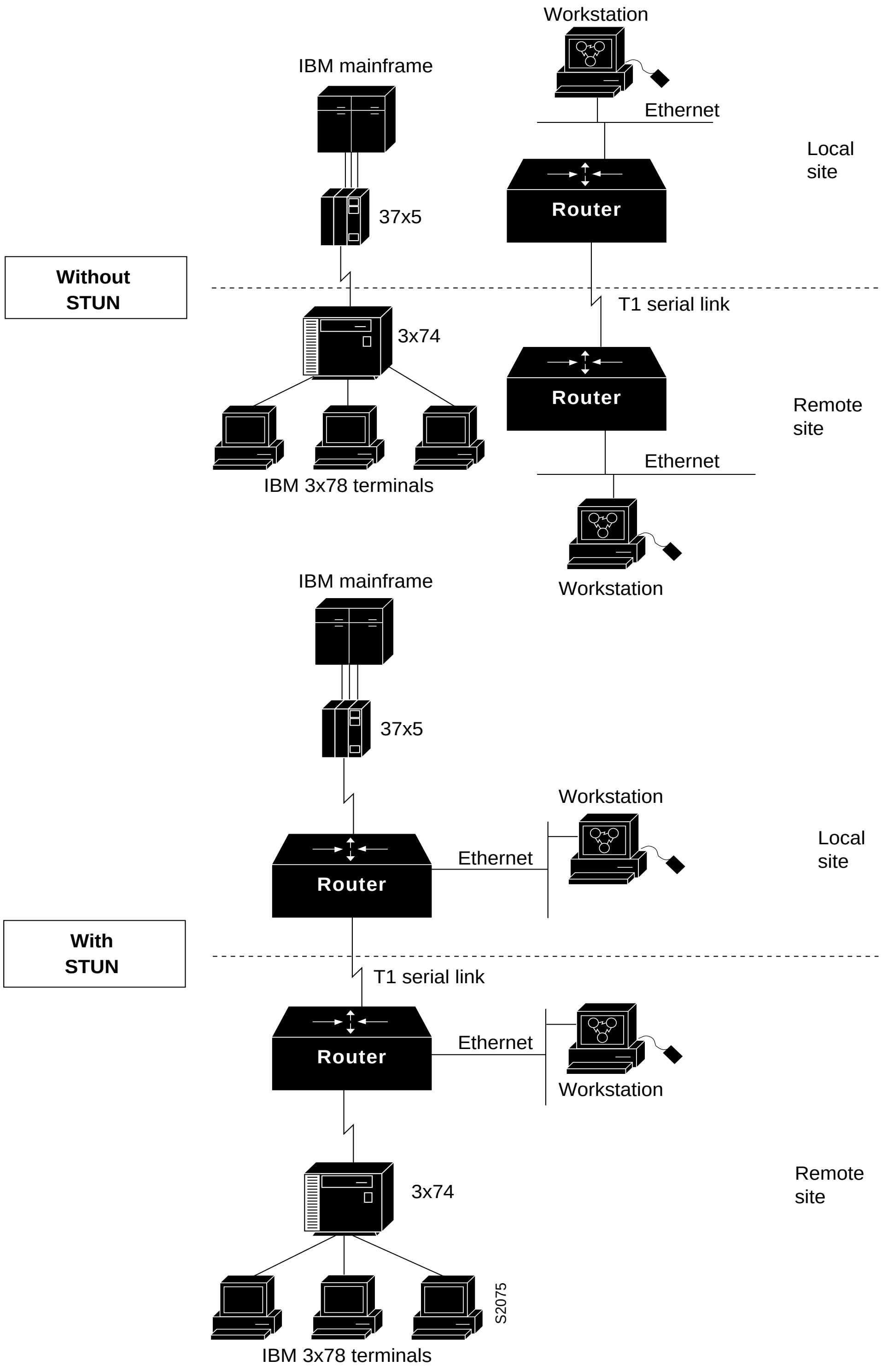

Figure 23-1 IBM Network Configuration with and without STUN

The STUN Network

STUN operates in two modes: passthrough and local acknowledgment. Figure 23-2 shows the difference between passthrough mode and local acknowledgment mode.

The upper half of Figure 23-2 shows STUN configured in passthrough mode. In passthrough mode, the routers act as a wire and the SDLC session remains between the end stations. In this mode, STUN provides a straight pass-through of all SDLC traffic, including control frames.

The lower half of Figure 23-2 shows STUN configured in local acknowledgment mode. In local acknowledgment mode, the routers terminate the SDLC sessions and send only data across the WAN. Control frames no longer travel the WAN backbone networks.

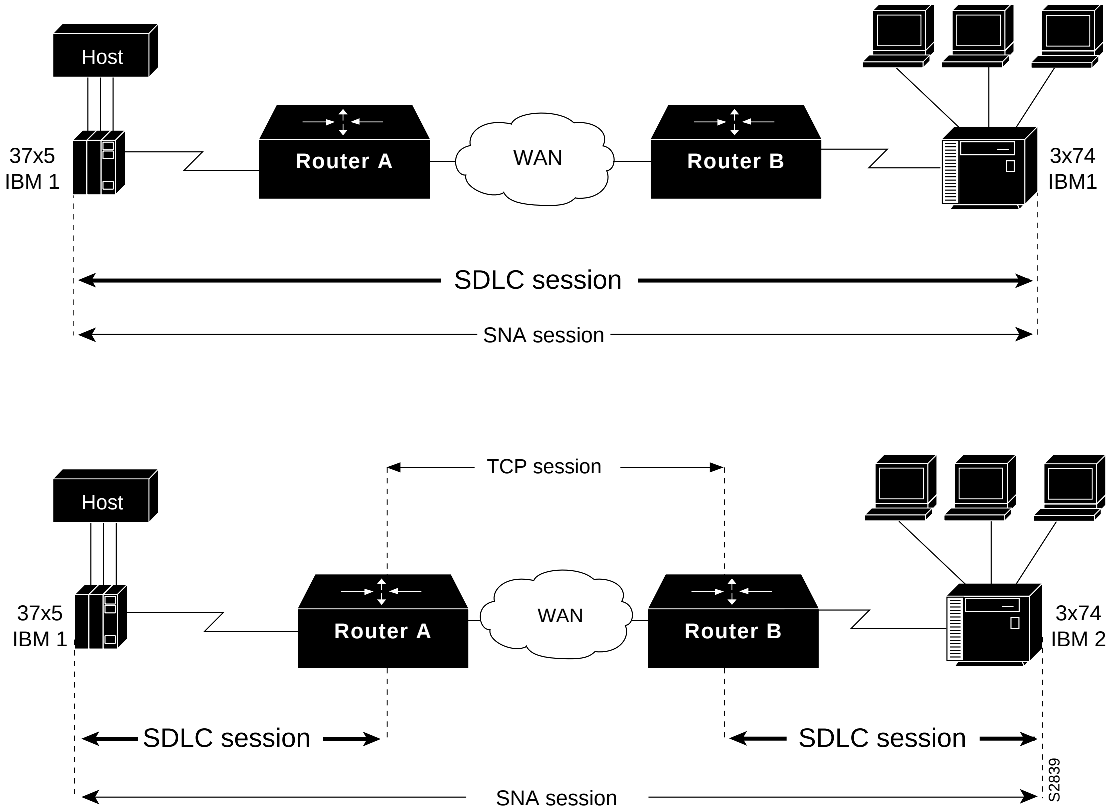

Figure 23-2 Comparison of STUN in Passthrough Mode and Local Acknowledgment Mode

Note To enable STUN 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 STUN local acknowledgment feature also provides priority queuing for TCP-encapsulated frames.

STUN Configuration Task List

To configure and monitor STUN, or STUN Local Acknowledgment, complete the tasks in the following sections:

- Enable STUN
- Configure SDLC Broadcast
- Specify a STUN Protocol Group
- Enable STUN Interfaces and Place in STUN Group
- Establish the Frame Encapsulation Method
- Configure STUN with Multilink Transmission Groups
- Set up Traffic Priorities
- Monitor STUN Network Activity

See the end of the chapter for configuration examples.

Enable STUN

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

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

When configuring redundant links, ensure that the STUN peer names you choose on each router are the IP addresses of the most stable interfaces on each router, such as a loopback or Ethernet interface. See "STUN Configuration Examples" later in this chapter.

Configure SDLC Broadcast

The SDLC Broadcast feature allows SDLC broadcast address FF to be replicated for each of the STUN peers, so each of the end stations receives the broadcast frame. For example, in Figure 23-3, the FEP views the end stations 1, 2 and 3 as if they are on an SDLC multidrop link. Any broadcast frame sent from FEP to Router A is duplicated and sent to each of the downstream routers (B and C).

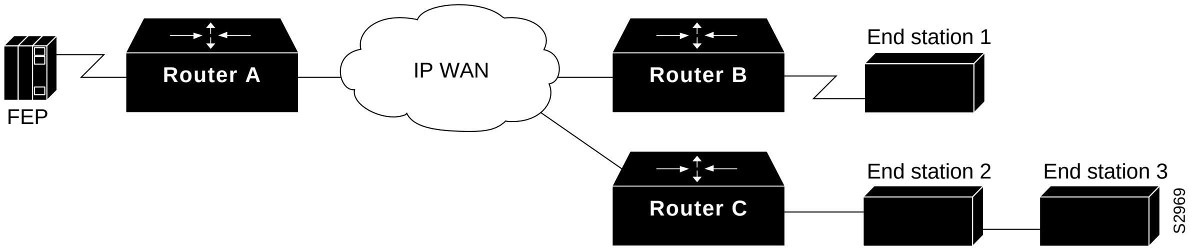

Figure 23-3 SDLC Broadcast across Virtual Multidrop Lines

To enable SDLC Broadcast, perform the following task in interface configuration mode:

Task	Command
Enable SDLC Broadcast.	**sdlc virtual-multidrop**

Enable SDLC broadcast only on the router that is configured to be the secondary station on the SDLC link (Router A in Figure 23-3).

You must also configure SDLC address FF on Router A for each of the STUN peers. To do so, perform the following task in global configuration mode:

Task	Command
Configure SDLC address FF on Router A for each STUN peer.	**stun route address** *address-number* **tcp** *ip-address* **[local-ack] [priority] [tcp-queue-max]**

Specify a STUN Protocol Group

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.

There are three predefined STUN protocols:

- Basic
- SDLC
- SDLC transmission group

You also can specify a custom STUN protocol.

You must specify either the SDLC protocol or the SDLC transmission group protocol if you want to use the STUN Local Acknowledgment feature.

Note Before you can specify a custom protocol, you must first define the protocol; see the section "Create and Specify a Custom STUN Protocol" later in this chapter for the procedure.

Specify a Basic STUN Group

The basic STUN protocol is unconcerned with details of serial protocol addressing and is used when addressing is unimportant. Use this when your goal with STUN is to replace one or more sets of point-to-point (not multidrop) serial links by using a protocol other than SDLC. Perform the following task in global configuration mode:

Task	Command
Specify a basic protocol group and assign a group number.	**stun protocol-group** *group-number* **basic**

Specify an SDLC Group

You can specify SDLC protocol groups to associate interfaces with the SDLC protocol. The SDLC STUN protocol is used for placing the routers in the midst of either point-to-point or multipoint (multidrop) SDLC links. To define an SDLC protocol group, perform the following task in global configuration mode:

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

If you specify an SDLC protocol group, you cannot specify the **stun route all** command on any interface of that group.

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.

Specify an SDLC Transmission Group

An SNA transmission group is a set of lines providing parallel links to the same pair of SNA front-end-processor (FEP) devices. This provides redundancy of paths for fault tolerance and load sharing. To define an SDLC transmission group, perform the following task in global configuration mode:

Task	Command
Specify an SDLC protocol group, assign a group number, and create an SNA transmission group.	**stun protocol-group** *group-number* **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 a transmission group, see “Example of Configuring Transmission Groups” later in this chapter.

Create and Specify a Custom STUN Protocol

The STUN implementation allows you to create your own STUN protocols. To define a custom protocol and tie STUN groups to the new protocol, perform the following tasks in global configuration mode:

Task		Command
Step 1	Create a custom protocol.	**stun schema** *name* **offset** *constant-offset* **length** *address-length* **format** *format-keyword*
Step 2	Specify the custom protocol group and assign a group number.	**stun protocol-group** *group-number* **schema**

Enable STUN Interfaces and Place in STUN Group

You must enable STUN on serial interfaces and place these interfaces in the protocol groups you have defined. 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
Step 1	Enable STUN function on a serial interface.	**encapsulation stun**
Step 2	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 Frame Encapsulation Method

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

- HDLC encapsulation without local acknlwledgment
- TCP encapsulation without local acknowledgment
- TCP encapsulation with SDLC local acknowledgment and optional priority queuing

Configure HDLC Encapsulation

You can encapsulate SDLC or HDLC 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 global configuration mode:

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

Note You can only forward all traffic if you are using basic STUN protocol groups.

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 tasks:

Task		Command
Step 1	Enter interface configuration mode.	See Table 2-1.

Task		Command
Step 2	Forward all TCP traffic for this IP address.	**stun route all tcp** *ip-address*
Step 3	Enter global configuration mode.	See Table 2-1.
Step 4	Specify TCP encapsulation.	**stun route address** *address-number* **tcp** *ip-address* [**local-ack**] [**priority**] [**tcp-queue-max**]

Use the **no** form of these commands to disable forwarding of all TCP traffic.

This configuration is typically used when the two routers may be connected via an IP network as opposed to a point-to-point link. Otherwise, HDLC should always be used.

Configure TCP Encapsulation with SDLC Local Acknowledgment and Priority Queuing

You can only 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 specified an SDLC or SDLC transmission group.

SDLC local acknowledgment provides local termination of the SDLC session so that control frames no longer travel the WAN backbone networks. This means that time-outs are less likely to occur.

Figure 23-4 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.

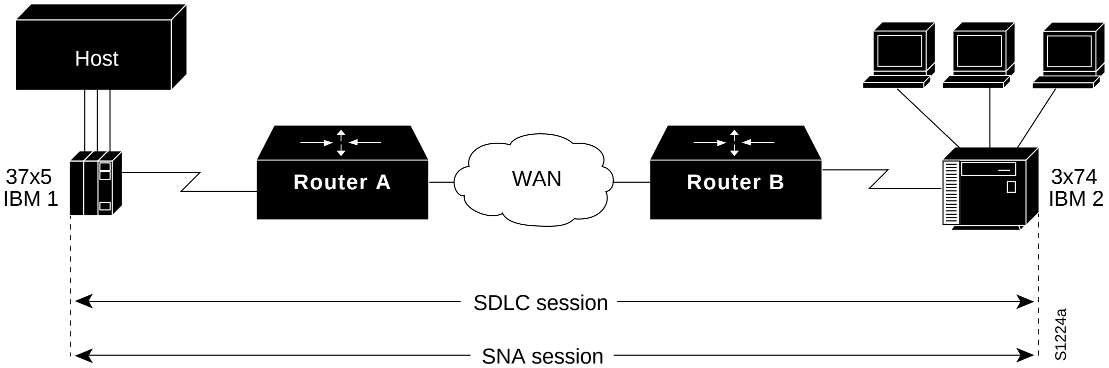

Figure 23-4 SDLC Session without Local Acknowledgment

With SDLC local acknowledgment, the SDLC session between the two end nodes is not end-to-end but instead terminates at the two local routers, as shown in Figure 23-5. 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.

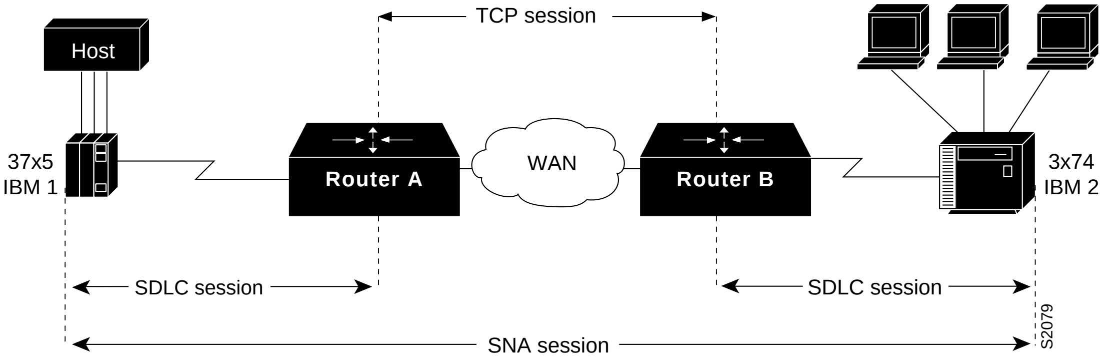

Figure 23-5 SDLC Session with Local Acknowledgment

To configure TCP encapsulation with SDLC local acknowledgment and priority queuing, perform the following tasks:

- Assign the router an SDLC primary or secondary role.
- Enable the SDLC local acknowledgment feature.
- Establish priority queuing levels.

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.

For example, in the IBM environment, an FEP is the primary station and cluster controllers are secondary stations. If the router is connected to a cluster controller, it should appear as an FEP and must therefore be assigned the role of a primary SDLC node. If the router is connected to an FEP, it should appear as a cluster controller and must therefore be assigned the role of a secondary SDLC node. Routers connected to SDLC primary end-stations must play the role of an SDLC secondary and routers attached to SDLC secondary end stations must play the role of an SDLC primary station.

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**] [**tcp-queue-max**]

The **stun route address 1 tcp local-ack priority tcp-queue-max** interface configuration command enables local acknowledgment and TCP encapsulation. Both these options are required to use transmission groups . You should specify the SDLC address with the echo bit turned off for transmission group interfaces. The SDLC broadcast address 0xFF is routed automatically for transmission group interfaces. The **priority** keyword creates multiple TCP sessions for this route. The **tcp-queue-max** keyword sets the maximum size of the outbound TCP queue for the SDLC. The default TCP queue size is 100. The value for **hold-queue in** should be greater than the value for **tcp-queue-max**.

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 “Configuring Serial Link Address Prioritization Using STUN TCP/IP Encapsulation Example” 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 [tcp-queue-max]**

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

Configure STUN with Multilink Transmission Groups

You can configure multilink SDLC transmission groups across STUN connections between IBM communications controllers such as IBM 37x5s. Multilink transmission group allow you to collapse multiple WAN leased lines into one leased line.

SDLC multilink transmission groups provide the following features:

- Network control program (NCP) SDLC address allowances, including echo and broadcast addressing.
- Remote NCP load sequence. After a SIM/RIM exchange but before a SNRM/UA exchange, NCPs send numbered I-frames. During this period, I-frames are not locally acknowledged but instead are passed through. After the SNRM/UA exchange, local acknowledgment occurs.
- Rerouting of I-frames sent from the router to the NCP if a link is lost in a multilink transmission group.
- Class of service based on transmission group sequencing is performed in the router, which preserves the class of service specified by the sending NCP.
- Flow control rate tuning causes a sending NCP to "feel" WAN congestion and hold frames that would otherwise be held in the router waiting to be transmitted on the WAN. This allows the NCP to perform its class-of-service algorithm more efficiently based on a greater knowledge of network congestion.

STUN connections that are part of a transmission group must have local acknowledgment enabled. Local acknowledgment keeps SDLC poll traffic off the WAN and reduces store-and-forward delays through the router. It also might minimize the number of NCP timers that expire due to network delay. Also, these STUN connections must go to the same IP address. This is because SNA transmission groups are parallel links between the same pair of IBM communications controllers.

Design Recommendations

This section provides some recommendations that are useful in configuring SDLC multilink transmission groups.

The bandwidth of the WAN should be larger than or equal to the aggregate bandwidth of all serial lines to avoid excessive flow control and ensure no degradation in response time. If other protocols also are using the WAN, ensure that the WAN bandwidth is significantly greater than the aggregate SNA serial line bandwidth to ensure that the SNA traffic does not monopolize the WAN.

When you are using a combination of routed transmission groups and directly connected NCP transmission groups, you need to plan the configuration carefully to ensure that SNA sessions do not stop unexpectedly. Assuming that hardware reliability is not an issue, from a software point of view, single-link routed transmission group are as reliable as direct NCP-to-NCP single-link transmission groups.This is true because neither the NCP nor the router can reroute I-frames when a transmission group has only one link. Additionally, multilink transmission group directed between NCPs and multilink transmission group through router are equally reliable. Both can perform rerouting.

However, you might run into problems if you have a configuration in which two NCPs are directly connected (via one or more transmission group links) and one link in the transmission group is routed. The NCPs will view this as a multilink transmission group. However, the router views the transmission group as a single-link transmission group. A problem can arise in the following situation: Assume that an I-frame is being transmitted from NCP A (connected to router A) to NCP B (connected to router B) and that all SDLC links are currently active. Router A will acknowledge the I-frame sent from NCP A and will send it over the WAN. If, before the I-frame reaches router B, the SDLC link between router B and NCP B goes down, router B will attempt to reroute the I-frame on another link in the transmission group when it receives the I-frame. However, because this is a single-link transmission group, there are no other routes, and router B drops the I-frame. NCP B will never receive this I-frame because router A acknowledged its receipt, and NCP A marked it as transmitted and deleted it. NCP B detects a gap in the transmission group sequence numbers and waits to receive the missing I-frame. It will wait forever for this I-frame, and in the meantime will not send or receive any other frames. This means that NCP B is technically inoperational and that all SNA sessions through NCP B will be lost.

One final design recommendation note concerns a configuration in which one or more lines of an NCP transmission group are router and one or more lines are directly connected between NCPs. If the network delay associated with one line of an NCP transmission group is different from the delay of another line in the same NCP transmission group, the receiving NCP will spend additional time resequencing PIUs.

Set up Traffic Priorities

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

- Enable Class of Service
- Assign Queuing Priorities
- Prioritize STUN Traffic over All Other Traffic

Enable Class of Service

SNA class of service allows prioritization of FEP (NCP) traffic. To enable the class-of-service feature, perform the following task in global configuration mode:

Task	Command
Force the router to read the Format Identification 4 (FID4) frame to prioritize traffic.	**stun cos-enable**

When used in conjunction with the **priority** keyword on the **stun route** command, the **stun cos-enable** command causes SNA network priority traffic to flow at a higher priority than other SNA data traffic. You can only use the class-of-service feature with SDLC transmission group traffic.

To disable class of service, use the **no stun cos-enable** command.

Note Before completing this task, you must first establish local acknowledgment and priority levels.

Assign Queuing Priorities

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

- Serial interface address or TCP port
- Logical unit (LU) address

Prioritize by Serial Interface Address or TCP Port

You can prioritize traffic on a per-serial-interface address or TCP port basis. You might 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 one of the following tasks in global configuration mode:

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

Figure 23-6 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.

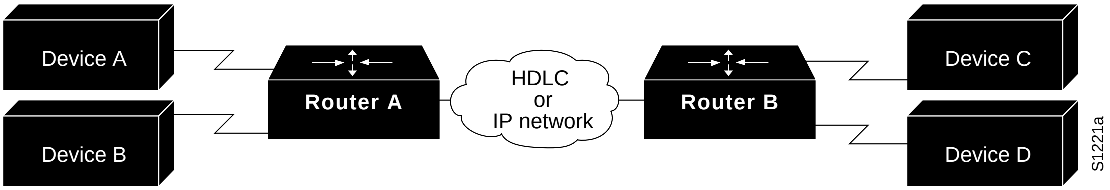

Figure 23-6 Serial Link Address Prioritization

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

For an example of how to prioritize traffic, see “Configuring Serial Link Address Prioritization Using STUN TCP/IP Encapsulation Example” 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 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-number address-number queue-keyword*

In Figure 23-7, LU address prioritization can be set so that particular LUs receive data in preference to others or so that LUs have priority over the printer, for example.

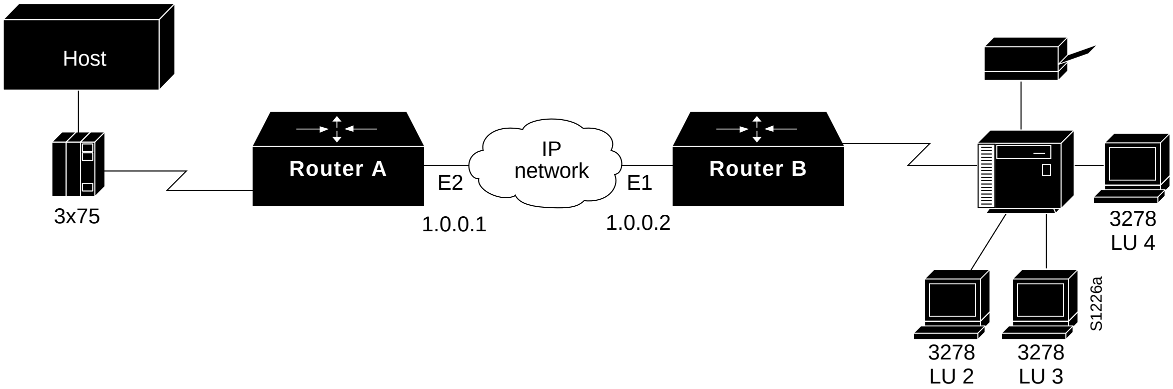

Figure 23-7 SNA LU Address Prioritization

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

For an example of how to prioritize traffic, see “Configuring Serial Link Address Prioritization Using STUN TCP/IP Encapsulation Example” 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-number* **stun** *queue-keyword* **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 “Configuring Serial Link Address Prioritization Using STUN TCP/IP Encapsulation Example” later in this chapter.

Monitor STUN Network Activity

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

Task	Command
List the status display fields for STUN interfaces.	**show stun**

STUN Configuration Examples

The following sections provide STUN configuration examples:

- SDLC Broadcast Configuration Example

- Configuring STUN Priorities Using HDLC Encapsulation Example
- Configuring Serial Link Address Prioritization Using STUN TCP/IP Encapsulation Example
- Configuring STUN Multipoint Implementation Using a Line-Sharing Device Example
- Configuring STUN Local Acknowledgment Example
- Configuring LOCADDR Priority Groups—Simple Example
- Configuring LOCADDR Priority Groups for STUN Example
- Configuring Transmission Groups Example

SDLC Broadcast Configuration Example

In the following example, an FEP views end stations 1, 2, and 3 as if they were on an SDLC multidrop link. Any broadcast frame sent from the FEP to Router A is duplicated and sent to each of the downstream routers (B and C):

```
stun peer-name xxx.xxx.xxx.xxx
stun protocol-group 1 sdlc
interface serial 1
encapsulation stun
stun group 1
stun sdlc-role secondary
sdlc virtual-multidrop
sdlc address 1
sdlc address 2
sdlc address 3
stun route address 1 tcp yyy.yyy.yyy.yyy local-ack
stun route address 2 tcp zzz.zzz.zzz.zzz local-ack
stun route address 3 tcp zzz.zzz.zzz.zzz local-ack
stun route address FF tcp yyy.yyy.yyy.yyy
stun route address FF tcp zzz.zzz.zzz.zzz
```

Configuring STUN Priorities Using HDLC Encapsulation Example

Assume that the link between Router A and Router B in Figure 23-8 is a serial tunnel that uses the simple serial transport mechanism. 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.

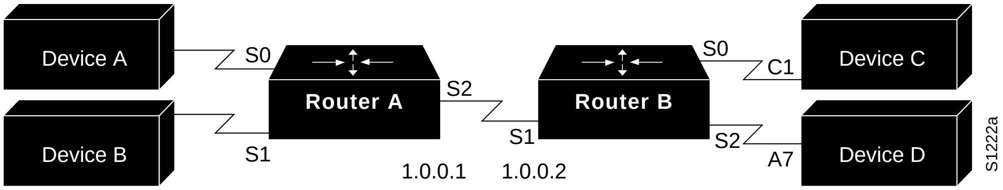

Figure 23-8 STUN Simple Serial Transport

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

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
```

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

Assume that the link between Router A and Router B is a serial tunnel that uses the TCP/IP encapsulation as shown in Figure 23-9. 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.

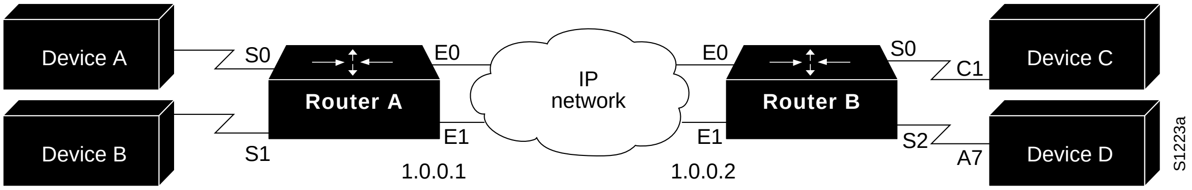

Figure 23-9 STUN TCP/IP Encapsulation

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
```

Configuring STUN Multipoint Implementation Using a Line-Sharing Device Example

In Figure 23-10, 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.

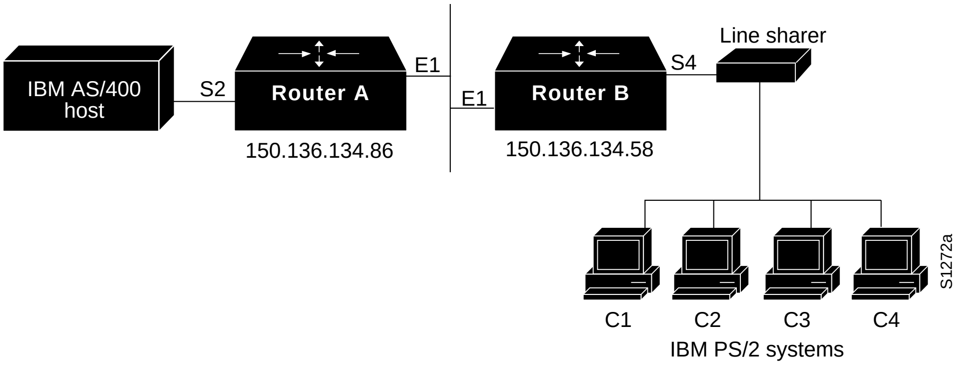

Figure 23-10 STUN Communication Involving a Line-Sharing Device

The configuration file for the routers shown in Figure 23-10 follows.

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
stun remote-peer-keepalive

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
stun remote-peer-keepalive
!
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
sdlc line-speed 9600
! 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
```

Configuring STUN Local Acknowledgment Example

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
stun remote-peer-keepalive
!
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
stun remote-peer-keepalive
!
interface Serial 0
no ip address
encapsulation stun
stun group 1
stun sdlc-role primary
sdlc line-speed 19200
sdlc address C1
stun route address C1 tcp 150.136.64.92 local-ack
clockrate 19200
```

Configuring LOCADDR Priority Groups—Simple Example

The following 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
```

Configuring LOCADDR Priority Groups for STUN Example

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
```

Configuring Transmission Groups Example

Figure 23-11 shows two routers that support a double-link transmission group. Router A is the SDLC primary router, and Router B is the SDLC secondary router. The IBM 1 37x5 is acting as the SDLC secondary 37x5, and the IBM 2 37x5 is acting as the SDLC primary 37x5.

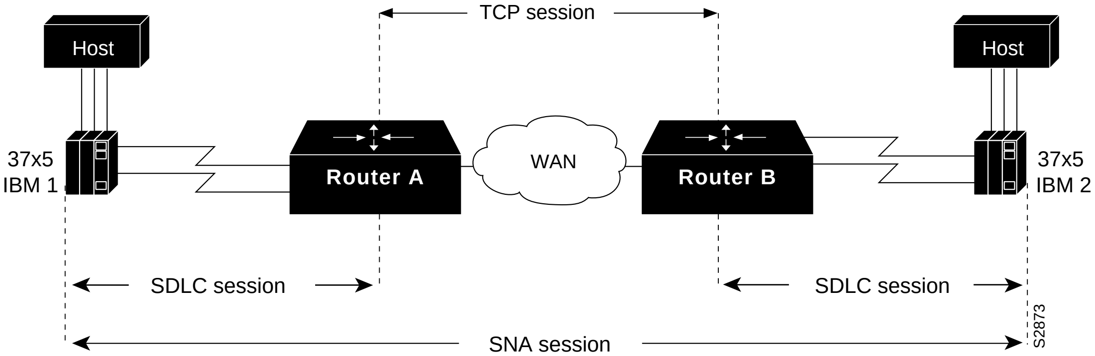

Figure 23-11 Double-Link Transmission Group

The configuration files for the two routers shown in Figure 23-11 follow.

Router A (SDLC Primary Router)

```
stun peer-name 150.136.112.13
stun remote-peer-keepalive
stun protocol-group 91 sdlc-tg
stun cos-enable
!
interface serial 1
mtu 4400
hold-queue 150 in
no ip address
encapsulation stun
stun group 91
stun sdlc-role primary
sdlc line-speed 56000
sdlc address 01 echo
stun route address 1 tcp 150.136.112.10 local-ack priority tcp-queue-max 75
!
interface serial 2
mtu 4400
hold-queue 150 in
no ip address
encapsulation stun
stun group 91
stun sdlc-role primary
sdlc line-speed 56000
sdlc address 02 echo
stun route address 2 tcp 150.136.112.10 local-ack priority tcp-queue-max 75
```

Router B (SDLC Secondary Router)

```
stun peer-name 150.136.112.10
stun remote-peer-keepalive
stun protocol-group 91 sdlc-tg
stun cos-enable

interface serial 1
mtu 4400
hold-queue 150 in
no ip address
encapsulation stun
stun group 91
stun sdlc-role secondary
sdlc address 01 echo
stun route address 1 tcp 150.136.112.13 local-ack priority tcp-queue-max 75

interface serial 2
mtu 4400
hold-queue 150 in
no ip address
encapsulation stun
stun group 91
stun sdlc-role secondary
sdlc address 02 echo
stun route address 2 tcp 150.136.112.13 local-ack priority tcp-queue-max 75
```