

# Chapter 1

## *XRemote Configuration and Management*

---

# 1

This chapter describes how to configure the XRemote capabilities of Network Computing Devices, Inc. (NCD) X Window terminals on the Cisco protocol translator. You will find the following information in this chapter:

- An overview of Cisco-specific XRemote implementation issues.
- Configuration procedures for making connections.
- EXEC commands for troubleshooting XRemote operation.

Making XRemote connections is described in the chapter “Protocol Translator User Commands.”

The commands to configure XRemote are entered in the privileged configuration mode, entered with the **configure** command. For more information about the configuration mode, see the chapter “Startup and Basic Configuration.”

A command summary is included at the end of the chapter.

---

### *The X Window System*

This chapter does not fully address the details of the X Window System and X-based computing environments. The following are suggested sources for more complete definitions and implementation descriptions:

- *X Library Reference Guide*, Ira Chayut and Camile Cook. A System Publications, Inc.
- *X Tool Kit Reference Manual*, Ira Chayut and Camile Cook. A System Publications, Inc.
- *X User Reference Guide*, Ira Chayut and Camile Cook. A System Publications, Inc.
- *X Window System User's Guide for Version 11*, Tim O'Reilly, Valerie Quercia, and Linda Lamb. O'Reilly and Associates, Inc.

The X Window System™, typically called X, is a network-based graphics window system developed at the Massachusetts Institute of Technology (MIT) in 1984 for workstations running UNIX. It was originally written in the C programming language; however, X is not tied to a specific operating environment, programming language, or hardware configuration.

Previous window systems were *kernel-based* and were thus closely linked to the operating system running on a workstation itself. They typically have run only on discrete systems, such as a single workstation. The X Window System is not part of any operating system, but instead, is composed of application programs. Thus, the X Window System enables flexible, graphics-based network computing across a wide range of operating systems and hardware platforms.

## *X and the Client–Server Model*

The underlying architecture of the X Window System is based on a *client-server* model. The system is split into two parts: *display servers* and *clients*. Display servers provide specific display capabilities and track user input; clients are application programs that perform specific tasks. These two parts can reside on the same computer, or can be separated over a network. In an X terminal environment, such as in NCD terminal implementations, the display server resides on the display station and the client resides on a host computer.

Because X employs this functional partitioning and is independent of both hardware and operating environment, X terminal users can access different types of computers to simultaneously access several applications and resources in a multivendor environment. A user at an X terminal can run and display a calendar program on a VAX, a spreadsheet program on a PC, and a compiler on a workstation concurrently.

## *How XRemote Works*

XRemote is a protocol developed specifically to optimize support for X over a serial communications link. Its compression and decompression algorithms are designed to handle bit-mapped displays and windowing systems.

There are two basic parts to XRemote:

- Local PROM-based elements
- The XRemote helper process

The PROM-based elements consist of the *X server* and *boot monitor*. These are responsible for booting the display station, managing local processes, and handling communications with the host computer. The helper process allows the XRemote display station to communicate, through the protocol translator, with other hosts on the network. PROMs that implement these elements are available from NCD.

---

**Note:** XRemote protocol translation support is disabled on an IGS system whenever routing is enabled.

---

---

## Cisco's Implementation of XRemote

XRemote enables a user of an NCD network display station to run the X Window System via 9600 baud (and faster) modem connections with superior performance compared with conventional serial protocols, such as SLIP (Serial Line Internet Protocol). The NCD terminal must have a compatible XRemote PROM set installed.

Figure 1-1 illustrates XRemote network accessibility via Cisco protocol translators.

*Figure 1-1* Typical XRemote Configuration



Cisco's implementation of XRemote is fully compatible with the NCD XRemote protocol. Remote access to fonts is provided in these ways:

- Using the industry-standard protocol for transporting X traffic over TCP/IP networks.
- Using the Digital Equipment Corporation protocol for transporting X traffic over LAT networks.

Remote access to fonts is provided in these ways:

- Using the Internet standard Trivial File Transfer Protocol (TFTP) for TCP/IP networks.
- Using the Digital Equipment Corporation protocol for font access via LAT.

A single XRemote user can use any combination of TCP/IP and LAT client connections, and any combination of TFTP and LAT font access.

---

## Configuring XRemote

To make a host connection via NCD's XRemote feature and a Cisco protocol translator, you must complete some basic configuration steps.

- Step 1:** Configure the X terminal and protocol translator for fonts to be used and other special parameters.
- Step 2:** Attach cables and set up your modem for use with XRemote (access over asynchronous lines only), or cable the NCD terminal directly to the Cisco protocol translator.
- Step 3:** Set up the NCD terminal's X Server, Network, and Serial Parameters. Refer to your NCD manual for specific details of configuring these parameters.

The following sections describe these tasks. Descriptions of the EXEC commands to maintain and monitor TN3270 connections follow these sections.

### Font Selection

Depending on the fonts to be used on the terminal, you may need to specify the location of a font server. The NCD terminal contains a small set of built-in fonts in local ROM. It is recommended that these fonts be used, as loading fonts over a serial line can increase application startup time. The default for an NCD terminal is to use built-in fonts, unless you log in using DECwindows over LAT. When using DECwindows over LAT, the standard DECwindows fonts are used automatically.

#### *Accessing Nonresident Fonts via TFTP*

When an X terminal application requests a font that is not stored in the NCD ROM, the X terminal makes a request for a font file from the protocol translator. The protocol translator uses the trivial file transfer protocol (TFTP) to load the font from the font server, and then passes the font to the X terminal using the XRemote protocol. The process of loading fonts from the protocol translator to the X terminal can take 30 to 45 seconds, depending on the size of the font file.

An X server can display only the fonts it finds in the directories in its font path. The X server's default font path includes only the built-in fonts. To access fonts stored on a host, you must add the host's font directories to the X server's font path. To do this, use the UNIX command **xset** with the **fp+** argument to add fonts to the end of the server's font path.

For example, to allow your display station to access the 100 dots per inch (dpi) fonts found in the standard font directory, run the following command (at the host system prompt):

```
host_prompt%xset fp+ /usr/lib/x11/ncd/fonts/100dpi
```

For more information, refer to the *NCDware XRemote User's Manual*.

## *LAT Font Selection*

Downloading of fonts occurs automatically when a remote DECwindows login session is initiated using the EXEC **xremote lat** command. Instead of relying on TFTP to download the fonts, the fonts are read in via the LAT protocol.

If you wish to use DECwindows fonts while running standard X applications on a UNIX host, you need to use the UNIX **xset** command or an application that issues an XSetFontPath request to set a font path. You may want to do this if you are primarily a TCP/IP user, but also run some DECwindows applications. The step to accomplish this follows:

- Execute **xset**, or the application to issue an XSetFontPath request, to set the following path:

*/LAT/SERVICE*

where *SERVICE* is a LAT service name with DECwindows support; case is not significant.

When the protocol translator sees a request for font files in that directory, instead of using TFTP for access, it uses LAT to access the specified service.

## *Protocol Translator Configuration*

Using three global configuration commands, you can set several protocol translator options that identify font loaders (hosts), set buffer size, and specify the number of font loader retries. These system configuration commands are available only when the protocol translator privilege level has been enabled, allowing access to the **configure** command. Once you have accessed the system configuration facility, use the **xremote** global configuration commands to modify these parameters. The only parameter that generally must be set is the font loader host identification (only if you intend to download fonts to the X terminal).

### *Specifying Buffer Size*

Use the global configuration command **xremote tftp buffersize** to set the protocol translator buffer size for specific font requirements. This command applies to both TFTP and LAT font access. This command has the following syntax.

```
xremote tftp buffersize [decimal value]  
no xremote tftp buffersize
```

This **xremote** command sets the buffer size used for loading font files to a specified value (in decimal bytes). When the X terminal requests that a font file be loaded, the protocol translator must first load the font file into an internal buffer before passing it to the X terminal. The default value of 70000 is adequate for most font files, but the size can be increased as necessary for nonstandard font files.

The buffer size may be set as low as 4096 bytes, and as large as the available memory on the protocol translator will allow. If you are using LAT font access, you should not lower the buffer size below the default, because the font directory for all of the LAT fonts (created internally) requires 70000 bytes.

The command **no xremote tftp buffersize** restores the buffer size to the default of 70000 bytes.

### *Identifying TFTP Font Servers*

Use the global configuration command **xremote tftp host** to add a specific TFTP font server as a source of fonts for the terminal (and the protocol translator). This command has the following syntax:

```
xremote tftp host [hostname]  
no xremote tftp host [hostname]
```

Each time a new host name is entered, the list on the protocol translator is updated. Font servers are queried in the order of their definition when the X terminal requests a font.

The command **no xremote tftp host** removes a font server from the list maintained by the protocol translator.

### *Determining TFTP Font Loader Retries*

Use the global configuration command **xremote tftp retries** to specify the number of retries the font loader will attempt before declaring an error condition. This command has the following syntax:

```
xremote tftp retries [value]  
no xremote tftp retries
```

The default value is three. Under certain conditions, you may need to increase this number, particularly if the font server(s) is(are) known to be heavily loaded. Acceptable values range from 1 to 15.

The command **no xremote tftp retries** restores the retries number to the default of three.

## *Making Modem Connections*

The following list outlines recommendations for modem setup when using XRemote to access remote resources via a Cisco protocol translator.

### *Modem Requirements*

In general, you can use any modem that provides acceptable performance for your application. The following guidelines apply to an XRemote operation using a modem:

- Disable any error correction and compression features of the modem. As XRemote implements its own compression and error correction, the modem's compression and error correction actually impair performance.

- If you must use a flow control mechanism, hardware flow control (such as RTS/CTS or DTR/DSR) is recommended. Software flow control (such as XON/XOFF) is discouraged.
- The modem should incur minimal delays in round-trip transmissions, even when transmitting small packets, and should be transparent to the data stream.
- The modem should provide true full-duplex transmission at 9600 baud or faster. Half-duplex modems are not suitable for use with XRemote.

---

## Monitoring XRemote Connections

You can use the EXEC command **show xremote** to list XRemote connections and monitor XRemote traffic through the protocol translator. The syntax for this command is as follows:

### **show xremote**

This command provides XRemote parameters applied to the entire system, as well as statistics that are pulled for all active XRemote connections.

The following illustrates screen output when XRemote is enabled on a protocol translator, and XRemote sessions are active.

```
PT#show xremote
XRemote server-wide parameters:
  Font buffersize:      72000           Font retries: 3
  Font memory errors:  0

TFTP font load statistics for host 131.108.1.111:
  Bytes read:          2697239         Files read: 258
  Network errors:      4               File errors: 0

LAT font load statistics for service WHEEL, incarnation 5:
  Bytes read           182401          Files read: 14
  Protocol errors:     0               Insufficient memory: 0

XRemote statistics for tty2:
  Current clients:     9               Total clients: 17
  Requesting client:   5               Current request size: 0
  Replying client:     6               Current reply size: 0
  XDM state:           10              Next timeout: 172460
  Retransmit counter:  0               Local UDP port: 53616
  Keepalive dormancy: 180              Session id: 94
  Unread input:        0               Unwritten output: 0
  Input buffer size:   1024            Output buffer size: 108
  Protocol version:    2               Line state: Connected
  Transmit packets:    50768           Receive packets: 49444
  Transmit errors:     0               Receive errors: 37
  Retransmissions:     403             Receive out of sequence: 76
  Round trip time:     383             Retransmit interval: 766
  Transmit window:     7               Receive window: 7
  Transmit next:       6               Receive next: 3
  Transmit unacked:    6               Receive unacked: 0
```

```

Connection 0 - TCP connection from 131.108.1.55 [Display Manager]
  Client state:      CS_ACTIVE      Byte order: MSBfirst
  Unread input:     0                Unwritten output: 0
  Input buffer size: 1024           Output buffer size: 1024

Connection 1 - TCP connection from 131.108.1.55
  Client state:      CS_ACTIVE      Byte order: MSBfirst
  Unread input:     0                Unwritten output: 0
  Input buffer size: 1024           Output buffer size: 1024

Connection 2 - TCP connection from 131.108.1.55
  Client state:      CS_ACTIVE      Byte order: MSBfirst
  Unread input:     0                Unwritten output: 0
  Input buffer size: 1024           Output buffer size: 1024

Connection 3 - LAT connection from WHEEL
  Client state:      CS_ACTIVE      Byte order: LSBfirst
  Unread input:     0                Unwritten output: 0
  Input buffer size: 1024           Output buffer size: 1024

```

You can use the EXEC command **show xremote line** with a line number to list XRemote connections and monitor XRemote traffic for specific lines on a protocol translator. The syntax for this command is as follows:

**show xremote line** *number*

The argument *number* is a decimal value representing the virtual terminal lines available on a protocol translator.

The following illustrates screen output resulting from a **show xremote line** command when XRemote is enabled on a protocol translator, and XRemote sessions are active. Only information specific to an individual terminal line is provided.

```

PT#show xremote line 3
Xremote statistics for tty3:
  Current clients:      11          Total clients: 19
  Requesting client:   10          Current request size: 0
  Replying client:     10          Current reply size: 0
  XDM state:           10          Next timeout: 173304
  Retransmit counter:  0           Local UDP port: 28384
  Keepalive dormancy: 180         Session id: 29
  Unread input:        0           Unwritten output: 0
  Input buffer size:   1024        Output buffer size: 108
  Protocol version:    2           Line state: Connected
  Transmit packets:    28875       Receive packets: 18644
  Transmit errors:     0           Receive errors: 13
  Retransmissions:    53          Receive out of sequence: 41
  Round trip time:     384         Retransmit interval: 768
  Transmit window:     7           Receive window: 7
  Transmit next:       2           Receive next: 7
  Transmit unacked:    2           Receive unacked: 0

Connection 0 - TCP connection from 131.108.1.27 [Display Manager]
  Client state:      CS_ACTIVE      Byte order: MSBfirst

```



```
Unread input:      0                Unwritten output: 0
Input buffer size: 1024             Output buffer size: 1024

Connection 1 - TCP connection from 131.108.1.27
Client state:      CS_ACTIVE         Byte order: MSBfirst
Unread input:      0                Unwritten output: 0
Input buffer size: 1024             Output buffer size: 1024

Connection 2 - TCP connection from 131.108.1.27
Client state:      CS_ACTIVE         Byte order: MSBfirst
Unread input:      0                Unwritten output: 0
Input buffer size: 1024             Output buffer size: 1024
```

---

## *Debugging XRemote*

Although several privileged EXEC debugging commands are provided with the Cisco XRemote implementation, they are generally used for internal Cisco technical support purposes. The following is a listing of these debug commands. For each **debug** command, there is a corresponding **undebug** command to disable the reports.

### **debug xremote-bytestream-flow**

The **debug xremote-bytestream-flow** command enables logging of XRemote bytestream state machine flow.

### **debug xremote-bytestream-protocol**

The **debug xremote-bytestream-protocol** command enables logging of XRemote bytestream protocol packets.

### **debug xremote-deltas**

The **debug xremote-deltas** command enables logging of XRemote delta compression information.

### **debug xremote-files**

The **debug xremote-files** command enables logging of XRemote file access.

### **debug xremote-flow**

The **debug xremote-flow** command enables logging of XRemote client flow.

**debug xremote-packet**

The **debug xremote-packet** command enables logging of XRemote packets.

**debug xremote-replies**

The **debug xremote-replies** command enables logging of XRemote replies.

**debug xremote-requests**

The **debug xremote-requests** command enables logging of XRemote requests.

**debug xremote-xdm**

The **debug xremote-xdm** command enables logging of XRemote XDMCP protocol and state machine events.

---

## *XRemote Global Command Summary*

The following XRemote global configuration commands specify system-wide parameters for XRemote support.

**[no] xremote buffersize** [*decimal value*]

Sets the buffer size used for loading font files to a specified value (in decimal). When the X terminal requests that a font file be loaded, the protocol translator must first load the font file into an internal buffer before passing it to the X terminal. The default value of 70000 is adequate for most font files, but the size can be increased as necessary for non-standard font files. The buffer size may be set as low as 4096 bytes, and as large as the available memory on the protocol translator will allow. The command **no xremote buffersize** restores the buffer size to the default of 70000 bytes.

**[no] xremote tftp host** [*hostname*]

Adds a specific font server as a source of fonts for the terminal (and the protocol translator). Each time a new host name is entered, the list on the protocol translator is updated. Font servers are queried in the order of their definition when the X terminal requests a font. The command **no xremote tftp host** removes a font server from the list maintained by the protocol translator.

**[no] xremote tftp retries** *value*

Specifies the number of TFTP retries the font loader will attempt before declaring an error condition. The argument *value* is an integer.

The default value is three. Under certain conditions, you may need to increase this number, particularly if the font servers are known to be heavily loaded. Acceptable values range from 1 to 15. The command **no xremote tftp retries** restores the retries number to the default of three.

