CHAPTER 11

Configuring X.25 and LAPB

This chapter describes how to configure connections through X.25 networks, including Link Access Procedure, Balanced (LAPB) connections. LAPB procedures are presented first for those users who only want to configure a simple, reliable serial encapsulation method. For a complete description of the commands mentioned in this chapter, refer to the "X.25 and LAPB Commands" chapter in the *Router Products Command Reference* publication. For historical background and a technical overview of X.25 and LAPB, see the *Internetworking Technology Overview* publication. To translate between X.25 and another protocol, refer to the *Protocol Translator Configuration Guide and Command Reference.*

Cisco's Implementation of LAPB and X.25

A group of specifications published by the International Telecommunication Union Telecommunication Standardization Sector (ITU-T) defines *Recommendations*, or specifications, for X.25. The ITU-T specifications basically specify connections between data terminal equipment (DTE) and data communications equipment (DCE) for remote terminal access and computer communications. The X.25 specifications include LAPB as the data link layer protocol and X.25 as the network layer protocol, or packet layer protocol (PLP) as it is also known. The ITU-T updates its specifications every four years, and the specifications dated 1980 and 1984 are the most common versions currently in use. Additionally, the International Standards Organization (ISO) has published ISO 7776:1986 as an equivalent to the LAPB standard, and ISO 8208:1989 as an equivalent to the ITU-T 1984 X.25 Recommendation packet layer. Our X.25 software follows the ITU-T 1984 X.25 Recommendation, except for its Defense Data Network (DDN) and Blacker Front End (BFE) operation, which follow the ITU-T 1980 X.25 Recommendation.

Note The ITU-T carries out the functions of the former Consultative Committee for International Telegraph and Telephone (CCITT).

In addition to providing remote terminal access, our X.25 software provides transport for LAN protocols—IP, DECnet, XNS, ISO CLNS, AppleTalk, Novell IPX, Banyan VINES, and Apollo Domain—and bridging. For information about these protocols, refer to the specific protocol chapters in the *Router Products Configuration Guide*.

Briefly, the Cisco Systems X.25 software provides the following capabilities:

- LAPB datagram transport—LAPB is a protocol that operates at Level 2 (the data link layer) of the OSI reference model. It offers a reliable connection service for exchanging data (in units called frames). Protocol datagrams (IP, DECnet, AppleTalk, and so forth) are carried over a reliable LAPB connection, or datagrams of several of these protocols are encapsulated in a proprietary protocol and carried over a LAPB connection. The LAPB connection is configured to carry a single protocol or multiple protocols.
- X.25 datagram transport—Protocol datagrams (IP, DECnet, AppleTalk, and so forth) are encapsulated inside packets on an X.25 virtual circuit. Mappings between X.25 addresses and protocol addresses allow these datagrams to be routed through an X.25 network, thereby allowing an X.25 public data network (PDN) to transport LAN protocols.
- X.25 switch—X.25 calls can be routed based on their X.25 addresses either between serial interfaces on the same router (local switching) or across an IP network to another router (remote switching, also called *tunneling*). Remote X.25 switching encapsulates the X.25 packet level inside a TCP connection, allowing X.25 equipment to be connected via a TCP/IP-based network.
- Connection-Mode Network Service (CMNS)—CMNS is a mechanism through which local X.25 switching can be extended to different media (Ethernet, FDDI, and Token Ring) by using OSI-based NSAP addresses. This implementation provides X.25 over LLC2 to allow X.25 over media other than serial interfaces. Cisco's CMNS implementation supports services defined in ISO Standards 8208 (packet level) and 8802-2 (frame level).
- DDN and BFE X.25—DDN-specified Standard Service is supported. The DDN X.25 Standard Service is the required protocol for use with DDN Packet-Switched Nodes (PSNs). The Defense Communications Agency (DCA) has certified Cisco Systems' DDN X.25 Standard Service implementation for attachment to the Defense Data Network. Cisco's DDN implementation also includes Blacker Front End and Blacker Emergency Mode operation.
- X.25 MIB—Specified in SNMP MIB Extension for X.25 LAPB (RFC 1381) and SNMP MIB Extension for the X.25 Packet Layer (RFC 1382) subsets are supported. The LAPB XID Table X.25 Cleared Circuit Table, and X.25 Call Parameter Table are not implemented. All values are read-only. To use the X.25 MIB, refer to the *Cisco Management Information Base (MIB) User Quick Reference,* or the RFCs.
- X.25 support—Our software supports all mandatory requirements of 1988 X.25.

Our X.25 implementation does not support fast switching.

Facility Handling in Encapsulated X.25 Virtual Circuits

The router either originates or accepts encapsulation virtual circuits (SVCs) in order to transport LAN traffic through an X.25 network.

When the router originates a CALL for LAN traffic encapsulation, the facilities in the CALL are controlled by the facilities configured for the interface and the map statement that specifies the LAN/X.25 encapsulation. Because a router can be attached to a Public Data Network (PDN), the interface and map configurations allow a number of facilities to be specified in outgoing CALLs. These facilities are specified in all originated CALLs relating to the given interface and map with one exception; the incoming and outgoing maximum packet sizes proposed will be lowered if the lower layer (LAPB) cannot support the specified DATA packet size.

When the router accepts an encapsulation CALL, many facilities are simply ignored. The maximum packet sizes will be lowered if the lower layer (LAPB) cannot support the sizes proposed. A reverse-charged CALL will be CLEARed if neither the interface nor the map allows it. A CALL that specifies a Network User Identification (NUID) will be CLEARed if the user authentication fails.

Facility Handling in Routed X.25 Virtual Circuits

Routed X.25 traffic might have facilities added, deleted, or modified by the router.

Standard (1984 X.25) Facilities

The following subsections describe how standard (1984 X.25) facilities are treated when routing a switched virtual circuit (SVC).

Flow Control Negotiation (Window Sizes and Maximum Packet Sizes)

Our routed X.25 implementation requires that the flow control parameters be negotiated end-to-end and actively intervenes in the negotiation process using the CALL/CALL ACCEPTED packet exchange to ensure this. This behavior is important because routed X.25 traffic is simply passed between the two interfaces (local or remote) without fragmentation, reassembly, or local flow control handling. As part of the flow control negotiation process, the following activities take place:

- The router will insert flow control parameters into the switched CALL if the requested values do not match the outgoing interface's defaults.
- The router will strip flow control parameters from the switched CALL if the requested values match the outgoing interface's defaults.
- Any requested maximum packet size that exceeds the capability of either interface will be lowered to the largest value that can be supported.
- Any requested window size that exceeds 7 will be lowered to 7 if a CALL is routed from a modulo 128 interface to a modulo 8 interface.
- A remotely routed CALL will have both the window sizes and packet sizes inserted into the CALL packet sent over the TCP connection. These values will be stripped from the outgoing CALL if they match the default values of the outgoing interface. Note that some earlier Cisco X.25 implementations did not insert the flow control parameters into the packet. Because X.25 cannot determine what the incoming interface's default values are, it will indicate the universally acceptable window and packet size values of 2/2 and 128/128 on the CALL ACCEPTED packet. The **x25 routing** flag **USE-TCP-IF-DEFS** can be configured to use the outgoing interface's default values as an aid to software migration.
- An outgoing CALL ACCEPTED packet will indicate the accepted flow control values if, for any reason, they are different from the values proposed for the incoming CALL.

Throughput Negotiation

An incoming Throughput facility will be forwarded.

Closed User Group Selection

A basic format Closed User Group selection facility will be forwarded; any other format of Closed User Group selection (extended format, CUG with outgoing access or Bilateral CUG) will be stripped.

Reverse Charging

An incoming Reverse Charging facility will be forwarded.

Fast Select

An incoming Fast Select facility will be forwarded.

Network User Identification (NUID)

An incoming NUID facility on a CALL packet will be forwarded; an NUID facility on a CALL ACCEPTED packet will be stripped.

Charging Information

Any Charging Information or Request will be stripped.

RPOA Selection

Any RPOA Selection will be stripped.

Called Line Address Modified Notification

A Called Line Address Modified notification will be forwarded.

Call Redirection Notification

A Call Redirection notification will be stripped.

Transit Delay Selection

An incoming Transit Delay facility will be forwarded.

ITU-T-Specified Marker Facilities

The following subsections describe how ITU-T-specified marker facilities are treated when routing an SVC.

Calling Address Extension

An incoming Calling Address Extension facility will be forwarded.

Called Address Extension

An incoming Called Address Extension facility will be forwarded.

Quality of Service Negotiation

Any of the Quality of Service facilities will be stripped.

Expedited Data Negotiation

An Expedited Data Negotiation facility will be stripped.

Local Marker Facilities Specified for DDN or BFE X.25

The following subsections describe how local marker facilities are treated when routing an SVC.

DDN Service Type

An incoming DDN Service Type facility will be stripped from a CALL, but inserted if a forwarded CALL ACCEPTED packet specifies a DDN precedence facility.

DDN Precedence

An incoming DDN Precedence facility will be forwarded through the router. However, both the input and output interfaces need to be configured for DDN X.25 encapsulation.

BFE Emergency Mode Addressing

An incoming BFE Emergency Mode Addressing facility will be stripped.

LAPB Configuration Task List

It is possible to use only LAPB as a serial encapsulation method. This can be done using a leased serial line. You must use one of the X.25 packet-level encapsulations when attaching to an X.25 network.

The LAPB standards distinguish between two types of hosts: data terminal equipment (DTE), and data circuit-terminating equipment (DCE). At Level 2, or the data link layer in the OSI model, LAPB allows for orderly and reliable exchange of data between a DTE and a DCE. A router using LAPB encapsulation can act as a DTE or DCE device at the protocol level, which is distinct from the hardware DTE or DCE level.

Using LAPB under noisy conditions can result in greater throughput than HDLC encapsulation. When LAPB detects a missing frame, the router retransmits the frame instead of waiting for the higher layers to recover the lost information. This behavior is good only if the host timers are relatively slow. In the case of quickly expiring host timers, however, you will discover that LAPB is spending much of its time transmitting host retransmissions. If the line is not noisy, the lower overhead of HDLC encapsulation is more efficient than LAPB. When using long delay satellite links, for example, the lock-step behavior of LAPB makes HDLC encapsulation the better choice.

To configure LAPB, complete the tasks in the following sections. The tasks in the first section are required; the remaining are optional.:

- Establish LAPB DCE or DTE Operation
- Set the LAPB Retransmission Timer and Frame Parameters
- Define the LAPB Hold Queue Size
- Monitor and Maintain LAPB and X.25

Examples of LAPB configurations are given at the end of this chapter.

Establish LAPB DCE or DTE Operation

Set the appropriate LAPB encapsulation to run datagrams over a serial interface. One end of the link must be DTE and the other must be DCE.

Task	Command
Specify a serial interface.	**interface serial** *unit*[1]

[1] This command is documented in the "Interface Commands" chapter in the *Router Products Command Reference* publication.

To select an encapsulation and the protocol if using a single protocol, or to select the multiple protocol operation, perform one or more of the following tasks in interface configuration mode:

Task	Command
Enable encapsulation of a single protocol on the line using DCE operation.	**encapsulation lapb-dce**
Enable encapsulation of a single protocol on the line using DTE operation.	**encapsulation lapb**
Set the line protocol.	**lapb protocol** *protocol*
Enable use of multiple protocols on the line using DCE operation.	**encapsulation multi-lapb-dce**
Enable use of multiple protocols on the line using DTE operation.	**encapsulation multi-lapb**[1]

[1] Multi-LAPB does not support bridging (SRB and transparent) or TCP header compression. Only protocols supported by LAPB encapsulation are supported by multiprotocol LAPB encapsulation.

For an example of configuring LAPB DCE operation, see the section "Typical LAPB Configuration Example" later in this chapter.

Set the LAPB Retransmission Timer and Frame Parameters

X.25 Level 2 or LAPB operates at the data link layer of the OSI reference model. LAPB specifies methods for exchanging data (in units called *frames*), detecting out-of-sequence or missing frames, retransmitting frames, and acknowledging frames.

When connecting to an X.25 network, use the N1 parameter value set by the network administrator. This value is the maximum number of bits in a LAPB frame, which determines the maximum size of an X.25 packet. When using LAPB over leased lines, the N1 parameter should be eight times the hardware maximum transmission unit (MTU) size. The other frame parameters are determined by the network configuration; see Table 11-1 for the default values.

The retransmission timer determines how long a transmitted frame can remain unacknowledged before the router polls for an acknowledgment. For X.25 networks, the router retransmission timer setting should match that of the network.

For leased-line circuits, the retransmission timer setting is critical. The timer setting must be large enough to permit a maximum-sized frame to complete one round trip on the link. If the timer setting is too small, the router will poll before the acknowledgment frame can return, which may result in duplicated frames and severe protocol problems; the design of LAPB assumes that a frame has been lost if it is not acknowledged within period T1. If the timer setting is too large, the router waits longer than necessary before requesting an acknowledgment, which reduces bandwidth.

Table 11-1 summarizes the LAPB parameters you can set in interface configuration mode.

Table 11-1 LAPB Parameters

Task (LAPB Parameter)	Ranges	Default	Command
Set the window size (k).	1–7 packets	7	**lapb k** *window-size*
Set maximum bits per frame (N1).	1088–32840 bits	12056	**lapb n1** *bits*
Set count for sending frames (N2).	1–255 tries	20	**lapb n2** *tries*
Set the retransmission timer (T1).	1–64000 milliseconds	3000	**lapb t1** *milliseconds*

For an example of configuring the LAPB T1 timer, see the section "Typical LAPB Configuration Example" later in this chapter.

Define the LAPB Hold Queue Size

To define the maximum number of frames to be held while the other side is not accepting data, perform the following task in interface configuration mode:

Task	Command
Define the size of a frame hold queue.	**lapb hold-queue** *queue-size*

X.25 Configuration Task List

To configure X.25, complete the tasks in one or more of the following sections, depending upon the X.25 application or task required for your network:

- Configure an X.25 Datagram Transport
- Configure X.25 Routing
- Configure CMNS Routing
- Configure DDN or BFE X.25
- Configure X.25 Level 3 Parameters and Special Features
- Monitor and Maintain LAPB and X.25

Note that all X.25 applications except DDN and BFE X.25 operation and routing require that an encapsulation method and an X.121 address be set; see the sections "Set X.25 DTE or DCE Operation" and "Set the X.25 Interface Address" for the tasks to set them.

Default parameters have been provided for X.25 operation; however, you can change the settings to meet the needs of your X.25 network or as defined by your X.25 service supplier. We also provide special and custom configuration settings to further optimize your X.25 network. See the section "Configure X.25 Level 3 Parameters and Special Features" later in this chapter for more information.

See the end of this chapter for examples of configuring X.25.

Configure an X.25 Datagram Transport

X.25 support is most commonly configured as a transport for datagrams across an X.25 network. This is accomplished by first establishing a mapping between protocol addresses (for example, IP or DECnet) and the X.121 addresses of the X.25 network. When datagrams for a particular destination are routed for the first time, a virtual circuit is set up to the appropriate X.121 address. The Call User Data portion of the initial Call Request identifies the protocol of the datagrams being carried by a particular virtual circuit. If multiple protocols are in use, multiple virtual circuits will be opened.

Figure 11-1 illustrates two routers sending data across an X.25 public data network (PDN).

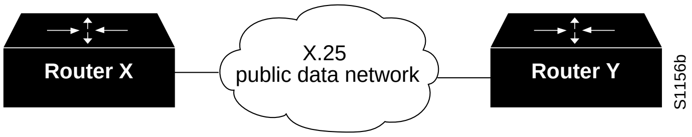


Figure 11-1 Transporting LAN Protocols across an X.25 PDN

You must perform the tasks in the following sections to configure your router as a transport for X.25 datagrams. Note that attachments to a PDN and X.25 datagram transport applications (except DDN and BFE operation) should set an interface address.

- Set X.25 DTE or DCE Operation
- Set the X.25 Interface Address

Perform the tasks in the following sections, as necessary, to complete the X.25 configuration for your network needs:

- Map Protocol Addresses to X.121 Addresses
- Establish a PVC
- Establish a Default VC Protocol
- Configure Subinterfaces

The following sections describe how to perform these configuration tasks. Configuring the X.25 parameters and special features, including TCP header compression and X.25 bridging, are described in the section “Configure X.25 Level 3 Parameters and Special Features” later in this chapter.

Set X.25 DTE or DCE Operation

A router using X.25 Level 3 encapsulation can act as a DTE or DCE protocol device. Default serial encapsulation is HDLC. To configure one of these encapsulation types, according to the needs of your X.25 service supplier, perform the following task in interface configuration mode:

Task	Command
Set X.25 DTE operation.	**encapsulation x25**
Set X.25 DCE operation.	**encapsulation x25-dce**

Typically a public data network will require attachment as a DTE. (This is distinct from the hardware interface DTE and DCE assignments.)

For an example of configuring X.25 DTE operation, see the section “Typical X.25 Configuration Example” later in this chapter.

Set the X.25 Interface Address

The X.121 address is assigned by the X.25 network service provider. To set the X.121 address, perform the following task in interface configuration mode:

Task	Command
Set the X.121 address.	**x25 address** *x.121-address*

For an example of configuring the X.25 interface address, see the section "Typical X.25 Configuration Example" later in this chapter.

Map Protocol Addresses to X.121 Addresses

The routers can use several methods to map protocol addresses to X.121 addresses used by X.25 as follows.

A router set up for DDN or BFE service uses a dynamic mapping technique to convert between IP and X.121 addresses. These techniques have been designed specifically for attachment to the DDN network and to Blacker encryption equipment. Their design, restrictions, and operation make them work well for these specific applications, but not for other networks.

You can establish the X.121 address of a particular network interface using the **x25 address** interface subcommand. A DDN or BFE interface will have an X.121 address generated from the interface IP address, which for proper operation should not be modified.

You can also set up explicit protocol-to-X.121 address mapping for a host. Because no defined protocol can dynamically determine such mappings, you must enter a mapping for each host with which the router will exchange traffic.

To establish a map, perform the following task in interface configuration mode:

Task	Command
Establish a protocol-to-X.121 address map.	**x25 map** *protocol-keyword protocol-address x.121-address* [*options*]

As an example, if you are running IP over X.25, you must define an IP address for the X.25 interface as well as map any destination host IP address to their X.121 address. IP addressing tasks are described in the "Configuring IP" chapter in the *Router Products Configuration Guide*.

When setting up the address map, you can include features such as enabling broadcasts and number of virtual circuits allowed as well as the user facility settings as part of the options. The following are the supported protocols and keywords for map definitions.

- IP—**ip**
- DECnet—**decnet**
- XNS—**xns**
- Novell IPX—**ipx**
- AppleTalk—**appletalk**
- VINES—**vines**
- Apollo Domain—**apollo**
- Bridging—**bridge**
- OSI Connectionless Network Service—**clns**

- OSI Connection-Mode Network Service—**cmns**
- TCP header compression—**compressedtcp**

For specific information about how to establish a protocol to run over X.25, refer to the appropriate protocol chapters in this publication.

The configuration for the Open Shortest Path First (OSPF) protocol can be greatly simplified by adding the optional **broadcast** keyword when doing this task. See the **x25-map** command description in the "X.25 and LAPB Commands" chapter of the *Router Products Command Reference* publication for more information.

Establish a PVC

Permanent virtual circuits (PVCs) are the X.25 equivalent of leased lines; they are never disconnected. You must specify the required network protocol-to-X.121 address map before you set up a PVC.

To establish a PVC, perform the following task in interface configuration mode:

Task	Command
Set a PVC.	**x25 pvc** (**encapsulating**) *circuit protocol-keyword protocol-address* [*option*]

See the previous section for a list of protocols; all but CMNS are supported by this task. For an example of configuring a PVC, see the section "Using a PVC to Exchange IP Traffic Example" later in this chapter.

Establish a Default VC Protocol

The Call Request packet that sets up a virtual circuit (VC) might contain a field called the Call User Data (CUD) field. Typically, the software uses the first byte of Call User Data to distinguish which high-level protocol will be carried by a particular virtual circuit.

Table 11-2 lists the hexadecimal values of the Call User Data and its corresponding network level protocol. The use of 0x81 for ISO 8473 (CLNS) and 0x01 for PAD are ISO standards. The use of 0xCC for IP is defined by RFC 877, and is recognized by ISO. The use of C0 00 80 C4 is defined by Banyan. The other values are meaningful only to our X.25 software. Most of the single-byte identifiers are padded to four bytes with three bytes of 0x00. BFE IP encapsulation requires that only one byte be used. CLNS may use one or five bytes as defined in ISO 8473.

Table 11-2 Protocols and Initial Byte of Call User Data

Protocol	Initial CUD Byte
PAD	0x01
ISO CLNS	0x81
Banyan VINES	0xC0 00 80 C4
DOD IP	0xCC
DECnet	0xD0
XNS	0xD1
AppleTalk	0xD2
Novell IPX	0xD3

Protocol	Initial CUD Byte
Apollo Domain	0xD4
Bridging	0xD5
TCP Header Compression (THC)	0xD8

You can specify that calls with unknown CUD or with no CUD are to be interpreted as IP or PAD calls by performing the following task in interface configuration mode:

Task	Command
Establish the default protocol.	**x25 default** *protocol*

Configure Subinterfaces

X.25 networks provide multiple point-to-point virtual circuits, either permanent (PVCs) or dynamic (SVCs), through a single physical serial interface. If all VCs are presented on the same interface, routing protocols that use split-horizon techniques to propagate updates are not able to route between the VCs. You can define subinterfaces to assign one ore more VC s to separate "virtual" interfaces. This allows routing protocols to route between VCs on separate subinterfaces.

As always, you can use any of the X.25 or LAPB configuration commands on a primary interface. Use the **x25 map** and **x25 pvc** encapsulation commands on secondary interfaces. (For more information about configuring subinterfaces, refer to the "Configuring Interfaces" chapter in the *Router Products Configuration Guide*.)

Note When configuring IP routing over X.25, you might need to make adjustments to accommodate split horizon effects. Refer to the "Configuring IP Routing Protocols" chapter in the *Router Products Configuration Guide* for details about how the router handles possible split horizon conflicts. By default, split horizon is enabled for X.25 networks.

Configure X.25 Routing

The X.25 software implementation allows virtual circuits to be routed from one X.25 interface to another and from one router to another. The routing behavior can be controlled with switching and tunneling commands, based on a locally built table.

Encapsulated LAN protocols can share an X.25 serial interface with the X.25 switching support. Switching or forwarding X.25 virtual circuits can be done two ways:

- Incoming calls received from a local serial interface running X.25 can be forwarded to another local serial interface running X.25. This is known as *local X.25 switching* because the router handles the complete path. It does not matter whether the interfaces are configured as DTE or DCE devices, because the software will take the appropriate actions.
- An incoming call also can be forwarded to another of our routers over a LAN using the TCP/IP protocols. Upon receipt of an incoming call, a TCP stream connection is established to the router that is acting as the switch for the destination. All X.25 packets are sent and received over this reliable data stream. Flow control is maintained end-to-end. This is known as *remote X.25 switching*, or *tunneling.* It does not matter whether the interfaces are configured as DTE or DCE, because the software will take the appropriate actions.

Running X.25 over TCP/IP provides a number of benefits. The datagram containing the X.25 packet can be switched by other routers using their high-speed switching abilities. X.25 connections can be sent over networks running only the TCP/IP protocols. The TCP/IP protocol suite runs over many different networking technologies, including Ethernet, Token Ring, T1 serial, and FDDI. Thus X.25 data can be forwarded over these media to another router, where it can be output to an X.25 interface.

When the connection is made locally, the switching configuration is used; when the connection is across a LAN, the tunneling configuration is used. The basic function is the same for both types of connections, but different configuration commands are required for the two types of connections.

The X.25 switching subsystem supports the following facilities and parameters:

- The D-bit ignored but passed through transparently
- Variable-length interrupt data (if not operating as a DDN or BFE interface)
- Flow Control Parameter Negotiation
    - — Window size up to 7, or 127 for modulo 128 operation
    - — Packet size up to 4096 (if the LAPB layers used are capable of handling the requested size)
- Basic Closed User Group selection
- Throughput class negotiation
- Reverse charging and fast select

The tasks for configuring these facilities are described in the section “Configure X.25 Level 3 Parameters and Special Features.

To configure X.25 tunneling on your router, perform the following tasks:

- Enable X.25 Routing
- Construct the X.25 Routing Table
- Translate X.25 Called and Calling Addresses
- Update the X.121 Address
- Configure a (Local) Switched PVC
- Configure a (Remote) Tunneled PVC

You also need to configure the X.25 parameters and special features, as required for your network. Each task is described in a following section.

Enable X.25 Routing

You can enable switching, or tunneling, (used for remote switching), allowing you to route X.25 traffic through a LAN. To enable local X.25 routing, perform the following task in global configuration mode:

Task	Command
Enable X.25 routing.	**x25 routing** [**use-tcp-if-defs**]

The **USE-TCP-IF-DEFS** flag is used by some routers that receive remote routed calls from older software versions; it might be needed if the originating router cannot be migrated to a new software release.

For an example of configuring X.25 routing, see the sections “X.25 Route Address Pattern Matching Example” and “X.25 Routing Example” later in this chapter.

Construct the X.25 Routing Table

The X.25 routing table is consulted when an incoming call is received that should be forwarded to its destination. Two fields are used to determine the route: the called X.121 network interface address (or destination host address) and the X.25 packet's CUD field. When the destination address and the CUD of the incoming packet fit the X.121 and CUD patterns in the routing table, the call is forwarded. Additionally, when local switching, you can translate called and calling addresses. To construct an X.25 routing table, you must perform one of the following tasks in global configuration mode:

Task	Command
Construct the X.25 routing table by interface type.	**x25 route** [*#position*] *x121-address* [**cud** *pattern*] **interface** *interface number*
or	
Construct the X.25 routing table by IP address (tunneling).	**x25 route** [*#position*] *x121-address* [**cud** *pattern*] **ip** *ip-address* [*ip-address2...ip-address6*]

Entries have as a destination an interface (local switching) or IP address (remote switching). Up to six remote IP addresses can be entered as alternates routes that will be tried in turn to get a successful connection.

Note It can take up to 50 seconds to try an alternate route due to TCP timings.

For an example of constructing the routing table, see the section "X.25 Routing Example" later in this chapter.

Translate X.25 Called and Calling Addresses

When interconnecting two separate X.25 networks, it is sometimes necessary to provide for address translation. Your X.25 switch supports translation of X.25 called and calling addresses for local switching. To translate addresses, perform the following task in global configuration mode:

Task	Command
Translate X.25 called and calling addresses when switching.	**x25 route** [*#position*] *x121-address* [**substitute-source** *rewrite-pattern*] [**substitute-dest** *rewrite-pattern*] [**cud** *pattern*] **interface** *interface number*

Note that address substitution is only performed on routes to an interface. When running X.25 over IP, address substitution can be performed on the destination IP system if the destination system is configured with the appropriate X.25 routing commands.

Update the X.121 Address

Some X.25 calls, when forwarded by the X.25 switching support, need the calling (source) X.121 address updated to that of the outgoing interface. This is necessary when forwarding calls from private data networks to public data networks; other networks might require either address to be suppressed.

To update or suppress the X.121 address, perform one of the following tasks in interface configuration mode:

Task	Command
Update the X.121 address.	**x25 use-source-address**
Suppress the calling (source) X.121 address in Call Request packets.	**x25 suppress-calling-address**
Suppress the called (destination) X.121 address in Call Request packets.	**x25 suppress-called-address**

Configure a (Local) Switched PVC

You can configure an X.25 PVC in the X.25 switching software. This means that DTEs that require permanent circuits can be connected to a router acting as an X.25 switch and have a properly functioning connection. X.25 RESETs will be sent to indicate when the circuit comes up or goes down. Both interfaces must define complementary locally switched PVCs.

To configure a locally switched PVC, perform the following task in interface configuration mode:

Task	Command
Configure a switched PVC.	**x25 pvc** *pvc-number1* **interface** *interface-name* **pvc** *pvc-number2* [*option*]

For an example of configuring a local switched PVC, see the section "Switching a PVC on the Same Router Example" later in this chapter.

Configure a (Remote) Tunneled PVC

A PVC can be forwarded to another router over a LAN using the TCP/IP protocols. When the interfaces come up, a TCP stream connection is established to the router that is acting as the switch for the destination. All X.25 packets will be sent and received over this reliable data stream. Flow control is maintained end-to-end. This is known as *remote PVC switching* or *tunneling.*

Running X.25 over TCP/IP provides a number of benefits. Other routers can switch IP datagrams containing the X.25 packets using the router's high-speed switching abilities. X.25 data can be sent over networks running only TCP/IP protocols. The TCP/IP protocol suite runs over many different networking technologies, including Ethernet, Token Ring, T1 serial, and FDDI. Thus X.25 data can be forwarded over these media to another of our routers where it can be output to an X.25 interface. Both interfaces must define complementary tunneled PVCs.

To connect a PVC across a TCP/IP LAN, perform the following task in interface configuration mode:

Task	Command
Connect a PVC across a TCP/IP LAN.	**x25 pvc** *pvc-number1* **tunnel** *ip-address* **interface serial** *string* **pvc** *pvc-number2* [*options*]

For an example of configuring a remote tunneled PVC, see the section "Establishing a Connection between Two PVCs Example" later in this chapter.

Configure CMNS Routing

The Connection-Mode Network Service (CMNS) provides a mechanism through which local X.25 switching can be extended to nonserial media by using OSI-based NSAP addresses. This implementation runs packet-level X.25 over frame-level LLC2.

Note For information about configuring LLC2 parameters, refer to the "Configuring LLC2 and SDLC Parameters" chapter in the *Router Products Configuration Guide*.

In addition, our CMNS implementation allows LAN-based OSI resources, such as a DTE host and a Sun workstation, to be interconnected to each other via the router's LAN interfaces *and* to a remote OSI-based DTE through a WAN interface (using, for example, an X.25 PSN).

Note CMNS is implicitly enabled whenever an X.25 encapsulation is included with a serial interface configuration.

All local mapping is performed by statically mapping MAC addresses and X.121 addresses to NSAP addresses.

Implementing CMNS routing involves completing the tasks in the following sections:

- Enable CMNS on an Interface
- Specify a CMNS Static Map of Addresses

Enable CMNS on an Interface

Enable CMNS on a nonserial interface by performing the following task in interface configuration mode:

Task	Command
Enable CMNS.	**cmns enable**

For an example of enabling CMNS on an interface, see the section "Enabling CMNS for X.121 and MAC Addresses Example" later in this chapter.

Specify a CMNS Static Map of Addresses

After enabling CMNS on a nonserial interface (or specifying X.25 encapsulation on a serial interface), you must map NSAP addresses to either MAC-layer addresses or X.121 addresses, depending on the application.

For CMNS support over dedicated serial links (such as leased lines), an X.121 address is not needed, but can be optionally included. You must specify the X.121 address for CMNS connections over a packet-switched network, and you must specify a MAC address for CMNS connections over a nonserial medium (Ethernet, FDDI, or Token Ring).

To map the NSAP addresses to either a MAC address or X.121 address, perform one of the following tasks in interface configuration mode:

Task	Command
Statically map NSAP address to non-serial MAC-layer address.	**x25 map cmns** *nsap-address mac-address*
Statically map NSAP address to X.25, with an optional X.121 destination address.	**x25 map cmns** *nsap-address X.121-address*

For an example of configuring a CMNS static map of addresses, see the section "Enabling CMNS for X.121 and MAC Addresses Example" later in this chapter.

Configure DDN or BFE X.25

The DDN X.25 protocol has two versions: Basic Service and Standard Service. Our X.25 implementation only supports the Standard Service. DDN X.25 Standard Service requires that the X.25 data packets carry IP datagrams. The DDN Packet Switch Nodes (PSNs) can extract the IP datagram from within the X.25 packet and pass data to another Standard Service host.

The DDN X.25 Standard is the required protocol for use with DDN PSNs. The Defense Communications Agency (DCA) has certified our DDN X.25 Standard implementation for attachment to the Defense Data Network. As part of the certification, our software is required to provide a scheme for dynamically mapping Internet addresses to X.121 addresses. See the section "DDN X.25 Dynamic Mapping" that follows for details on that scheme.

To enable DDN X.25 service, perform the tasks in the following sections:

- Enable DDN X.25
- Define IP Precedence Handling

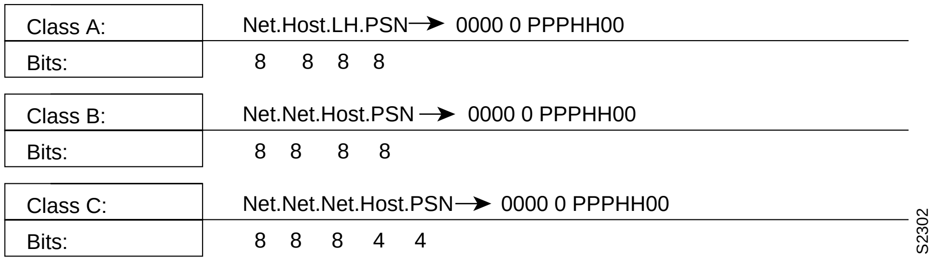
To enable BFE X.25 service, perform the task in the following section:

- Configure Blacker Front-End X.25

DDN X.25 Dynamic Mapping

The DDN X.25 standard implementation includes a scheme for dynamically mapping all classes of IP addresses to X.121 addresses without a table. This scheme requires that the IP and X.121 addresses conform to the formats shown in Tables 11-3 and 11-4. These formats segment the IP addresses into network (N), host (H), logical address (L), and PSN (P) portions. For the BFE encapsulation, the IP address is segmented into Port (P), Domain (D), and BFE ID number (B). The DDN algorithm requires that the host value be less than 64.

Table 11-3 DDN IP Address Conventions



Class A:	Net.Host.LH.PSN ➔ 0000 0 PPPHH00
Bits:	8 8 8 8
Class B:	Net.Net.Host.PSN ➔ 0000 0 PPPHH00
Bits:	8 8 8 8
Class C:	Net.Net.Net.Host.PSN ➔ 0000 0 PPPHH00
Bits:	8 8 8 4 4

Table 11-4 BFE IP Address Conventions

BFE Class A :	Net.unused.Port.Domain.BFE → 0000 0 PDDDBBB				
Bits:	8	1	3	10	10

S2823

The DDN conversion scheme uses the host and PSN portions of an IP address to create the corresponding X.121 address. Strictly speaking, the DDN conversion mechanism is limited to Class A IP addresses. However, the router can convert Class B and Class C addresses as well. As indicated, this method uses the last two octets of a Class B address as the host and PSN identifiers, and the upper and lower four bits in the last octet of a Class C address as the host and PSN identifiers, respectively. The BFE conversion scheme requires a Class A IP address.

The DDN conversion scheme uses a physical address mapping if the host identifier is numerically less than 64. (This limit derives from the fact that a PSN cannot support more than 64 nodes.) If the host identifier is numerically larger than 64, the resulting X.121 address is called a logical address. The DDN does not use logical addresses.

The format of physical DDN X.25/X.121 addresses is ZZZZFIIIHHZZ(SS), where each character represents a digit. ZZZZ represents four zeros, F is zero to indicate a physical address, III represents the PSN octet from the IP address padded with leading zeros, HH is the host octet from the IP address padded with leading zeros, and ZZ represents two zeros. (SS) represents the optional and unused subaddress.

The physical and logical mappings of the DDN conversion scheme always generate a 12-digit X.121 address. Subaddresses are optional; when added to this scheme, the result is a 14-digit X.121 address. The DDN does not use subaddressing.

Packets using routing and other protocols that require broadcast support can successfully traverse X.25 networks, including the DDN. This traversal requires the use of network protocol-to-X.121 maps, because the router must know explicitly where to deliver broadcast datagrams. (X.25 does not support broadcasts.) You can mark network protocol-to-X.121 map entries to accept broadcast packets; the router then sends broadcast packets to hosts with marked entries. For DDN or BFE operation, the router generates the interface X.121 addresses from the interface IP address using the DDN or BFE mapping technique.

Enable DDN X.25

Both DCE and DTE operation cause the router to specify the Standard Service facility in the Call Request packet, which notifies the PSNs to use Standard Service.

To enable DDN X.25, perform one of the following tasks in interface configuration mode, as appropriate for your network:

Task	Command
Set DDN X.25 DTE operation.	**encapsulation ddnx25**
Set DDN X.25 DCE operation.	**encapsulation ddnx25-dce**

For an example of enabling DDN X.25, see the section “DDN X.25 Configuration Example” later in this chapter.

Define IP Precedence Handling

Using Standard Service, the DDN can be configured to provide separate service for datagrams with high precedence values. If the router receives an IP packet with a nonzero Internet precedence field, it uses a different virtual circuit which requested the DDN-specified precedence mapping in the Call Request packet. Different virtual circuits are maintained based on the precedence mapping values and the number of virtual circuits permitted.

By default, the DDN X.25 software opens one virtual circuit for all types of service values. You can enable the precedence sensitivity feature by performing the following task in interface configuration mode:

Task	Command
Allow a new virtual circuit based on the type of service (TOS) field.	**x25 ip-precedence**

Some hosts send nonstandard data in the TOS field, thereby causing multiple, wasteful virtual circuits to be created.

Configure Blacker Front-End X.25

For environments that require a high level of security, your router software supports attachment to Defense Data Network Blacker Front-End (BFE) equipment and Blacker Emergency Mode operation.

Blacker Emergency Mode allows your BFE device and your router to function in emergency situations. When the router is configured to participate in emergency mode and the BFE device is in emergency mode, the router sends address translation information to the BFE device to assist it in sending information.

Our implementation of Blacker Emergency Mode adheres to the specifications outlined in the DCA Blacker Interface Control document, published March 21, 1989.

Your BFE device is configured to be in one of three possible modes as follows:

- Enters emergency mode when requested to by the network. If the router is configured to respond to a BFE device in emergency mode, or if the EXEC command **bfe enter** is used, the router sends address translation information to the BFE device.
- Never enters emergency mode.
- Notifies the router that the emergency mode window is open and waits for the router to tell it to enter emergency mode. If the router is configured to respond to a BFE in emergency mode, or if the EXEC command **bfe enter** is used, the router sends a special address translation packet to the BFE device. The “special” data includes a command to the BFE to enter emergency mode.

Perform these tasks to configure Blacker Emergency Mode:

- Set BFE encapsulation on the router attached to a BFE device.
- Provide address translation information to the BFE device.
- Define the circumstances under which the router will participate in emergency mode.
- Enter Blacker Emergency Mode using the **bfe** EXEC command.

The following tables describe these tasks.

BFE encapsulation operates to map between Class A IP addresses and the X.121 addresses expected by the BFE encryption device. To set BFE encapsulation, perform the following task in interface configuration mode:

Task	Command
Set BFE encapsulation on the router attached to a BFE device.	**encapsulation bfex25**

You must set up a table that provides the address translation information the router sends to the BFE when the BFE is in emergency mode. To do so, perform the following task in interface configuration mode:

Task	Command
Set up the table that lists the BFE nodes (host or gateways) to which the router will send packets.	**x25 remote-red** *host-ip-address* **remote-black** *blacker-ip-address*

You can define the circumstances under which the router participates in emergency mode and how it will participate in emergency mode. To do so, perform the following tasks in interface configuration mode:

Task	Command
Configure the circumstances under which the router will participate in emergency mode.	**x25 bfe-emergency** {**never** | **always** | **decision**}
Configure how a router configured as **x25 bfe-emergency decision** will participate in emergency mode.	**x25 bfe-decision** {**no** | **yes** | **ask**}

To set the router to participate in emergency mode or to end participation in emergency mode when your system is so configured, perform the following task in EXEC mode:

Task	Command
Set router to participate in emergency mode.	**bfe** {**enter** | **leave**} *interface-type number*

For an example of configuring Blacker Emergency mode, see the section “Setting Blacker Emergency Mode Example” at the end of this chapter.

Configure X.25 Level 3 Parameters and Special Features

The router software allows you to configure the standard Level 2 and Level 3 X.25 parameters and user facilities.

Note If you connect a router to an X.25 network, use the parameters set by the network administrator. Also, note that the X.25 Level 2 parameters described earlier in this chapter affect X.25 Level 3 operations.

This section describes the X.25 parameters, user facilities, and special features you can configure. Which tasks you perform depends upon the structure of your network and the requirements of the service provider. These parameters must be adjusted to match the values used by the X.25 network. It is common for networks to require values different from these defaults.

To configure the optional parameters, user facilities, and special features, perform one or more of the tasks described in the following sections:

- Configure Virtual Circuit Ranges
- Configure the Encapsulation Virtual Circuit Idle Time
- Increase the Number of Virtual Circuits Allowed
- Configure the Ignore Destination Time
- Set Default Window Sizes
- Set Default Packet Sizes
- Establish the Packet Acknowledgment Policy
- Configure the X.25 Level 3 Retransmission Timers
- Set X.25 TCP Header Compression
- Define X.25 Host Names
- Define the Default Protocol
- Configure X.25 User Facilities
- Set the Packet Number Modulo
- Suppress or Update the Calling Address
- Suppress the Called Address
- Disable Packet-Level Restarts
- Define the VC Packet Hold Queue Size
- Configure an Interface Alias
- Configure X.25 Bridging

Configure Virtual Circuit Ranges

The X.25 protocol maintains multiple connections over one physical link between a DTE and a DCE. These connections are called virtual circuits (VCs) or logical channels (LCs). X.25 can maintain up to 4095 VCs numbered 1 through 4095. An individual VC is identified by giving its logical channel identifier (LCI) or virtual circuit number (VCN). Many documents use the terms VC and LC, and VCN, LCN, and LCI interchangeably. Each of these terms refer to the virtual circuit number.

An important part of X.25 operation is the range of virtual circuit numbers. Virtual circuit numbers are broken into four ranges (listed here in numerically increasing order):

1. Permanent virtual circuits (PVCs)
2. Incoming-only circuits
3. Two-way circuits
4. Outgoing-only circuits

The incoming-only, two-way, and outgoing-only ranges define the VC numbers over which a switched virtual circuit (SVC) can be established by placing an X.25 call, much like a telephone network establishes a switched voice circuit when a call is placed.

The rules about DCE and DTE devices initiating calls are as follows:

- Only the DCE device can initiate a call in the incoming-only range.
- Only the DTE device can initiate a call in the outgoing-only range.
- Both the DCE device and the DTE device can initiate a call in the two-way range.

There is no difference in the operation of the SVCs except the restrictions on which a device can initiate a call. These ranges can be used to prevent one side from monopolizing the virtual circuits, which can be useful for X.25 interfaces with a small total number of SVCs available.

Six X.25 parameters define the upper and lower limit of each of the three SVC ranges. A PVC must be assigned a number less than the numbers assigned to the SVC ranges. An SVC range is not allowed to overlap another range.

Table 11-5 lists the virtual circuit types, their ranges, and defaults. Note that the values for these parameters must be the same on both ends of an X.25 link. For connection to a public data network (PDN), these values must be set to the values assigned by the network. An SVC range is unused if its lower and upper limits are set to 0; other than this use for marking unused ranges, VC 0 is not available.

Note Because the X.25 protocol requires the DTE and DCE to have identical VC ranges, if the interface is up, changes to the VC range limits will be held until the X.25 protocol RESTARTs the packet service.

Table 11-5 X.25 Virtual Circuits

Task (Parameter)	Range	Default
Set the LIC.	1–4095	0
Set the HIC.	1–4095	0
Set the LTC.	1–4095	1
Set the HTC.	1–4095	1024 (serial) 4095 (cmns)
Set the LOC.	1–4095	0
Set the HOC.	1–4095	0

For an example of configuring virtual circuit ranges, see the section “Setting Virtual Circuit Ranges Example” later in this chapter.

Configure the Encapsulation Virtual Circuit Idle Time

The router can clear a datagram transport SVC after a set period of inactivity. Routed SVCs are not timed for inactivity. You can set this time by performing the following task in interface configuration mode:

Task	Command
Set the idle time before an SVC is cleared.	**x25 idle** *minutes*

For an example of configuring the SVC idle timer, see the section “Typical X.25 Configuration Example” later in this chapter. See the section “Monitor and Maintain LAPB and X.25” later in this chapter for additional commands that clear virtual circuits.

Increase the Number of Virtual Circuits Allowed

For X.25 datagram transport, you can establish up to eight switched virtual circuits to one host for each protocol. To increase the number of VCs allowed, perform one or both of the following tasks in interface configuration mode:

Task	Command
Specify the maximum number of switched virtual circuits that can be open simultaneously to one host for each protocol.	**x25 nvc** *count*
Set up the LAN protocol-to-X.121 address mapping.	**x25 map** *protocol-keyword protocol-address X.121-address* **nvc** *count*

For an example of increasing the number of virtual circuits allowed, see the sections “Typical X.25 Configuration Example” and “DDN X.25 Configuration Example” later in this chapter.

Configure the Ignore Destination Time

Upon receiving a Clear for an outstanding datagram transport Call Request, the X.25 support code immediately tries another Call Request if it has more traffic to send. This action can overrun some X.25 switches.To define the number of minutes it takes to prevent calls from going to a previously failed destination by, perform the following task in interface configuration mode (incoming calls will still be accepted):

Task	Command
Configure the ignore destination time.	**x25 hold-vc-timer** *minutes*

Set Default Window Sizes

X.25 networks have a default input and output window size that is defined by the network administrator. You must set the router default input and output window sizes to match those of the network; see the note following the next section, "Set Default Packet Sizes." To set the default window sizes (default is 2), perform the following tasks in interface configuration mode:

Task	Command
Set the default number of packets the server can receive before sending an X.25 acknowledgment.	**x25 win** *packets*
Set the default number of sent packets that can remain unacknowledged before the VC uses its hold queue.	**x25 wout** *packets*

For an example of setting the default window sizes, see the sections "Typical X.25 Configuration Example" and "DDN X.25 Configuration Example" later in this chapter.

Set Default Packet Sizes

X.25 networks have a default maximum input and output packet size (default is 128) that is defined by the network administrator. To set the router default input and output maximum packet sizes to match those of the network, perform the following tasks in interface configuration mode:

Task	Command
Set the default input maximum packet size.	**x25 ips** *bytes*
Set the default output maximum packet size.	**x25 ops** *bytes*

To send a packet larger than the agreed X.25 packet size over an X.25 virtual circuit, a router must break the packet into two or more X.25 packets with the M-bit ("more data" bit) set. The receiving device collects all packets in the M-bit sequence and reassembles them into the original packet.

It is possible to define default packet sizes that cannot be supported by the lower layer (see the LAPB N1 parameter). The router will, however, negotiate lower maximum packet sizes for all SVCs so the agreed sizes can be carried.

For an example of setting the default maximum packet sizes, see the sections "Typical X.25 Configuration Example" and "DDN X.25 Configuration Example" later in this chapter.

Note Because the X.25 protocol requires the DTE and DCE to have identical default maximum packet sizes and default window sizes, if the interface is up, changes to the window and packet sizes will be held until the X.25 protocol RESTARTs the packet service.

Establish the Packet Acknowledgment Policy

You can instruct the router to send an acknowledgment packet when it has received a threshold of data packets it has not acknowledged, instead of waiting until its input window is full. A value of 1 will send an acknowledgment for each data packet received if it cannot be acknowledged in an outgoing data packet. This approach improves line responsiveness at the expense of bandwidth. A value of 0 restores the default behavior of waiting until the input window is full.

To establish the acknowledgment threshold, perform the following task in interface configuration mode:

Task	Command
Establish the threshold at which to acknowledge data packets.	**x25 th** *delay-count*

Configure the X.25 Level 3 Retransmission Timers

The X.25 Level 3 retransmission timers determine how long the router must wait before retransmitting various packets. You can set these timers independently.

To set the retransmission timers, perform any of the following tasks in interface configuration mode:

Task	Command
Set DTE T20 Restart Request.	**x25 t20** *seconds*
Set DCE T10 Restart Indication.	**x25 t10** *seconds*
Set DTE T21 Call Request.	**x25 t21** *seconds*
Set DCE T11 Incoming Call.	**x25 t11** *seconds*
Set DTE T22 Reset Request.	**x25 t22** *seconds*
Set DCE T12 Reset Indication.	**x25 t12** *seconds*
Set DTE T23 Clear Request.	**x25 t23** *seconds*
Set DCE T13 Clear Indication.	**x25 t13** *seconds*

For an example of setting the retransmission timers, see the section “DDN X.25 Configuration Example” later in this chapter.

Set X.25 TCP Header Compression

We support RFC 1144 TCP/IP header compression on serial lines using HDLC and X.25 encapsulation. The implementation of compressed TCP over X.25 uses one virtual circuit (VC) to pass the compressed packets. Any IP traffic (including standard TCP) is carried over separate IP encapsulation VCs.

To set X.25 TCP header compression, perform the following tasks in interface configuration mode:

Task	Command
Implement header compression on IP packets.	**ip tcp header-compression** [**passive**]
Allow a separate VC for compressed packets.	**x25 map compressedtcp** *ip-address x.121-address* [*option*]

Define X.25 Host Names

To define symbolic host names, perform the following task in global configuration mode:

Task	Command
Define a symbolic host name.	**x25 host** *name x.121-address* [**cud** *call-user-data*]

Define the Default Protocol

To identify calls with unknown or missing Call User Data as either IP or PAD calls, perform the following task in interface configuration mode:

Task	Command
Interpret incoming calls with unknown CUD.	**x25 default** *protocol*

Configure X.25 User Facilities

The X.25 software provides commands to support X.25 user facilities—options specified by the creators of the X.25 Recommendation—that allow you to implement features such as accounting, user identification, and flow control negotiation. The facilities configured by the **x25 map** commands are on a per-peer basis; all other facility commands specify the values sent for calls originated by the interface. Routed calls are not affected by the facilities specified for the outgoing interface.

To set the supported X.25 user facilities, perform one or more of the following tasks in interface configuration mode:

Task (User Facility)	Command
Select closed user group.	**x25 facility cug** *number* **x25 map** *protocol-keyword protocol-address* **cug** *number*
Set flow control parameter negotiation values to request on outgoing calls.	**x25 facility packetsize** *in-size out-size* **x25 map** *protocol-keyword protocol-address* **packetsize** *in-size out-size* **x25 facility windowsize** *in-size out-size* **x25 map** *protocol-keyword protocol-address* **windowsize** *in-size out-size*
Set reverse charging.	**x25 facility reverse** **x25 map** *protocol-keyword protocol-address* **reverse**
Allow reverse charging acceptance.	**x25 accept-reverse** **x25 map** *protocol-keyword protocol-address* **accept-reverse**
Select throughput class negotiation.	**x25 facility throughput** *in out* **x25 map** *protocol-keyword protocol-address* **throughput** *in out*
Select transit delay.	**x25 facility transit-delay** *number* **x25 map** *protocol-keyword protocol-address* **transit-delay** *number*
Set the Recognized Private Operation Agency (RPOA) to use.	**x25 facility rpoa** *name* **x25 map** *protocol-keyword protocol-address* **rpoa** *name*
Set the Cisco standard network user identification.	**x25 map** *protocol-keyword protocol-address* **nuid** *username password*
Set a user-defined network user identification allowing format determined by the network administrator	**x25 map** *protocol-keyword protocol-address* **nudata** *string*

Additionally, the D-bit is supported and passed through transparently. Both restricted and unrestricted fast select are also supported and are transparently handled by the software. No configuration is required for use of the D-bit or fast select facilities.

Set the Packet Number Modulo

Our implementation of X.25 supports the extended packet sequence numbering. To use the extended packet sequence, perform the following task in interface configuration mode:

Task	Command
Set the packet numbering modulo.	**x25 modulo** *modulus*

Note Because the X.25 protocol requires the DTE and DCE to have identical modulos, if the interface is up, changes to the modulo will be held until the X.25 protocol RESTARTs the packet service.

Suppress or Update the Calling Address

You can omit or update the calling (source) address in outgoing calls. Either of these options might be required by networks that expect only subaddresses or that expect the interface's address in the calling address field. To suppress or update the calling address, perform the appropriate task in interface configuration mode:

Task	Command
Omit the calling (source) X.121 address in outgoing calls.	**x25 suppress-calling-address**
Replace the calling (source) X.121 address in switched calls.	**x25 use-source-address**

Suppress the Called Address

You can omit the called (destination) address in outgoing calls. This option is required for networks that expect only subaddresses in the called address field. To suppress the called address, perform the following task in interface configuration mode:

Task	Command
Omit the called (destination) X.121 address in outgoing calls	**x25 suppress-called-address**

Disable Packet-Level Restarts

By default, a packet-level restart is performed when the link level is reset. This behavior can be disabled for networks that do not allow it, but disabling this behavior can cause anomalous packet layer behavior. To disable packet-level restarts, perform the following task in interface configuration mode:

Task	Command
Disable packet-level restarts.	**no x25 linkrestart**

Define the VC Packet Hold Queue Size

To define the maximum number of packets that can be held while a virtual circuit is unable to send data, perform the following task in interface configuration mode:

Task	Command
Define the VC packet hold queue size.	**x25 hold-queue** *queue-size*

Configure an Interface Alias

You can supply alias X.121 addresses for an interface. This allows the interface to accept data encapsulation calls to an address other than the address defined by the **x25 address** command.To configure an alias, perform the following task in interface configuration mode:

Task	Command
Supply an alias for other X.121 addresses that will be accepted on the interface.	**x25 route** [*#position*] *x121-pattern* [**cud** *pattern*] **alias** *interface number*

Configure X.25 Bridging

Our transparent bridging software supports bridging over X.25 VCs. To enable the X.25 bridging capability, perform the following task in interface configuration mode:

Task	Command
Define bridging of X.25 frames.	**x25 map bridge** *x.121-address* **broadcast** [*option*]

Monitor and Maintain LAPB and X.25

To monitor and maintain X.25 and LAPB, perform any of the following tasks in EXEC mode:

Task	Command
Clear all virtual circuits at once (calls originated, terminated, and switched by the router are cleared), or clear the single VC specified.	**clear x25-vc** *interface name number* [*lcn*]
Display CMNS information.	**show cmns** [*interface-name*]
Display operation statistics for an interface.	**show interfaces serial** *number*
Display CMNS connections over LLC2.	**show llc2**
Display the protocol-to-X.121 address map.	**show x25 map**
Display the one-to-one mapping of the host IP addresses and the remote BFE device's IP addresses.	**show x25 remote-red**
Display routes assigned by the **x25 route** command.	**show x25 route**
Display details of active VCs.	**show x25 vc** [*lcn*]

Note See the "X.25 Cause and Diagnostic Codes" appendix in the *Debug Command Reference* publication for a description of X.25 diagnostic codes that can appear in these **show** command displays.

LAPB and X.25 Configuration Examples

The following sections provide examples to help you understand how to configure LAPB and X.25 for your network:

- Typical LAPB Configuration Example
- Typical X.25 Configuration Example
- Setting Virtual Circuit Ranges Example
- Switching a PVC on the Same Router Example
- X.25 Routing Example
- X.25 Route Address Pattern Matching Example
- Using a PVC to Exchange IP Traffic Example
- Establishing a Connection between Two PVCs Example
- Enabling Remote Tunneling Example
- Enabling CMNS for X.121 and MAC Addresses Example
- Switching CMNS over a PDN Example
- Switching CMNS over Leased Lines Example
- DDN X.25 Configuration Example
- Setting Blacker Emergency Mode Example
- Configuring X.25 to Allow Ping Support over Multiple Lines Example
- Netbooting over X.25 Example

Typical LAPB Configuration Example

In the following example, the frame size (N1), window size (k), and maximum retransmission (N2) parameters retain their default values. The **encapsulation** interface configuration command sets DCE operation to carry a single protocol, IP by default, and the **lapb t1** interface configuration command sets the retransmission timer to 4,000 milliseconds (4 seconds) for a link with a long delay or slow connecting DTE device.

```
interface serial 3
encapsulation lapb-dce
lapb t1 4000
```

Typical X.25 Configuration Example

The following example shows the complete configuration for a serial interface connected to a commercial X.25 PDN for routing the IP protocol. The IP subnetwork address 131.108.9.0 has been assigned for the X.25 network.

Note When routing IP over X.25, the X.25 network must be treated as a single IP network or subnetwork. Map entries for routers with addresses on subnetworks other than the one on which the interface's IP address is stored are ignored by the routing software. Additionally, all routers using the subnet number should have map entries for all others. There are also issues with the broadcast flag, which apply both to IP and to other protocols with dynamic routing.

```
interface serial 2
ip address 131.108.9.1 255.255.255.0
!
encapsulation X25
!
! The ”bandwidth” command is not part of the X.25
! configuration; it is especially important to understand that it does not
! have any connection with the X.25 entity of the same name.
!”bandwidth” commands are used by IP routing processes (currently only IGRP),
! to determine which lines are the best choices for traffic.
! Since the default is 1544 Kbaud, and X.25 service at that rate is not generally
! available, most X.25 interfaces that are being used with IGRP in a
! real environment will have ”bandwidth” settings.
!
! This is a 9.6 Kbaud line:
!
bandwidth 10
!
! These Level 3 parameters are default flow control values; they need to
! match the PDN defaults. The values used by an SVC are negotiable on a per-call basis:
!
x25 win 7
x25 wout 7
x25 ips 512
x25 ops 512
!
! You must specify an X.121 address to be assigned to the X.25
! interface by the PDN.
!
x25 address 31370054065
!
! The following Level 3 parameters have been set to match the network.
! You generally need to change some Level 3 parameters, most often
! those listed below. You might not need to change any Level 2
! parameters, however.
!
x25 htc 32
x25 idle 5
x25 nvc 2
!
! The following commands configure the X.25 map. If you want to exchange
! routing updates with any of the routers, they would need
! ”broadcast” flags.
! If the X.25 network is the only path to the routers, static routes are
! generally used to save on packet charges. If there is a redundant
! path, it might be desirable to run a dynamic routing protocol.
!
x25 map IP 131.108.9.3 31370019134 ACCEPT-REVERSE
! ACCEPT-REVERSE allows collect calls
x25 map IP 131.108.9.2 31370053087
!
! If the PDN cannot handle fast back-to-back frames, use the
!”transmitter-delay” command to slow down the interface.
!
transmitter-delay 1000
```

Setting Virtual Circuit Ranges Example

The following example sets the following VC ranges: 5 to 20 dedicated to incoming calls only (from the DCE to the DTE), 25 to 1024 for either incoming or outgoing calls, no VC range dedicated to outgoing calls (from the DTE to the DCE). Up to four permanent virtual circuits can be defined on VCs 1 through 4.

```
x25 lic 5
x25 hic 20
x25 ltc 25
```

Switching a PVC on the Same Router Example

In the following example, there is a PVC connected between two serial interfaces on the same router. In this type of interconnection configuration, the destination interface must be specified along with the PVC number on that interface. To make a working PVC connection, two commands must be specified, each pointing to the other.

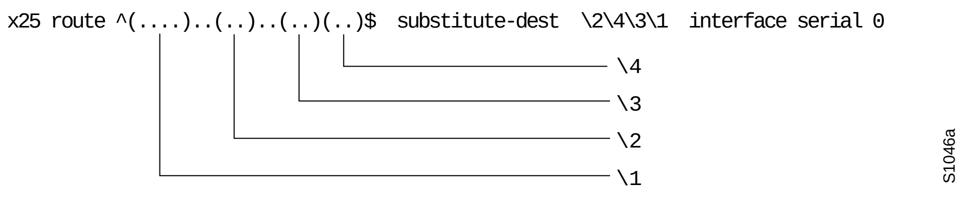
```
interface serial 0
encapsulation x25
x25 ltc 5
x25 pvc 1 interface serial 1 pvc 4
!
interface serial 1
encapsulation x25
x25 ltc 5
x25 pvc 4 interface serial 0 pvc 1
```

X.25 Route Address Pattern Matching Example

The following example shows how to indicate that X.25 calls to addresses whose first four Data Network Identification Code (DNIC) digits are 1111 should be routed to interface serial 3, but that the DNIC field in the addresses presented to the equipment connected to that interface should be changed to 2222. The \1 in the rewrite pattern indicates the portion of the original address matched by the digits following the 1111 DNIC.

```
x25 route ^1111(.*) substitute-dest 2222\1 interface serial 3
```

The following example shows a more contrived command intended to illustrate the power of the rewriting scheme:



```
x25 route ^(....)..(..)..(..)(..)$  substitute-dest  \2\4\3\1  interface serial 0
```

It causes all X.25 calls with 14-digit called addresses to be routed through interface serial 0. The incoming DNIC field would be moved to the end of the address. The fifth, sixth, ninth, and tenth digits would be deleted, and the thirteenth and fourteenth would be moved before the eleventh and twelfth.

X.25 Routing Example

The following examples illustrate how to enable an X.25 switch, how to enable call forwarding, and how to configure a router on a Tymnet/PAD switch to accept and forward calls.

This first example shows how to enable X.25 switching, as well as how to enter routes into the X.25 routing table:

```
! Enable X.25 forwarding
x25 routing
!
! Enter routes into the table. Without a positional parameter, entries
! are appended to the end of the table
x25 route ^100$ interface serial 0
x25 route 100 cud ^pad$ interface serial 2
x25 route 100 interface serial 1
x25 route ^3306 interface serial 3
x25 route .* ip 10.2.0.2
```

The routing table forwards calls for X.121 address 100 out interface serial 0. Otherwise, if the X.121 address contains 100 anywhere within it and contains no Call User Data, or the Call User Data is not the string “pad”, it is forwarded onto serial 1. If the X.121 address contains the digits 100 and the Call User Data is the string “pad”, the call is forwarded onto serial 2. All X.121 addresses that do not match the first three routes are checked for a DNIC of 3306 as the first four digits. If they do match, they are forwarded over serial 3. All other X.121 addresses will match the fifth entry, which is a match-all pattern and will have a TCP connection established to the IP address 10.2.0.2. The router at 10.2.0.2 will then route the call according to its X.25 routing table.

This second example configures a router that sits on a Tymnet PAD/switch to accept calls and have them forwarded to a DEC VAX system. This feature permits running X.25 network over a generalized, existing IP network, thereby making it unnecessary to get another physical line for one protocol. The router positioned next to the DEC VAX system is configured with X.25 routes, as follows:

```
x25 route vax-x121-address interface serial 0
x25 route .* ip cisco-on-tymnet-ipaddress
```

This routes all calls to the DEC VAX X.121 address out to serial 0, where the VAX is connected running PSI. All other X.121 addresses are forwarded to the *cisco-on-tymnet* address using its IP address. This takes all outgoing calls from the VAX and sends them to *cisco-on-tymnet* for further processing.

On the router named *cisco-on-tymnet*, you would enter these commands:

```
x25 route vax-x121-address ip cisco-on-vax
x25 route .* interface serial 0
```

This forces all calls with the VAX X.121 address to be sent to the router with the VAX connected to it. All other calls with X.121 addresses are forwarded out to Tymnet. If Tymnet can route them, a CALL ACCEPTED packet is returned, and everything proceeds normally. If Tymnet cannot handle it, it clears the call, and the CLEAR REQUEST packet is forwarded back toward the VAX.

Using a PVC to Exchange IP Traffic Example

The following example, illustrated in Figure 11-2, demonstrates how to use the PVC to exchange IP traffic between Router X and Router Y.

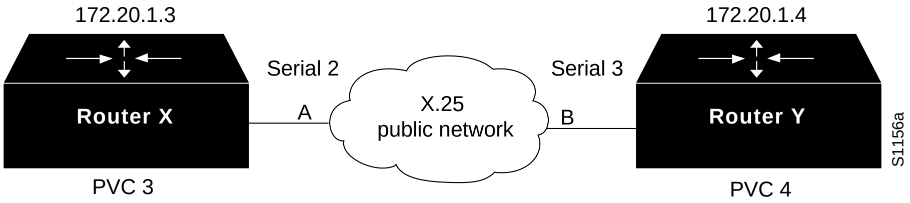


Figure 11-2 Establishing an IP Encapsulation PVC through an X.25 Network

Configuration for Router X

```
interface serial 2
ip address 131.108.1.3 255.255.255.0
x25 map ip 131.108.1.4 0
x25 pvc 4 ip 131.108.1.4
```

Configuration for Router Y

```
interface serial 3
ip address 131.108.1.4 255.255.255.0
x25 map ip 131.108.1.3 0
x25 pvc 4 ip 131.108.1.3
```

In this example, the PDN has established a PVC through its network connecting PVC number 3 of access point A to PVC number 4 of access point B. On Router X, a connection is established between Router X and Router Y’s IP address, 131.108.1.4. On Router Y, a connection is established between Router Y and Router X’s IP address, 131.108.1.3.

Establishing a Connection between Two PVCs Example

In the following example, a connection is established between two PVCs across a LAN. Because the connection is remote (across the LAN), the tunneling command is used. This example establishes a PVC between Router X, Serial 0, PVC1 and Router Y, Serial 1, PVC 2. Figure 11-3 provides a visual representation of the configuration.

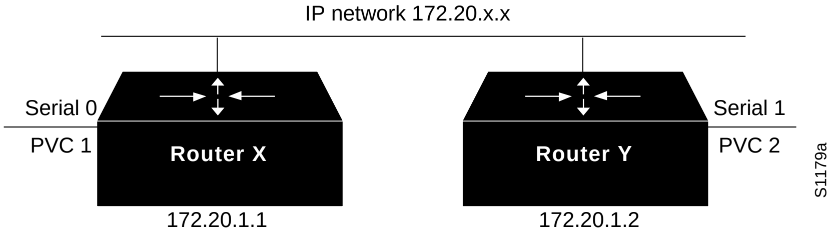


Figure 11-3 X.25 Tunneling Connection

Configuration for Router X

```
interface serial 0
x25 pvc 1 tunnel 131.108.1.2 interface serial 1 pvc 2
```

Configuration for Router Y

```
interface serial 1
x25 pvc 2 tunnel 131.108.1.1 interface serial 0 pvc 1
```

Enabling Remote Tunneling Example

In Figure 11-4, the connection between points A and B is switched, and the connections between points A or B and C are tunneled.

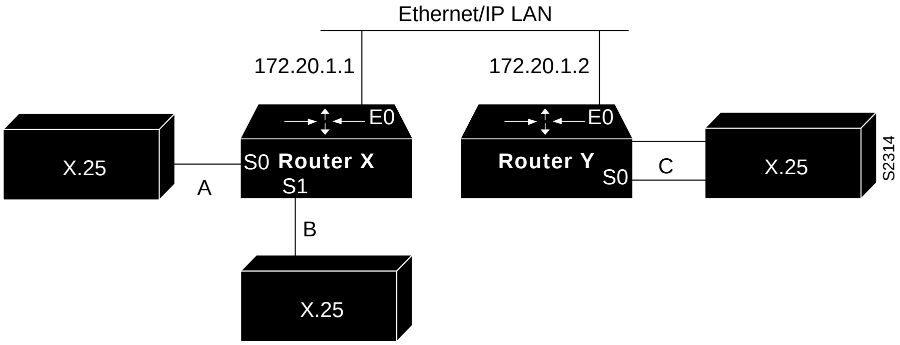


Figure 11-4 Local Switching and Remote Tunneling PVCs

Configuration for Router X

```
interface ethernet 0
ip address 131.108.1.1 255.255.255.0
!
interface serial 0
x25 ltc 5
x25 pvc 1 interface serial 1 pvc 1
x25 pvc 2 tunnel 131.108.1.2 interface serial 0 pvc 1
!
interface serial 1
x25 ltc 5
x25 pvc 1 interface serial 0 pvc 1
x25 pvc 2 tunnel 131.108.1.2 interface serial 0 pvc 2
```

Configuration for Router Y

```
interface ethernet 0
ip address 131.108.1.2 255.255.255.0
!
interface serial 0
x25 ltc 5
x25 pvc 1 tunnel 131.108.1.1 interface serial 0 pvc 2
x25 pvc 2 tunnel 131.108.1.1 interface serial 1 pvc 2
```

Enabling CMNS for X.121 and MAC Addresses Example

The following examples illustrate enabling CMNS and configuring X.121 and MAC address mappings:

```
interface ethernet 0
cmns enable
x25 map cmns 38.8261.1000.0150.1000.17 0000.0c00.ff89
! Above maps NSAP to MAC-address on Ethernet0
!
interface serial 0
encapsulation x25
x25 map cmns 38.8261.1000.0150.1000.18 3110451
! Above maps NSAP to X.121-address on Serial0
! assuming the link is over a PDN
!
interface serial 1
encapsulation x25
x25 map cmns 38.8261.1000.0150.1000.20
! Above specifies cmns support for Serial1
! assuming that the link is over a leased line
```

Switching CMNS over a PDN Example

The following example depicts switching CMNS over a packet-switched public data network (PDN). Figure 11-5 illustrates the general network topology for a CMNS switching application where calls are being made between resources on opposite sides of a remote link to Host A (on an Ethernet) and Host B (on a Token Ring), with a PDN providing the connection.

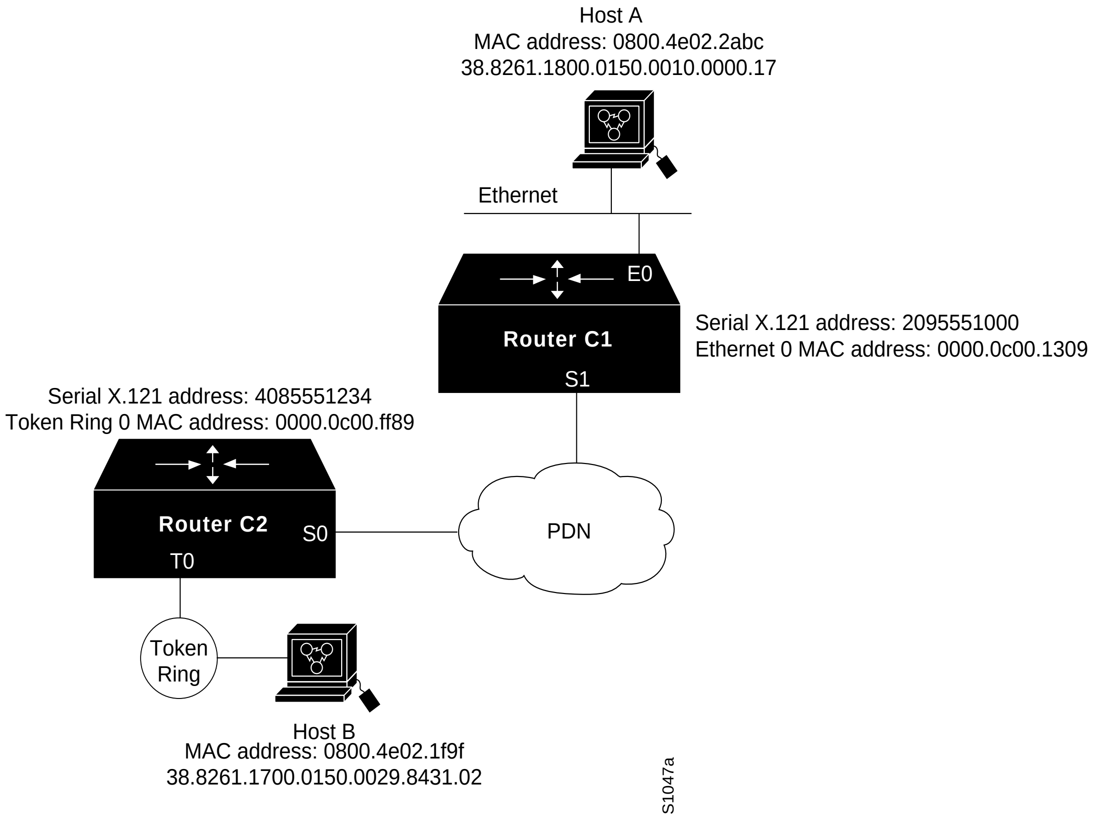

Host A
MAC address: 0800.4e02.2abc
38.8261.1800.0150.0010.0000.17
Ethernet
E0
Router C1
S1
Serial X.121 address: 2095551000
Ethernet 0 MAC address: 0000.0c00.1309
Serial X.121 address: 4085551234
Token Ring 0 MAC address: 0000.0c00.ff89
Router C2
S0
T0
PDN
Token Ring
Host B
MAC address: 0800.4e02.1f9f
38.8261.1700.0150.0029.8431.02
S1047a

Figure 11-5 Example Network Topology for Switching CMNS over a PDN

The following configuration listing allows resources on either side of the PDN to call Host A or Host B. This configuration allows traffic intended for the remote NSAP address specified in the **x25 map cmns** commands (for the serial ports) to be switched through the serial interface for which CMNS is configured.

Configuration for Router C2

```
! This configuration specifies that any traffic from any other
! interface intended for any NSAP address with NSAP prefix 38.8261.17
! will be switched to MAC address 0800.4e02.1f9f
! through Token Ring 0
!
interface token 0
cmns enable
x25 map cmns 38.8261.17 0800.4e02.1f9f
!
! This configuration specifies that traffic from any other interface
! on Cisco Router C2 that is intended for any NSAP address with
! NSAP-prefix 38.8261.18 will be switched to
! X.121 address 2095551000 through Serial 0
!
interface serial 0
encapsulation x25
x25 address 4085551234
x25 map cmns 38.8261.18 2095551000
```

Configuration for Router C1

```
! This configuration specifies that any traffic from any other
! interface intended for any NSAP address with NSAP 38.8261.18
! will be switched to MAC address 0800.4e02.2abc through Ethernet 0
!
interface ethernet 0
cmns enable
x25 map cmns 38.8261.18 0800.4e02.2abc
!
! This configuration specifies that traffic from any other interface
! on Cisco Router C1 that is intended for any NSAP address with
! NSAP-prefix 38.8261.17 will be switched to X.121 address
! 4085551234 through Serial 1
!
interface serial 1
encapsulation x25
x25 address 2095551000
x25 map cmns 38.8261.17 4085551234
```

Switching CMNS over Leased Lines Example

The following example illustrates switching CMNS over a leased line. Figure 11-6 illustrates the general network topology for a CMNS switching application where calls are being made by resources on the opposite sides of a remote link to Host C (on an Ethernet) and Host D (on a Token Ring), with a dedicated leased line providing the connection.

The following configuration listing allows resources on either side of the leased line to call Host C or Host D. This configuration allows traffic intended for the remote NSAP address specified in the **x25 map cmns** commands (for the serial ports) to be switched through the serial interface for which CMNS is configured.

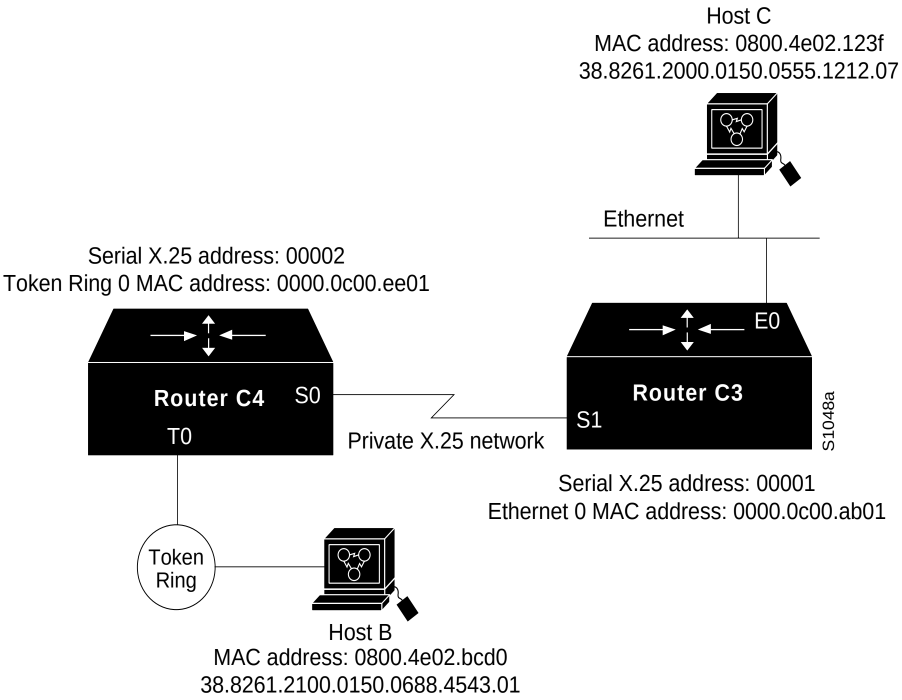

Host C
MAC address: 0800.4e02.123f
38.8261.2000.0150.0555.1212.07
Ethernet
Serial X.25 address: 00002
Token Ring 0 MAC address: 0000.0c00.ee01
E0
Router C4
S0
T0
Router C3
S1
Private X.25 network
S1048a
Serial X.25 address: 00001
Ethernet 0 MAC address: 0000.0c00.ab01
Token Ring
Host B
MAC address: 0800.4e02.bcd0
38.8261.2100.0150.0688.4543.01

Figure 11-6 Example Network Topology for Switching CMNS over a Leased Line

A key difference for this configuration compared with the previous example is that with no PDN, the specification of an X.121 address in the **x25 map cmns** command is not necessary. The specification of an X.25 address also is not needed, but is included for symmetry with the previous example.

Configuration for Router C4

```
! This configuration specifies that any traffic from any other
! interface intended for any NSAP address with NSAP 38.8261.21
! will be switched to MAC address 0800.4e02.bcd0 through Token Ring 0
!
interface token 0
cmns enable
x25 map cmns 38.8261.21 0800.4e02.bcd0
!
! This configuration specifies that traffic from any other interface
! on Cisco Router C4 that is intended for any NSAP address with
! NSAP-prefix 38.8261.20 will be switched through Serial 0
!
interface serial 0
encapsulation x25
x25 address 00002
x25 map cmns 38.8261.20
```

Configuration for Router C3

```
! This configuration specifies that any traffic from any other
! interface intended for any NSAP address with NSAP 38.8261.20
¡ will be switched to MAC address 0800.4e02.123f through Ethernet 0
!
interface ethernet 0
cmns enable
x25 map cmns 38.8261.20 0800.4e02.123f
!
! This configuration specifies that traffic from any other interface
! on Cisco Router C3 that is intended for any NSAP address with
! NSAP-prefix 38.8261.21 will be switched through Serial 1
!
interface serial 1
encapsulation x25
x25 address 00001
x25 map cmns 38.8261.21
```

DDN X.25 Configuration Example

The following example illustrates how to configure a router interface to run DDN X.25:

```
interface serial 0
ip address 192.31.7.50 255.255.255.240
encapsulation DDNX25
x25 win 6
x25 wout 6
x25 ips 1024
x25 ops 1024
x25 t20 10
x25 t21 10
x25 t22 10
x25 t23 10
x25 nvc 2
x25 map IP 192.31.7.49 000000010300 BROADCAST
```

Setting Blacker Emergency Mode Example

In the following example, interface serial 0 is configured to require an EXEC command from the administrator before it participates in emergency mode. The host IP address is 21.0.0.12, and the address of the remote BFE unit is 21.0.0.1. When the BFE enters emergency mode, the router will prompt the administrator for the EXEC command **bfe enter** to direct the router to participate in emergency mode.

```
interface serial 0
ip address 21.0.0.2 255.0.0.0
encapsulation bfex25
x25 bfe-emergency decision
x25 remote-red 21.0.0.12 remote-black 21.0.0.1
x25 bfe-decision ask
```

Configuring X.25 to Allow Ping Support over Multiple Lines Example

For **ping** commands to work in an X.25 environment (when load sharing over multiple serial lines), you must include entries for all adjacent interface IP addresses in the **x25 map** command for each serial interface. The following example illustrates this point.

Consider two routers, Router A and Router B, communicating with each other over two serial lines via an X.25 PDN (see Figure 11-7) or over leased lines. In either case, all serial lines must be configured for the same IP subnet address space. In order to allow for successful **ping** commands, the configuration might be as in the lists that follow. In any event, a similar configuration is required for the same subnet IP addresses to work across X.25.

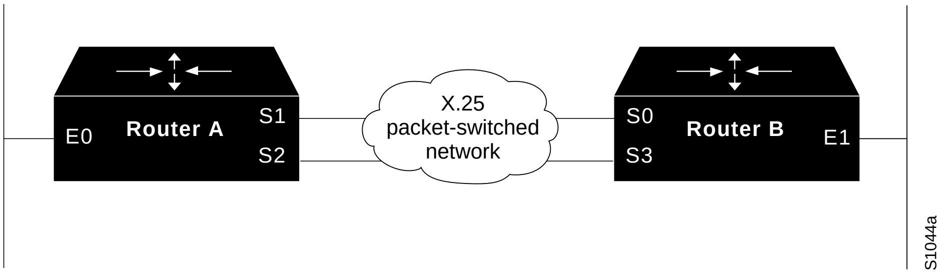


Figure 11-7 Parallel Serial Lines to X.25 Network

Note All four serial ports configured for the two routers in the following configuration example must be assigned to the same IP subnet address space. In this case, the subnet is 131.108.170.0.

Configuration for Router A

```
interface serial 1
ip 131.108.170.1 255.255.255.0
x25 address 31370054068
x25 map ip 131.108.170.3 31370054065
x25 map ip 131.108.170.4 31370054065
!
interface serial 2
ip 131.108.170.2 255.255.255.0
x25 address 31370054069
x25 map ip 131.108.170.4 31370054067
x25 map ip 131.108.170.3 31370054067
! allow either destination address
x25 31370054068 alias serial2
x25 31370054069 alias serial1
```

Configuration for Router B

```
interface serial 0
ip 131.108.170.3 255.255.255.0
x25 address 31370054065
x25 map ip 131.108.170.1 31370054068
x25 map ip 131.108.170.2 31370054068
!
interface serial 3
ip 131.108.170.4 255.255.255.0
x25 address 31370054067
x25 map ip 131.108.170.2 31370054069
x25 map ip 131.108.170.1 31370054069
! allow either destination address
x25 31370054065 alias serial3
x25 31370054067 alias serial0
```

Netbooting over X.25 Example

When netbooting over X.25, you cannot netboot via a broadcast. You must netboot from a specific host. Also, an **x25 map** command must exist for the host that you netboot from.The **x25 map** command is used to map an IP address into an X.121 address. There must be an **x25 map** command that matches the IP address given on the **boot system** command line. The following is an example of such a configuration:

```
boot system gs3-k.100 131.108.126.111
!
interface Serial 1
ip address 131.108.126.200 255.255.255.0
encapsulation X25
x25 address 10004
x25 map IP 131.108.126.111 10002 broadcast
lapb n1 12040
clockrate 56000
```

In this case, 10002 is the X.121 address of the remote router that can get to host 131.108.126.111.

The remote router must have the following **x25 map** entry:

```
x25 map IP 131.108.126.200 10004 broadcast
```

This entry allows the remote router to return a boot image (from the netboot host) to the router netbooting over X.25.