

# Chapter 1

## Configuring Translation

---

# 1

This chapter focuses on the configuration of Cisco's "one-step method" protocol translation facility. In addition, it provides configuration examples for typical protocol translation environments. The following information is provided:

- Overview of one-step translation methods supported by Cisco's protocol translation software.
- A definition of the **translate** command and its various options.
- A summary of a generalized environment used to illustrate protocol translation in this chapter.
- Illustrative application examples for protocol translation. Applications are separated into discrete sections.
- Example configurations for each of the applications described.

---

### *Cisco's Protocol Translation Options*

Cisco's protocol translation (also called application gateway) function translates virtual terminal protocols to allow devices running dissimilar protocols to communicate. Cisco protocol translation software supports Telnet (TCP), LAT, and X.25. One-step protocol translation software performs bi-directional translation between any of the following protocol pairs:

- Telnet <--> LAT
- Telnet <--> X.25
- LAT <--> X.25

---

**Note:** TN3270 and XRemote are also supported by the protocol translation software. However, to translate between these and other supported protocols, you must use the two-step method. Refer to the chapter “Using the Protocol Translator,” for information concerning two-step translations in general. Refer to the “Configuring TN3270” and “Configuring XRemote” chapters for information specific to TN3270 and XRemote, respectively.

---

The translation of virtual terminal protocols allows *connectivity* to devices running different protocol stacks. This option does not perform translation between other protocol services such as file transfer protocols.

Protocol translation has previously been available with the CPT product. The CPT is a stand alone unit that supports up to 100 concurrent translation sessions. Protocol translation is now also offered as a software option for the IGS multiprotocol router/bridge. This software option allows concurrent routing, bridging, and protocol translation in a single unit.

When concurrently performing protocol translation with either routing or bridging, the IGS supports up to 32 transparent translation sessions. If routing and bridging are not configured, the IGS supports up to 100 transparent translation sessions. This allows the IGS to match the current CPT product’s functionality. If the IGS is not configured for “one-step” protocol translation using the **translate** command, it can still handle up to 100 virtual terminal sessions using the “two-step” method. Refer to “Using the Protocol Translator” early in this manual for more information concerning one-step and two-step protocol translation procedures. If you use the IGS for routing or bridging and then decide to use the full 100 translation sessions, you must explicitly turn routing or bridging off and reboot the system to see the additional lines. With routing and bridging disabled, the IGS by default uses the proxy Address Resolution Protocol (ARP) to find the router. You can also configure a default router or use the Gateway Discovery Protocol.

---

**Note:** This chapter assumes a working knowledge of Cisco protocol translator and routing software conventions. It touches on specific configuration procedures for specifying system-wide facilities, as well as **translate** command examples. Before continuing with this chapter, be sure that you are familiar with system configuration, system management, LAT, TCP, and X.25 discussion provided in this manual. It also includes references to X.25 routing. Refer to the Cisco *Router Products Configuration and Reference* publication for a complete discussion of X.25 routing features.

---

---

## *Configuring One-Step Translations*

The ability to make single-step user connections to resources through Cisco protocol translators is made possible by the **translate** global subcommand. This section defines the **translate** command specifications and provides some notes regarding use of protocol-specific incoming and outgoing addressing options. A lengthy example of configuration file translation specifications is provided at the end of this section.

## Using the Translate Command

The **translate** command causes incoming LAT, TCP, or X.25 requests for connections to a specified destination address/host name to result in the automatic conversion to the specified outgoing connection type. Subsequently, the transmission of a connection request to the address specified in the **translate** command is executed.

---

**Note:** The translate command does not support TN3270 or XRemote one-step translations. To use the protocol translator to convert these protocols, you must use the two-step method as described in the chapter “Using the Protocol Translator.” Also, refer to Part III, “Transmission Protocols,” for details concerning connection services and options available for all protocols supported by the protocol translator.

---

The **translate** subcommand has the following general format:

```
translate inaddress [inoptions] outaddress [outoptions][globaloptions]
```

The **translate** subcommand causes incoming connections to the specified input protocol and address to result in the automatic formation of an outgoing connection to the specified protocol and address. Options *inaddress* and *outaddress* have the following format:

```
tcp ipaddress  
lat servicename  
x25 X.121address
```

You must specify a protocol name (**tcp**, **lat**, or **x25**).

The following brief descriptions summarize the differences between these arguments and their respective options.

### *Telnet/TCP Translation*

The *inaddress*/*outaddress* argument *ipaddress* is either the standard form (0.0.0.0 to 255.255.255.255) or the name of an ip host that can be resolved via Domain Name Service, IEN116 name service, or by explicit specification in an **ip host** command.

The options for the **tcp** argument include:

```
port nnn  
binary
```

The **port** option is used for outgoing TCP connections, or for matching only this port on incoming connections. The default port is port 23 (Telnet). For incoming connections, port 0 will match **any** port.

The **binary** option is for negotiating Telnet binary mode on the Telnet connection. This was the default in previous versions of the protocol translator software, and is set automatically if you enter an old format **translate** subcommand.

### *LAT Translation Options*

The argument *servicename* is an alphanumeric string that identifies a LAT service. The application of “servicename” may differ somewhat depending on whether it is being used on the input portion (inaddress) or the destination (outaddress) portion of the translate command. When used on the inaddress portion of a translate command the servicename is essentially the name of the service that users specify when trying to make a translated connection. This name can, but does not need to, match the name of final destination resource. This can be useful when making remote translated connections.

The options for the outgoing **lat** argument include:

**node** *nodename*  
**port** *portname*

---

**Note:** These apply only to the destination.

---

Using the **node** keyword followed by a LAT node name causes a connection to be made to a specified node which offers a service. By default, the connection is made to the highest-rated node which offered the service.

The keyword **port** specifies a destination LAT port name in the format of the remote system. This parameter is usually ignored in most timesharing systems, but is used by terminal servers which offer reverse-LAT services.

The **node** and **port** options can be combined.

### *X.25 Translation Options*

The **x25** inaddress/outaddress argument *X.121 address* can be up to 14 decimal numbers in length. It must conform to specifications provided in the *CCITT 1984 Red Book*. This number generally consists of a portion that is administered by the PDN and a portion that is locally assignable. You must be sure that the numbers that you assign are in agreement with addresses assigned to you by the X.25 service provider.

The X.121 addresses used in the `translate` command will generally be “sub-addresses” of the X.121 address of the X.25 network interface. Typically, the interface address will be a 12 digit number. Any additional digits are interpreted as a “sub-address.” The PDN still routes these calls to the interface, and the protocol translator itself is responsible for dealing with the extra digits in an appropriate way.

The options for the **x25** argument include:

**cud** *call-user-data*  
**profile** *profile*  
**reverse**

The option **cud** *call-user-data* sends the specified *call-user-data* text as part of the call user data (CUD) in an outgoing call request (after the protocol identification bytes).

The option **profile** *name* causes the X.3 PAD parameters to be set as defined in the profile created by the **x29 profile** subcommand, as defined in the “Configuring X.25 and X.3” chapter.

The option **reverse** provides reverse charging for X.25 on a per-call rather than a per-interface basis.

### *Global Translation Options*

Global options for the **translate** subcommand are as follows:

**access-class** *nnn*  
**local**  
**login**  
**quiet**  
**max-users** *nnn*

The **access-class** *nnn* keyword allows the **translate** subcommand to be used by source hosts that match the access list parameters. This is currently supported only for incoming TCP and X.25 connections. Here, the argument *nnn* is the number previously assigned to an access list (*nnn* is an integer).

The **local** keyword allows protocol negotiations (such as **ECHO** and **linemode**) to not be translated.

The **login** keyword requires that the user log in before the outgoing connection is made. This type of login is specified on the VTY's using the **login** command.

The **quiet** keyword suppresses printing of user-information messages.

The **max-users** keyword limits the number of simultaneous users of the translation to *nnn* (*nnn* is an integer).

The following is an example of the **translate** command format:

```
translate tcp 132.100.2.34 port 25 stream x25 31370012561 cud mail  
reverse access-class 22
```

---

## *Protocol Translation Configuration Examples*

The following examples show configuration files taken from operating installations. You may use them to help you understand how to configure your application.

The first example sets up incoming Telnet sessions to a public access catalog running on a UNIX host. The host is configured for 32 virtual circuits:

```
! The following commands set up the network services:
no service config
!   disables TFTP autoloading of configuration files
service domain
!   enables Domain Name System-based host-name-to-address translation
service pad
!   enables incoming PAD connections
!
! The following commands configure the interfaces:
interface ethernet 0
!   establishes the Ethernet interface
ip address 92.45.81.5 255.255.255.0
!   specifies the interface address and subnet mask
ip broadcast-address 92.45.81.255
!   specifies the Internet address to use for broadcasts
!
interface serial 0
!   establishes the X.25 interface
no ip address
!   disables IP processing on this interface
encapsulation x25
!   specifies X.25 DTE encapsulation
x25 address 100
!   specifies the X.25 address
!
! The following commands specify X.25 Level 2 and 3 parameters:
x25 htc 28
x25 hoc 28
x25 hic 28
!
name-server 92.45.81.2
!   specifies the server address for name and address resolution
domain-name Cisco.com
!   defines the default domain name
!
default-gateway 92.45.81.4
!   sets the gateway address
hostname cpt1
!   specifies "cpt1" as the name for prompts (example: cpt1>)
!
! The following commands establish addresses that form automatic
! PAD/Telnet connections. The "local" keyword provides for
! local PAD parameter change requests.
translate 92.45.81.128 10100 local
translate 92.45.81.129 10100
translate 92.45.81.254 10167 local
!
! These commands begin configuration requests for the console
! line (line 0):
```

```

line console 0
!   establishes console line
password secret
!   sets password for the console line
line vty 0 100
!   establishes virtual terminal lines
login
password alsoSecret
!   establishes a password for the virtual terminal lines
exec-timeout 2
!   sets the user input interval to 2 minutes
session-timeout 5
!   sets the session interval to 5 minutes
dispatch-character 10
!   defines ^-J as the character that transmits a packet
telnet transparent
!   causes the protocol translator to send a RETURN/NUL
!   instead of RETURN/LINE FEED
!
end

```

The following configuration example sets up incoming X.25 network service:

```

no service config
!   disables TFTP autoloading of configuration files
no service domain
!   disables Domain Name System-based host-name-to-address translation
!
! These commands set up the Ethernet interface:
interface Ethernet 0
address 182.32.159.50 255.255.255.0
!
! These commands set up the X.25 interface:
interface Serial 0
no ip address
encapsulation x25
x25 address 2
x25 hic 73
x25 htc 73
x25 hoc 73
!
! This command establishes the addresses that form automatic
! PAD/Telnet connections.
translate 182.32.159.51 201
!
name-server 255.255.255.255
!   specifies the server address for name and address resolution
ip alias 182.32.159.55 6101
!   assigns 182.32.159.55 to the service provided on TCP port 6101
end

```

Use the EXEC **show translate** command described in this section to obtain displays of translation activity.

## *Examining Translation Configuration*

The command **show translate** is used to show information about any protocol translation information configured using the **translate** configuration command. The command syntax is:

### **show translate**

The format for the output of the command is:

```
Translate From: X25 3137005406515 access-class 5
To:    TCP 199.31.7.17 Port 23
0/10 users active, 5 peak, 20 total, 1 failures
```

The first line contains the source protocol for the translation, and the second line contains the destination protocol and global options, all as specified in the original **translate** configuration command.

The last line in the display describes usage statistics for that particular translation. The first number 0 is the number of currently active users acting through the translate command. Following the slash, 10 is the max-users global parameter from the configuration (0 if no maximum was specified). The peak number 5 is the largest number of users concurrently using this translation. The total 20 is the total number of connections through this translation, and failures 1 is the number of attempted translations that failed for any reason (destination host not responding, access failure, login failure, resource exhaustion, and so on.)

More sample outputs:

```
Translate From: LAT LUCY
To:    TCP 132.100.2.113 Port 23
1/0 users active, 1 peak, 1 total, 0 failures
Translate From: X25 3137005406515 access-class 5
To:    TCP 199.31.7.17 Port 23
0/0 users active, 0 peak, 0 total, 0 failures
Translate From: LAT MATHOML
To:    TCP 199.31.7.17 Port 23
2/0 users active, 2 peak, 3 total, 0 failures
Translate From: TCP 132.100.1.25 Port 23
To:    X25 31370 Profile echo
0/0 users active, 0 peak, 0 total, 0 failures
```

---

## *Debugging Translation*

There is one privileged debugging command for translation: **debug translate**. It is described in this section and is used to troubleshoot translation activity. Generally, you enter this commands during troubleshooting sessions with Cisco Customer Engineers.



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

### **debug translate**

The **debug translate** command shows the startups of each type of protocol translation, and indicates failures of the **translate** command.

---

## *Application Environment Summary*

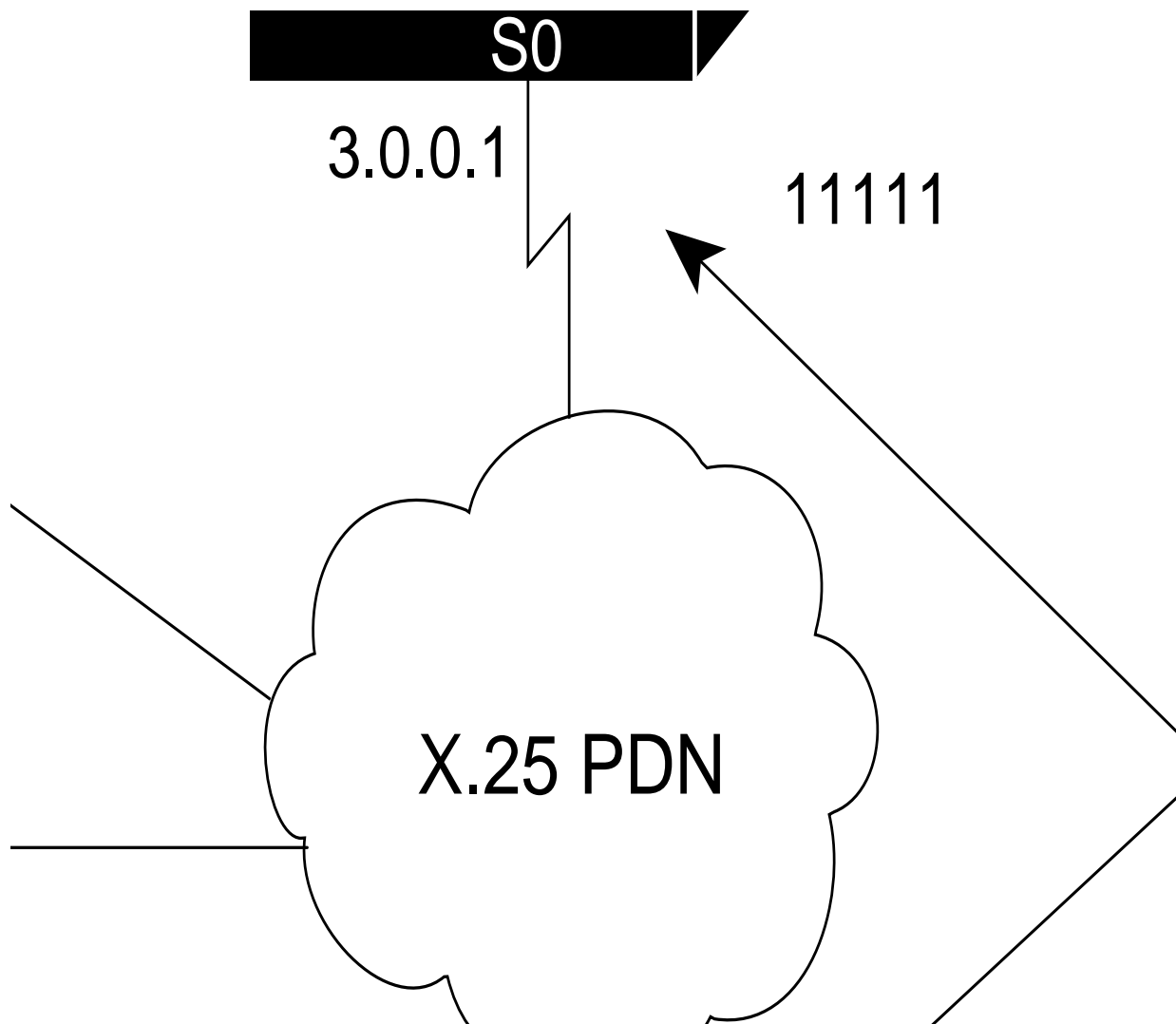
The remainder of this chapter is devoted to illustrating protocol translation examples. Most of these application examples are defined within the context of the network illustrated in Figure 1-1. The application environment described here focuses on IGS implementations; however, in applications solely featuring protocol translation, a CPT can be used as well.

---

**Note:** In the application illustrations that follow throughout this chapter, source and destination device icons used to illustrate the flow of translated information are shown with black type in outlined shapes. Other elements in the environment are shown with reverse type on solid black shapes (as shown for all elements in Figure 1-1).

---

*Figure 1-1* Summary Diagram Showing IGS Systems as Protocol Translators



The following example listings provide the basic global configuration subcommands and interface configuration subcommands to setup IGS-A (connected to Network A) and IGS-B (connected to Network B) illustrated in Figure 1-1. Refer to the IP and X.25 chapters under Transmission Protocols in this manual for more information concerning configuration details for these protocols.

**Example (IGS-A):**

```

!*****IGS ROUTER A - GENERAL CONFIGURATION*****
!
Interface ethernet 0
ip address 1.0.0.2 255.255.0.0
!
! enable LAT on interface
lat enabled
!
interface serial 0
encapsulation X.25
x25 address 1111
! The following parameters may depend on your network

```

```

x25 facility packetsize 512 512
x25 facility windowsize 7 7
!
! IP address and MAP command needed only if routing IP
ip address 3.0.0.1 255.255.0.0
x25 map ip 3.0.0.2 22222 broadcast
!
! Set up IP routing
router igrp 100
network 1.0.0.0
network 3.0.0.0
!
! Advertise as available for connections via LAT
! Use this name (IGS-A) if connecting via 2-step method
! (for connecting directly to a specific Protocol Translator)
lat service IGS-A enable
!
! Set up some IP host names/addresses
ip host IGS-A 1.0.0.2 3.0.0.1
ip host TCP-A 1.0.0.1
ip host TCP-B 2.0.0.1
ip host IGS-B 3.0.0.2 2.0.0.2

```

The partial configuration example provided above for IGS-A outlines baseline configuration for the unit's Ethernet and serial interfaces. It configures support for IP, LAT, and X.25. For more information concerning these global configuration commands and interface configuration subcommands, refer to the chapter "Configuring the System." The emphasis in this chapter is on using the **translate** command variations with this basic configuration (and the one provided for IGS-B) as the framework or context for discussing protocol translation options. Refer to the Cisco *Router Products Configuration and Reference* publication for information about router functions included in this example.

---

**Note:** The examples that follow focus on creating configurations that support "one-step" protocol translation. These connections can also be made using the "two-step" protocol translation method. Refer to "Using the Protocol Translator" early in this manual for more information concerning one-step and two-step protocol translation procedures.

---

**Example (IGS-B):**

```

!*****IGS ROUTER B - GENERAL CONFIGURATION*****
!
Interface ethernet 0
ip address 2.0.0.2 255.255.0.0
!
! enable LAT on interface
lat enabled
!
interface serial 0
encapsulation X.25
x25 address 22222
! The following parameters may depend on your network
x25 facility packetsize 512 512

```

```
x25 facility window size 7 7
!
! IP address and MAP command needed only if routing IP
ip address 3.0.0.2 255.255.0.0
x25 map ip 1.0.0.2 11111 broadcast
!
! Set up IP routing
router igrp 100
network 2.0.0.0
network 3.0.0.0
!
! advertise as available for connections via LAT
! Use this name (IGS-B) if connecting via 2-step method
! (for connecting directly to a specific Protocol Translator)
lat service IGS-B enable
!
! Set up some IP host names/addresses
ip host IGS-A 3.0.0.1 1.0.0.2
ip host TCP-A 1.0.0.1
ip host TCP-B 2.0.0.1
ip host IGS-B 2.0.0.2 3.0.0.2
```

---

**Note:** IP host names used to identify specific hosts can be specified either explicitly by using the **ip host** global configuration command, or by using Domain Name Service (DNS) facilities.

---

As with the IGS-A, the partial configuration provided for IGS-B outlines baseline configuration for the unit's Ethernet and serial interfaces. IGS-B is configured to support IP, LAT, and X.25. For more information concerning these global configuration commands and interface configuration subcommands, refer to the chapters "Configuring the System" and "Transmission Protocols." Again, the emphasis in this chapter will be on using **translate** command variations with this basic configuration as the framework or context for discussing protocol translation options.

The example configuration segments that follow are partial and representative in nature. They would be part of the complete configuration file for an individual protocol translator/router.

---

## *Local LAT to TCP*

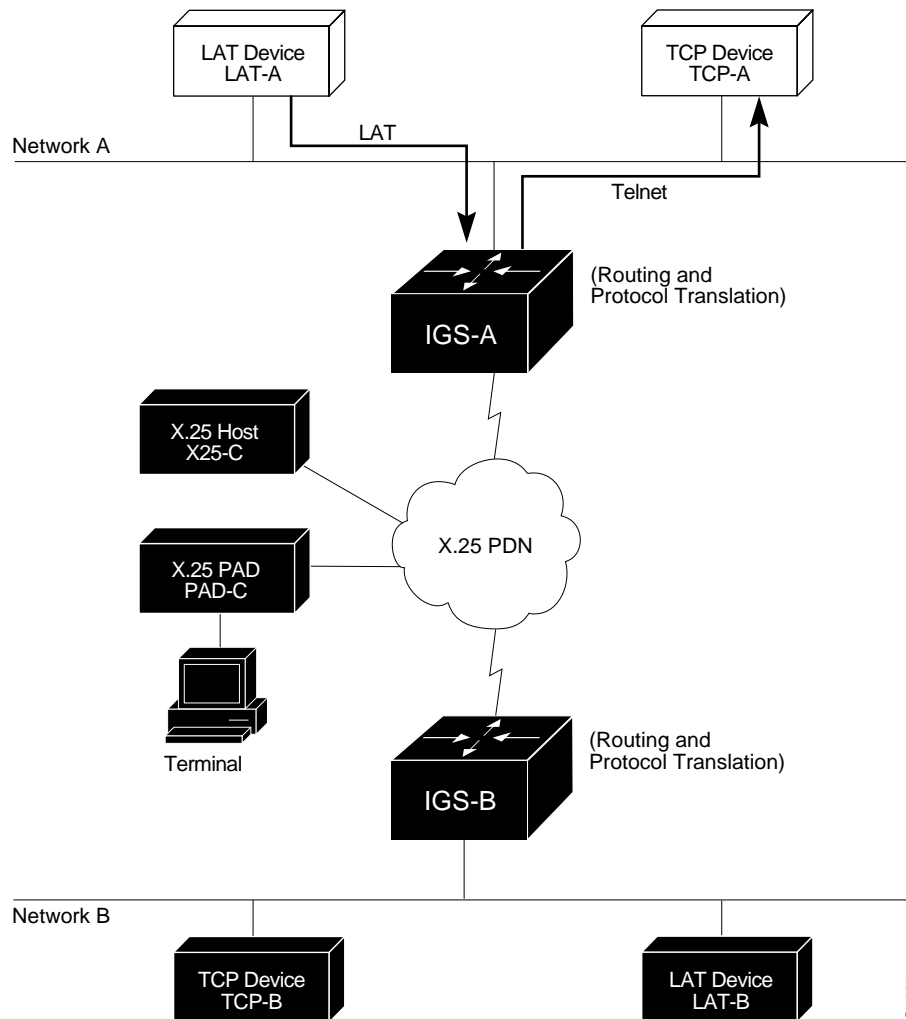
The IGS can translate between LAT and TCP (Telnet) traffic to allow communication among resources in these protocol environments. In Figure 1-2, the LAT device on Network A (LAT-A) is shown connecting to a TCP device (TCP-A).

---

**Note:** In the context of examples described in this chapter, a LAT device can be a LAT terminal server or LAT host, while a TCP device can be either a TCP terminal server or TCP host.

---

**Figure 1-2** Local LAT to TCP Translation



The following example configuration illustrates use of the **translate** global configuration subcommand to translate from LAT to TCP. It is applied to IGS-A.

**Example:**

```
! Translate LAT connections to TCP for connectivity to TCP-A
translate lat TCP-A tcp TCP-A
! Optional additional commands
lat service TCP-A ident Protocol Translation to TCP-A
```

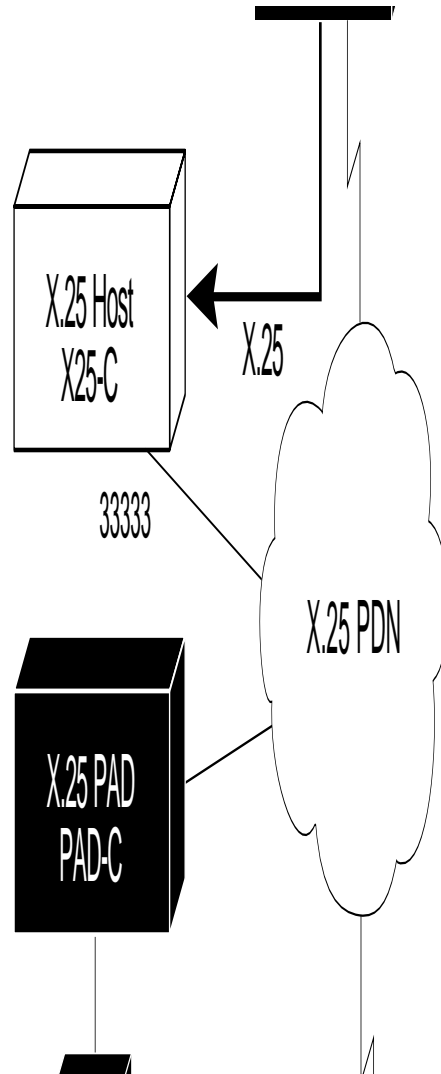
In the last command above, the text string “Protocol Translation to TCP-A” is text that is used as an identification for the LAT service named TCP-A (sent to other servers on the local network).

---

## *LAT to X.25 Host*

The IGS protocol translation option allows LAT devices to communicate with X.25 hosts through an X.25 PDN. In the application illustrated in Figure 1-3, a LAT device (LAT-A) is talking to an X.25 host (X25-C) with its LAT traffic translated to X.25.

*Figure 1-3* LAT to X.25 Host Translation



The following example configuration illustrates use of the **translate** global configuration subcommand to translate from LAT to X.25. It is applied to IGS-A. This configuration example includes an example of using *reverse* charging for connections. Setting the translate command to reverse charges causes the protocol translator to instruct the PDN to charge the destination for the connection. It is essentially a collect call. This reversal of charges must be pre-arranged with the PDN and destination location (on an administrative basis), or the call will not be accepted.

**Example:**

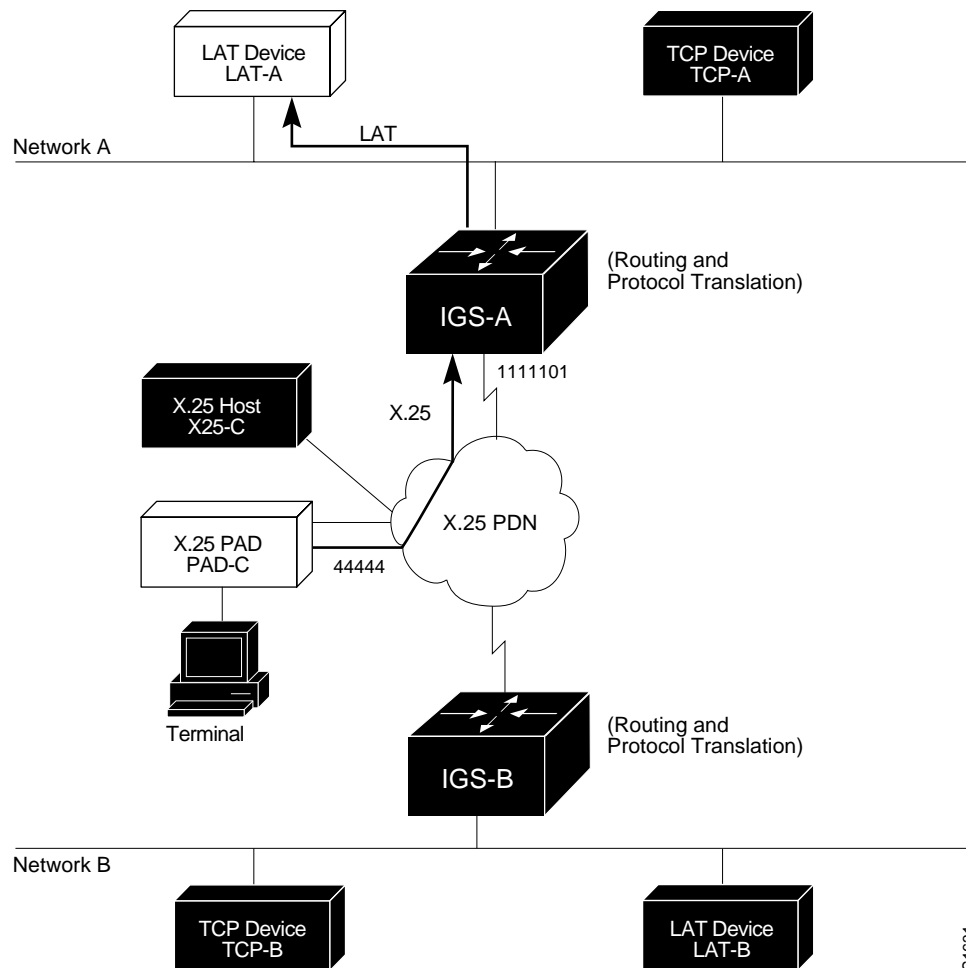
```
! Translate LAT to X.25 host, with reverse charging
translate lat X25-C x25 33333 reverse
!
! Specify optional X.25 hostname
x25 host X25-C 33333
```

---

## *X.25 PAD to LAT*

In addition to allowing LAT devices to communicate with X.25 hosts as illustrated in Figure 1-3, IGS protocol translation allows terminals connected to X.25 PADs to communicate with LAT devices on a remote LAN (Figure 1-4). X.25 PAD terminals make a call using an X.121 address which is translated to a LAT address. To the PAD terminal user, the connection appear to be a direct connection to a host on the X.25 PDN. The IGS also supports X.29 access lists, which allow you to restrict LAN resources (LAT or TCP) available to the PAD user.

**Figure 1-4** X.25 PAD to LAT Translation



S1631

### *Example (IGS-A):*

The following example configuration illustrates use of the **translate** global configuration subcommand to translate from an X.25 PAD to a LAT-device on a Network A. It is applied to IGS-A. The configuration example includes an access list that limits remote LAT-access through IGS-A to connections from PAD-C.

```

! Define X25 access list to only allow pad-c
x25 access-list 1 permit ^44444
x25 access-list 1 deny .*
!
! Set up translation
translate x25 1111101 lat LAT-A access-class 1

```

The preceding configuration example typifies the use of access lists in Cisco internetworking systems. The first two lines define the scope of “access-list 1”. The first line specifies that access-list 1 will permit all calls from X.121 address 44444. The caret symbol (^) specifies that the leading 4 is the beginning of the address number. Refer to the “Pattern Matching” appendix for details concerning the use of special characters in defining X.121 addresses. The second line of the definition explicitly denies calls from any other number.



This access list is then applied to all incoming traffic on the serial port for IGS-A (X.121 address 1111101) with the third configuration line in the example. However, it applies only to the **translate** command at the end of the example above. This **translate** command specifies that incoming X.25 packets on the serial line (with address 1111101) is translated to LAT and sent to LAT-A if it passes access-list 1 restrictions.

If multiple X.25 translate commands are defined, each must include unique X.121 address. It is also important that the CCITT protocol used in transferring packets match among X.121 addresses. This is specified in the protocol identification field of call-user data. This field specifies whether a packet is routed, translated, or handled as a virtual terminal connection.

---

**Note:** The X.121 address 1111101 used in the preceding example may be a subaddress of the address 11111 originally assigned to this serial port on IGS-A earlier in this chapter. However, that is *not* a requirement. The number to use in this command is negotiated (administratively) between the customer site's network management personnel and the PDN service provider. The number specified must match the number to which your PDN is sending data. That number may or may not be the number (or a subaddress of the number) administratively assigned to the Cisco protocol translator. It is up to the site administrator and the PDN to agree on a number to be used, as it is possible that the PDN can be configured to place calls on a given line that are intended for a destination that does *not* match the number assigned by site administration in the configuration file. Refer to the *1984 CCITT Red Book* specifications for more information concerning X.121 addresses.

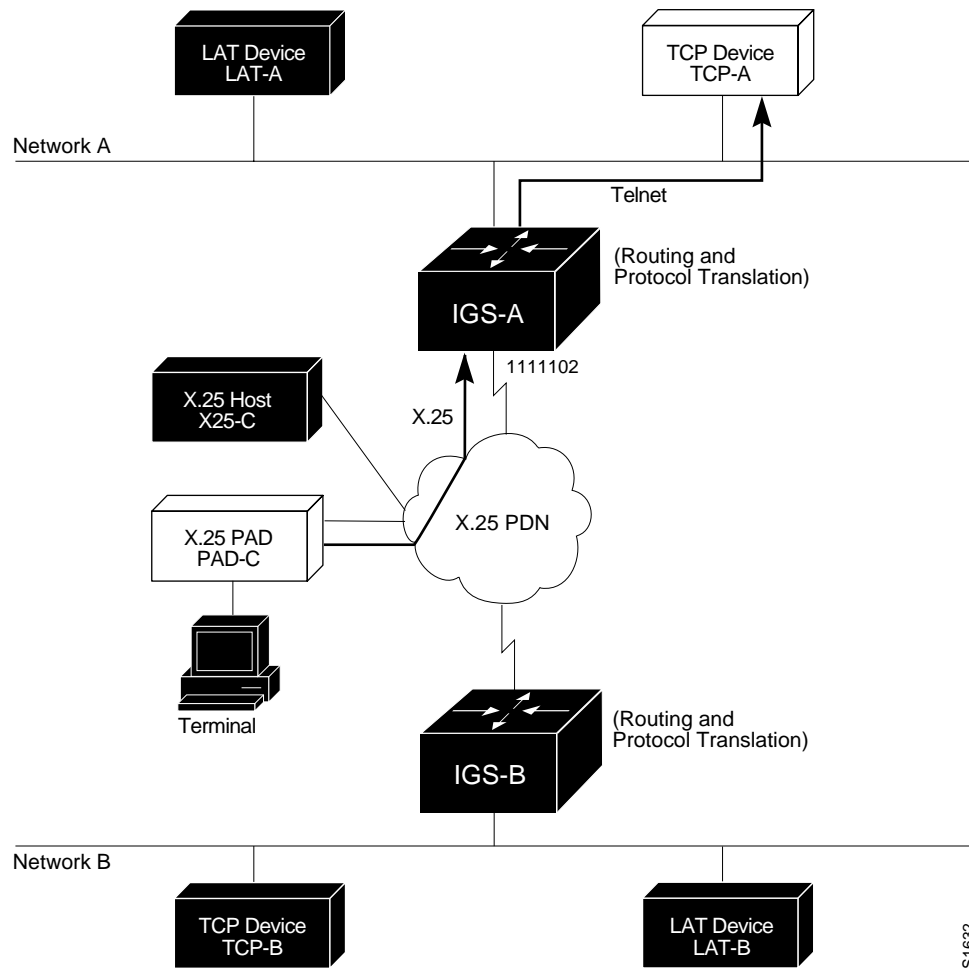
---

---

## *X.25 PAD to TCP*

Making a translated connection from an X.25 PAD to a TCP device (Figure 1-5) is analogous to the preceding X.25 PAD to LAT example. Instead of translating to LAT, the IGS-A configuration includes a statement to translate to Telnet (called TCP in the configuration syntax of Cisco protocol translation software). Note that a protocol translator can include statements supporting both translations (X.25 PAD to LAT and X.25 PAD to TCP). Different users on the same PAD can talk to X.25, LAT, or TCP devices.

*Figure 1-5* X.25 PAD to TCP Translation



S1632

The following example configuration illustrates use of the **translate** global configuration subcommand to translate from an X.25 PAD to a TCP-device on a Network A. It is applied to IGS-A.

**Example:**

```
! Set up translation
translate x25 1111102 tcp TCP-A
```

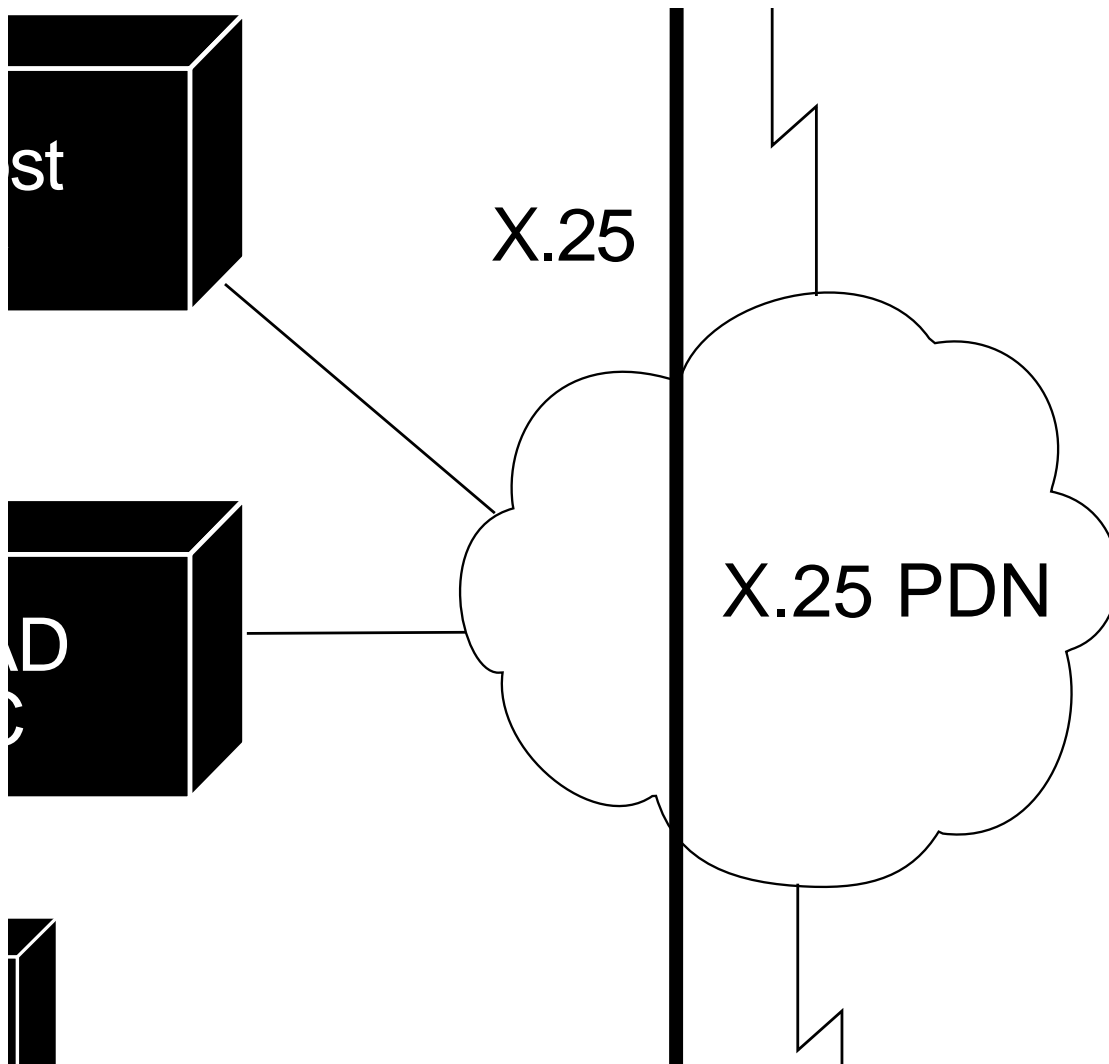
---

## LAT to LAT (via X.25 Translation)

Protocol translators can be used to provide transparent connectivity between LAT devices on different networks via an X.25 PDN. Figure 1-6 illustrates this application. In this application, the LAT device on Network A (LAT-A) first makes a virtual connection to the IGS on Network A using the LAT protocol.

The IGS translates the LAT packets into X.25 packets and sends them through the X.25 network to the IGS connected to Network B. This IGS translates the X.25 packets back to LAT packets and establishes a virtual connection to the LAT device on Network B (LAT-B). These handoffs are essentially handled transparently when configured for a “one-step” translation as described in the “Using the Protocol Translator” chapter.

*Figure 1-6* LAT to LAT via an X.25 PDN



The following configuration example illustrates the use of the **translate** global configuration subcommand to translate from LAT to X.25 and from X.25 to back to LAT to allow connection service to a LAT-device on a Network B from a LAT device (LAT-A) on Network A. It includes configurations for both IGS-A and IGS-B.

**Example (IGS-A):**

```
! Translate LAT to X.25
translate lat DISTANT-LAT x25 2222201
```

**Example (IGS-B):**

```
! Translate X.25 to LAT
translate x25 2222201 lat LAT-B
```

In the translate statement for IGS-A, *DISTANT-LAT* defines a LAT service name for IGS-A. When a connection to LAT-B is attempted by a user on device LAT-A, *DISTANT-LAT* is the target specified in a connection command.

In the **translate** configuration command for IGS-B, the LAT service on the target host (LAT-B) is *LAT-B*. *IGS-B* translates the incoming X.25 packets from 2222201 to LAT and transparently relays these packets to LAT-B.

An example connection request follows.

**Example Connection Command:**

```
local>connect DISTANT-LAT
```

With the example configurations specified for IGS-A and IGS-B above, the user connection command `connect DISTANT-LAT` (or any similar connection command) results in a connection attempt from LAT-A on Network A to LAT-B on Network B.

A translation configuration intended to force IGS-B to send information back from LAT-B to LAT-A is symmetrical to the prior configuration. The following examples illustrate this corresponding configuration (not shown in Figure 1-6):

**Example (IGS-B):**

```
! Translate LAT to X.25
translate lat FAR-LAT x25 1111103
```

**Example (IGS-A):**

```
! Translate X.25 to LAT
translate x25 1111103 lat LAT-A
```

---

**Note:** You can use the same name (for example, “LAT-B”) in the **translate** configuration subcommand for both IGS-A and IGS-B, but this symmetry is not required. Each protocol translator operates independently. The key is the common X.121 address used in both commands. If you prefer to have unique service names, set the names in each of the protocol translators to be the same.

---

---

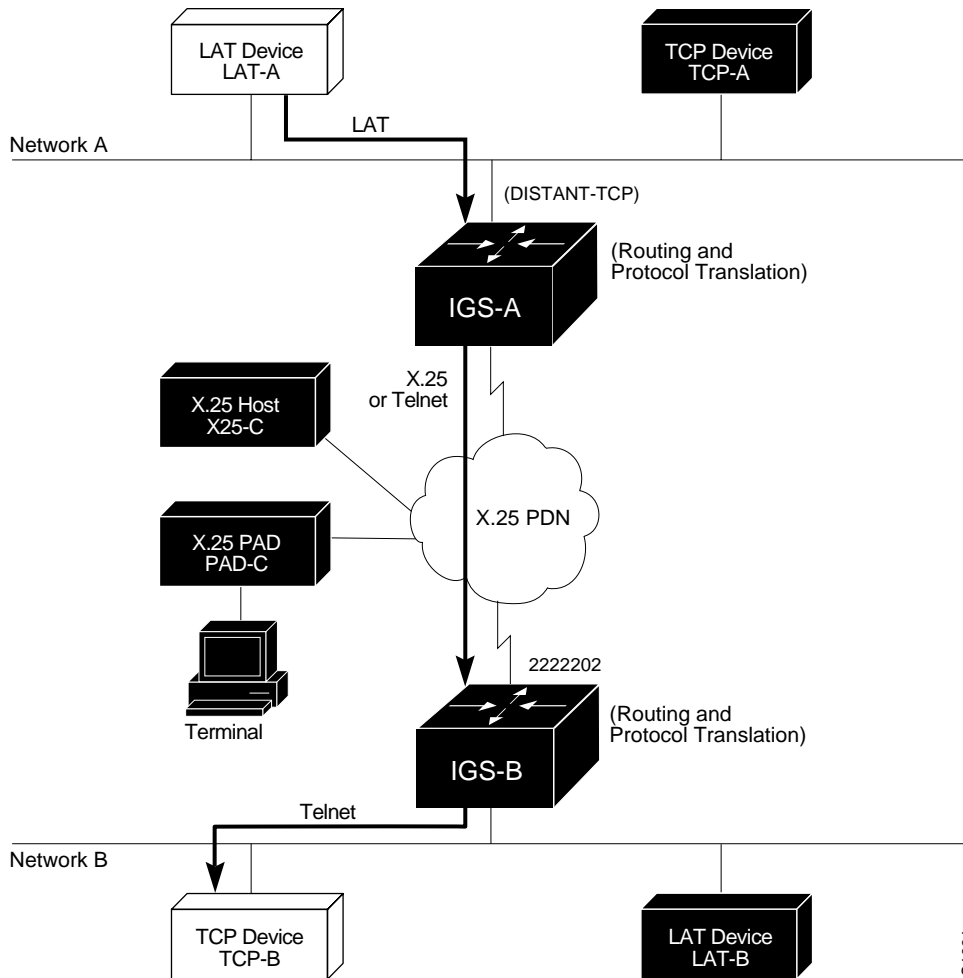
## *LAT to TCP (via X.25)*

Cisco protocol translators can be used to provide transparent connectivity between LAT and TCP devices on different networks via an X.25 PDN. In the application illustrated in Figure 1-7, the traffic from the LAT device on Network A is communicating with the TCP device on Network B. There are two ways to provide this connectivity.

The LAT traffic from Network A can be translated into either X.25 packets or Telnet packets to be sent out on the X.25 PDN. If the traffic is translated to Telnet by IGS-A, the packets are encapsulated within X.25 frames. In general, translating the traffic directly to X.25 is more efficient in this application since no encapsulation is necessary. X.25 packets have only five bytes of header information, while Telnet over X.25 includes 45 bytes of header information.

If the traffic is translated from LAT directly into X.25 frames by IGS-A, then IGS-B on Network B translates incoming packets intended for device TCP-B into Telnet. If IGS-A converts LAT to TCP (Telnet), then the Telnet traffic is being encapsulated in X.25 and sent on the X.25 network; the IGS on Network B strips off the encapsulation and routes the TCP packet. In this case, protocol translation is not needed on IGS-B.

*Figure 1-7* LAT to TCP via X.25



S1634

The following configuration examples illustrate use of the **translate** global configuration subcommand to translate from LAT to X.25 (IGS-A) and from X.25 to Telnet (IGS-B) allowing connection service to a TCP-device on a Network B (TCP-B) from a LAT-device on Network A (LAT-A). Example configurations are for both IGS-A and IGS-B.

**Example (IGS-A):**

```
! Translate LAT to X.25
translate lat DISTANT-TCP x25 2222202
```

**Example (IGS-B):**

```
! Translate X.25 to TCP
translate x25 2222202 tcp TCP-B
```

In the translate statement for IGS-A, *DISTANT-TCP* defines a LAT service name for IGS-A. When a connection to TCP-B is attempted by a user on device LAT-A, *DISTANT-TCP* is the target specified in a connection command.

In the **translate** configuration command for IGS-B, the Telnet service on the target host (TCP-B) is *TCP-B*. IGS-B translates the incoming X.25 packets from 2222202 to Telnet packets and transparently relays these packets to TCP-B.

An example connection request follows.

**Example:**

```
local>connect DISTANT-TCP
```

With the example configurations specified for IGS-A and IGS-B above, the user connection command `connect DISTANT-TCP` (or similar connection command) results in a connection attempt from LAT-A on Network A to TCP-B on Network B.

---

**Note:** You can use the same name (for example, *TCP-B*) in the **translate** configuration subcommand for both IGS-A and IGS-B, but this symmetry is not required. Each protocol translator operates independently. The key is the common X.121 address used in both commands. If you prefer to have unique service names, set the names in each of the protocol translators to be the same.

---

---

## *LAT to LAT over Frame Relay or SMDS*

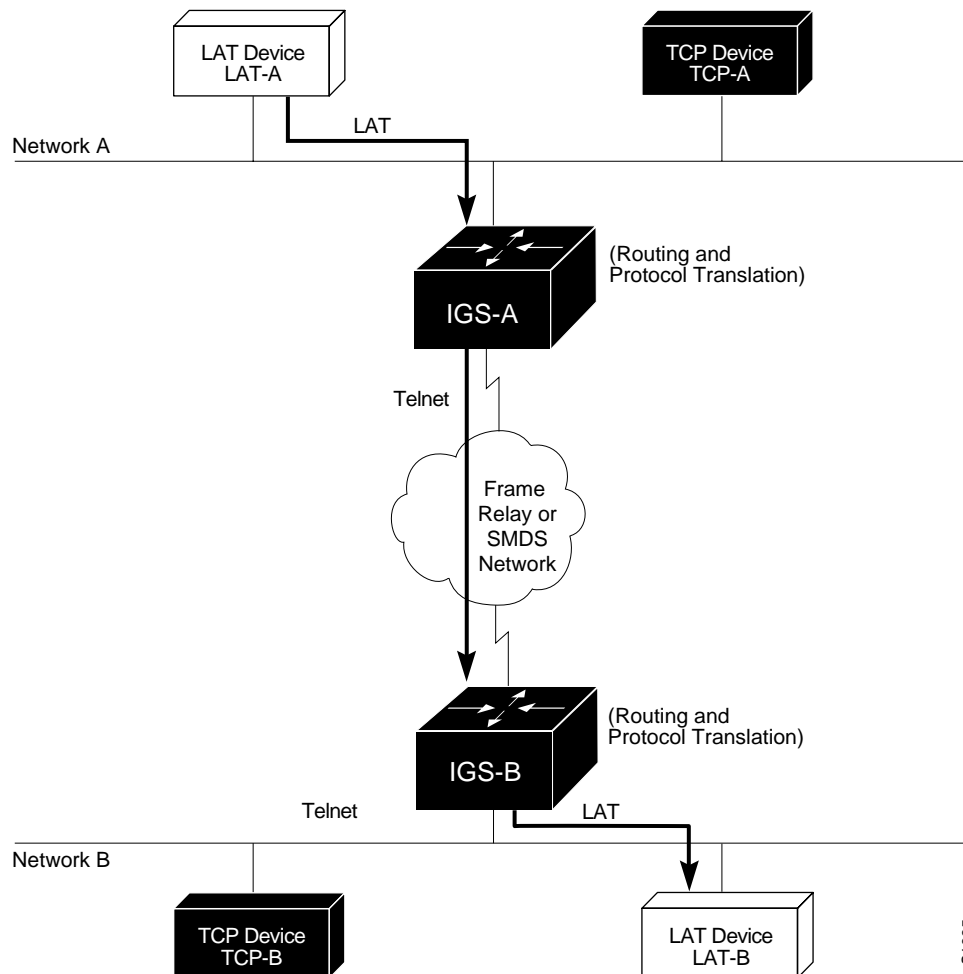
To transport LAT traffic over a Frame Relay or SMDS network, LAT must first be translated to TCP (Telnet). The TCP traffic is routed over the Frame Relay network and then retranslated back to LAT on the IGS on Network B. Refer to Figure 1-8 and the associated partial configuration example.

---

**Note:** The interface configurations for a Frame Relay or SMDS implementation specified differ from the specifications at the beginning of this chapter. For more information concerning Frame Relay and SMDS configuration, refer to the “Configuring Packet Switch Node Software” chapter of the Cisco *Router Products Configuration and Reference* publication.

---

**Figure 1-8** LAT to LAT over Frame Relay or SMDS



S1635

The following configuration example illustrates use of the **translate** global configuration subcommand to translate from LAT to LAT when the WAN uses Frame Relay or SMDS. Under this configuration, the IGS is routing encapsulated packets translated from LAT to TCP (Telnet) over the Frame Relay or SMDS network. Packets are then translated back to LAT on the other side of the Frame Relay or SMDS network. Example translation configurations for both IGS-A and IGS-B are included, but these examples do not include complete interface configuration specifics.

**Example (IGS-A):**

```
! Translate LAT to TCP/Telnet
translate lat DISTANT-LAT tcp IGS-B
```

**Example (IGS-B):**

```
! Translate TCP to LAT
translate tcp IGS-B lat LAT-B
```



---

**Note:** You can use the same name (for example, LAT-B) in the **translate** configuration subcommand for both IGS-A and IGS-B, but this symmetry is not required. Each protocol translator operates independently. The key is the common IP name used in both commands.

---

---

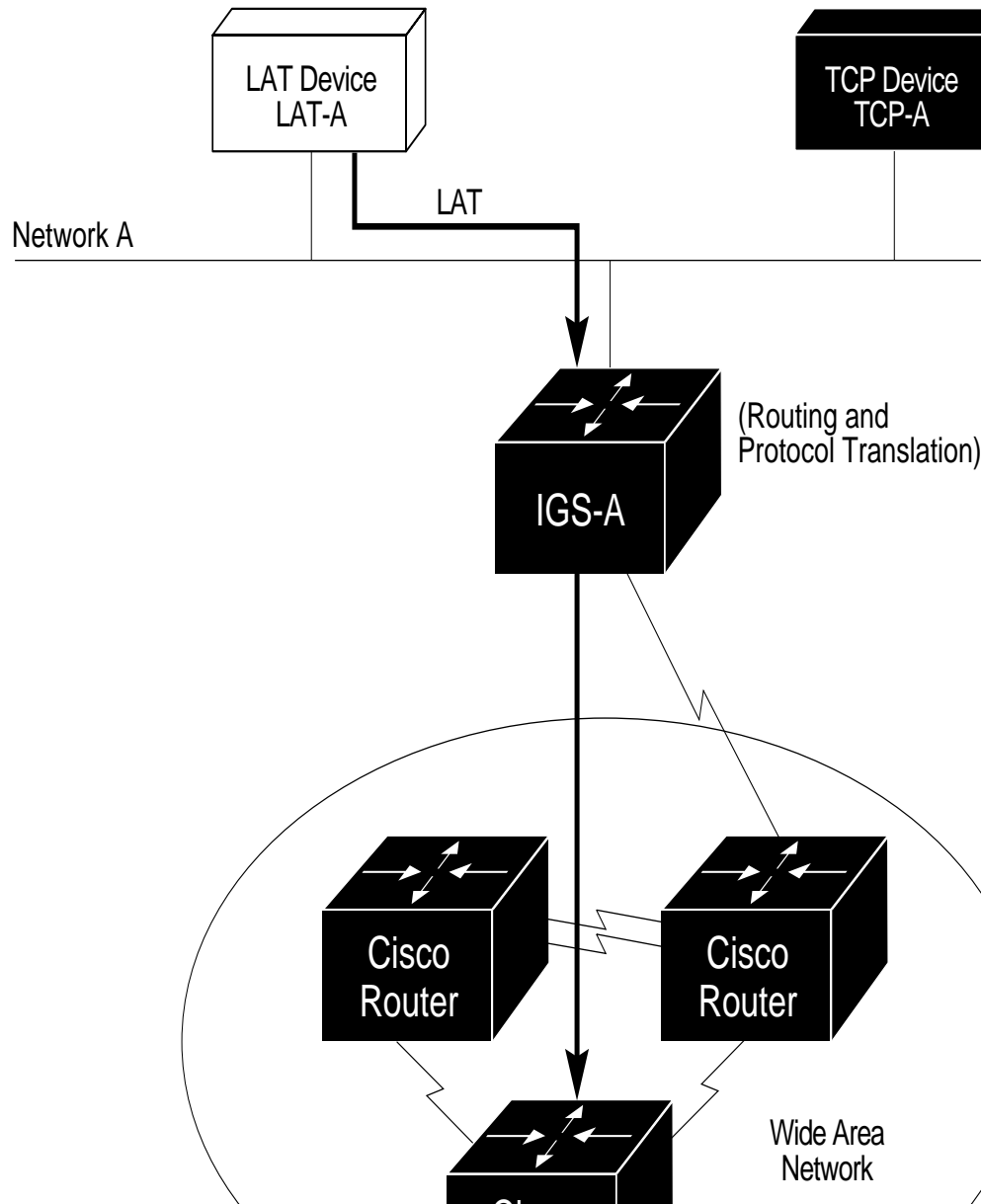
## *LAT to LAT (over an IP WAN)*

The IGS with protocol translation can also be used to connect LAT devices over a Wide Area backbone that only allows routable protocols (Figure 1-9). This dilemma faces organizations with LAT networks that are either isolated or are on their own internetwork.

With the IGS, LAT traffic can be translated to TCP and then routed on the WAN as Telnet traffic. The LAT connections stay local between the LAT device and the IGS. Connections are thus not susceptible to delays on the WAN.

This reduces the amount of traffic on the WAN since only the data from specific LAT sessions is forwarded on the WAN rather than all the LAT protocol status information packets.

**Figure 1-9** LAT to LAT over an IP WAN



The following configuration example illustrates use of the **translate** configuration subcommand to translate from LAT to LAT when an IP WAN media is used. Under this configuration, the IGS is routing encapsulated packets translated from LAT to TCP (Telnet) over the WAN network. Packets are then translated back to LAT on the other side of the WAN network. Example translation configurations for both IGS-A and IGS-B are included, but these examples do not include specifics of configuration for routers in the WAN. The examples shown are essentially the same configurations for the protocol translation as with the preceding Frame Relay example.

### *Example (IGS-A)*

```
! Translate LAT to TCP/Telnet
translate lat DISTANT-LAT tcp IGS-B
```

### **Example (IGS-B)**

```
! Translate TCP to LAT
translate tcp IGS-B lat LAT-B
```

---

**Note:** You can use the same name (for example, *LAT-B*) in the **translate** configuration subcommand for both IGS-A and IGS-B, but this symmetry is not required. Each protocol translator operates independently. The key is the common IP name used in both commands.

---

---

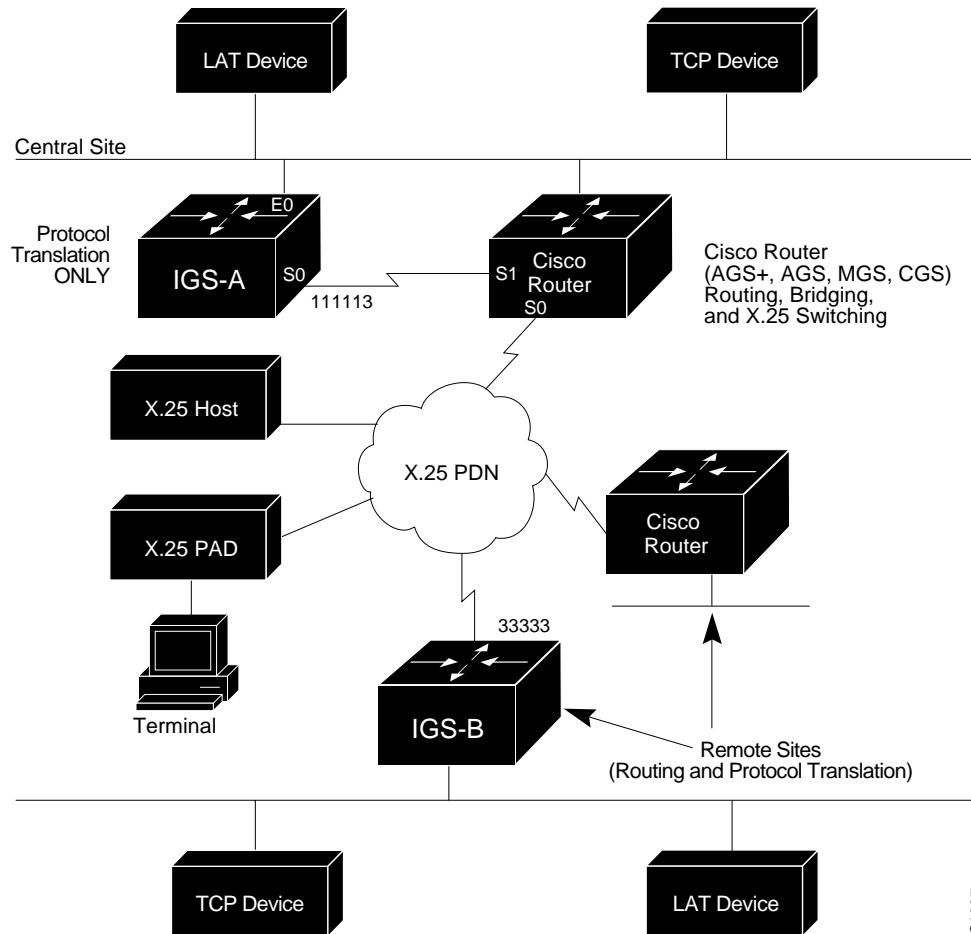
## **Central Site Protocol Translation**

When an organization requires the translation of protocols among multiple remote sites over an X.25 network, with retranslation at a central site, an IGS may be required. The IGS in this configuration allows more than 32 concurrent translation sessions.

To support this application, an IGS is directly connected back-to-back (Figure 1-10) to another Cisco router (for example, AGS+, AGS, MGS, or CGS). This second Cisco router acts as an X.25 switch, by sending X.25 packets to the IGS while concurrently routing and bridging other protocols.

As the IGS is operating only as a protocol translator, it is capable of supporting 100 concurrent protocol translation sessions as opposed to the 32 that are supported when the IGS is routing, bridging, and protocol translating.

**Figure 1-10** Central Site Protocol Translation Example



S1637

The following configuration example illustrates the configuration of IGS and AGS+ systems to support translating protocols over an X.25 network among multiple sites.

The “other” Cisco router is configured to act as an X.25 switch to send X.25 packets to the IGS while concurrently routing and bridging other protocols. This example also illustrates use of the **translate** global configuration subcommand to translate LAT, TCP, and X.25 over an X.25 WAN media. Under this configuration, IGS-A can translate LAT or TCP (Telnet) traffic into X.25 packets for transmission over an X.25 PDN network. Packets are then translated back to LAT or TCP on the other side of the WAN. Example configurations are included for both IGS-A and the Cisco router connected to the WAN at the central location; however, complete configurations details are not provided.

***Example (Basic Configuration for IGS-A):***

```
Interface ethernet 0
ip address 1.0.0.2 255.255.0.0
!
! enable LAT on interface if concurrently routing (8.3 feature)
lat enable
!
interface serial 0
encapsulation X.25
! note that this is subaddress 3 of 11111
x25 address 111113
! The following parameters may depend on your network
x25 facility packetsize 512 512
x25 facility windowsize 7 7
no ip address
```

***Example (“Other” Central Site Cisco Router Configuration):***

```
! Interface to WAN
interface serial 0
x25 address 11111
x25 route ^11113 int ser 1
ip address ....
! Interface to IGS
int ser 1
x25 route .* interface ser 0
no ip address
```

***Example (Translate Configuration for IGS-A):***

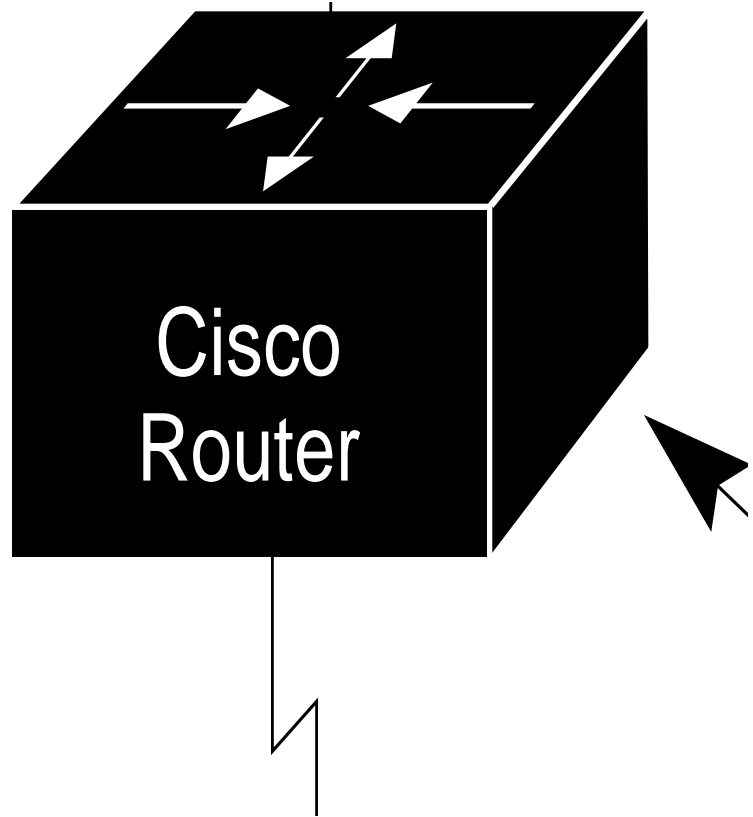
```
no ip routing
! Note subaddress of subaddress 11111(3(3))
translate x25 1111133 tcp tcpdevice
translate lat TCP-B x25 3333301
translate lat tcp-device tcp tcp-device
! etc...any translate commands needed by application
```

---

## ***Stand-Alone LAT to TCP Translation***

For sites needing a high amount of local TCP-to-LAT translation sessions, the IGS can be set up to only use an Ethernet port. This application allows 100 concurrent translation sessions. In the applications illustrated in Figure 1-11, any other Cisco router, including an IGS, can be used to interconnect network segments performing bridging and/or routing.

***Figure 1-11*** IGS as Stand-alone Protocol Translator



*Translation Example (IGS-A only):*

```
Interface ethernet 0
ip address 1.0.0.2 255.255.0.0
!
! enable LAT on this interface
lat enabled
!
interface serial 0
shutdown
no ip routing
default-gateway 1.0.0.100
!
translate lat TCP-A tcp TCP-A
translate lat TCP-B tcp TCP-B
translate tcp LAT-A lat lat-z
! etc...translate commands as required
```

---

## Translation Global Configuration Command Summary

This section lists summarizes the global system configuration command **translate**:

**translate** *inaddress* [*inoptions*] *outaddress* [*outoptions*] [*globaloptions*]

Causes incoming connections to the specified input protocol and address to result in the automatic formation of an outgoing connection to the specified protocol and address. Options *inaddress* and *outaddress* have the format:

**tcp** *ipaddress*

**lat** *servicename*

**x25** *X.121address*

You must specify a service name (**tcp**, **lat**, or **x25**).

The **translate** command causes incoming LAT, TCP, or X.25 connection requests to specified destination address/host name to result in the automatic conversion to the specified outgoing connection type, and subsequently the transmission of a connection request to the address specified in the **translate** command.

---

**Note:** The translate command does not support TN3270 or XRemote one-step translations. To use the protocol translator to convert these protocols, you must use the two-step method as described in the “Using the Protocol Translator” chapter. Also, refer to earlier chapters in this manual for details concerning connection services and options available for all protocols supported by the protocol translator.

---

The *tcp* *inaddress*/*outaddress* argument *ipaddress* is either the standard form (0.0.0.0 to 255.255.255.255) or the name an ip host that can be resolved either via Domain Name Service, IEN116 name service, or by explicit specification in an **ip host** command.

The options for the **tcp** argument include:

**port** *nnnn*

**binary**

The **port** option is used for outgoing TCP connections, or for matching only this port on incoming connections. The default port is port 23 (Telnet). For incoming connections, port 0 will match **any** port.

The **binary** option is for negotiating Telnet binary mode on the Telnet connection. This was the default in previous versions of the protocol translator software, and is set automatically if you enter an old format **translate** subcommand.

The options for the **x25** argument include:

**cud** *call-user-data*

**profile** *profile*

**reverse**

The option **cud** *call-user-data* sends the specified *call-user-data* text as part of the call user data (CUD) in an outgoing call request (after the protocol identification bytes).

The option **profile** *name* causes the X.3 PAD parameters to be set as defined in the profile created by the **x29 profile** subcommand, as defined in the “Configuring X.25 and X.3” chapter.

The option **reverse** provides reverse charging for X.25 on a per-call rather than a per-interface basis.

The options for the outgoing **lat** argument include:

**node** *nodename*

**port** *portname*

Using the **node** keyword followed by a LAT node name causes a connection to be made to a specified node which offers a service. By default, the connection is made to the highest-rated node which offered the service.

**port** *portname*

The keyword **port** specifies a destination LAT port name in the format of the remote system. This parameter is usually ignored in most timesharing systems, but is used by terminal servers which offer reverse-LAT services.

The **node** and **port** options can be combined.

Global options for the **translate** subcommand are as follows:

**access-class** *nnn*

**local**

**login**

**quiet**

**max-users** *nnn*

The **access-class em nnn** keyword allows the **translate** subcommand to be used by source hosts that match the access list parameters. This is currently supported only for incoming TCP and X.25 connections.

The **local** keyword allows protocol negotiations (such as **ECHO** and **linemode**) to not be translated.

The **login** keyword requires that the user log in before the outgoing connection is made. This type of login is specified on the VTY's using the **login** command.

The **quiet** keyword suppresses printing of user-information messages.

The **max-users** keyword limits the number of simultaneous users of the translation to *nnn* (*nnn* is an integer).