**C H A P T E R  5**

# Terminal or Telecommuting Service Connections Using Protocol Translation

This chapter describes the methods you can use to connect 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 protocol listed in earlier chapters in this publication, as well as with X.3 PAD, which is described in the "X.3 PAD Connections" section later in this chapter. The commands you use to make these connections and exit from them are listed in the "Terminal Service Connections" and "Telecommuting Service Connections" chapters.

This chapter describes the additional tasks required to perform protocol translation from one host to another using a communication or access server, or a router with protocol translation. Specifically, it contains the following sections:

- Protocol Translation Methods
- X.3 PAD Connections
- Protocol Translation Session Examples

## Protocol Translation Methods

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

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

The server product 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 SLIP/PPP and X.25 PAD environments
- TN3270 to LAT, X.25, SLIP/PPP, and TCP (Telnet)

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

## The One-Step Translation Method

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

The one-step method provides transparent protocol translation. When connecting to the remote network host, the user enters the translation 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 product, the server product 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.

To support connections in each direction, the network administrator must include **translate** command statements in the configuration file. Refer to the *Protocol Translator Configuration Guide and Command Reference* publication for information about configuring a server product for one-step connections and for information about the **translate** command.

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.

## The Two-Step Translation Method

In general, you use the two-step process when you want to use protocol translation for one-time connections. The network administrator must first have configured the server product for the transmission protocols you will be using.

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 product. The difference is that when two devices are connected via asynchronous serial lines you do not encounter wiring complexity, unreliability, management problems, and performance bottlenecks.

Also, the two-step process allows another level of security over the one-step translation method when TACACS 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 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%
```

# X.3 PAD Connections

The following sections describe X.3 PAD connection tasks:

* Make a PAD Connection

* Switch between Connections

* Exit a PAD Session

A PAD is a packet assembler/disassembler, which is a device that receives a character stream from one or more terminals, assembles the character stream into packets, and sends the data packets out to a host. A PAD can also do the reverse. 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 PAD connections open at the same time and switch between them. You can also exit a connection and return to the user EXEC prompt at any point.

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

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

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

### Syntax Description

| | |
|---|---|
| *X.121-address* | Specifies the X.121 address of the X.25 host. |
| *hostname* | 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. You don't have to enter the **pad** command; just the address is enough to start connection.) |
| **/cud** *text* | (Optional) Includes the specified *text* in the Call User Data field of the outgoing Call Request Packet. |
| **/debug** | (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. |

/**profile** *name*        (Optional) Sets X.3 PAD parameters for the *name* script. Same as issuing the **x29 profile** global configuration command when translating X.25.

/**reverse**            (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 the "Monitoring and Managing Connections" chapter.

## Switch between Connections

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

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

**Step 1**    Press **Ctrl-^ X** to escape the current connection and return to the EXEC prompt.

**Step 2**    Use the **where** command to list the open connections. The system displays information about all open connections associated with the current terminal line.

**Step 3**    Type the **resume** command and the connection number.

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 server product:

   **resume** [*connection*] [*keyword*]

### Syntax Description

*connection*           (Optional) The name or number of the connection; the default is the most recent connection.

*keyword*             (Optional) One of the options listed in Table 5-1.

Local X.3 parameters can also be changed when resuming a connection. Refer to "Set X.3 PAD Parameters" in the "Monitoring and Managing Connections" chapter later in this publication.

**Table 5-1**        **Resume Options**

| Option | Description |
| --- | --- |
| /**debug** | Prints parameter changes and messages. On a server product, 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. |

| Option | Description |
|---|---|
| **/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 the "Monitoring and Managing Connections" chapter for a list of 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 a 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 translation methods.

## Using the One-Step Method for TCP (Telnet) to X.25 Host Connections

The following is an example of one-step protocol translation. A UNIX workstation user makes a connection to a remote X.25 host named host1 over an X.25 PDN. The server product automatically converts the Telnet connection request to an X.25 connection request and transmits it 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 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 and Command Reference* publication.

The server product accepts the Telnet connection and immediately forms an outgoing connection with remote host1 as defined in a **translate** command in your server product'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 server product 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 that 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 sequence 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 server product then drops the TCP connection, leaving the user at the UNIX system prompt.

## 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 server product, 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 server product named orion is accessible, it returns a login message and you enter your login name and password.

**Step 2** Connect from the server product 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 server product immediately sets the PAD to single character mode with local echoing, because this is the behavior the server product 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 server product, but not echoed locally or displayed. After the password is verified, the PAD again requests local echoing from the server product.

**Step 3** Complete this sample session by logging off, which returns you to the server product system EXEC prompt.

**Step 4** Execute the **quit** EXEC command and the server product drops the network connection to the PAD.

## Changing Parameters and Settings Dynamically

The following example shows how to make a dynamic change during a session. This example presumes a need to edit information on a remote host and 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 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 the following 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 is set to a dispatch character, and the original PAD connection is resumed.