

Configuring Protocol Translation

This chapter describes how to configure protocol translation connections on the communication server. It assumes you understand how to use the configuration software. It provides procedures for specifying system-wide facilities, as well as application examples. Before continuing with this chapter, be sure that you are familiar with the information provided in the X.25, Telnet, LAT, TN3270, and XRemote configuration chapters in this publication.

For a complete explanation of the **translate** command, refer to the *Access and Communication Servers Command Reference* publication. For more information about making connections and establishing translation sessions, see the *Cisco Access Connection Guide*.

In the context of this chapter, a communication server set up to run protocol translation software is referred to as a protocol translator.

Note Telnet is a remote terminal protocol that is part of the Transmission Control Protocol/Internet Protocol (TCP/IP) suite. The descriptions and examples in the following sections use the term TCP as a reference to Telnet functionality.

Support for Terminal Connections and Protocol Translation

A communication server set up for protocol translation uses the International Telecommunication Union Telecommunication Standardization Sector (ITU-T) Recommendation X.25 for transferring raw data over X.25 networks. The X.25 software supports both commercial and DDN versions. Protocol translators also support X.25 as a transport mechanism for IP packets, and X.3- and X.29-based PAD connections. This allows the protocol translators to connect to an X.25 public data network (PDN). This X.25 connection allows transport of TCP/IP packets across the X.25 packet-switching network in the same way as would occur on a protocol translator.

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

Communication servers without protocol translation do not support an X.25 PAD function, so they cannot communicate with hosts directly connected to the X.25 PDN. Communication servers encapsulate TCP/IP packets in X.25 packets for transfer over a packet-switching network. These packets can be received by communication servers. Only servers with protocol translation include PAD capabilities, but other servers can communicate with protocol translators using the TCP/IP or

LAT protocols. Server products with protocol translation can translate TCP/IP and LAT into X.25, and then communicate with X.25 hosts. Server products with protocol translation support all PAD standards (X.28, X.29, and X.3). The connection to the packet-switching network is through a synchronous line.

Connections to a PAD are made using EXEC commands. You can configure PAD parameter profiles that can be used to set PAD parameters with other commands, and you can configure access lists to control X.25 network access. Both these features use the message fields defined in Recommendation X.29, which describes procedures for exchanging data between PADs or between a PAD and a DTE.

Cisco's Protocol Translation Methods

The protocol translation software attempts to provide transparent protocol translation between systems running disparate protocols. It enables terminal users on one network to access hosts on another network, despite differences in the native protocol stacks associated with the originating device and the target host.

A communication or access server supports virtual terminal connections in both directions between the protocols in the following list. You can configure the server to translate automatically between them. This is called the one-step translation method.

- X.25 and Local Area Transport (LAT)
- X.25 and Telnet sessions using the Transmission Control Protocol (TCP)
- LAT and TCP/Telnet

The software supports limited connections in both directions between the following protocols. Connecting between these protocols requires that you first connect to a protocol translator, then to the host to which you want to connect. This is called the two-step translation method.

- XRemote to X.25 PAD environments (XRemote must use the two-step method)
- TN3270 to LAT, X.25, and TCP/Telnet (TN3270 must use the two-step method)

The following sections describe the two-step and the one-step translation methods.

See the end of this chapter for protocol translation application and session examples.

Two-Step Method

In general, you use the two-step process when you want to use protocol translation for one-time connections or when you want to use a protocol translator as a general-purpose gateway between two types of networks (for example, X.25 PDN and TCP/IP). Your network administrator must first have configured the server for the transmission protocols you will be using.

Note You must use the two-step method for translations of TN3270 and XRemote.

With the two-step connection process, you can modify the parameters of either network connection, even while a session is in process. This process is similar to connecting a group of terminal lines from a PAD to a group of terminal lines from a TCP server. The difference is that you do not encounter the wiring complexity, unreliability, management problems, and performance bottlenecks that occur when two devices are connected via asynchronous serial lines.

Also, the two-step process allows another level of security over the one-step translation method when TACACS and password protection is enabled. These security features are described in the “Managing the System” chapter earlier in this publication.

To connect to the remote network host running a foreign protocol, follow this procedure:

Step 1 Make a network connection to a protocol translator using the EXEC command for the protocol running at your local terminal.

These commands are listed earlier in this guide. X.3 PAD connections are described later in this chapter.

Step 2 When the protocol translator prompt appears, connect to the remote host using the EXEC command for the protocol running on the remote host (such as TN3270 or XRemote).

Example

The following example shows a connection from a local UNIX host “sankara” to a protocol translator “pt” as the first step in a two-step translation process.

```
sankara% telnet pt
```

The following example shows a connection from a protocol translator “pt” to a host named “ibm3278” as the second step in a two-step translation process.

```
pt> tn3270 ibm3278
ibm3278%
```

One-Step Method

In general, you use the one-step method when network users repeatedly log on to the same remote network hosts through a server. This connection is more efficient and enables the server to act upon greater knowledge of the protocols in use because the server acts as a network connection rather than as a terminal.

The one-step method provides transparent protocol conversion. When connecting to the remote network host, the user enters the connection command to the remote network host, but does not need to specify protocol translation. The network administrator creates a configuration that defines a connection and the protocols to be translated. The user performs one step to connect with the host.

When you make a one-step connection to the server, the server determines which host the connection is for and which protocol that host is using. It then establishes a new network connection using the protocol required by that host.

A disadvantage of the one-step method is that the initiating computer or user does not know that two networking protocols are being used. This means that parameters of the foreign network protocols cannot be changed after connections are established. The exception to this limitation is the set of parameters common to both networking protocols. Any parameter in this set can be changed from the first host to the final destination.

To create one-step protocol translation connection specifications, perform the following global configuration task:

Task	Command
Create the connection specifications for one-step protocol translation.	translate <i>protocol incoming-address</i> [<i>in-options</i>] <i>protocol</i> <i>outgoing-address</i> [<i>out-options</i>] [<i>global-options</i>]

Protocol Translation Application Examples

This section provides protocol translation examples for the following applications:

- Local LAT-to-TCP Translation
- LAT-to-TCP Translation over a WAN
- LAT-to-LAT Translation over a WAN

Note In the application illustrations that follow throughout the remainder of 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.

Local LAT-to-TCP Translation

Figure 20-1 shows a simple LAT-to-TCP translation across an Ethernet network. The configuration file for the communication server follows the figure. The name *TCPA* is the logical name given to the device *TCP-A*.

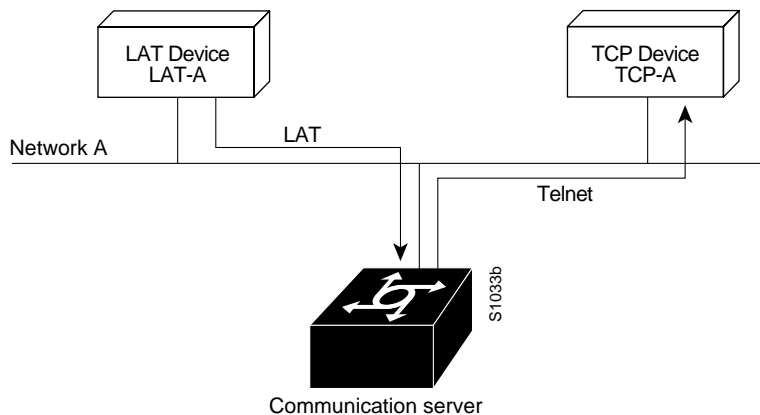


Figure 20-1 Local LAT-to-TCP Translation

Configuration for Communication Server

```
interface ethernet 0
ip address 1.0.0.2 255.255.0.0
!
! enable LAT on this interface
lat enabled
!
translate lat TCPA tcp TCP-A
```

LAT-to-TCP Translation over a WAN

Figure 20-2 shows a configuration that allows translation of LAT to TCP and transmission across an IP-based WAN. The configuration file for CS A follows the figure. The logical LAT service name *distant-TCP* is the name given to device *TCP-B*.

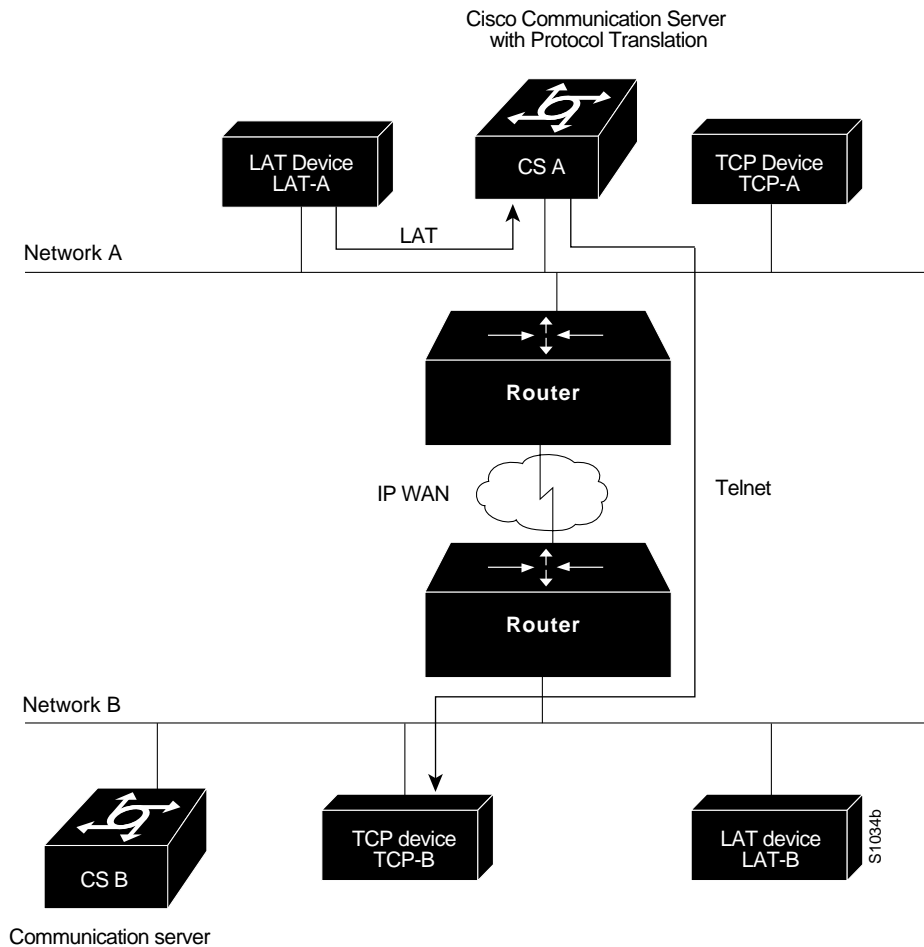


Figure 20-2 LAT-to-TCP Translation over a WAN

Configuration for CS A

```
interface ethernet 0
ip address 1.0.0.2 255.255.0.0
!
! enable LAT on this interface
lat enabled
!
translate lat distant-TCP tcp TCP-B
```

LAT-to-LAT Translation over a WAN

In Figure 20-3, LAT can be transported to a remote LAT device by translating the packets to TCP format and using Telnet to send them across the WAN. The configuration files for CS A and CS B follow the figure. The logical name *TS-B1* is the name given to device *TS-B*.

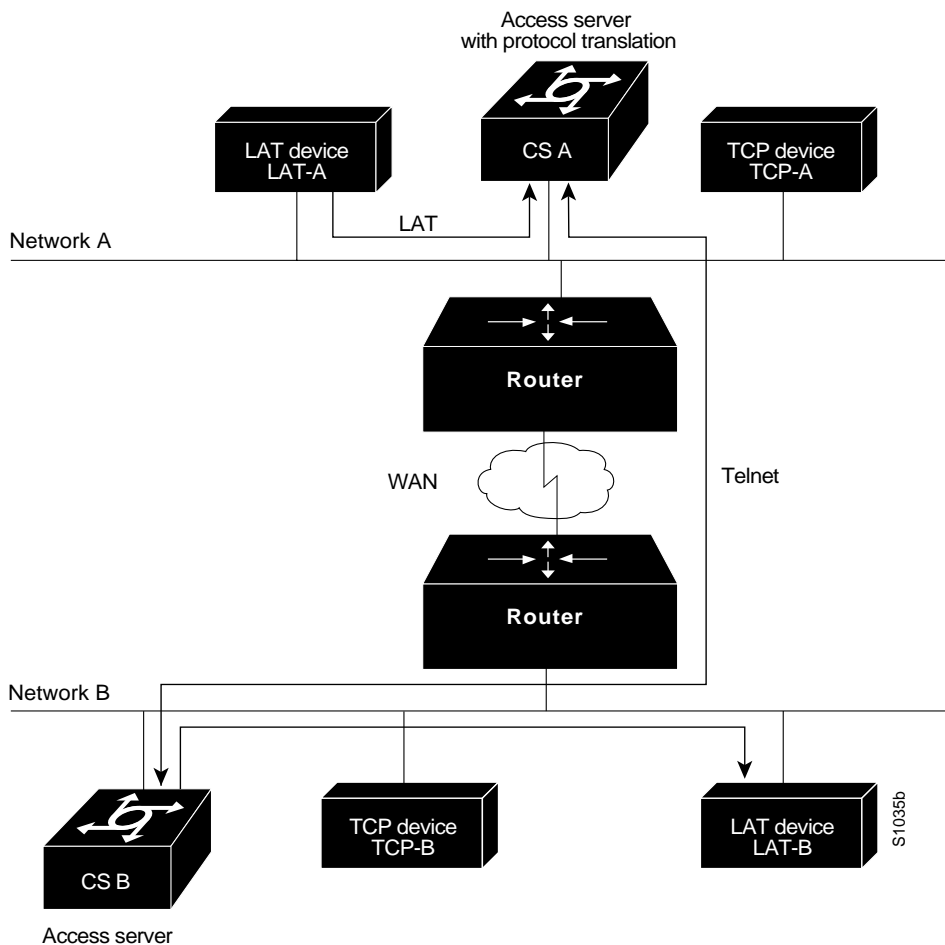


Figure 20-3 LAT-to-LAT Translation over a WAN

Configuration for CS A

```
interface ethernet 0
ip address 1.0.0.2 255.255.0.0
!
! enable LAT on this interface
lat enabled
!
translate lat distant-LAT tcp TS-B1
```

Configuration for CS B

```
interface ethernet 0
ip address 2.0.0.2 255.255.0.0
!
! enable LAT on this interface
lat enabled
!
translate lat TS-B1 lat LAT-B
```

Protocol Translation Session Examples

This section illustrates how to make connections for protocol translation using the one-step and two-step methods.

Using the One-Step Method for TCP-to-X.25 Host Connections

This example illustrates one-step protocol translation featuring a UNIX workstation user making a connection to a remote X.25 host named *host1* over an X.25 PDN. The protocol translator automatically converts the Telnet connection request to an X.25 connection request and transmits the request as specified in the system configuration.

- A connection is established by entering the **telnet** EXEC command at the UNIX workstation system prompt, as follows:

```
unix% telnet host1
```

Note This example implicitly assumes that the name *host1* is known to the UNIX host (obtained via DNS, IEN116, or a static table) and is mapped to the IP address used in a **translate** command.

The protocol translator accepts the Telnet connection and immediately forms an outgoing connection with remote *host1* as defined in a **translate** command in your protocol translator's active configuration file.

Next, *host1* sets several X.3 parameters, including local echo. Since the Telnet connection is already set to local echo (at the UNIX host), no changes are made on the TCP connection.

The *host1* connection prompts for a user name, then *host1* sets the X.3 parameters to cause remote echo (the same process as setting X.3 PAD parameter 2:0), and prompts for a password. The protocol translator converts this to a Telnet option request on the UNIX host, which then stops the local echo mode.

At this point the user is connected to the PAD application and the application will set the X.3 PAD parameters (although they can always be overridden by the user). When the user is finished with the connection, he enters the escape character to exit back to the host connection, then enters the appropriate command to close the connection.

The *host1* host immediately closes the X.25 connection. The protocol translator then drops the TCP connection, leaving the user back at the UNIX system prompt.

Using the Two-Step Method for TCP-to-PAD Connections

To use the two-step method, perform the following steps:

- Step 1** Connect directly from a terminal or workstation to a protocol translator.

For example, you might make the following connection requests at a UNIX workstation as a first step to logging into a database called *Information Place* on an X.25 PDN:

```
unix% telnet orion
```

If the protocol translator named *orion* is accessible, it returns a login message and you enter your login name and password.

Step 2 Connect from the protocol translator to *Information Place*, which is on an X.25 host. You connect to an X.25 host using the **pad** EXEC command followed by the service address:

```
orion> pad 71330
```

Once the connection is established, the protocol translator immediately sets the PAD to single character mode with local echoing, since this is the behavior the protocol translator expects. The PAD responds with its login messages and a prompt for a password:

```
Trying 71330...Open
Welcome to the Information Place
Password:
```

Because the password should not echo on your terminal, the PAD requests remote echoing so that characters will be exchanged between the PAD and the protocol translator, but not echoed locally or displayed. After the password is verified, the PAD again requests local echoing from the protocol translator, which it does from then on.

To complete this sample session, you log off, which returns you to the protocol translator system EXEC prompt. From there, you execute the EXEC **quit** command and the protocol translator drops the network connection to the PAD.

Changing Parameters and Settings Dynamically

The following example illustrates how to make a dynamic change during a protocol translation session. In this example, you need to edit information on remote host *Information Place*. Suppose that you need to change the X.3 PAD parameters that define the editing characters from the default Delete key setting to the Ctrl-D sequence.

Step 1 Enter the escape sequence to return to the system EXEC prompt:

```
ctrl ^ x
```

Step 2 Enter the **resume** command with the **/set** keyword and the desired X.3 parameters. X.3 parameter 16 sets the Delete function. ASCII character 4 is the Ctrl-D sequence.

```
CS> resume /set 16:4
```

The session resumes with the new settings, but now you notice that information is not being displayed correctly. You might want to set the **/debug** switch to check that your parameter setting has not been changed by the host PAD.

Step 3 Enter the escape sequence to return to the system EXEC prompt, then enter the resume command with the **/debug** switch.

```
CS> resume /debug
```

The **/debug** switch provides helpful information about the connection.

You can also set a packet dispatch character or sequence using the **terminal dispatch-character** command.

The following example shows how to set ESC (ASCII character 27) as a dispatch character:

```
CS> terminal dispatch-character 27
```

To return to the PAD connection, simply enter the following:

```
CS> resume
```