

Connecting to a Host through a Protocol Translator

This chapter describes the methods you can use to connect from a host running one protocol (such as Telnet with TCP/IP) to a host running another protocol (such as LAT). This process is called protocol translation, and allows devices running dissimilar protocols—such as X.25 and TCP/IP—to communicate. Protocol translation does not permit translation between other services such as file transfer protocols.

You can make a protocol translation connection using any of the protocols listed in Chapter 3, as well as with X.3 PAD, which is described in the later section “X.3 PAD Connections.” The commands you use to make and exit from these connections are listed in Chapter 3. This chapter describes the additional tasks required to perform protocol translation from one host to another using a protocol translator, a communication server with protocol translation, or a router with protocol translation. Specifically, it contains the following information:

- Protocol translation methods
- X.3 PAD connections
- Protocol translation session examples

Protocol Translation Methods

The protocol translator supports virtual terminal connections in both directions between the protocols in the following list. You can configure the protocol translator to automatically translate between the following types of connections. (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 protocol translator supports limited connections in both directions between the following protocols:

- XRemote to X.25 PAD environments
- TN3270 to LAT, X.25, and TCP/Telnet

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

Connecting between these protocols requires that you specifically tell the protocol translator to translate each time you make a connection. (This is called the two-step translation method.)

This section describes both the two-step and the one-step translation method.

The Two-Step Method

This section describes how and when to make a two-step protocol translator connection. The network administrator must first have configured the protocol translator for the transmission protocols you will be using.

In general, use the two-step process when you want to use protocol translation for one-time connections.

With the two-step connection process, you can individually 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 protocol translator. The difference is that you do not encounter the wiring complexity, unreliability, management problems, and performance bottlenecks that occur when two separate 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 configuration guide for your server product.

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 specific to the protocol running at your local terminal.

These commands are listed in Chapter 3, “Connecting to a Host Through a Communication Server,” except X.3 PAD connections, which are described in this chapter in the later section, “X.3 PAD Connections.”

Step 2 When the protocol translator prompt appears, connect to the remote host using the EXEC command specific to the protocol running on the remote host.

Example

In the following example, the local terminal is an IBM 3278 host running TN3270 and the remote network host is running rlogin. The name of the protocol translator is ganges, and the remote UNIX host is krishna.

```
ibm3278% tn3270 ganges
```

If the protocol translator named sankara is accessible, it returns a login message and you enter your login name and password. After you enter your name and password, connect to the remote host.

```
pt> rlogin krishna
krishna%
```

The One-Step Method

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

The one-step method provides transparent protocol conversion. In other words, when connecting to the remote network host, the user enters the connection command to the remote network host, but need not specify protocol translation at connection time. The network administrator creates a configuration that defines a connection and the protocols to be translated. This is called the *one-step* method because the user performs one step to connect with the host.

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

To support connections in each direction, the network administrator must specifically include **translate** command statements in the configuration file. See the *Protocol Translator Configuration Guide* for information about configuring a protocol translator for one-step connections. See the publication *Protocol Translator Command Reference* for information about the **translate** command.

One disadvantage of using 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 the 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.

X.3 PAD Connections

This section describes the following tasks;

- Making a PAD connection during protocol translation sessions involving X.25
- Switching between sessions
- Exiting a PAD connection

A PAD is a packet assembler/disassembler, which is a device that receives a character stream from a terminal or group of terminals, assembles the character stream into packets, and sends the data packets out to a host. A PAD can also do the reverse. That is, it can take data packets from a network host and translate them into a character stream that can be understood by the terminals. A PAD is defined by CCITT Recommendations X.3, X.28, and X.29.

Make a PAD Connection

You can have several concurrent PAD connections open and switch back and forth between them. You can also exit out of a connection and return to the EXEC prompt at any point.

To open a new connection, exit out of the current connection by typing the escape sequence to return to the system command prompt, then open the new connection.

To log in to a PAD, enter the following command:

```
pad {X.121-address | hostname} [/cud text] [/debug] [/reverse]
```

Syntax Description

<i>X.121-address</i>	Specifies the X.121 address of the X.25 host.
<i>hostname</i>	Specifies the X.25 host name if the host-to-address mapping has been set with the X.25 host command. (The pad command supports one-word connections. In other words, you are not required to enter the pad command; just the address is enough to start connection.)
<i>/cud text</i>	(Optional.) Includes the specified <i>text</i> in the Call User Data field of the outgoing Call Request Packet.
<i>/debug</i>	(Optional.) Causes the informational level of logging messages to be printed whenever the remote host changes an X.3 parameter setting or sends any other X.29 control packet.
<i>/reverse</i>	(Optional.) Causes reverse charge calls to be accepted on a per-call, rather than a per-interface, basis.

To display information about packet transmission and X.3 PAD parameter settings, enter the **show x25 pad** command. This command is described in Chapter 5, “Managing and Monitoring Connections.”

Switch between Connections

You can have several concurrent connections open and switch back and forth between them. The number of connections that can be opened is defined by the **session-limit** command, which is described in the publication *Protocol Translator Command Reference*.

You can switch between connections by escaping out of one connection and resuming a previously opened connection, as follows:

- Step 1** Escape out the current connection and return to the EXEC prompt by pressing **Ctrl-^ X**.
- Step 2** List the open connections using the **where** command. You display information about all open connections associated with the current terminal line.
- Step 3** Type the **resume** command and the connection number to make the connection.

You can also resume the previous connection by pressing the Return key.

The **where** command has no additional syntax. The **resume** command has the following syntax when used on the communication server:

```
resume [connection] [keyword]
```

Syntax Description

<i>connection</i>	(Optional.) The name or number of the connection; the default is the most recent connection.
<i>keyword</i>	(Optional.) One of the options listed in Table 4-1.

Local X.3 parameters can also be changed when resuming a connection. See the section “Set X.3 PAD Parameters” in Chapter 5.

Table 4-1 Resume Options

Option	Description
/debug	Prints parameter changes and messages. On a protocol translator, this option displays informational messages whenever the remote host changes an X.3 parameter or sends an X.29 control packet.
/echo	Performs local echo.
/line	Enables line-mode editing.
/nodebug	Prevents parameter changes and messages from being printed.
/noecho	Disables local echo.
/noline	Disables line mode and enables character-at-a-time mode, which is the default.
/nostream	Disables stream processing.
/set	Sets X.3 connection options. Refer to “Set X.3 PAD Parameters” in Chapter 5 for a list of these connection options.
/stream	Enables stream processing.

The **Ctrl-^ X**, **where**, and **resume** commands are available with all supported connection protocols.

Exit a PAD Session

You can issue any of the following commands to terminate an active terminal session:

exit

quit

logout

Protocol Translation Session Examples

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

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

In the following example, the user connects directly from a terminal or workstation on a TCP/IP network to a protocol translator, and then to a database called Information Place on an X.25 packet data network. The database has a service address of 71330.

Step 1 Make the following connection requests at a UNIX workstation as a first step to logging into the database Information Place:

```
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 the database 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.

Step 3 To complete this sample session, you log off, which returns you to the protocol translator system EXEC prompt.

Step 4 Execute the EXEC **quit** command and the protocol translator drops the network connection to the PAD.

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

The following 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.

Step 1 Establish a connection 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 using DNS, IEN116, or a static table) and is mapped to the IP address used in a **translate** command as defined in the *Protocol Translator Configuration Guide*.

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.

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

Step 2 Enter a username, then a password when the host1 connection prompts for them. The protocol translator converts this to a Telnet option request on the UNIX host, which then stops the local echo mode.

At this point you are connected to the PAD application and the application will set the X.3 PAD parameters (although they can always be overridden using the **resume** or **x3** commands).

Step 3 When you are finished with the connection, enter the escape character to exit back to the host connection, then enter the appropriate command to close the connection.

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

Changing Parameters and Settings Dynamically

The following example illustrates how to make a dynamic change during a session. In this example, you need to edit information on a remote host named 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. (Refer to Appendix A for a list of ASCII characters.)

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.

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

The session resumes with the new settings. If the information is not being displayed correctly, you can 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.

```
pt> 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, as shown in this example:

Step 1 Set the ESC key (ASCII character 27) to a dispatch character by entering the following command:

```
pt> terminal dispatch-character 27
```

Step 2 Return to the PAD connection by entering the following command:

```
pt> resume
```

The ESC key has been set to a dispatch character and the original PAD connection has been resumed.