CHAPTER 3

Connecting to a Host through a Communication Server

This chapter describes how to make terminal-to-host connections using a communication server. To make protocol translation connections, refer to Chapter 4, “Connecting to a Host through a Protocol Translator.” You can make the following types of connections from a terminal connected to the communication server:

- LAT
- Telnet and UNIX rlogin
- TN3270
- SLIP
- PPP
- XRemote

These tasks are outlined in the following sections. See the end of this chapter for examples of XRemote connections on an X display terminal.

LAT Connections

Digital Equipment Corporation’s (Digital’s) Local Area Transport (LAT) protocol is most often used to connect communication servers to Digital hosts. LAT is a Digital-proprietary protocol. Your communication server uses LAT technology licensed from Digital. You can perform the following LAT connection tasks from your communication server:

- Make a LAT connection
- Switch between connections
- Use Digital commands
- Exit a LAT Session

These tasks are described in the following sections.

Make a LAT Connection

To connect to a LAT host, enter the **lat** EXEC command at the system prompt. The **lat** command has the following syntax:

lat *name* [**node** *nodename* | **port** *portname* | **/debug**]

Syntax Description

name	LAT learned service name.
node *nodename*	(Optional.) Specifies a connection to a particular LAT node that offers a service. If you do not include the node option, the node with the highest rating offering the service is used. Use the EXEC command **show lat nodes** to display information about all known LAT nodes.
port *portname*	(Optional.) Specifies a destination LAT port name. This keyword is ignored in most timesharing systems, but is used by communication servers offering *reverse LAT* services. Reverse LAT involves connecting to one communication server from another. In this case, the target communication server runs the host portion of the protocol. Enter the port name in the format of the remote system in place of the *portname* argument.
/debug	(Optional.) A switch that, when enabled, prints parameter changes and other special messages on the terminal.

The LAT protocol must be explicitly specified when your preferred transport is set to **none** or to another protocol. (If your preferred transport is set to **lat**, you can use the **connect** command.) You can quit the connection by entering **Ctrl-C**, or complete the connection by entering the password for a given service.

Examples

The following example establishes a LAT connection from the communication server named cs to the host eng2:

```
cs> lat eng2
Trying ENG2...Open
         ENG2 – VAX/VMS V5.2
Username: JSmith
Password:
    Welcome to VAX/VMS version V5.2 on node ENG2
    Last interactive login on Friday,  6-APR-1990 19:46
```

The system informs you of its progress by displaying the messages “Trying <system>...” and then “Open.” If the connection were not successful, you would receive a failure message.

The following example establishes a LAT connection from the communication server named cs to the our-modems service and specifies port 24, which is a special modem:

```
cs> lat our-modems port 24
```

The following example establishes a LAT connection from the communication server named cs to the our-modems service and specifies a node named eng:

```
cs> lat our-modems node eng
```

The following example uses the LAT session debugging capability:

```
cs> lat Eng2 /debug
Trying ENG2...Open
        ENG2 – VAX/VMS V5.2
 Username: JSmith
 Password:
    Welcome to VAX/VMS version V5.2 on node ENG2
    Last interactive login on Tuesday, 10-APR-1990 19:02
[Set Flow out off, Flow in on, Format 8:none, Speed 9600/9600]
[Set Flow out off, Flow in on, Format 8:none, Speed 9600/9600]
$ set ter/speed=2400
[Set Flow out off, Flow in on, Format 8:none, Speed 2400/2400]
```

A variety of LAT events are reported, including all requests by the remote system to set local line parameters. The messages within brackets ([ ]) above are the messages produced by the remote system setting line characteristics to operating system defaults.

Define a Group List for Outgoing LAT Connections

You can temporarily define the list of services to which you or another user can connect. You do so by defining the group code lists used for connections from specific lines The group code range entered *must be* a subset of the line's configured group's code range. Use the following command:

terminal lat out-group {*groupname* | *number* | *range*}

Syntax Description

groupname	Specifies the name of the group that has access to the system through the specified line.
number	Specifies the number of the group that has access to the system through the specified line.
range	Indicates the range of group numbers. Separate the range of numbers with a hyphen.

Example

```
cs> terminal lat out-group 4, 6-189
```

Switch between Sessions

You can have several concurrent LAT sessions open and switch back and forth between them.

To open a subsequent session, first enter the escape sequence to quit out of the current session, which by default is **Ctrl-^ X**, then open a new session.

To list the available LAT services, issue the following command:

show lat services

For sample output to this command, see Chapter 5, "Managing and Monitoring Connections."

For more information about switching between LAT sessions, refer to Chapter 5, "Managing and Monitoring Connections."

Use Digital Commands on the Communication Server

Your communication server supports a subset of Digital commands, including the following:

Task	Command
List EXEC commands.	**help**
Close the active session.	**logout**

Exit a LAT Session

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

exit

quit

logout

Telnet and UNIX rlogin Connections

Two Transmission Control Protocol/Internet Protocols (TCP/IP), Telnet and rlogin, are available for making connections to a host.

Telnet, a virtual terminal protocol that is part of the TCP/IP protocol suite, allows for connections to hosts. Telnet is the more widely used protocol.

The rlogin protocol is a remote login service developed for the BSD UNIX system. It provides better control and output suppression than Telnet, but can only be used when the host (typically, a UNIX system) supports rlogin. Our implementation of rlogin does not subscribe to the trusted host model. That is, a user cannot automatically log onto a UNIX system from the communication server, but must provide a user ID and a password for each connection.

This implementation of Telnet and rlogin provides these connections services:

- Make Telnet connections
- Execute special Telnet escape sequences
- Make rlogin connections
- Switch between Telnet and rlogin connections
- Exit Telnet and rlogin sessions

These tasks are described in the following sections.

Make Telnet Connections

To log into a host that supports Telnet, enter either of the following commands:

connect *host* [*port*] [*keyword*]

telnet *host* [*port*] [*keyword*]

Syntax Description

host A host name or an Internet address.

port (Optional.) A decimal TCP port number; the default is the Telnet server port (decimal 23) on the host.

keyword (Optional.) One of the options listed in Table 3-1.

Table 3-1 Telnet Connection Options

Option	Description
/route *path*	Specifies loose source routing. The argument *path* is a list of host names or Internet addresses that specify network nodes, ending with the final destination.
/line	Enables Telnet line mode. In this mode, the communication server does not send any data to the host until you press Return. You can edit the line using the standard communication server command editing characters (Backspace, Delete, **Ctrl-U, Ctrl-W**). The **/line** keyword is a local switch; the remote server is not notified of the mode change.
/debug	Enables Telnet debugging mode.
/stream	Turns on *stream* processing, which enables a raw TCP stream with no Telnet control sequences. A stream connection does not process Telnet options, and may be appropriate for connections to ports running UUCP and other non-Telnet protocols.

With our implementation of TCP/IP, you are not required to enter the commands **connect** or **telnet** to establish a Telnet connection. You can just enter the learned host name, as long as the host name is different from a command word for the communication server.

To display a list of the available hosts, enter the following command:

show hosts

To display the status of all TCP connections, enter the following command:

show tcp

The communication server assigns a logical name to each connection; several commands use these names to identify connections. The logical name is the same as the host name, unless that name is already in use or you change the connection name with the EXEC command **name-connection**. If the name is already in use, the communication server assigns a null name to the connection.

Examples

The following example routes packets from the source system to kl.sri.com, then to 10.1.0.11, and finally to mathom:

```
cs> connect mathom /route:kl.sri.com 10.1.0.11 mathom
```

The following example connects to a host with logical name mathom:

```
cs> mathom
```

Execute Special Telnet Escape Sequences

The Telnet software supports special Telnet commands in the form of Telnet sequences that map generic terminal control functions to operating system-specific functions.

To issue a special Telnet command, enter the escape sequence (usually **Ctrl-^**) and then a command character. You can enter the command character as you hold down Ctrl or with Ctrl released; you can type either uppercase or lowercase letters.

Table 3-2 lists the special Telnet commands.

Table 3-2 Special Telnet Commands

Task	Escape Sequence
Break	**Ctrl-^ B**
Interrupt Process (IP)	**Ctrl-^ C**
Erase Character (EC)	**Ctrl-^ H**
Abort Output (AO)	**Ctrl-^ O**
Are You There? (AYT)	**Ctrl-^ T**
Erase Line (EL)	**Ctrl-^ U**

At any time during an active Telnet session, you can list the Telnet commands by entering the following command at the system prompt:

Ctrl-^ ?

This is done by typing the escape sequence followed by a question mark. This command displays an online table of the special Telnet commands, for quick reference.

A sample of this list follows (the Ctrl key is represented by the first caret [^]).

```
CS> ^^?
[Special telnet escape help]
^^B  sends telnet BREAK
^^C  sends telnet IP
^^H  sends telnet EC
^^O  sends telnet AO
^^T  sends telnet AYT
^^U  sends telnet EL
```

Make rlogin Connections

You can have several concurrent rlogin connections open and switch back and forth between them.

To open a new connection, exit out of the current connection by typing the escape sequence, which by default is **Ctrl-^ X**, to return to the system command prompt, then open a new connection.

To log in to a UNIX host using rlogin, enter the following command:

rlogin *host* [*debug*]

Syntax Description

host Specifies the host name or Internet address.

debug (Optional.) Enables debugging output from the rlogin protocol.

Example

The following example makes an rlogin connection to a host at address 108.33.21.2 and enables the message mode for debugging:

```
cs> rlogin 108.33.21.2 debug
```

Switch between Telnet and rlogin Sessions

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

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

Step 1 Escape out the current session and return to the EXEC prompt by pressing **Ctrl-^ X**.

Step 2 List the open sessions using the **where** command. You display information about all open sessions associated with the current terminal line.

Step 3 Type the **resume** command and the session number to make the connection.

You can also resume the previous session 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

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 3-3.

Table 3-3 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	Cancels printing of parameter changes and messages.
/noecho	Disables local echo.
/noline	Disables line mode and enables character-at-a-time mode, which is the default.
/nostream	Disables stream processing.
/set *parameter:value*	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.

Examples

The following example shows how to resume connection 2:

```
CS> resume 2
```

You can omit the command name and simply type the connection number to resume that connection. The following example illustrates how to resume connection 3:

```
CS> 3
```

Exit Telnet and rlogin Sessions

You can issue any of the following commands to terminate an active Telnet or rlogin session:

exit

quit

logout

TN3270 Connections

You connect to a host using TN3270 terminal emulation at the EXEC prompt. The system administrator must configure a default terminal emulation file that permits the terminal to correctly communicate with the host. Refer to the publications *Communication Server Configuration Guide* and *Communication Server Command Reference* to specify alternate terminal emulations. The administrator can also specify custom terminal emulations.

Unlike Telnet and LAT connections, you *must* enter the command **tn3270** to make a connection to an IBM 3278 host. To begin a TN3270 session, enter the following command:

tn3270 *hostname*

Syntax Description

hostname Name of a specific host on a network that is reachable by the communication server. The default terminal emulation mode allows access using a VT100 emulation.

Example

The following example establishes a terminal session with an IBM host named finance:

```
cs> tn3270 finance
```

You can issue any of the following commands to terminate an active TN3270 session:

exit

quit

logout

SLIP Connections

You make most connections to the remote host that supports the Serial Line Internet Protocol (SLIP) at the EXEC prompt. The system administrator can configure SLIP to expect a specific address or to provide one for you. It is also possible to set up SLIP in a mode that compresses packets for more efficient use of the line.

To make a SLIP connection, enter the following command:

slip {**/default |** *remote-ip-address* | *remote-name*} [**@***tacacs-server*]} [**/routing**] [**/compressed**]

Syntax Description

/default	(Optional.) Makes a SLIP connection when a default address has been configured.
remote-ip-address	Specifies the IP address of the client workstation or PC.
remote-name	Specifies the name of the client workstation or PC.
@*tacacs-server*	(Optional.) Specifies the IP address or IP host name of the TACACS server to which the user's TACACS authentication request is sent.
/routing	(Optional.) Indicates that the remote system is a router. Line must be configured for asynchronous routing using SLIP encapsulation.
/compressed	(Optional.) Indicates that IP header compression should be negotiated. The system administrator must have configured the system with the **ip tcp header-compression passive** command for this command to be valid in EXEC mode. The command **ip tcp header-compression** forces header compression on or off.

If you specify an address for the TACACS server, either using **default**, or *tacacs-server*, the address must be the first parameter in the command after you type **slip**. If you do not specify an address or enter **default**, you will be prompted for an IP address or host name. You can enter **default** at this point to use the default address configured for the line.

If you do not use the *tacacs-server* argument to specify a TACACS server for SLIP address authentication, the TACACS server specified at login (if any) will be used for the SLIP address query.

To allow optimal use of bandwidth on a line, SLIP allows compressing of the SLIP packets using Van Jacobson TCP header compression as defined in RFC 1144. The default is to not compress the packets. The configuration file must have header compression on and the **slip /compressed** EXEC command must be explicitly entered for header compression to occur.

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

exit

quit

logout

Examples

The following example illustrates how to make a connection when a default IP address has been assigned. Once a correct password is entered, you are placed in SLIP mode, and the IP address is displayed.

```
cs> slip
Password:
Entering SLIP mode.
Your IP address is 192.31.7.28, MTU is 1524 bytes
```

The following example illustrates the prompts displayed and the response required when dynamic addressing is used to assign the SLIP address:

```
cs> slip
IP address or hostname? 192.31.6.15
Password:
Entering SLIP mode
Your IP address is 192.31.6.15, MTU is 1524 bytes
```

In the following example, the address 192.31.6.15 has been assigned as the default. Password verification is still required before SLIP mode can be enabled.

```
cs> slip default
Password:
Entering SLIP mode
Your IP address is 192.31.6.15, MTU is 1524 bytes
```

The following example illustrates the implementation of header compression on the interface with the IP address 128.66.2.1:

```
cs> slip 128.66.2.1 /compressed
Password:
Entering SLIP mode.
Interface IP address is 128.66.2.1, MTU is 1500 bytes.
Header compression will match your system.
```

In the previous example, the interface is configured for **ip tcp header-compression passive**, which permitted the user to enter the **/compressed** keyword at the EXEC mode prompt. The message "Header compression will match your system" indicates that the user indeed specified compression.

If the line was configured for **ip tcp header-compression on**, this line would read "Header compression is On."

The following example specifies a TACACS server parlance for address authentication:

```
cs> slip 1.0.0.1@parlance
Password:
Entering SLIP mode.
Interface IP address is 1.0.0.1, MTU is 1500 bytes
Header compression will match your system.
```

PPP Connections

You can make asynchronous connections using the Point-to-Point Protocol (PPP). To start a PPP connection, enter the following command at the EXEC prompt:

ppp {**/default** | {*remote-ip-address* | *remote-name*} [**@***tacacs-server*]} [**/routing**]

Syntax Description

/default	(Optional.) Makes PPP connection when a default address has been configured.
remote-ip-address	Specifies an IP address of the client workstation or PC. This parameter can only be specified if the line is set for dynamic addresses using the line configuration command **async address dynamic**.
remote-name	Specifies the name of the client workstation or PC. This parameter can only be specified if the line is set for dynamic addresses using the line configuration command **async address dynamic**.
@*tacacs-server*	(Optional.) Specifies an IP address or IP host name of the TACACS server to which the user's TACACS authentication request is sent.
/routing	(Optional.) Indicates that the remote system is a router and that routing messages should be exchanged over the link. The line must be configured for asynchronous routing using PPP encapsulation.

If you specify an address for the TACACS server, either using **default** or *tacacs-server*, the address must be the first parameter in the command after you type **ppp**. If you do not specify an address or enter **default**, you will be prompted for an IP address or host name. You can enter **default** at this point to use the default address configured for the line.

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

exit

quit

logout

Example

The following example shows a line that is in asynchronous mode using PPP encapsulation. The name of the PC is ntpc (assuming that the name ntpc is in the DNS so that it can be resolved to a real IP address). The person typing this command is on the PC running a terminal emulator program.

```
CS> ppp ntpc@server1
```

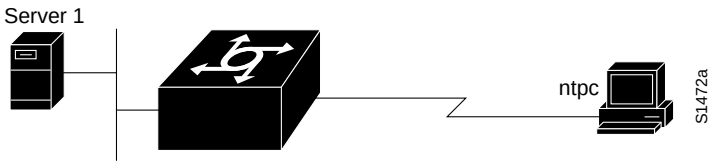

Figure 3-1 Using the PPP EXEC Command

XRemote Connections

The XRemote protocol permits you to connect to remote hosts via TCP/IP and LAT using an X display station and modem. You make connections in any one of the following ways:

- Automatically, using the X Display Manager Control Protocol (XDMCP) for TCP/IP networks
- Automatically, using DECwindows login for LAT networks
- Manually, using a step-by-step access process

The following sections outline steps required for starting up XRemote in several typical environments. When possible, use the automated processes. Before any of these tasks can be performed, the system administrator must have configured a path for loading fonts.

Automatic Session Startup—XDMCP Server

If your host computer supports a server for XDMCP, such as the *xdm* program included in X11R4 or later, you can use automatic session startup to make an XRemote session connection. To do so, enter the following command:

xremote xdm [*hostname*]

Syntax Description

hostname (Optional.) Computer host name.

This command causes an XDMCP session startup request to be made to the computer specified. If you do not specify a host name, a broadcast message is sent to all hosts. The first host to respond by starting up a session is used.

The communication server and X terminal stay in XRemote mode until either the display manager terminates the session, or a reset request is received from the X terminal.

Example

The following example starts a session with a remote host named star.

```
cs> xremote xdm an
```

Automatic Session Startup—DECwindows Login via LAT

If your host computer supports DECwindows login sessions, you can use automatic session startup to make an XRemote session connection. Once the system administrator at the remote host has configured support for DECwindows over LAT, you can use the EXEC command **xremote lat** to initiate the connection. The command has the following syntax:

xremote lat *service*

Syntax Description

service Name of the desired LAT service.

After you issue this command, expect the following to occur:

- The communication server down-line loads several initial fonts for the DECwindows login display.
- The terminal displays the DIGITAL logo and DECwindows login box.

When the login box appears, log into the system. Upon completion of login, more fonts are loaded, then the remote session starts up.

Note Due to heavy font usage, DECwindows applications might take longer than expected to start when using XRemote. This additional time is only necessary at startup. Once the application starts, performance and access times should be as expected.

Example

The following example begins connection with a LAT service named WHIRL:

```
cs> xremote lat WHIRL
```

Manual XRemote Session Startup

If you are not using a host computer that supports XDMCP or LAT, you must use manual session startup. Manual session startup involves several steps:

Step 1 Enable XRemote manually on the communication server port.

Step 2 Connect to the host computer.

Step 3 Set the location of the X display.

Step 4 Start up client applications.

Step 5 Return to the EXEC prompt.

Step 6 Enable XRemote manually again on the communication server port.

Step 7 What to do if a session terminates.

You can also establish XRemote sessions between communication servers and exit XRemote sessions. The following sections describe these tasks. See the “XRemote Examples” section at the end of this chapter for XRemote connection examples.

Enable XRemote Manually

To prepare the communication server for manually starting an XRemote session, enter the **xremote** EXEC command at the system prompt.

xremote

This command begins the instructions that prompt you through the connection.

Example

The following example illustrates how a successful manual XRemote session begins:

```
dialup> xremote
XRemote enabled; your display is dialup:2006
Start your clients and type XRemote again
```

The system replies with a message informing you of your X display location This information will be used to inform the host of the location of your X display server.

If no clients are found, you see the following message:

```
No X clients waiting - check that your display is darkstar:2006
```

Check your hosts to determine if an error was made when starting the session. The most likely cause is an improperly specified display location. Another possible cause is the host computer not recognizing the name of the communication server through which you are attempting to make a connection.

Connect to the Host Computer

You can connect to a host using one of the following connection commands, and log in as usual:

telnet

lat

rlogin

Set the Location of the X Display

At this point, you are logged in to the remote host.

Note If you are using a version of Telnet on the remote host that supports the "X Display Location" option (RFC 1096), skip this step and proceed to the section "Start Up Client Applications."

Inform the host computer of the location of your X display, which was provided to you by the communication server when you enabled XRemote manually.

For most versions of the UNIX operating system, the X display location is set by using the **setenv** command to set the environment variable DISPLAY. Refer to your UNIX system's online **X(1)** manual page for more information.

On VAX/VMS, use the **SET DISPLAY** command. Refer to the *VMS DCL Dictionary* for more information.

Note You must install either the TCP/IP transport from Digital or a third-party TCP/IP transport. Contact your VAX/VMS system administrator for the appropriate TCP/IP transport name.

Start Up Client Applications

Now you must start your client applications for your host operating system.

The communication server accepts the X connection attempt from the client application and places the client into a dormant state.

Return to the EXEC Prompt

If it is possible to log off the host computer and keep your X clients running in the background, you can do so now. This conserves resources on the host and communication server that would otherwise be inaccessible until you exited from XRemote state.

If you cannot log off the host computer and keep your clients running, escape back to the communication server prompt using the escape sequence (**Ctrl**-^ **X** by default).

Re-enable XRemote Manually

Begin a manual remote session again; see the step "Enable XRemote Manually." If the X clients connected successfully, the session will be put into XRemote mode, and the clients will be allowed to complete their startup.

If no clients were found, you will see the following message:

```
No X clients waiting - check that your display is darkstar:2018
```

Check your hosts to determine if an error was made when starting the session. The most likely cause is an improperly specified display location. Another possible cause is the host computer not recognizing the name of the communication server through which you are attempting to make a connection.

What to Do if a Session Terminates

In manual operation, the communication server and X terminal remain in XRemote mode until either all clients disconnect or a reset request is received from the X terminal.

If a session terminates during startup, it might be because you invoked transient X clients which set some parameters and disconnected (such as **xset** or **xmodmap**). There must always be one session open or the connection will be reset.

Establish XRemote Sessions Between Communication Servers

A communication server allows a user of an X display server that does not support XRemote to run the XRemote protocols. An X display server (such as a PCX, MACX or UNIX workstation) connected to an Ethernet network can dial out through a communication server on a conventional modem to access an X client program on a host residing on another network. The communication server provides the server-side helper process.

To run XRemote, connect to one of the XRemote ports.

Note The NCD helper process does not support X display devices that use a maximum request and response size larger than 64 kilobytes.

Find out from your administrator whether the connection from your X display terminal is configured as an individual line or a rotary connection. To connect to an individual line, telnet from the X display terminal to port 9000 plus the decimal value of the line number. To make a rotary connection, telnet from the X display server to port 10000 plus the decimal value of the line number. See the publications *Communication Server Configuration Guide* and the *Communication Server Command Reference* for information about how to configure individual lines and rotary connections.

Figure 3-2 illustrates a configuration in which a display server is not running XRemote. In this configuration, the server-side XRemote helper is running on Communication Server 1, and the client-side XRemote helper is running on Communication Server 2.

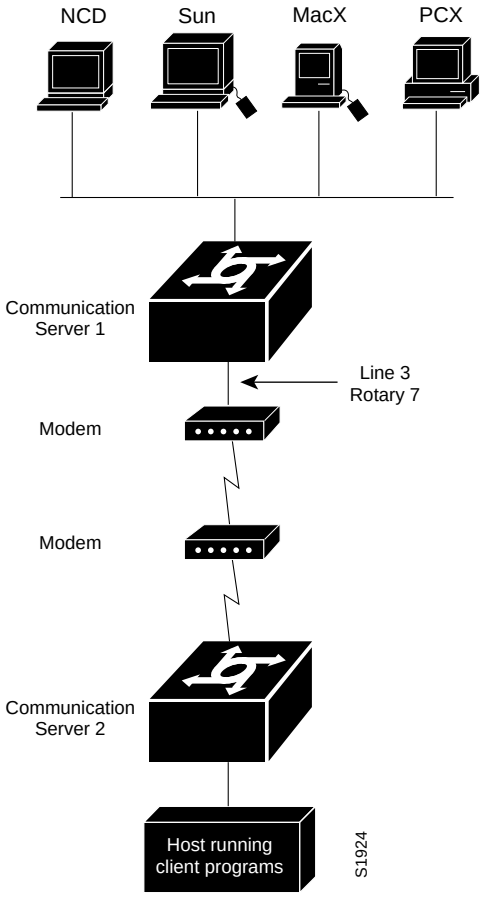

Figure 3-2 XRemote Session between Communication Servers

Exit XRemote Sessions

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

exit

quit

logout

XRemote Examples

Use the examples in this section to understand how to make XRemote connections.

Connecting an X Display Terminal

The following example shows a connection from an X display terminal through a communication server to a host running client programs.

1 Enter the **xremote** command at the EXEC prompt:

```
dialup> xremote
```

2 Read and follow the instruction from the host:

```
XRemote enabled; your display is dialup:2006
Start your clients and type XRemote again
```

3 Connect to the client:

```
dialup> telnet eureka
Trying EUREKA.NOWHERE.COM (252.122.1.55)... Open

SunOS UNIX (eureka)
```

4 Log in at the prompt:

```
login: deal
Password:

Last login: Fri Aug  7 17:17:46 from dialup.nowhere.com
SunOS Release  (SERVER+FDDI+DBE.patched) #14: Fri Jul 31 10:37:29 PDT 1992
```

5 At the client prompt, enter the display name from step 2 and the **xterm** command:

```
eureka% setenv DISPLAY dialup:2006
eureka% xterm &
[1] 15439
```

6 Disconnect from the client:

```
eureka% logout

[Connection to EUREKA closed by foreign host]
```

7 Begin the XRemote session:

```
dialup> xremote
Entering XRemote
```

The communication server and X terminal stay in XRemote mode until either the display manager terminates the session or a reset request is received from the X terminal:

```
Connection closed by foreign host.
eureka%
```

Making XRemote Connections between Communication Servers

This section provides two examples of XRemote connections between communication servers.

The following steps show how an XRemote connection is established for a configuration like the one shown in Figure 3-2. These steps assume that the administrator has set the user's display environment variable to identify the user's X display terminal.

1. From the PCX, MACX, or UNIX machine in Figure 3-2, the user connects to port 9003 on Communication server 1 (CS1). If the administrator has configured a rotary number 7, the user connects to port 10007. (Refer to the publications *Communication Server Configuration Guide* and *Communication Server Command Reference* for more information about rotary groups.)
2. CS1 connects the user to a modem.
3. The modem calls Communication Server 2 (CS2).
4. The user enters **xremote** at the CS2 prompt.
5. The user connects to the remote host from CS2 using the **telnet** command.
6. The user starts the X client program that will run on the remote host and will display on the X display server (PCX, MACX, or UNIX host).
7. The user escapes from the remote host back to the CS2, or logs out if clients were run in the background, and enters **xremote** again at the CS2 prompt.

The following example shows the steps to make an XRemote connection between communication servers. The number 9016 in the first line of the display indicates a connection to individual line 16. If the administrator had configured a rotary connection, the user would enter 10000 plus the number of the rotary instead of 9016.

Step 1 Enter the **telnet** command to make the connection:

```
space% telnet golden-road 9016
Trying 192.31.7.84 ...
Connected to golden-road.cisco.com.
Escape character is '^]'.
```

Step 2 Supply the password for TACACS verification:

```
User Access Verification

Password:
Password OK

 --- Outbound XRemote service ---
Enter X server name or IP address: innerspace
Enter display number [0]:

 Connecting to tty16... please start up XRemote on the remote system
```

Step 3 Dial in to the remote system using the modem, and then log in:

```
atdt 13125554141
DIALING
RING
CONNECT 14400

User Access Verification
Username: deal
Password:
  Welcome to the cisco dial-up communication server.
```

Step 4 Enter the **xremote** command at the EXEC prompt, and then read and follow the instructions from the host:

```
dialup> xremote
XRemote enabled; your display is dialup:2006
Start your clients and type XRemote again
```

Step 5 Connect to the client:

```
dialup> telnet sparks
Trying SPARKS.NOWHERE.COM (252.122.1.55)... Open

SunOS UNIX (sparks)

login: deal
Password:
Last login: Fri Aug  7 17:17:46 from dialup.nowhere.com
SunOS Release  (SERVER+FDDI+DBE.patched) #14: Fri Jul 31 10:37:29 PDT 1992
```

Step 6 At the client prompt, enter the display name from step 4 and the **xterm** command:

```
sparks% setenv DISPLAY dialup:2006
sparks% xterm &
[1] 15439
```

Step 7 Disconnect from the client:

```
sparks% logout

[Connection to SPARKS closed by foreign host]
```

Step 8 Begin the XRemote session:

```
dialup> xremote
Entering XRemote
```

Once the connection is closed by the foreign host, the Xterm window appears on the local workstation screen:

```
Connection closed by foreign host.
sparks%
```