

# Chapter 1

## Configuring X.25

---

# 1

Cisco Systems terminal servers support the use of X.25 as a transport mechanism for Internet packets. Cisco Systems servers use X.25 for transferring raw data over X.25 networks. Refer to the X.25 Recommendation, published by the International Telegraph and Telephone Consultative (CCITT). The following information is provided in this chapter:

- X.25 interface and encapsulation methods
- Configuring X.25 parameters
- Building X.29 access lists
- Monitoring X.25 traffic and protocol specifications with the terminal server

---

### *Cisco's Implementation of X.25*

Cisco terminal servers have an option that allows connection to an X.25 Public Data Network (PDN). Like Cisco routers, this X.25 connection allows transport of the TCP/IP packets across the X.25 packet switching network.

Cisco terminal servers do not support an X.25 PAD function, so they are unable to communicate with hosts directly connected to the X.25 PDN. The terminal servers encapsulate TCP/IP packets in X.25 packets for transfer over a packet switching network. These packets can be received by Cisco routers, TRouters, or other Cisco terminal servers.

The connection to the packet switching network is through a synchronous line.

---

**Note:** To connect to an X.25 host on a PDN, Cisco's IGS or CPT protocol translators can be used. Terminal servers communicate with protocol translators via the Telnet protocols and the CPT translates this into X.25, which allows connection to an X.25 host. The CPT supports all PAD functions (X.28, X.29, and X.3).

---

---

## X.25 Encapsulation Methods

The **encapsulation** subcommand of the interface configuration command sets up X.25 processing with default parameters on a serial interface. The command syntax is:

**encapsulation** *method*

This subcommand takes one argument, *method*, which specifies the type of X.25 support (commercial or DDN) and whether the connection is DTE (Data Terminal Equipment) or DCE (Data Communications Equipment). DDN X.25, a variant of X.25 with automatic Internet-to-X.121 address conversion, is used primarily to pass Internet traffic on the Defense Data Network. Most connections to X.25 switches are DTE. Refer to the “Configuring the System” chapter or a list of serial encapsulations supported with the terminal server.

### *Example:*

The following configuration commands set up a DTE connection to a commercial X.25 network on a serial interface:

```
interface serial 0
encapsulation x25
```

To adjust the Level 2 (LAPB) parameters of an X.25 interface, use the **lapb** subcommand of the **interface** configuration command.

---

## Configuring the X.25 Parameters

The Cisco terminal server software provides subcommands to configure the standard Level 2 and Level 3 X.25 parameters and user facilities.

This section discusses the commands and procedures needed to set these parameters including interface addresses, virtual circuit channel sequences, and addresses.

### *Setting the X.25 Level 2 (LAPB) Parameters*

X.25 Level 2, or LAPB (Link Access Procedure, Balanced), is a data encapsulation protocol that operates at Level 2 (the data link level) 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.

LAPB parameters are set with the **lapb** interface subcommand. The interface must be running with either a LAPB or X.25 encapsulation method specified by the **encapsulation** interface subcommand. The command syntax is:

**lapb** *parameter value*

This subcommand takes two required arguments, *parameter* and *value*.

The argument *parameter* is one of several keywords described in the following text, and the argument *value* is a decimal number representing a period of time, a bit count, or a frame count, depending on parameter. Table 1-1 summarizes the LAPB parameters.

**Table 1-1** LAPB Parameters

Parameter	Value	Value Range	Default
k	<i>frames</i>	1-7	7
n1	<i>bits</i>	1-16384	12000
n2	<i>times</i>	1-255	20
t1	<i>milliseconds</i>	1-64000	3000

S1613

---

**Note:** The LAPB th value is not included as a configurable parameter; the value is always F0.

---

### *Setting the Retransmission Timer*

The retransmission timer determines how long a transmitted frame can remain unacknowledged before the terminal server polls for an acknowledgment. To set the limit for the retransmission timer (the LAPB T1 parameter), use the **lapb t1** subcommand. The command syntax is:

**lapb t1** *milliseconds*

The argument *milliseconds* is the number of milliseconds from 1 through 64000. The default value is 3000 milliseconds.

For X.25 networks, the terminal server retransmission timer setting should match that of the network. Mismatched retransmission timers can cause excessive retransmissions and an effective loss of bandwidth.

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 terminal server will poll before the acknowledgment frame can return, which results in an effective loss of bandwidth. If the timer setting is too large, the terminal server waits longer than necessary before requesting an acknowledgment, which also reduces bandwidth.

To determine an optimal value for the retransmission timer, use the privileged EXEC command **ping** to measure the round-trip time of a maximum-sized frame on the link. Multiply this time by a safety factor that takes into account the speed of the link, the link quality, and the distance. A typical safety factor is 1.5. Choosing a larger safety factor can result in slower data transfer if the line is noisy. However, this disadvantage is minor compared to the excessive retransmissions and effective bandwidth reduction caused by a timer setting that is too small.

### *Setting Frame Parameters*

To specify the maximum number of bits a frame can hold, use the **lapb n1** interface subcommand:

#### **lapb n1** *bits*

The **n1** keyword specifies the maximum number of bits (N1) a frame can hold. The argument *bits* is the number of bits from 1 through 12000, and must be a multiple of eight. The default value is 12000 bits (1500 bytes).

When connecting to an X.25 network, use the N1 parameter value set by the network administrator, which is the maximum size of an X.25 packet. When using LAPB over leased lines, the N1 parameter should be eight times the MTU.

To specify the maximum number of times an acknowledgment frame can be retransmitted, use the **lapb n2** interface subcommand:

#### **lapb n2** *retries*

The argument *retries* is the retransmission count from 1 through 255. The default value is 20 retransmissions.

To specify the maximum permissible number of outstanding frames, called the “window size,” use the **lapb k** interface subcommand:

#### **lapb k** *window-size*

The argument *window-size* is a packet count from 1 to 7. The default value is 7 packets.

### **Example:**

The following configuration commands change the LAPB window size (the K parameter) to 3 packets:

```
interface serial 0
lapb k 3
```

To adjust the Level 3 (packet level) parameters of an X.25 interface, use the **x25** subcommand of the **interface** configuration command.

## Setting the X.25 Level 3 Parameters

This section describes the keywords for the **x25** subcommand of the interface configuration command, used to adjust certain X.25 Level 3 parameters. 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 the Cisco defaults.

---

**Note:** If you connect a terminal server to an X.25 network, use the parameters set by the network administrator. Also, note that the X.25 Level 2 parameters described in “Setting the X.25 Level 2 (LAPB) Parameters” earlier in this section affect X.25 Level 3 operations.

---

### Setting the X.25 Interface Address

To set the X.121 address of a particular network interface, use the **x25 address** subcommand. The address is assigned by the X.25 network:

**x25 address** *X.121-address*

The argument *X.121-address* is a variable-length X.121 address.

The value must match that assigned by the X.25 network.

#### **Example:**

The following configuration commands set the X.121 address for the interface:

```
interface serial 0
x25 address 00000123005
```

To display the current X.25 parameter settings, use the EXEC command **show interfaces**. Refer to the “Monitoring X.25” section later in this chapter for more information.

### Configuring the Virtual Circuit Channel Sequence

An important part of X.25 operation is the virtual circuit channel sequence. This sequence is a range of virtual circuit channel numbers broken into four groups (listed here in numerically increasing order):

1. Permanent virtual circuits
2. Incoming calls
3. Incoming and outgoing (two-way) calls
4. Outgoing calls

Several X.25 parameters determine the numerical ranges of the last three groups; the range for permanent virtual circuits falls numerically below the incoming call range.

X.25 communications devices use the virtual circuit channel sequence when allocating virtual circuits. When initiating a call, these devices must search for an available channel in the sequence in one of two ways. For outgoing calls (made by a DTE device), the search starts at the upper end of the outgoing call range and proceeds in the direction of decreasing channel numbers. The search continues until the device finds an available channel or reaches the lower limit of the two-way call range.

For incoming calls (handled by a DCE device), the search for channel numbers to allocate starts at the lower end of the incoming call range, and proceeds in the direction of increasing channel numbers. The search continues until the device finds an available channel or reaches the upper limit of the two-way call range. The DTE and DCE devices fail to find an available channel only if the overall sequence range is very small or when all of the channels are in use.

To set the upper and lower limit parameters of the channel ranges, use the **x25** subcommand keywords listed in Table 1-2. Each keyword takes a channel number as its argument. Note that the values for these parameters must be the same on both ends of an X.25 link.

The default lower limit for all channel ranges on the terminal server is 1; the default upper limit for all ranges is 1024. These defaults reflect a simple approach to allocating the channel ranges: assign the same low value to all lower limits, and assign the same high value to all upper limits. This approach causes the three ranges to overlap and become one large range.

**Table 1-2** Range Limit Keywords for the Virtual Circuit Channel Sequence

Keyword	Limit Type	Range	Default
lic	Lower limit, incoming call range	1-4095	1
hic	Upper limit, incoming call range	1-4095	1024
ltc	Lower limit, two-way call range	1-4095	1
htc	Upper limit, two-way call range	1-4095	1024
loc	Lower limit, outgoing call range	1-4095	1
hoc	Upper limit, outgoing call range	1-4095	1024

S1614

### *Setting the Highest Incoming Channel*

The **hic** keyword sets the highest incoming channel (HIC).

**x25 hic channel**

The argument *channel* is a channel number from 1 through 4095. The default value is 1024.

### *Setting the Highest Outgoing Channel*

The **hoc** keyword sets the highest outgoing channel (HOC).

**x25 hoc channel**

The argument *channel* is a channel number from 1 through 4095. The default value is 1024.

### *Setting the Highest Two-Way Channel*

The **htc** keyword sets the highest two-way channel (HTC).

**x25 htc channel**

The argument *channel* is a channel number from 1 through 4095. The default value is 1024.

### *Setting the Lowest Incoming Channel*

The **lic** keyword sets the lowest incoming channel (LIC).

**x25 lic channel**

The argument *channel* is a channel number from 1 through 4095. The default value is 1.

### *Setting the Lowest Outgoing Channel*

The **loc** keyword sets the lowest outgoing channel (LOC).

**x25 loc channel**

The argument *channel* is a channel number from 1 through 4095. The default value is 1.

### *Setting the Lowest Two-Way Channel*

The **ltc** keyword sets the lowest two-way channel (LTC).

**x25 ltc channel**

The argument *channel* is a channel number from 1 through 4095. The default value is 1.

### **Example:**

These commands set the following channels:

```
x25 hic 20
x25 htc 20
x25 hoc 20
```

Table 1-3 illustrates the effect on channels 01 through 20 of the commands in the preceding example.

**Table 1-3** Example Command Effects on Channels

<b>01-20:</b>	— incoming	S1615
<b>01-20:</b>	— either incoming or outgoing	
<b>01-20:</b>	— outgoing	

### *Maintaining Virtual Circuits*

The terminal server can clear a Switched Virtual Circuit (SVC) after a set period of inactivity. To set this period, use the **x25 idle** interface subcommand. The command syntax is:

**x25 idle** *minutes*

The argument *minutes* is the number of minutes in the period. The default value is 0 (zero), which causes the terminal server to keep the SVC open indefinitely. Both calls originated and terminated by the terminal server are cleared.

To clear all virtual circuits at once, use the privileged EXEC command **clear x25-vc**. This command takes an interface type keyword (usually **serial**) and a unit number as arguments to identify the interface with which the virtual circuits are associated.

To clear a particular virtual circuit, the **clear x25-vc** command takes the two arguments described above and a logical circuit number (LCN) value as a third argument to specify the circuit.

To increase throughput across networks, you can establish up to eight switched virtual circuits to a host. To specify the maximum number of switched virtual circuits that can be open simultaneously to one host, use the **x25 nvc** interface subcommand:

**x25 nvc** *count*

The argument *count* is a circuit count from 1 to 8; the default is 1.

To establish or delete Permanent Virtual Circuits (PVCs), use the **x25 pvc** interface configuration subcommand. The syntax for the command is as follows:

**[no] x25 pvc** *circuit protocol-keyword protocol-address*

The argument *circuit* is a virtual circuit channel number and it must be less than the lower limit of the incoming call range in the virtual circuit channel sequence (set using the **lic** keyword). The argument *protocol-keyword* can be one of the protocol types mentioned above. The argument *protocol-address* is that of the host at the other end of the PVC.

---

**Note:** You must specify the required network-protocol-to-X.121 address mapping with an **x25 map** subcommand before you can set up a PVC.

---



To configure a PVC for a given interface, use the following versions of the **x25 pvc** interface subcommand. The syntax is as follows:

```
x25 pvc pvc-number interface interface-name pvc-number  
x25 pvc pvc-number ip address interface-name pvc-number
```

The argument *pvc-number* is the PVC number that will be used on the local interface (as defined by the primary interface command). The argument *interface-name* is the interface type and unit number (serial 0, for example), as specified by the **interface** or **ip address** keywords.

---

**Note:** When configuring X.25 to use a PVC, you must ensure that no traffic is sent toward a remote terminal server between the time the **x25 map** command is issued and the time that **x25 pvc** command is issued. Otherwise, the local system will create a switched virtual circuit (SVC), and then the PVC command will not be allowed. Map entries with the **broadcast** attribute are particularly likely to get traffic, due to routing protocol traffic. The simplest way to ensure that no traffic is sent while configuring is to shut down the interface while configuring it for a PVC.

---

### *Configuring the Ignore VC Timer*

Upon receiving a Clear Request for an outstanding Call Request, the X.25 support code immediately tries another Call Request, if it has more traffic to send. This can overrun some X.25 switches. To prevent this problem, use the **x25 hold-vc-timer** configuration command:

```
x25 hold-vc-timer minutes
```

The argument *minutes* is the number of minutes it takes to prevent calls to a previously failed destination. Incoming calls will still be accepted.

### *Configuring the X.25 Level 3 Retransmission Timers*

The X.25 Level 3 retransmission timers determine how long the terminal server must wait before retransmitting various Call Request packets. You can set these timers independently using the **x25** subcommand keywords listed in Table 1-4. Each keyword requires a time value in seconds as its argument. The last column shows the default timer values, in seconds. Four of the timers apply to DTE devices, and the other four apply to DCE devices.

**Table 1-4** Retransmission Timer Keywords and Defaults

<b>Keyword (DTE/DCE)</b>	<b>Affected Request Packet</b>	<b>Time(seconds) (DTE/DCE)</b>
t20/t10	Restart Request	180/60
t21/t11	Call Request	200/180
t22/t12	Reset Request	180/60
t23/t13	Clear Request	180/60

S1616

### *Setting DTE Restart Request Retransmission Timer*

The **t20** sets the limit for the Restart Request retransmission timer (T20) on DTE devices.

**x25 t20** *seconds*

The argument *seconds* is a time value in seconds. The default is 180 seconds.

### *Setting DCE Restart Request Retransmission Timer*

The **t10** keyword sets the limit for the Restart Request retransmission timer (T10) on DCE devices.

**x25 t10** *seconds*

The argument *seconds* is a time value in seconds. The default value is 60 seconds.

### *Setting the DTE Call Request Retransmission Timer*

The **t21** keyword sets the limit for the Call Request retransmission timer (T21) on DTE devices.

**x25 t21** *seconds*

The argument *seconds* is a time value in seconds. The default value is 200 seconds.

### *Setting the DCE Call Request Retransmission Timer*

The **t11** keyword sets the limit for the Call Request retransmission timer (T11) on DCE devices.

**x25 t11** *seconds*

The argument *seconds* is a time value in seconds. The default value is 180 seconds.

### *Setting the DTE Reset Request Retransmission Timer*

The **t22** keyword sets the limit for the Reset Request retransmission timer (T22) on DTE devices.

**x25 t22 seconds**

The argument *seconds* is a time value in seconds. The default value is 180 seconds.

### *Setting the DCE Reset Request Retransmission Timer*

The **t12** keyword sets the limit for the Reset Request retransmission timer (T12) on DCE devices.

**x25 t12 seconds**

The argument *seconds* is a time value in seconds. The default value is 60 seconds.

### *Setting the DTE Clear Request Retransmission Timer*

The **t23** keyword sets the limit for the Clear Request retransmission timer (T23) on DTE devices.

**x25 t23 seconds**

The argument *seconds* is a time value in seconds. The default value is 180 seconds.

### *Setting the DCE Clear Request Retransmission Timer*

The **t13** keyword sets the limit for the Clear Request retransmission timer (T13) on DCE devices.

**x25 t13 seconds**

The argument *seconds* is a time value in seconds. The default value is 60 seconds.

### *Updating 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.

Outgoing calls forwarded over a specific interface can have their calling X.121 address updated by using the **x25 use-source-address** subcommand:

**x25 use-source-address**

## Setting X.25 Packet Sizes

X.25 networks use maximum input and output packet sizes set by the network administration. You can set the terminal server input and output packet sizes to match those of the network with the **x25** subcommand keywords **ips** and **ops**, respectively:

**x25 ips** *bytes*  
**x25 ops** *bytes*

The argument *bytes* is a byte count in the range of 128 through 4096. The default value is 128 bytes.

Larger packet sizes are better, because smaller packets require more overhead processing.

---

**Note:** Set the **x25 ips** and **x25 ops** keywords to the same value unless your network supports asymmetry between input and output packets.

---

To send a packet larger than the X.25 packet size over an X.25 virtual circuit, a terminal server 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 with the M-bit set and reassembles them.

## Setting the Flow Control Modulus

X.25 supports flow control with a sliding window sequence count. The window counter restarts at zero upon reaching the upper limit, which is called the window modulus. To set the window modulus, use the **x25 modulo** interface subcommand:

**x25 modulo** *modulus*

The argument *modulus* is either 8 or 128. The default value is 8. The value of the modulo parameter must agree with that of the device on the other end of the X.25 link.

## Configuring Packet Acknowledgment

To specify upper limits on the number of outstanding unacknowledged packets, use the **x25** subcommand keywords **win** (for input window) and **wout** (for output window):

**x25 win** *packets*  
**x25 wout** *packets*

The argument *packets* is a packets count. The packet count for **win** and **wout** can range from 1 to the *window modulus*. The default value is 2 packets.

The **x25 win** determines how many packets the terminal server can receive before sending an X.25 acknowledgment; the number is one less than **win**. The **wout** limit determines how many sent packets can remain unacknowledged before the terminal server uses a hold queue. To maintain high bandwidth utilization, assign these limits the largest number that the network allows.

---

**Note:** Set **win** and **wout** to the same value unless your network supports asymmetry between input and output window sizes.

---

You can instruct the terminal server to send acknowledgment packets when it is not busy sending other packets, even if the number of input packets has not reached the **win** count. This approach improves line responsiveness at the expense of bandwidth. To enable this option, use the **x25 th** subcommand:

**x25 th** *delay*

The argument *delay* must be between 0 (zero) and the input window size. A value of 1 sends one Receiver Ready acknowledgment per packet at all times. The default value is 0 (zero), which disables the delayed acknowledgment strategy.

The terminal server sends acknowledgment packets when the number of input packets reaches the count you specify, providing there are no other packets to send. For example, if you specify a count of 1, the terminal server can send an acknowledgment per input packet.

## *Setting the X.25 Level 3 Facilities*

This section describes the **x25** keywords for setting the X.25 Level 3 Facilities.

### *Setting the Default Protocol*

To set the default protocol, use the **x25 default** interface subcommands:

**x25 default** *protocol*  
**no x25 default** *protocol*

The **default** keyword specifies the protocol assumed by the terminal server to interpret incoming calls with unknown Call User Data. The argument *protocol* can be the keyword **ip**.

There is no initial default value. If you do not use the **x25 default** subcommand, the terminal server clears any incoming calls with unknown Call User Data.

If you use this subcommand, the terminal server assumes that these calls use the specified protocol.

Use the **no x25 default** subcommand to remove the protocol specified.

This subcommand is used when an incoming call is received without identifying CALL USER DATA. Normally, the call is cleared when this subcommand is used; the incoming call is assumed to contain the specified default protocol.

### *Suppressing the Calling Address*

To omit the calling address in outgoing calls, use the **x25 suppress-calling-address** interface subcommands:

```
x25 suppress-calling-address  
no x25 suppress-calling-address
```

The **suppress-calling-address** keyword omits the calling (source) X.121 address in Call Request packets. This option is required for networks that expect only subaddresses in the calling address field. The calling address is sent by default.

Use the **no x25 suppress-calling-address** subcommand to reset this subcommand to the default state.

### *Suppressing the Called Address*

To omit the called address in outgoing calls, use the **x25 suppress-called-address** interface subcommands:

```
x25 suppress-called-address  
no x25 suppress-called-address
```

The **suppress-called-address** keyword omits the called (destination) X.121 address in Call Request packets. This option is required for networks that expect only subaddresses in the called address field. The called address is sent by default.

Use the **no x25 suppress-called-address** subcommand to reset this subcommand to the default state.

### *Defining a Packet Hold Queue*

To define the number of packets to be held until a virtual circuit is established, use the **x25 hold-queue** interface subcommand:

```
x25 hold-queue queue-size  
no x25 hold-queue [queue size]
```

The argument *queue-size* defines the number of packets. By default, this number is zero. Use the **no x25 hold-queue** command without an argument to remove this command from the configuration file; enter the command with a value of zero to return to the default.

### *Accepting Reverse Charge Calls*

To instruct the terminal server to accept all reverse charge calls, use the **x25 accept-reverse** interface subcommands:

```
x25 accept-reverse  
no x25 accept-reverse
```

The **accept-reverse** keyword causes the interface to accept reverse charge calls by default. This behavior can also be configured on a per-peer basis using the **x25 map** subcommand:

The **no** form disables this facility.

### *Forcing Packet-Level Restarts*

To force a packet-level restart when the link level is restarted, use the **x25 linkrestart** interface subcommand:

```
x25 linkrestart  
no x25 linkrestart
```

The **linkrestart** keyword restarts X.25 Level 2 (LAPB) when errors occur. This behavior is the default and is necessary for networks that expect this behavior. The **no** option disables this facility.

### *Setting the Packet Network Carrier*

To set the packet network carrier, use the following interface subcommand:

```
x25 rpoa name number
```

The **x25 rpoa** interface subcommand specifies a list of transit Recognized Private Operating Agencies (RPOAs) to use, referenced by name. The argument *name* must be unique with respect to all other RPOA names. It is used in the **x25 facility** and **x25 map** interface subcommands. The argument *number* is a number that is used to describe an RPOA.

### *Setting X.25 Parameters on a Per-Call Basis*

To override interface settings on a per-call basis, use the **x25 facility** interface subcommand:

```
x25 facility parameter value  
no x25 facility parameter value
```

The **facility** keyword enables X.25 facilities, which are sent between DTE and DCE devices to negotiate certain link parameters.

The arguments *parameter* and *value* specify one of the following:

- **cug** *number*—Specifies a Closed User Group *number* from 1 through 99 to provide an extra measure of network security.
- **packetsize** *in-size out-size*—Sets the size in bytes of input packets (*in-size*) and output packets (*out-size*). Both values should be the same.
- **reverse**—Reverses charges on all Call Request packets from the interface.
- **windowsize** *in-size out-size*—Sets the packet count for input windows (*in-size*) and output windows (*out-size*). Both values should be the same.
- **throughput** *in out*—Sets the requested throughput values for input and output throughput across the network.

- **rpoa name**—Specifies the list of transit RPOAs to use in outgoing Call Request packets.

### *Setting IP Precedence*

To enable or disable the ability to open a new virtual circuit based on the IP Type of Service (TOS) field, use the **x25 ip-precedence** interface subcommand.

#### **[no] x25 ip-precedence**

The **no x25 ip-precedence** option disables the ability to open a new virtual circuit based on the IP Type (TOS) field. By default, Cisco terminal servers open one virtual circuit for each type of service.

## *Address Mapping Issues*

The Cisco terminal servers can use three methods to map Internet addresses to X.121 addresses used by X.25.

---

**Note:** Mapping of IP addresses to X.121 addresses is necessary only when running IP over X.25.

---

A terminal server set up for DDN service uses the dynamic mapping technique specified in Appendix A of the DDN X.25 Host Interface Specification, available from the DDN Network Information Center (NIC). This technique has two severe limitations: it applies only to Class A Internet addresses, and it ignores the third octet of the Internet address; this technique works well for the DDN, but not for any other network.

You can establish the X.121 address of a particular network interface using the **x25 address** subcommand of the **interface** configuration command. If you do not specify the address, the terminal server uses the DDN mapping technique to obtain the X.121 address of an interface. This command is described earlier in this chapter in the “Setting the X.25 Level 3 Parameters” section.

You can set up the Internet-to-X.121 address mapping for the host using the **x25 map** subcommand of the **interface** configuration command. Because no defined protocol can dynamically determine such mappings, you must enter a mapping for each host with which the protocol translator will exchange traffic. The command syntax is as follows:

**[no] x25 map protocol-keyword protocol-address X.121-address [option1]... [option]**

For the terminal server, the argument *protocol-keyword* can only be **ip**. The *address* arguments specify the network-protocol-to-X.121 mapping.

The *option* arguments add certain features to the mapping specified, and can be any of the following, up to six. (They must be specified in the order listed).

- **reverse**—Specifies reverse charging for outgoing calls.



- **accept-reverse**—Causes the protocol translator to accept incoming reverse-charged calls. If this option is not present, the protocol translator clears reverse charge calls.
- **broadcast**—Causes the protocol translator to direct any broadcasts sent through this interface to the specified X.121 address.
- **cug number**—Specifies a Closed User Group number (from 1 to 99) for the mapping in the outgoing call.
- **nvc count**—Sets the number of virtual circuits (VCs) for this map/host. The default *count* is the **x25 nvc** setting of the interface. A maximum number of eight VCs can be configured for a single map/host.
- **packetsize in-size out-size**—Specifies input packet size *in-size* and output packet size *out-size* for the mapping in the outgoing call.
- **window-size in-size out-size**—Specifies input window size *in-size* and output window size *out-size* for the mapping in the outgoing call.
- **throughput in out**—Requests the amount of bandwidth through the X.25 network.
- **modulo size**—Specifies the maximum window size for this map. The argument *size* permits windows of 8 or 128 on the same interface.
- **nuid username password**—Specifies that a network ID facility be sent in the outgoing call with the specified user name and password.

You can define a static host-name-to-address mapping in the host cache using the **x25 host** subcommand. The command syntax is:

```
x25 host name 121-address [cud call-user-data]
no x25 host name
```

The argument *name* is the host name.

The keyword **cud call-user-data** sets the call user data field in the X.25 call request packet. This keyword is optional

---

## *X.25 Configuration Example*

The following list shows sample commands for configuring a terminal server interface to run X.25:

### *Example:*

```
interface serial 0
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
```

---

## Maintaining X.25

To clear all virtual circuits at once, use the privileged EXEC command **clear x25-vc**. This command takes an interface type keyword (usually **serial**) and a unit number as arguments to identify the interface with which the virtual circuits are associated.

To clear a particular virtual circuit, the **clear x25-vc** command takes the two arguments described above and a logical circuit number (LCN) value as a third argument:

```
clear x25 vc interface unit [lcn]
```

The **clear x25-vc** command clears all X.25 virtual circuits at once. The argument *interface* is the interface type. The argument *unit* is the interface unit number. The optional argument *lcn* clears the specified virtual circuit.

---

## Monitoring X.25

The terminal server provides EXEC **show** commands to provide information on interface operation and virtual circuit operation. Use the EXEC command **show interfaces** to display interface parameters and statistics. Use the EXEC command **show x25 vc** to display virtual circuit parameters and statistics.

### Displaying Interface Parameters and Statistics

To display parameter information for an interface using X.25 Level 3 protocol, use the EXEC command **show interfaces**. The command syntax is:

```
show interfaces
```

The following is a sample command output for a serial interface configured for DDN X.25 encapsulation:

```
Serial 0 is up, line protocol is up
Hardware type is slow Serial
Internet address is 192.31.7.50, subnet mask is 255.255.255.240
MTU 1006 bytes, BW 56 Kbit, DLY 20000 usec, rely 255/255, load 4/255
Encapsulation is DDNX25, loopback is not set
LAPB state is CONNECT, T1 300, N1 8232, N2 20, K 7
VS 4, VR 1, RCNT 0, Remote VR 4, Retransmissions0
IFRAMEs 296326/297717 RNRs 0/0 REJJs 7/0 SABMs 2/1 FRMRs 0/0 DISCs 0/0
X25 address 000000020300, state R1, modulo 8, idle 0, timer 0, nvc 2
window size: input 6, output6, Packet size: input 1024, output 1024
```

```

Timer: T20 10 T21 10 T22 10 T23 10 TH 0
Channels: Incoming 1-1024 Two-wy 1-1024 Outgoing 1-1024
RESTARTs 3/5 CALLS 2+1/1+0 DIAGs 0/0
Last input 0:00:00, output 0:00:00, output hang never
Output queue max 10, 0 drops; input queue 0/75, 0 drops
Five minute input rate 1157 bits/sec, 10 packets/sec
Five minute output rate 1011 bits/sec, 10 packets/sec
570893 packets input, 39013225 bytes, 0 no buffer
Received 0 broadcasts, 0 runts, 0 giants
16 input errors, 16 CRC, 0 frame, 0 overrun, 0 ignored, 0 abort
593482 packets output, 142141161 bytes
0 output errors, 0 collisions, 0 interface resets, 0 restarts

```

In the command output certain fields, such as `IFRAMES` on the eighth line and `RESTARTs` on the thirteen line, show counts of the frame type using the format “sent/received.” For example, `SABMs 2/1` indicates that two SABM frames were sent and one was received.

On the first line, the `LCI` field displays the virtual circuit number. The `State` field displays the state of the virtual circuit (which is independent of the states of other virtual circuits); `D1` is the normal ready state. (See the CCITT X.25 Recommendation for a description of virtual circuit states.) The `Interface` field shows the interface used for the virtual circuit.

On the ninth line, the `idle` field shows the number of minutes the terminal server waits before closing idle switched virtual circuits; a value of zero indicates that the terminal server holds circuits open indefinitely. In the `timer` field, a non-zero value indicates the number of seconds until the X.25 process timer expires.

On the thirteenth line the `CALLs` field indicates the number of Call Request messages sent and received, and the number of send/receive failures. The format is “sent + failures/received + failures.”

## *Displaying Mapping of Internet to X.121 Addresses*

The EXEC command **show x25 map** displays the mapping of Internet to X.121 addresses. The command syntax is:

```
show x25 map
```

The following is a sample command output:

```

Serial0: 192.31.7.49 000000010300 PERMANENT, BROADCAST, 2 LCNS: 1024, 1*
Serial0: 000000020300 INTERFACE

```

Each line of output shows the interface name, the Internet address, the X.121 address, and the type of mapping entry.

Address-mapping types are `PERMANENT` (entered with `x25 map interface` subcommand), `INTERFACE` (indicating the address of a terminal server network interface), and `DDN` (derived using the DDN mapping standard). If broadcasts are enabled for an address mapping, the keyword `BROADCAST` also appears in the output line.

If applicable, the line also shows the count of any logical channel numbers (LCNs) and a list of the `LCNs` for that address. An asterisks (\*) indicates the current `LCN`.

## Displaying Virtual Circuit Parameters and Statistics

The EXEC command **show x25 vc** displays the details of the active X.25 switched virtual circuits. The command syntax is:

```
show x25 vc [lcn]
```

Use the *lcn* argument to the **show x25 vc** command to examine a particular virtual circuit.

The following is a sample command output:

```
LC1: 1024, State: D1, Interface: Serial0
Started 5d04, last input 0:00:00, output 0:00:00
Connected to [192.31.7.49] <-> 000000010300 Precedent: 0
Window size input: 6, output: 6
Packet size input: 1024, output: 1024
PS: 3 PR: 3 ACK: 0 Remote PR: 3 RCNT: 1 RNR: FALSE
Retransmits: 0   Timer (secs): 0   Reassembly (bytes): 0
Held Fragments/Packets: 0/0
Bytes 6073909/3151247 Packets 147971/27955 Resets 0/0 RNRs 0/0 REJs 0/
0 INTs 0/0
```

On the first line, the `LCI` field displays the virtual circuit number. The `State` field displays the state of the virtual circuit (which is independent of the states of other virtual circuits); `D1` is the normal ready state. (See the CCITT X.25 Recommendation for a description of virtual circuit states.) The `Interface` field shows the interface used for the virtual circuit.

On the second line, the `Started` field shows the time elapsed since the virtual circuit was created, the `last input` field shows time of last input, and the `output` field shows time of last output.

On the third line, the `Connected to` field shows in brackets the network-protocol address and then the X.121 address. The `Precedent` field, which appears only if you have specified DDN encapsulation, indicates IP precedence.

The fourth and fifth lines show window and packet sizes. These sizes can differ from the interface default values if facilities were offered and accepted in the Call Request and Call Accepted packets.

On the sixth line, the `PS` and `PR` fields show the current send and receive sequence numbers, respectively. The `Remote PR` field shows the last `PR` number received from the other end of the circuit. The `RCNT` field shows the count of unacknowledged input packets. The `RNR` field shows the state of the Receiver Not Ready flag; this field is true if the network sends a receiver-not-ready packet.

On the seventh line, the `Retransmits` field shows the number of times a packet has been retransmitted. The `{Timer}` field shows a non-zero time value if a packet has not been acknowledged or if virtual circuits are being timed for inactivity. The `Reassembly` field shows the number of bytes received for a partial packet (a packet in which the “more data” bit is set).

On the eighth line, the “Fragments” part of the `Held Fragments/Packets` field shows the number of X.25 packets being held. (In this case, “Fragments” refers to the X.25 fragmentation of higher-level data packets.) The “Packets” part of the `Held Fragments/Packets` field shows the number of higher-level Protocol packets currently being held pending the availability of resources, such as the establishment of the virtual circuit.

On the last line, the `Bytes` field shows the total numbers of bytes sent and received. The `Packets`, `Resets`, `RNRs`, `REJs`, and `INTs` fields show the total sent and received packet counts of the indicated types. (RNR is Receiver Not Ready, REJ is Reject, and INT is Interrupt).

---

## Debugging X.25

These are the privileged X.25 debugging commands. The EXEC **debug** commands described in this section are used to troubleshoot X.25 connections. Generally, you enter these commands during troubleshooting sessions with Cisco Customer Engineers.

For each **debug** command, there is a corresponding **undebug** command that turns the display off.

### **debug lapb**

To enable logging of LAPB (X.25 Level 2) transactions, use the **debug lapb** EXEC command.

### **debug x25**

To log all the X.25 circuit activities and traffic, use the **debug x25** EXEC command. The **undebug x25** command ends this logging.

### **debug x25-events**

To log X.25 events, use the **debug x25-events** EXEC command.

This command enables the monitoring of all X.25 traffic but does not display information about X.25 data or acknowledgment packets. Unlike the **debug x25** command, the **debug x25-events** command does not report all data activity, thus reducing the volume of output.

### **debug x25-vc [lcn]**

To log X.25 activity on virtual circuits, use the **debug x25-vc** EXEC command.

This command enables logging of X.25 activity on all virtual circuits. To log this activity for a particular virtual circuit, specify an LCN by adding the optional argument *lcn*.

The following is a sample **debug x25-vc** command output:

```
Serial0: X25 I R1 RESTART (5) 8 lci 0 cause 7 diag 250
Serial0: X25 O R1 RESTART CONFIRMATION (3) 8 lci 0
Serial0: X25 O P2 CALL REQUEST (19) 8 lci 1
From(14): 31250000000101 To(14): 31109090096101
Facilities (0)
Serial0: X25 O P6 CLEAR REQUEST (5) 8 lci 1 cause diag 122
```

For each event, the first field identifies the interface on which the activity occurred, and the second field indicates that it was an X.25 event. The third field indicates whether the X.25 packet was input (I) or output (O). The fourth field is the state of the interface: R1 is the normal ready state, R2 indicates the DTE not-ready state, and R3 indicates the DCE not-ready state.

The fifth field is the type of the X.25 packet that triggered the event, and the sixth field (in parentheses) gives the total length of the X.25 packet in bytes. The seventh field is the window modulus. The eighth field (labeled *lci*) shows the virtual circuit number. The ninth field (labeled *cause*) gives the cause code, and the tenth field (labeled *diag*) gives the diagnostic code.

For Call Request and Call Connected packets, the terminal server shows additional information on separate lines. The *From* field shows the calling X.121 address and the *To* field shows the called X.121 address. The number in parentheses after the field name [(14)] specifies the number of digits in the address. The *Facilities* field indicates the length (in bytes) of the requested facilities and the facilities contents.

---

## X.25 Interface Subcommand Summary

This section provides an alphabetic list of all the interface subcommands used in the X.25 interface:

### **encapsulation** *encapsulation-type*

This subcommand takes one argument, *encapsulation-type*. For the purposes of an X.25 network configuration, this command specifies the type of X.25 support (commercial or DDN) and whether the connection is DTE (Data Terminal Equipment) or DCE (Data Communications Equipment). DDN X.25, a variant of X.25 with automatic Internet-to-X.121 address conversion, is used primarily to pass Internet traffic on the Defense Data Network. Most connections to X.25 switches are DTE. Refer to the “Configuring the System” chapter for a list of serial encapsulations supported.

### **lapb** *parameter value*

This subcommand takes two required arguments, *parameter* and *value*.

The argument *parameter* is one of several keywords described in the following text, and the argument *value* is a decimal number representing a period of time, a bit count, or a frame count, depending on parameter. Table 1-1 summarizes the LAPB parameters.

**lapb k** *window-size*

Specifies the maximum permissible number of outstanding frames, called the “window size.”

The argument *window-size* is a packet count from 1 to 7. The default value is 7 packets.

**lapb n1** *bits*

Specifies the maximum number of bits a frame can hold.

The **n1** keyword specifies the maximum number of bits (N1) a frame can hold. The argument *bits* is the number of bits from 1 through 12000, and must be a multiple of eight. The default value is 12000 bits (1500 bytes).

**lapb n2** *retries*

Specifies the maximum number of times an acknowledgment frame can be retransmitted.

The argument *retries* is the retransmission count from 1 through 255. The default value is 20 retransmissions.

**lapb t1** *milliseconds*

Sets the limit for the retransmission timer (the LAPB T1 parameter).

The argument *milliseconds* is the number of milliseconds from 1 through 64000. The default value is 3000 milliseconds.

**[no] x25 accept-reverse**

Instructs the terminal server to accept all reverse charge calls. This behavior can also be configured on a per-peer basis using the **x25 map** subcommand. The **no** variation resets the default state.

**x25 address** *X.121-address*

Sets the X.121 address of a particular network interface. The address is assigned by the X.25 network. The argument *X.121-address* is a variable-length X.121 address.

**[no] x25 default** *protocol*

Specifies or removes the protocol assumed by the terminal server to interpret calls with unknown Call User Data. The argument *protocol* can be set to **ip**.

**[no] x25 facility** *parameter value*

Overrides interface settings on a per-call basis and enables X.25 facilities, which are sent

between DTE and DCE devices to negotiate certain link parameters.

The arguments *parameter* and *value* may be as follows:

- **cug number**—Specifies a Closed User Group *number* from 1 through 99 to provide an extra measure of network security.
- **packetsize** *in-size out-size*—Sets the size in bytes of input packets (*in-size*) and output packets (*out-size*). Both values should be the same.
- **reverse**—Reverses charges on all Call Request packets from the interface.
- **window size** *in-size out-size*—Sets the packet count for input windows (*in-size*) and output windows (*out-size*). Both values should be the same.
- **throughput** *in out*—Sets the requested throughput values for input and output throughput across the network.
- **rpoa name**—Specifies the list of transit RPOAs to use in outgoing Call Request packets.

#### **x25 hic** *channel*

Sets the highest incoming channel (HIC). The argument *channel* is a channel number from 1 through 4095. The default value is 1024.

#### **x25 hoc** *channel*

Sets the highest outgoing channel (HOC). The argument *channel* is a channel number from 1 through 4095. The default value is 1024.

#### **x25 hold-vc-timer** *minutes*

Prevents overruns on X.25 switches for traffic through the VCs. The argument *minutes* is the number of minutes to prevent calls to a previously failed destination. Incoming calls will still be accepted.

#### **x25 htc** *channel*

Sets the highest two-way channel (HTC). The argument *channel* is a channel number from 1 through 4095. The default value is 1024.

#### **x25 idle** *minutes*

Sets the period of inactivity once an SVC has been cleared. The argument *minutes* is the number of minutes in the period. The default value is 0 (zero), which causes the terminal server to keep the SVC open indefinitely. Both calls originated and terminated by the terminal server are cleared.



**[no] x25 host *name 121-address [cud call-user-data]***

Defines a static host-name-to-address mapping in the host cache.

The argument *name* is the host name.

**[no] x25 ip-precedence**

Enables or disables the ability to open a new virtual circuit based on the IP Type of Service (TOS) field. By default, Cisco terminal servers open one virtual circuit for each type of service.

**x25 ips *bytes***

**x25 ops *bytes***

Set the terminal server packet size to match those of the network. The **ips** keyword specifies the terminal server input packet size while the keyword **ops** specifies the terminal server output packet size. The argument *bytes* is a byte count in the range of 128 through 1024. The default value is 128 bytes.

**x25 lic *channel***

Sets the lowest incoming channel (LIC). The argument *channel* is a channel number from 1 through 4095. The default value is 1.

**[no] x25 linkrestart**

Forces a packet-level restart when the link level is restarted and restarts X.25 Level 2 (LAPB) when errors occur. This behavior is the default and is necessary for networks that expect this behavior. The **no** variation resets the default state.

**x25 loc *channel***

Sets the lowest outgoing channel (LOC). The argument *channel* is a channel number from 1 through 4095. The default value is 1.

**x25 ltc *channel***

Sets the lowest two-way channel (LTC). The argument *channel* is a channel number from 1 through 4095. The default value is 1.

**[no] x25 map *protocol-keyword protocol-address X.121-address [option1 ... option]***

Specifies a network-protocol-to-X.121 address mapping such as Internet-to-X.121 or DECnet-to-X.121.

The argument *protocol-keyword* can be one protocol type:

## ip

The *address* arguments specify the network-protocol-to-X.121 mapping.

The *option* arguments add certain features to the mapping specified, and can be any of the following, up to six. (They must be specified in the order listed).

- **reverse**—Specifies reverse charging for outgoing calls.
- **accept-reverse**—Causes the terminal server to accept incoming reverse-charged calls. If this option is not present, the terminal server clears reverse charge calls.
- **broadcast**—Causes the terminal server to direct any broadcasts sent through this interface to the specified X.121 address.
- **cug number**—Specifies a Closed User Group number (from 1 to 99) for the mapping in the outgoing call.
- **nvc count**—Sets the number of virtual circuits (VCs) for this map/host. The default *count* is the **x25 nvc** setting of the interface. A maximum number of eight VCs can be configured for a single map/host.
- **packetsize** *in-size out-size*—Specifies input packet size *in-size* and output packet size *out-size* for the mapping in the outgoing call.
- **window size** *in-size out-size*—Specifies input window size *in-size* and output window size *out-size* for the mapping in the outgoing call.
- **throughput** *in out*—Requests the amount of bandwidth through the X.25 network.
- **modulo** *size*—Specifies the maximum window size for this map. The argument *size* permits windows of 8 or 128 on the same interface.
- **nuid** *username password*—Specifies that a network ID facility be sent in the outgoing call with the specified user name and password.

### **x25 modulo** *modulus*

Sets the modulus. The argument *modulus* is either 8 or 128. The default value is 8. The value of the modulo parameter must agree with that of the device on the other end of the X.25 link.

### **x25 nvc** *count*

Specifies the maximum number of switched virtual circuits that can be open simultaneously to one host. The argument *count* is a circuit count from 1 to 8; the default is 1.

### **[no] x25 pvc** *circuit protocol-keyword protocol-address*

Establishes or deletes Permanent Virtual Circuits (PVCs). The argument *circuit* is a virtual circuit channel number and it must be less than the lower limit of the incoming call range in the virtual circuit channel sequence (set using the **lic** keyword). The argument *protocol-keyword* can be one of the protocol types mentioned above. The argument *protocol-address* is that of the host at the other end of the PVC.

---

**Note:** You must specify the required network-protocol-to-X.121 address mapping with an **x25 map** subcommand before you can set up a PVC.

---

**x25 pvc** *pvc-number* **interface** *interface-name* *pvc-number*  
**x25 pvc** *pvc-number* **ip address** *interface-name* *pvc-number*

Configures a PVC for a given interface. The argument *pvc-number* is the PVC number that will be used on the local interface (as defined by the primary interface command). The argument *interface-name* is the interface type and unit number (serial 0, for example), as specified by the **interface** or **ip address** keywords.

**x25 rpoa name** *n*

Specifies a list of transit RPOAs to use, referenced by name. The argument *name* must be unique with respect to all other RPOA names. It is used in the **x25 facility** and **x25 map** interface subcommands. The argument *n* is a number that is used to describe an RPOA.

**[no] x25 suppress-calling-address**

Omits the calling (source) X.121 address in Call Request packets. This option is required for networks that expect only subaddresses in the calling address field. The calling address is sent by default. The **no** variation resets the default state.

**x25 t10** *seconds*

Sets the limit for the Restart Request retransmission timer (T10) on DCE devices. The argument *seconds* is a time value in seconds. The default value is 60 seconds.

**x25 t11** *seconds*

Sets the limit for the Call Request retransmission timer (T11) on DCE devices. The argument *seconds* is a time value in seconds. The default value is 180 seconds.

**x25 t12** *seconds*

Sets the limit for the Reset Request retransmission timer (T12) on DCE devices. The argument *seconds* is a time value in seconds. The default value is 60 seconds.

**x25 t13** *seconds*

Sets the limit for the Clear Request retransmission timer (T13) on DCE devices. The argument *seconds* is a time value in seconds. The default value is 60 seconds.

**x25 t20** *seconds*

Sets the limit for the Restart Request retransmission timer (T20) on DTE devices. The argument *seconds* is a time value in seconds. The default is 180 seconds.

**x25 t21** *seconds*

Sets the limit for the Call Request retransmission timer (T21) on DTE devices. The argument *seconds* is a time value in seconds. The default value is 200 seconds.

**x25 t22** *seconds*

Sets the limit for the Reset Request retransmission timer (T22) on DTE devices. The argument *seconds* is a time value in seconds. The default value is 180 seconds.

**x25 t23** *seconds*

Sets the limit for the Clear Request retransmission timer (T23) on DTE devices. The argument *seconds* is a time value in seconds. The default value is 180 seconds.

**x25 th** *delay*

Instructs the terminal server to send acknowledgment packets when it is not busy sending other packets, even if the number of input packets has not reached the **win** count, which improves line responsiveness at the expense of bandwidth.

The argument *delay* must be between 0 (zero) and the input window size. A value of 1 sends one Receiver Ready acknowledgment per packet at all times. The default value is 0 (zero), which disables the delayed acknowledgment strategy.

**x25 use-source-address**

Updates outgoing calls forwarded over a specific interface.

**x25 win** *packets*

**x25 wout** *packets*

Set the upper limits on the number of outstanding unacknowledged packets.

The **win** keyword specifies the upper limits of the number of outstanding unacknowledged packets in the input window. The **wout** keyword specifies the upper limits of the number of outstanding unacknowledged packets in the output window.

The argument *packets* is a packets count. The packet count for **win** and **wout** can range from 1 to the *window modulus*. The default value is 2 packets.