CTIX™ X.25 NETWORK GATEWAY MANUAL

First Edition (September 1987) 73–00287–A

Convergent Technologies makes no representations or warranties with respect to the contents hereof and specifically disclaims any implied warranties of merchantability or fitness for any particular purpose. Further, Convergent Technologies reserves the right to revise this publication and to make changes from time to time in its content without being obligated to notify any person of such revision or changes.

This document was produced using the Document Designer Series.

CONTENTS

LIST OF FIGURES

UPPERCASE IOCTL command names appear in UPPERCASE.

Underline Manual titles are underlined.

User-entered variables are underlined.

[ ] Command options are enclosed in square brackets [ ]. Do not type the [ ] when you specify command options.

| A vertical bar between values specifies an either/or choice. Specify only one value.

/ A slash separates or delimits individual values.

A slash separates directory names in a CTIX pathname.

- A hyphen specifies a range of values and is placed between the upper and lower limits of the specified range.

1 INTRODUCTION

Chapter 1 introduces the CTIX X.25 Network Gateway and the CTIX X.25 Network Gateway Manual. This chapter contains the following information:

- an introduction to wide area networking and the CTIX X.25 Network Gateway

- an overview outlining the basic features and functions of the CTIX X.25 Network Gateway

- a brief description of Gateway components

- an introduction defining the purpose, scope, intended audience, and content of this manual

> **NOTE**
>
> The CTIX X.25 Network Gateway is referred to as the Gateway throughout the remainder of this manual.

OVERVIEW

The Gateway provides a packet-switching method for wide area networks (WANs). Wide area networking is vital to achieving the information processing and sharing goals of most organizations. Packet switching is a useful technology when implementing data communications between remote workgroups or other organizations. Packet switching is an efficient and relatively inexpensive method of reliably delivering data.

Since its adoption by the International Telephone and Telegraph Consultative Committee (CCITT) standards body, the X.25 packet switching recommendations have been implemented for most U.S. and international public data networks (PDNs).

WHAT IS THE X.25 NETWORK GATEWAY?

Overview

The Gateway enables users of S/MT and S/PC systems to enjoy the advantages of a packet-switched network and to build distributed information systems.

The Gateway is easy to install, configure, and use. The Gateway enables CTIX application programs to communicate over an X.25 PDN on one or more leased lines. Line speed limitations vary among S/MT and S/PC systems.

Features and Functions

The following is a partial list of features designed to enhance the Gateway's functionality:

- The Programmatic Interface provides a simple method of developing new programs to access the Gateway.

- The X.25 Configuration Utility is provided for administration and configuration of the Gateway. The X.25 Configuration Utility administers all supporting configuration files and provides a simple option selection method for enabling and disabling the Gateway.

- The X.25 line can be activated, deactivated, and reconfigured without affecting the operations occurring on another X.25 line.

- Detailed error messages are provided.

- "Nationalizable" messages are provided. All printable messages can be translated into other languages.

Components

The Gateway consists of the following components:

- X.25 Loadable Character Device Driver
- X.25 Configuration Utility
- Programmatic Interface
- Library
- x25d daemon
- x25start

The X.25 Loadable Character Device Driver is the software that performs the functions of the packet, frame, and physical layer protocols. These protocols provide the character device driver interface to the CTIX applications. The driver is sometimes referred to as the "engine" of the Gateway. For this reason, the term driver is sometimes used throughout this manual interchangeably with the term Gateway.

The X.25 Configuration Utility is an easy-to-follow configuration tool that allows you to administer Gateway functions, X.25 parameters, and supporting files. You can use the X.25 Configuration Utility to select and modify line and network parameters, as well as X.25 packet and frame level parameters. The X.25 Configuration

Utility also notifies you when parameter combinations are incompatible or when required selections for proper Gateway operation are missing.

In addition to the X.25 Configuration Utility, default configuration parameter files are available with network-specific parameters, for example, for the Telenet or PSS networks. These facilities allow system administrators to install, configure, and successfully use the Gateway without an in-depth knowledge of X.25.

The Programmatic Interface is the interface between the Gateway and a CTIX application. It allows the application to establish a communications connection for data transfer. The Programmatic Interface uses the five basic system calls: **open**, **close**, **read**, **write**, and **ioctl**.

The Library is a collection of predefined Gateway functions that includes x25open, x25accept, and x25call.

The x25d daemon is a background process that periodically scans the Gateway for incoming calls and virtual circuit status. x25d executes programs on incoming calls based upon entries in the **proc_table**. x25d can accept an incoming call when the subaddress has an application program attached to it. x25d maintains a log file (**/usr/lib/x25/x25d.log**), which includes a record of incoming calls, the number of active virtual circuits, and various error conditions.

x25start is a background process that initiates the loadable character device driver.

Chapter 2, "CTIX X.25 Network Gateway," provides an in-depth description of the Gateway's components and how they function in relation to one another.

MANUAL OVERVIEW

PURPOSE AND SCOPE

This manual provides technical information and operating procedures required by system administrators and programmers who are responsible for installing, configuring, operating, maintaining, monitoring, and troubleshooting the Gateway. A description of the Programmatic Interface and instructions for creating user application programs to run in conjunction with the Gateway are also provided for programmers.

"Contents Guide" provides a content overview of each chapter and appendix to assist you in determining where the appropriate information required to complete your specific task is located.

Gateway-specific data communications terms are listed and defined in Terminology boxes in Chapters 2 and 4.

The Glossary provides definitions of the technical terminology used throughout this manual.

Wherever possible, examples and sample exercises are provided to guide you through the various procedures presented in this manual.

AUDIENCE

The CTIX X.25 Network Gateway Manual addresses the information requirements of system administrators and programmers.

NOTE

In this manual, the terms system administrator and programmer are used to define job functions, responsibilities, and required technical knowledge. These terms are not intended to reflect actual job titles or descriptions.

System Administrator

In this manual, the term system administrator refers to anyone who is responsible for installing, configuring, operating, monitoring, maintaining, and troubleshooting the Gateway.

The information presented in this manual assumes that the system administrator is knowledgeable of the CTIX operating system, is a CTIX "super-user," and is familiar with general communications concepts. The system administrator must be familiar with the hardware components to which the Gateway is configured.

With the aid of this manual, the system administrator must interpret error messages and status codes that may be received during Gateway operation.

Programmer

In this manual, the term programmer refers to anyone who is responsible for creating CTIX application programs to operate in conjunction with the Gateway.

The information presented in this manual assumes that the programmer's knowledge is equivalent to that listed above for system administrators. In addition, it is assumed that programmers possess both an understanding of the X.25 protocol and the required expertise to write CTIX application programs.

CONTENTS GUIDE

Chapter 1, "Introduction," introduces the Gateway and provides a description of the CTIX X.25 Network Gateway Manual.

Chapter 2, "CTIX X.25 Network Gateway," provides a description of Gateway components, including the CTIX Loadable Character Device Driver, X.25 Configuration Utility, Programmatic Interface, Library, x25d, and x25start. A concepts of operation section and a functional block diagram is also included.

Chapter 3, "Using the X.25 Configuration Utility," provides complete tutorial procedures for using the Gateway's X.25 Configuration Utility. A complete description of all available configuration parameters is also provided.

Chapter 4, "Programmatic Interface," presents a complete description of the Gateway's Programmatic Interface and the **open**, **close**, **read**, **write**, and **ioctl** basic system calls. Instructions for writing a CTIX application program using X.25 and a sample program are also included.

Chapter 5, "Programmatic Interface ioctl Formats," presents a complete description of the supported **ioctl** formats supported during Gateway operation.

Chapter 6, "Troubleshooting," provides a complete list of error messages and status codes associated with the Gateway.

Appendix A presents an overview of X.25 concepts.

Appendix B provides a complete description of supported packet formats.

Appendix C provides a listing of the SERVER LUN, G2 LUN, and USER LUN **ioctl** codes.

Appendix D provides lists of virtual key names used on several kinds of terminals and keyboards.

The Glossary defines technical terms used throughout this manual; an index follows.

2 CTIX X.25 NETWORK GATEWAY

Chapter 2 describes the Gateway and its components. This chapter contains the following information:

- terminology and definitions
- a description of the CTIX loadable character device driver
- a description of the X.25 Configuration Utility and its purpose during Gateway operation
- a description of the Programmatic Interface and its purpose
- a description of the X.25 Library functions, including x25open, x25accept, and x25call
- a description of the x25d daemon
- a description of x25start
- concepts of Gateway operation

WHAT IS X.25?

X.25 is a set of telecommunications recommendations established by the International Telephone and Telegraph Consultative Committee (CCITT). Its specific purpose is to define the interface between packet mode DTE (data terminal equipment) and DCE (data circuit equipment), allowing the packet mode DTE to access a public data network (PDN).

TERMINOLOGY

Gateway

A gateway provides a communications path or connection point to a specified communications system or network.

X.25 Loadable Character Device Driver (x25.o)

The Loadable Character Device Driver is software that performs the functions of the physical, frame, and packet layers (Levels 1, 2, and 3 respectively) of the X.25 protocol. It is sometimes referred to as the "engine" of the Gateway.

X.25 Configuration Utility (x25man)

The X.25 Configuration Utility provides a series of configuration menus designed to configure the Gateway and administer its operation.

Programmatic Interface

The Programmatic Interface is a standard CTIX device driver interface that allows the application program to access the device driver. The Programmatic Interface uses the basic system calls: **open**, **close**, **read**, **write**, and **ioctl**.

Public data network

Public Data Networks provide data communication services to the public for a subscription fee.

x25d Daemon

The x25d daemon is a background process that periodically scans the Gateway for incoming calls and virtual circuit status. x25d also spawns processes according to the specifications of the **proc_table**. Status information gathered by x25d is recorded in the **/usr/lib/x25/x25d.log** file.

X.25 Library

The X.25 Library is a collection of predefined functions, including an easy method for virtual circuit establishment. Supported library functions include x25open, x25accept, and x25call.

x25start

x25start is a background process that runs the Loadable Character Device Driver as a process.

WHAT IS A GATEWAY?

A gateway is a collection of software programs designed to provide a communications path or connection point between two similar or dissimilar communications systems. In the case of the X.25 Network Gateway, the Gateway provides communication between your CTIX system and a packet-switching X.25 public data network. It may also provide point-to-point communication between two S/MT or S/PC systems, using the DTE/DCE capabilities of the Gateway.

GATEWAY OVERVIEW

The Gateway enables CTIX application programs to communicate with an X.25 public data network (PDN). Communications occur over one or more full-duplex (FDX) lines. Line speeds vary between S/MT and S/PC systems.

The Gateway provides a Programmatic Interface to the CTIX application programs. The Programmatic Interface allows the application to establish a communications connection and enables data transfer.

Data is transferred through the Gateway in one of the following modes of operation:

- packet mode
- data mode

Packet mode refers to a mode of operation in which the CTIX application programs may specify X.25 packet transfer over the Gateway. To achieve successful communications while operating in packet mode, the configuration parameters and protocol variables must conform with both the Gateway, which is configured to your system, and the PDN requirements.

Data mode refers to a mode of operation in which the CTIX application program transfers only pure data. When operating in data mode, the Gateway itself performs the segmenting and packetizing of data. The application program is not required to observe X.25 procedures and packet formats.

Packet mode and data mode are both modes of operation relevant to a particular virtual circuit. Either mode of operation can be used interchangeably with either the X.25 or X.29 adapters. The X.25 adapter interprets and transmits only X.25 packets. The X.29 adapter interprets and transmits X.29 message packets. (See Data Communication Networks: Services and Facilities, Terminal Equipment and Interfaces.)

GATEWAY COMPONENT OVERVIEW

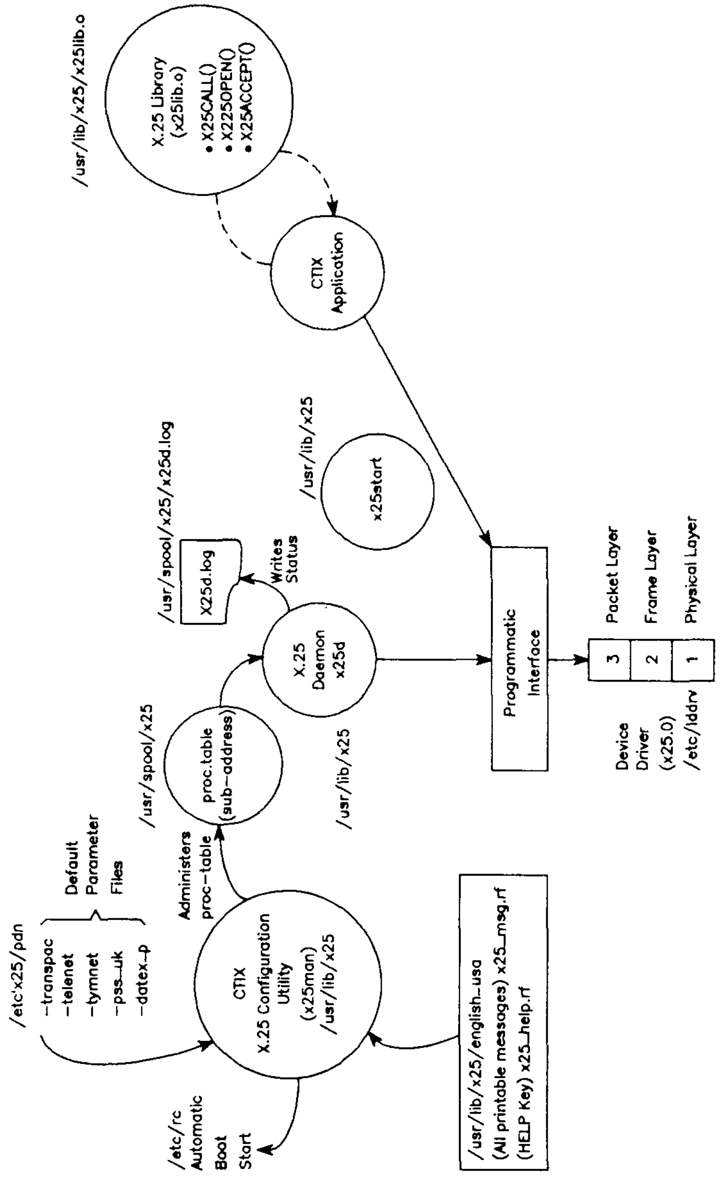
The Gateway consists of the following components:

- o Loadable Character Device Driver (x25.o)
- o X.25 Configuration Utility (x25man)
- o Programmatic Interface
- o X.25 Library (x25lib.o)
- o x25d daemon
- o x25start

Figure 2-1 shows the relationship between Gateway components.

X.25 LOADABLE CHARACTER DEVICE DRIVER

The X.25 Loadable Character Device Driver is the software that performs the functions of the packet (Level 3), frame (Level 2), and physical (Level 1) layers of the X.25 protocol. The protocols provide the character device driver interface to the CTIX application running on your system.



287.2-1

Figure 2-1. Relationships Between Gateway Components

X.25 CONFIGURATION UTILITY

The X.25 Configuration Utility (x25man) consists of an easy-to-follow series of Gateway configuration menus. Using the menus enables you to configure and fully utilize the functions of your Gateway and the services provided by the PDN to which you subscribe. The X.25 Configuration Utility allows you to administer the overall operation of your Gateway and perform the following functions:

- activate the Gateway
- deactivate the Gateway
- select and modify various line and network parameters
- select and modify X.25 packet and frame parameters.

The X.25 Configuration Utility also provides a number of facilities that enable you to choose how incoming calls are answered by the Gateway. For example, you can choose whether the X.25 software or the application will answer an incoming call. You can also decide whether the application will interface to the X.25 packet mode directly or through a virtual terminal interface to the X.25 software.

Each PDN operates in a slightly different fashion; therefore, during Gateway configuration it is recommended that you make use of the available parameter file associated with the PDN to which you are subscribing. In doing so, you ensure proper communication between the PDN and your Gateway. See Chapter 3, "Using the X.25 Configuration Utility," for additional information.

PROGRAMMATIC INTERFACE

The Gateway provides a standard CTIX device driver interface called the Programmatic Interface. The Programmatic Interface uses the **open**, **close**, **read**, **write**, and **ioctl** basic system calls. The device driver is represented by special files consisting of major and minor device numbers. X.25 devices are represented by a pathname and appear in the following form:

/dev/x250nn

where:

nn represents the two digits of the minor device number that corresponds to the Logical Unit Number (LUN) referred to throughout this manual.

The Programmatic Interface uses the following five basic system calls:

open The **open** function opens a CTIX X.25 special file, or LUN. It allocates a LUN for communication over the X.25 network. The **open** function does not initiate packet transmission.

close The **close** function closes a CTIX X.25 special file, or LUN. The LUN is deallocated, and the associated virtual circuit is cleared.

write The **write** function presents data buffers (or packets) to the Gateway for subsequent transmission to the X.25 line.

read The **read** function receives data buffers (packets) that were previously received by the Gateway on the X.25 line.

ioctl Many **ioctl** commands are defined by CTIX. Additional **ioctl** commands are defined for the Gateway to provide various mechanisms to control parameters and inquire about the status of the Gateway.

See Chapter 4, "Programmatic Interface," for additional information about the **read**, **write**, **open**, **close**, and **ioctl** system calls. See Chapter 5, "Programmatic Interface ioctl Formats," for additional information about the **ioctl** formats supported by the Gateway.

X.25 LIBRARY

The X.25 Library is a set of predefined Gateway functions. The following library functions are supported by the Gateway:

- x25open
- x25call
- x25accept

SYNOPSIS

```
x25accept(path, line, lcn, adapter, mode)
char *path;
int line, lcn, adapter, mode;
```

where:

line — is the line number (specified by x25d).

lcn — is the virtual circuit number (specified by x25d).

adapter — is the requested adapter (X.25 or X.29).

mode — is the requested mode (packet or data).

DESCRIPTION

The x25accept function performs all of the necessary actions to accept an incoming call. The following steps are required to accept an incoming call:

1. Open a free LUN.

2. Read the incoming call.

3. Write a standard Call Accept packet.

4. Set the requested mode (either packet or data).

RETURN VALUES

If successful, a valid file descriptor for the LUN is returned. In the case of an error, -1 is returned.

SYNOPSIS

```
x25call(line, adapter, mode, number)
int line, adapter, mode;
char number[];
```

where:

line — is the line number to use.

adapter — is the adapter to use (A_X25 or A_X29), as defined in the file **/usr/include/sys/x25.h**).

mode — is the specified mode (data or packet).

number — is a null-terminated string representing the called DTE address.

DESCRIPTION

The x25call function performs the following functions required to establish a virtual circuit by sending an outgoing call:

- o opens a free LUN
- o sends an IOX25MODE for the line and adapter number specified in the argument
- o sends an IOX25OPN

- sends an outgoing call to the address specified in the number string specified in the argument

- waits for a call confirmation packet

- sets the transmission mode to the value of the mode argument (data or packet mode)

The address specified in the argument is a string of ASCII digits (null-terminated) that represent the remote DTE address.

RETURN VALUES

If successful, a valid file descriptor for the LUN is returned. In the case of an error, -1 is returned.

SYNOPSIS

```
x25open(lun_path, line, adapter, lcn)
char lun_path[];
int line, adapter, lcn;
```

where:

lun_path is the path name of the LUN returned. The lun_path field should accommodate a minimum of 15 characters.

line is the logical line to use (0, 1, 2, or 3).

adapter is the adapter to use (A_X25 or A_X29), as defined in the file **/usr/include/sys/x25.h**).

lcn is the logical unit number to use. The value of lcn should be -1 if the LUN will be used for an outgoing call.

DESCRIPTION

The x25open function performs the following actions:

- opens an LUN to send an outgoing call

- opens an LUN to accept an incoming call
- opens an LUN for data exchange (if it is a permanent virtual circuit)

In the case of an incoming call, lcn must be initialized to the value specified by x25d when the application is activated.

In the case of an outgoing call, lcn must be set to -1. The X.25 software is responsible for choosing a free LUN.

The available device name (**/dev/x250**nn) is returned in the string "lun_path."

RETURN VALUES

If successful, a file descriptor is returned to the opened LUN. In the case of an error, -1 is returned.

x25d DAEMON

The x25d daemon is a background process that periodically scans the Gateway for incoming calls and virtual circuit status. x25d uses the SERVER LUN to extract information from the Gateway. It then records the information to the **x25d.log** file.

Incoming calls are received by x25d. x25d reads the Incoming Call packet from the SERVER LUN to determine its subaddress (the subaddress is the last two digits of the incoming call address). Identifying the subaddress enables x25d to determine the receiving application process. If there is no entry in the **proc_table** for the specified address, the incoming call is returned with a Clear. Otherwise, the application program defined by the entry is spawned off (becomes an independent process) from x25d with a set of arguments.

The CTIX application program corresponding to a certain X.25 subaddress is specified in the **proc_table**. You may specify up to 100 entries in the **proc_table** that correspond to a defined subaddress range from 00 to 99.

NOTE

> The **proc_table** is read only once per minute. For this reason, new entries may not be immediately active.

The **proc_table** entry allows parameters to be passed on to the application, along with the line number and logical channel number (LCN) parameters from the incoming call. A total of 12 user definable parameters may be specified. The parameters are passed on as string values in the "argv" and "argc" arguments.

Two flags are defined for each **proc_table** entry. One of the flags enables you to specify whether the application program will answer the incoming call (that is, transmit the Call Accept packet) or whether the incoming call will be automatically answered by x25d prior to spawning the application program.

The other flag is used to specify use of the virtual terminal interface.

x25d updates the status of the Gateway in the **x25d.log** file. The following information is maintained in the **x25d.log** file in a continuous fashion:

- information on incoming calls

- the number of active virtual circuits

- a report of various error conditions

An IOX25LST is periodically issued to the Gateway. IOX25LST extracts the status on the number of virtual circuits and lines. When a change is detected, x25d writes an update to the **x25d.log** file.

Periodically purge the **x25d.log** file to prevent continuous growth. This procedure is easily accomplished with an entry in the **crontab** file.

x25start

x25start is a background process that initiates the loadable character device driver.

CONCEPT OF OPERATION

The following section is designed to provide programmers who are responsible for creating CTIX applications with an understanding of how the Gateway operates. In particular, this section describes how the components of the Gateway work in relation to the Programmatic Interface.

LOGICAL CHANNELS AND LOGICAL UNITS

A virtual circuit is designated by a logical channel number (LCN), which is used by the packet level (Level 3) of the X.25 protocol and represented by a logical unit (LU). An LU is referred to by the logical unit number (LUN).

An application opens the LUN. Once the application successfully opens the LUN, CTIX retrieves a file descriptor for the LUN. From this point on, the application uses the file descriptor to access the LUN. (The LUN corresponds to the minor device number of the X.25 driver).

Although the LCN and LUN numbers for a virtual circuit may differ, they are associated with one another for the duration of that particular virtual circuit. For example, to establish a call, the Gateway may select LCN 5 and LUN 4. The association between the LCN and LUN continues only for the duration of the virtual circuit.

INIT LUN, G2 LUN, and SERVER LUN are LUNs allocated for special purposes.

INIT LUN is allocated for the initial loading of the configuration parameters.

SERVER LUN provides services to all LUNs, including the automatic allocation of LUNs to application processes.

G2 LUN serves as a secondary SERVER LUN in the Gateway.

All other LUNs are referred to as USER LUN and are used for the applications operation on the virtual circuits. Special LUN pathnames are defined in **/usr/include/sys/x25.h**.

The system call functions and their arguments are described in Chapter 4, "Programmatic Interface ioctl Formats."

DATA TRANSFER

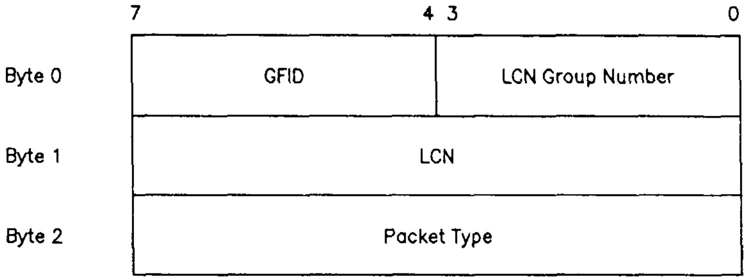
Packet Mode

When the Gateway is operating in packet mode, application programs can transmit and receive X.25 packets by using the **read** and **write** system calls of the Programmatic Interface. The packet itself is contained in the buffer portion of the argument and is specified with a 3-byte packet header (the first 3 bytes of the buffer), followed by an optional data portion. Figure 2-2 shows a sample packet.

```
int i;
unsigned char gfid, len, packtype;
unsigned char packet[MAX_DATA_SIZE+3];
unsigned char *data;

packet[0] = gfid;        /*General format ID     */
packet[1] = lcn;         /*Logical channel number*/
packet[2] = packtype;    /*Packet type           */
for(i=3; i<(MAX_DATA_SIZE+3); i++)
   packet[i] = *data++;
                         /*User data             */
```

Figure 2-2. Sample Packet

Figure 2-3 shows the packet header format. In byte 0, General Format Identifier (GFID), the application program may specify the Q bit for X.29 messages and the D bit for end-to-end data confirmation. These bits are transparently transmitted by the Gateway. The logical channel group and the logical channel number in byte 1 are supplied by the Gateway when writing packets and should always be set to 0. When packets are read from the Gateway, they contain the actual values used in the communications.



	7 4	3 0
Byte 0	GFID	LCN Group Number
Byte 1	LCN	
Byte 2	Packet Type	

Byte 0 contains both GFID and LCN Group Number.

287.2-3

Figure 2-3. Packet Header Format

The third byte must be encoded to identify the packet type of the transmitted packet. The following lists identify packet types that may be read or written from the Gateway.

NOTE

Using packet types that are not specified in the following list prompts an error return to the application program.

The following is a list of packet types that can be read from the Gateway:

- Call Connected
- Clear Indication
- Data Packet
- Interrupt
- Reset Indication

The following is a list of packet types that can be written to the Gateway:

- Call Request
- Clear Request
- Data Packet
- Interrupt
- Reset Request

The Call Request packet can contain an optional 16-byte data field. If the Fast Select facility is supported by the DCE, Fast Select data may also be transferred.

The data packet can contain any type of data bytes (the entire contents are transparently transferred by the Gateway). The maximum size of the data packet must be specified as one of the following:

- 16
- 32
- 64
- 128
- 256
- 512
- 1024

The application must obey the packet size configured or negotiated by the Gateway. If the application attempts to write a packet larger than the Gateway is configured to accommodate, an error is returned. The Gateway defaults to 128 data bytes and supports a maximum size of 1024 data bytes. The Data Packet Type field is always set to 0 for **read** and **write** system calls; that is, there is no P(R), N(R) encoding/decoding at the application level.

Data Mode

The data mode of the X.25 adapter is entered into from packet mode by issuing the IOX25DATA **ioctl** command. When operating in data mode, only user data can be read or written to the Gateway. The advantages of using data mode are that the application is not required to specify the packet header and there is no restriction on the size of the transferred data buffer. The application is also not required to have any knowledge of X.25 protocols.

If the buffer being written is larger than the specified maximum data size of the data packet, the Gateway automatically segments the buffer into data packets of the specified maximum size and sets the M bit (more bit). This procedure continues as long as additional data remains in the buffer. During a **read** operation, the Gateway concatenates data packets with the M bit set.

A user-definable signal, SIGBAND (**/usr/include/sys/x25.h**), is issued to the application program when any packet other than a data packet is received while the Gateway is operating in data mode. When the application program detects the signal, it monitors for any exception conditions, such as Clear, Reset, or Interrupt, on the virtual circuit. The application may use an IOX25STR **ioctl** command to determine the type of exception or change to packet mode for error recovery (for example, to reestablish the connection).

Figure 2-4 shows the X.25 Network Gateway concept of operation.

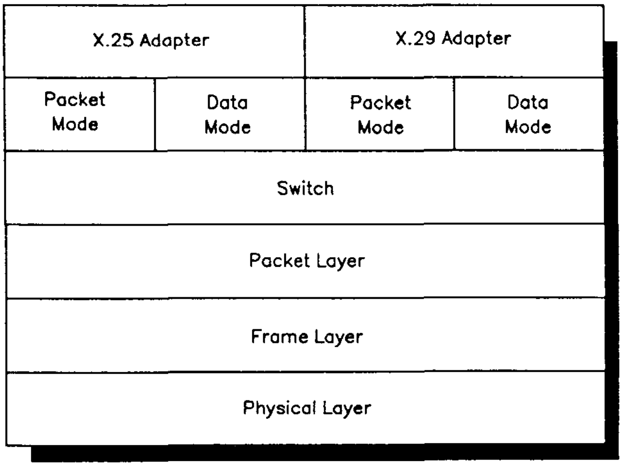


Figure 2-4. Concept of Operation

3 USING THE X.25 CONFIGURATION UTILITY

Chapter 3 describes the X.25 Configuration Utility and how it is used to configure the Gateway. This chapter contains the following information:

- a reference to the Release Notice for complete Gateway installation procedures
- an overview of the X.25 Configuration Utility
- a complete description of the X.25 Configuration Utility menus
- tutorial operating procedures for using the X.25 Configuration Utility menus to configure the Gateway

GATEWAY INSTALLATION

Refer to the Release Notice for MightyFrame CTIX X.25 Network Gateway, for complete installation procedures.

X.25 CONFIGURATION UTILITY OVERVIEW

The X.25 Configuration Utility is designed to provide an easy-to-follow method of configuring and administering Gateway operation. The X.25 Configuration Utility enables you to complete the following configuration procedures:

- activate or deactivate the Gateway on single or multiple lines
- modify line and network parameters

- modify X.25 packet and frame level parameters to fully utilize the services of the public data network (PDN)

- determine the manner in which incoming calls are answered by the CTIX application program

There are various public data networks to which you may subscribe. The predetermined parameters and protocols for each of these networks are slightly different. Therefore, when using the X.25 Configuration Utility to configure your Gateway, you must specify the parameter values and protocols of your Gateway to correspond with the specified subscription parameters of the network to which your Gateway is connected.

The X.25 Configuration Utility provides facilities that enable you to choose how incoming calls are answered by the application. You can specify whether the X.25 software or the application itself should accept an incoming call. You can also specify whether the application directly interfaces with the X.25 packet mode or goes through the virtual terminal interface to the X.25 software. (The virtual terminal interface is part of the CTIX X.25 Terminal/Host Adapter.)

The X.25 Configuration Utility directly controls or interfaces with the following X.25 tables, processes, or files:

- X.25 proc_table

  The X.25 **proc_table** is located in **/usr/spool/x25/proc_table** and consists of various entries that determine how incoming calls are answered, based upon the subaddress. The **proc_table** is managed by the X.25 Configuration Utility and used by x25d.

- default parameter files

  The default parameter files are located in the **/etc/x25/pdn** directory. Each file consists of binary format parameters specific to the network that your Gateway is connected to. These parameters are used as the default values for the initialization table. A default parameter file exists for each PDN to which your Gateway may be connected. The default parameter files are managed and used by the X.25 Configuration Utility.

- x25d daemon

  The x25d daemon is a background process designed to address and determine routing for all incoming calls and log statuses. When an incoming call is received, x25d scans the **proc_table** and dispatches the call to the appropriate application.

- x25start

  x25start runs as a background process and initiates the Gateway driver.

The sole responsibility of the X.25 Configuration Utility is to manage the Gateway's initialization table and **proc_table**. If the default parameter file is properly specified, operation of the Gateway may be as simple as completing the following steps:

1. Select the appropriate default parameter file.

2. Assign a RS-232-C port for connection to the public data network.

3. Allocate virtual circuit information for two-way virtual circuits only.

Several menus are provided to assist you in specifying line and network, and packet and frame level parameters for special network requirements.

The default parameter file is modified and downloaded to the driver. You may also choose to save a copy of the configuration file that you have created. Choose the Configure X.25 Network Gateway option, located on the X.25 Configuration Utility Main Menu, to create your own Configuration file.

The x25d process is initiated by the X.25 Configuration Utility to service incoming calls. When an incoming call arrives, x25d detects the last two digits of the called address (contained in the Incoming Call packet) and scans the **proc_table** to dispatch the appropriate application to the call. If you have specified that x25dshould answer incoming calls, x25d does so and enters into data mode for the application. Otherwise, x25d passes the line and LCN directly onto the application. The application then opens a LUN and reads the Incoming Call packet. When the application is configured to interface with X.25 packet mode, it can transmit and receive X.25 packets. When the application is configured to use the virtual terminal interface, it must first issue the IOX25PKT **ioctl** command before reading and writing data bytes.

The tree structure of the X.25 Configuration Utility is shown below.

Configure X.25 Network Gateway

- Default Parameter File
  - datex_p
  - pss_uk
  - telenet
  - transpac
  - tymnet
- Line and Network Parameters
  - Port Number
  - Network Type
  - Line Type
  - Clock
  - Auto Start
- Packet Level Parameters
  - Modulus
  - Default Packet Size
  - Maximum Packet Size
  - D-bit Used
  - Default Window Size
  - Maximum Window Size
  - T20 Timer
  - T21 Timer
  - T22 Timer
  - T23 Timer
  - Reset Request
  - Clear Request
  - Configuration of VCs
    - Number of Permanent VCs
    - LC Group Number for Permanent VCs
    - Starting LC Number for Permanent VCs
    - Number of Incoming-Only VCs
    - LC Group Number for Incoming-Only VCs
    - Starting LC Number for Incoming-Only VCs
    - Number of Two-Way VCs

- Configuration of VCs (continued)
  - LC Group Number for Two-Way VCs
  - Starting LC Number for Two-Way VCs
  - Number of Outgoing-Only VCs
  - LC Group Number for Outgoing-Only VCs
  - Starting LC Number for Outgoing-Only VCs
- Configuration of Facilities
  - Fast Select
  - Incoming Calls Barred
  - Reverse Charging

Frame Level Parameters

- Modulus
- Window Size
- Tl Timer
- N2
- T3 Timer

Start X.25 Network Gateway

Shut Down X.25 Network Gateway

Reconfigure X.25 Line

Status of X.25 Network Gateway

Hardware State

Frame Level State

Number of Virtual Circuits

Line State

Line Parameter

Frame Level Parameters

Configure proc table

View Active Entries

Add Entry
Sub-Address
Answer Mode
Interface Mode
Command
Arguments

Delete Entry

Disable Entry

Enable Entry

View All Entries

Configure X.25 Network Gateway Boot Sequence

MENU OPERATION

HOW TO DISPLAY THE X.25 CONFIGURATION UTILITY MAIN MENU

The default parameter files of the Gateway are located in **/etc/x25/pdn**. The X.25 Configuration Utility resides in **/usr/lib/x25**.

To initiate the X.25 Configuration Utility, complete the following steps:

1. Specify the **LANG** (language) parameter. For example, to display the X.25 Configuration Utility menus in English, enter the following:

   LANG=english_usa ; export LANG

   (Use **setenv** for the C shell.)

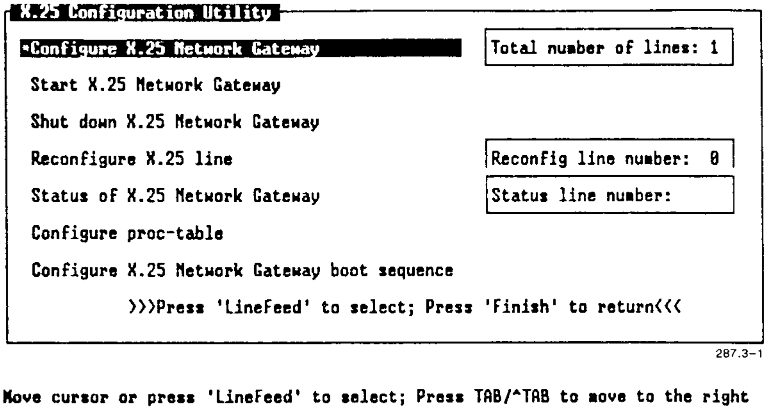
2. Press **Return**.

3. At the command line enter:

   ```
   /usr/lib/x25/x25man
   ```

4. Press **Return.**

The X.25 Configuration Utility Main Menu, as shown in Figure 3-1, appears on your terminal. You are now ready to begin the Gateway configuration process.


X.25 Configuration Utility
•Configure X.25 Network Gateway
Start X.25 Network Gateway
Shut down X.25 Network Gateway
Reconfigure X.25 line
Status of X.25 Network Gateway
Configure proc-table
Configure X.25 Network Gateway boot sequence
287.3-1

Total number of lines: 1

Reconfig line number: 0

Status line number:

>>>Press 'LineFeed' to select; Press 'Finish' to return<<<

Move cursor or press 'LineFeed' to select; Press TAB/^TAB to move to the right

Figure 3-1. X.25 Configuration Utility Main Menu

The X.25 Configuration Utility Main Menu displays the following configuration options:

- Configure X.25 Network Gateway

- Start X.25 Network Gateway

- Shut down X.25 Network Gateway

- Reconfigure X.25 Line

- Status of X.25 Network Gateway
- Configure proc_table
- Configure X.25 Network Gateway boot sequence

CURSOR MOVEMENT INSTRUCTIONS

Press the **Up** or **Down** cursor keys to move the highlighted cursor up or down on the displayed menu. Press the **Up** or **Down** cursor keys to display the desired value in a selected edit field.

Press **Tab** to move the highlighted cursor from side to side on the displayed menu.

Menus consist of the following field types:

- Menu fields are fields displayed on the menu that you cannot modify.
- Edit fields are fields displayed on the menu where information can be entered into the system.

Instructions for positioning the cursor and moving from menu to menu are located at the bottom of each menu as it is displayed on your screen.

During Gateway configuration, press **Help** for additional information about a specific parameter field of the X.25 Configuration Utility.

For informatioon about corresponding key names on different kinds of terminals and keyboards, see Appendix D.

HOW TO SELECT A CONFIGURATION OPTION FROM THE X.25 CONFIGURATION UTILITY MAIN MENU

To select the configuration option of your choice, complete the following steps:

1. Display the X.25 Configuration Utility Main Menu.

2. Press the **Up** or **Down** cursor keys until the highlighted cursor is positioned over the configuration option of your choice.

3. Press **Enter**. (The appropriate key name is displayed at the bottom of the menu.)

NOTE

> You must specify the total number of lines or the line number if you select one of the following X.25 Configuration Utility Main Menu configuration options:
>
> - Configure X.25 Network Gateway
> - Reconfigure X.25 line
> - Status of X.25 Network Gateway
>
> Specifying the line number is optional when you select the Status of X.25 Network Gateway option from the X.25 Configuration Utility Main Menu. When you specify a line number, you will receive detailed status information about the specified line. If not, you will receive general status information about all of the X.25 lines.

To specify the line number on the X.25 Configuration Utility Main Menu, as shown in Figure 3-1, complete the following steps:

1. Press the **Up** or **Down** cursor keys until the highlighted cursor is positioned over one of the following configuration options:

   - Configure X.25 Network Gateway
   - Reconfigure X.25 line
   - Status of X.25 Network Gateway

2. Press **Tab** to move the highlighted cursor to the edit field located to the right of the selected configuration option.

3. Type in the appropriate value.

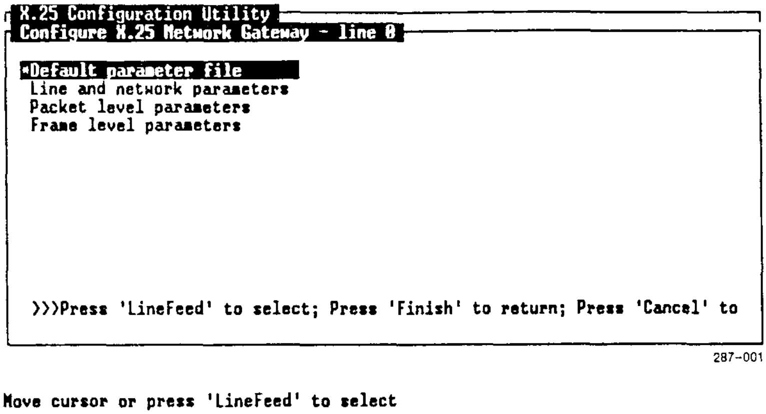
4. Press **Enter.**

CONFIGURE X.25 NETWORK GATEWAY

If you selected the Configure X.25 Network Gateway option from the X.25 Configuration Utility Main Menu, the Configure X.25 Network Gateway Menu, as shown in Figure 3-2, appears on your screen. You may choose one of the following parameter options from the Configure X.25 Network Gateway Menu:

- Default Parameter File
- Line and Network Parameters
- Packet Level Parameters
- Frame Level Parameters

If you are configuring for two lines, you must specify the number **2** on the X.25 Configuration Utility Main Menu.



```
X.25 Configuration Utility
Configure X.25 Network Gateway - line 0

*Default parameter file
 Line and network parameters
 Packet level parameters
 Frame level parameters

 >>>Press 'LineFeed' to select; Press 'Finish' to return; Press 'Cancel' to
```

287-001

```
Move cursor or press 'LineFeed' to select
```

Figure 3-2. Configure X.25 Network Gateway Menu

> **NOTE**
>
> If you press **Cancel** while you are configuring from the Configure X.25 Network Gateway Menu, none of the currently specified configuration parameters are saved.

To select a configuration option from the Configure X.25 Network Gateway Menu, as shown in Figure 3-2, complete the following steps:

1. Press the **Up** or **Down** cursor keys until the highlighted cursor is positioned over the configuration option of your choice.

2. Press **Enter**. (The appropriate key name is displayed at the bottom of the menu.)

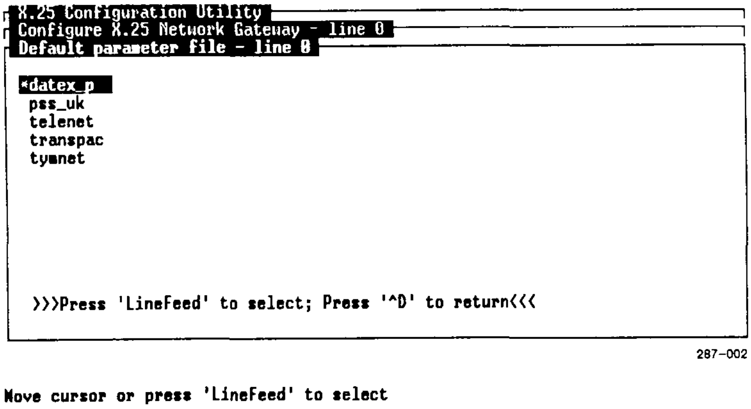
If you are configuring two lines, begin with Line 0. When Line 0 configuration is complete, the sequence automatically repeats itself, allowing you to configure Line 1.

DEFAULT PARAMETER FILE

Choosing the Default parameter file option allows you to access a list of default files. The default parameter files consist of both common and network-specific parameters, which enable your system to properly operate with the specified PDN. These parameters are used as defaults by the X.25 Configuration Utility during configuration of the Gateway. The Gateway provides default files for the following networks:

o datex_p
o pss_uk
o telenet
o transpac
o tymnet

If you select the Default parameter file option from the Configure X.25 Network Gateway Menu, a complete list of the default parameter files, located in the **/etc/x25/pdn** directory, are displayed on the Default Parameter File Menu, as shown in Figure 3-3.



```
X.25 Configuration Utility
Configure X.25 Network Gateway - line 0
Default parameter file - line 0

*datex_p
 pss_uk
 telenet
 transpac
 tymnet

 >>>Press 'LineFeed' to select; Press '^D' to return<<<

Move cursor or press 'LineFeed' to select
```

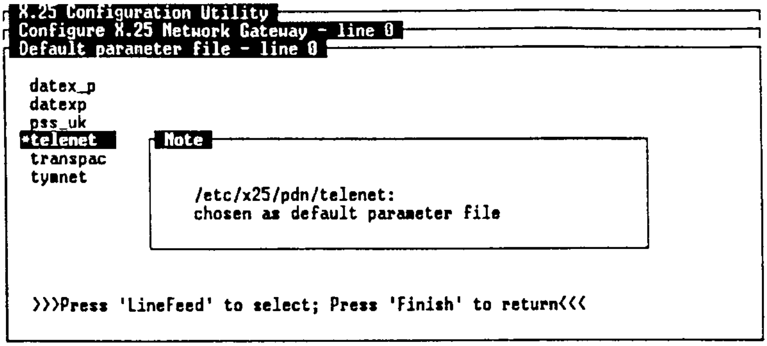
Figure 3-3. Default Parameter File Menu with datex_p Highlighted

To select a default file option from the Default Parameter File Menu, complete the following steps:

1. Press the **Up** or **Down** cursor keys until the highlighted cursor is positioned over the default parameter file of your choice.

2. Press **Enter**. (The appropriate key name is displayed at the bottom of the menu.)

Figure 3-4 shows an example of the Default Parameter File Menu after you have selected the telenet default parameter file. Press **Help** if you require additional information before choosing a default parameter file. A brief message is displayed for each default parameter file.



```
X.25 Configuration Utility
Configure X.25 Network Gateway - line 0
Default parameter file - line 0

 datex_p
 datexp
 pss_uk
*telenet      Note
 transpac
 tymnet
                   /etc/x25/pdn/telenet:
                   chosen as default parameter file


 >>>Press 'LineFeed' to select; Press 'Finish' to return<<<
```

287-003

Figure 3-4. Default Parameter File Menu with telenet Highlighted

NOTE

If you do not specify a default parameter file, a previously configured parameter file is used as the default. If you are configuring for the first time, the X.25 Network Gateway provides a set of default parameter files. It is recommended that you indicate a selection before continuing configuration of the remaining X.25 Configuration Utility parameters. If no default parameter file option is specified, you are responsible for configuring all of the X.25 parameters.

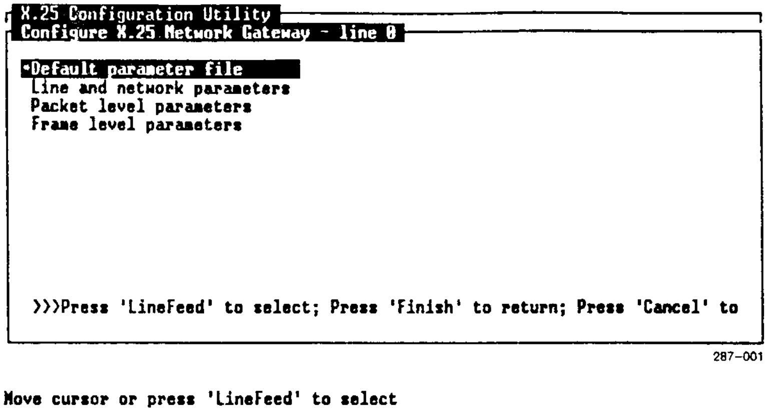
WARNING

System Administrators and the X.25 Configuration Utility are responsible for maintaining the integrity of the **/etc/x25/pdn** directory. NO OTHER FILES EXCEPT FOR THOSE WHICH COME WITH THE SYSTEM SOFTWARE OR THOSE CREATED BY THE X.25 CONFIGURATION UTILITY SHOULD RESIDE IN THE **/etc/x25/pdn** DIRECTORY.

LINE AND NETWORK PARAMETERS

Choosing the Line and network parameters option from the Configure X.25 Network Gateway Menu, as shown in Figure 3-5, allows you to configure the following line and network parameters for a specified X.25 line:

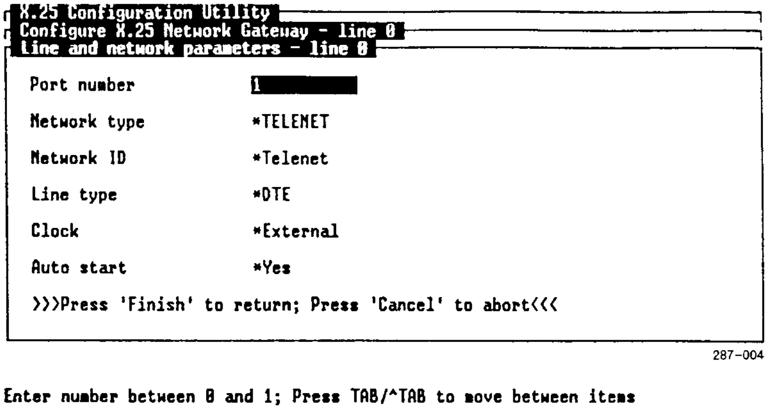
- Port number
- Network type
- Network ID
- Line type
- Clock
- Auto start



```
X.25 Configuration Utility
Configure X.25 Network Gateway - line 0

*Default parameter file
 Line and network parameters
 Packet level parameters
 Frame level parameters



 >>>Press 'LineFeed' to select; Press 'Finish' to return; Press 'Cancel' to
                                                                    287-001
Move cursor or press 'LineFeed' to select
```

Figure 3-5. Configure X.25 Network Gateway Menu

Figure 3-6 shows the Line and Network Parameters Menu.



```
X.25 Configuration Utility
Configure X.25 Network Gateway - line 0
Line and network parameters - line 0

 Port number             1

 Network type            *TELENET

 Network ID              *Telenet

 Line type               *DTE

 Clock                   *External

 Auto start              *Yes

 >>>Press 'Finish' to return; Press 'Cancel' to abort<<<
                                                                    287-004
Enter number between 0 and 1; Press TAB/^TAB to move between items
```

Figure 3-6. Line and Network Parameters Menu

For each option shown on the Line and Network Parameters Menu, shown in Figure 3-6, complete the following steps:

1. Press **Tab** until the highlighted cursor is positioned over the edit or menu field, located to the right of the configuration option, you wish to specify.

   Instructions for completing the selected parameter edit or menu field appear at the bottom of the menu.

2. Read the instructions provided at the bottom of the menu and either enter the appropriate value or press the **Up** or **Down** cursor keys until the desired selection is displayed in the edit field.

3. Repeat this process until all of the options shown on the Line and Network Parameter Menu are completed. If you do not enter values in any of the designated fields, the default values are assumed by the X.25 Configuration Utility.

4. Press **Exit**. (The appropriate key name is displayed at the bottom of the menu.)

The following sections describe each of the option fields appearing on the Line and Network Parameter Menu (shown in Figure 3-6).

PORT NUMBER

Specifying the Port number function on the Line and Network Parameters Menu allows you to assign an RS-232-C port for X.25 operation.

NOTE

Port numbering begins with 1, whereas X.25 logical line numbering begins with 0. Therefore, X.25 line 0 may be assigned to port 1.

To specify the Port number, complete the following steps:

1. Press **Tab** until the highlighted cursor is positioned over the edit field to the right of the Port number option.

2. Type either **0** or **1**.

3. Press **Tab** to position the cursor over the next menu field,

   or

   press **Exit** to return to the Configure X.25 Network Gateway Menu. (The appropriate key name is displayed at the bottom of the menu.)

NOTE

There is no default for the Port number function. You must select a port that is not running a current process. For example, do not select the same port where a login process is active. Remove the **getty** from the port in this case.

NETWORK TYPE

Specifying the Network type option from the Line and Network Parameters Menu allows you to designate the network being used. Network type allows you to select different protocol options required by various PDNs.

To specify the Network type option, complete the following steps:

1. Press **Tab** until the highlighted cursor is positioned over the menu field located to the right of the Network type option.

2. Press the **Up** or **Down** cursor keys until the desired network name appears in the menu field. (Each time you press the **Up** or **Down** cursor keys, a different network name is displayed in the field.)

3. Press **Tab** to position the cursor over the next menu field,

   or

   press **Exit** to return to the Configure X.25 Network Gateway Menu. (The appropriate key name is displayed at the bottom of the menu.)

NOTE

There is no default for the Network type function. You must specify a network type for proper Gateway operation.

NETWORK ID

The Network ID consists of a four-digit data network identification code (DNIC) that uniquely identifies each PDN in the various countries.

> **NOTE**
>
> Network ID information is used only by the application.

To select the Network ID option from the Line and Network Parameters Menu, complete the following steps:

1. Press **Tab** until the highlighted cursor is positioned over the menu field located to the right of the Network ID option.

2. Press the **Up** or **Down** cursor keys until the desired network name appears in the menu field.

3. Press **Tab** to position the cursor over the next menu field,

   or

   press **Exit** to return to the Configure X.25 Network Gateway Menu. (The appropriate keyname is displayed at the bottom of the menu.)

LINE TYPE

Specifying the Line type function enables you to select the line type (DTE or DCE) for both the X.25 packet and frame level of the specified X.25 line.

To specify the Line type option from the Line and Network Parameters Menu, complete the following steps:

1. Press **Tab** until the highlighted cursor is positioned over the menu field located to the right of the Line type option.

2. Press the **Up** or **Down** cursor keys until the desired value (**DTE** or **DCE**) appears in the menu field.

3. Press **Tab** to position the cursor over the next menu field,

   or

   press **Exit** to return to the Configure X.25 Network Gateway Menu. (The appropriate keyname is displayed at the bottom of the menu.)

If you are connecting two systems, configure one system as DTE and the other system as DCE.

The default for the Line type function is DTE (the required mode when connecting to a PDN).

CLOCK

Specifying the Clock function allows you to select the internal or external clock for the designated X.25 line.

Select the internal clock only when you are connecting two systems point-to-point without a modem or special cable. Otherwise, specify the default external clock.

> **NOTE**
>
> A special cable is required if you are installing a point-to-point connection between the DCE and DTE.

To specify Clock from the Line and Network Parameters Menu, complete the following steps:

1. Press **Tab** until the highlighted cursor is positioned over the menu field located to the right of the Clock option.

2. Press the **Up** or **Down** cursor keys until the desired value (**external** or **speed**) appears in the menu field.

3. Press **Tab** to position the cursor over the next menu field,

   or

   press **Exit** to return to the Configure X.25 Network Gateway Menu.

AUTO START

Specifying the Auto start option allows the X.25 software to attempt to bring up the line as soon as it downloads the initialization table. When the Auto start option is not selected, the application must issue a specific **ioctl** command to bring up the line.

To specify the Auto start option from the Line and Network Parameters Menu, complete the following steps:

1. Press **Tab** until the highlighted cursor is positioned over the menu field located to the right of the Auto start option.

2. Press the **Up** or **Down** cursor keys until the desired value (**yes** or **no**) appears in the menu field.

3. Press **Tab** to position the cursor over the next edit field,

   or

   press **Exit** to return to the Configure X.25 Network Gateway Menu. (The appropriate keyname is displayed at the bottom of the menu.)

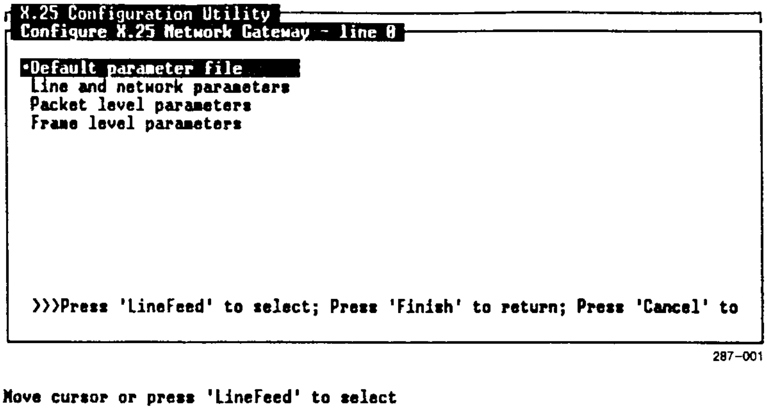
PACKET LEVEL PARAMETERS

The Packet level parameters option is located on the Configure X.25 Network Gateway Menu, as shown in Figure 3-7. This option allows you to specify the following packet level related parameters for the X.25 line:

- Modulus (8 or 128)
- Default packet size (16 to 1024)
- Maximum packet size (16 to 1024)
- D-Bit used (yes or no)
- Default window size (1 to 7 or 1 to 127)
- Maximum window size (1 to 7 or 1 to 127)
- T20 (restart request) timer (1 to 32767)

- T21 (call request) timer (1 to 32767)
- T22 (reset request) timer (1 to 32767)
- T23 (clear request) timer (1 to 32767)
- Reset request (number of retries, 1 to 255)
- Clear request (number of retries, 1 to 255)

- Configuration of VCs
- Configuration of Facilities



```
X.25 Configuration Utility
Configure X.25 Network Gateway - line 0

•Default parameter file
 Line and network parameters
 Packet level parameters
 Frame level parameters

>>>Press 'LineFeed' to select; Press 'Finish' to return; Press 'Cancel' to

Move cursor or press 'LineFeed' to select
```

287-001

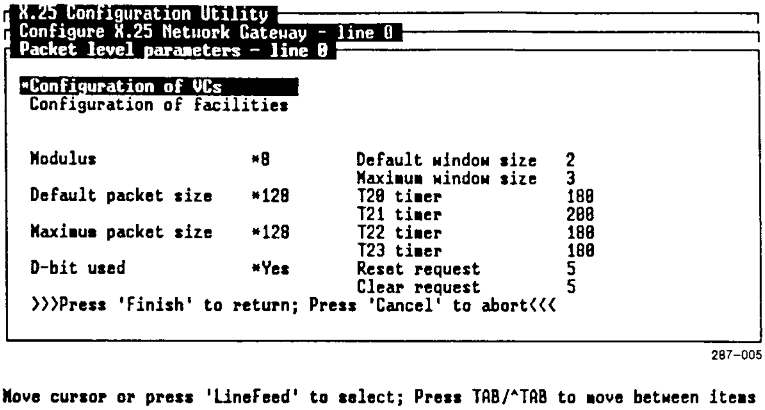
Figure 3-7. Configure X.25 Network Gateway Menu

To select the Packet level parameters options from the Configure X.25 Network Gateway Menu, shown in Figure 3-7, complete the following steps:

1. Press the **Up** or **Down** cursor keys until the highlighted cursor is positioned over the Packet level parameters option.

2. Press **Enter.** (The appropriate key name is displayed at the bottom of the menu.)

The Packet Level Parameters Menu, shown in Figure 3-8, appears on your terminal screen.



```
X.25 Configuration Utility
Configure X.25 Network Gateway - line 0
Packet level parameters - line 0

*Configuration of VCs
 Configuration of facilities


 Modulus                    *8        Default window size  2
                                      Maximum window size  3
 Default packet size        *128      T20 timer            100
                                      T21 timer            200
 Maximum packet size        *128      T22 timer            100
                                      T23 timer            100
 D-bit used                 *Yes      Reset request        5
                                      Clear request        5
 >>>Press 'Finish' to return; Press 'Cancel' to abort<<<

                                                            287-005
Move cursor or press 'LineFeed' to select; Press TAB/^TAB to move between items
```

Figure 3-8. Packet Level Parameters Menu with Menu Configuration of VCs Highlighted

To select the Configuration of VCs or Configuration of facilities options from the Packet Level Parameters Menu, complete the following steps:

1. Press the **Up** or **Down** cursor keys until the highlighted cursor is positioned over the configuration option of your choice.

2. Press **Enter**. (The appropriate key name is displayed at the bottom of the menu.)

To specify any of the other options shown on the Packet Level Parameters Menu, complete the following steps:

1. Press **Tab** until the highlighted cursor is positioned over either the menu or the edit field of your choice.

2. Press the **Up** or **Down** cursor keys until the desired value appears in the menu field,

   or

   type the appropriate value in the edit field.

4. Press **Tab** to position the cursor over the next menu or edit field of your choice,

   or

   press **Exit** to return to the Configure X.25 Network Gateway Menu. (The appropriate key name is displayed at the bottom of the menu.)

MODULUS

Modulus sequence numbering is used to number X.25 data packets. It begins with 0 and increments by 1 each time a data packet is transmitted or received. It proceeds up to modulus minus 1 and then cycles back to 0 to begin again. Sending and receiving sides are handled separately.

Specifying the Modulus option allows you to select an X.25 packet level sequence numbering modulus of 8 (for normal sequence numbering) or 128 (for extended sequence numbering).

To specify the Modulus option on the Packet Level Parameters Menu, complete the following steps:

1. Press **Tab** until the highlighted cursor is positioned over the menu field located to the right of the Modulus option.

2. Press the **Up** or **Down** cursor keys until the desired value appears in the menu field.

3. Press **Tab** to position the cursor over the next menu field,

   or

   press **Exit** to return to the Configure X.25 Network Gateway Menu. (The appropriate key name is displayed at the bottom of the menu.)

The default for the Modulus option is 8.

DEFAULT PACKET SIZE

Packet size is the actual size of user data in a data packet. It does not include X.25 packet and frame level headers.

Default packet size is the maximum-size data packet that is transferred, unless the packet size is negotiated during call establishment. In this case, the negotiated packet size is in effect for the entire period of time that the virtual circuit is active.

Specifying the Default packet size option allows you to select the default packet size that will be transmitted or received. You must select one of the following packet sizes (in bytes):

16
32
64
128
256
512
1024

To specify the Default packet size option on the Packet Level Parameters Menu, complete the following steps:

1. Press **Tab** until the highlighted cursor is positioned over the menu field located to the right of the Default packet size option.

2. Press the **Up** or **Down** cursor keys until the desired value appears in the menu field.

3. Press **Tab** to position the cursor over the next menu field,

   or

   press **Exit** to return to the Configure X.25 Network Gateway Menu. (The appropriate key name is displayed at the bottom of the menu.)

The default for the Default packet size option is 128 bytes.

MAXIMUM PACKET SIZE

The Maximum packet size function represents the maximum packet size that can be negotiated during call establishment. Without negotiation, the default packet size becomes the maximum.

Specifying the Maximum packet size option allows you to select the maximum packet size that can be negotiated during call establishment. You must select one of the following numbers (in bytes):

16
32
64
128
256
512
1024

The specified maximum packet size must be greater than or equal to the default packet size.

To specify the Maximum packet size option on the Packet Level Parameters Menu, complete the following steps:

1. Press **Tab** until the highlighted cursor is positioned over the menu field located to the right of the Maximum packet size option.

2. Press the **Up** or **Down** cursor keys until the desired value appears in the menu field.

3. Press **Tab** to position the cursor over the next menu field,

   or

   press **Exit** to return to the Configure X.25 Network Gateway Menu. (The appropriate key name is displayed at the bottom of the menu.)

The default for the Maximum packet size option is equal to that specified for the Default packet size option.

D-BIT USED

The D-bit is used to request end-to-end confirmation of data packet delivery. In other words, any acknowledgment of data packets is actually coming from the remote system rather than from the DCE to which the local system is connected.

Specifying the D-bit used option allows you to decide whether or not the use of end-to-end confirmation of data delivery is allowed.

To specify the D-bit used option on the Packet Level Parameters Menu, complete the following steps:

1. Press **Tab** until the highlighted cursor is positioned over the menu field located to the right of the D-bit used option.

2. Press the **Up** or **Down** cursor keys until the desired value appears in the menu field.

3. Press **Tab** to position the cursor over the next edit field,

   or

   press **Exit** to return to the Configure X.25 Network Gateway Menu. (The appropriate key name is displayed at the bottom of the menu.)

The default for the D-bit used option is no (do not use the D-bit).

NOTE

It is the responsibility of the user application to implement the D-bit protocol.

DEFAULT WINDOW SIZE

Each X.25 data packet has a sending sequence number. Explicit acknowledgment for each packet is not required. An acknowledgment is not restricted to a single data packet: it can acknowledge an entire data packet or a range of data packets (not to exceed the maximum window size) that have been received since the last acknowledgment was sent. Based on the modulus (8, as an example), one system can send as many as 7 (0 to 6) data packets without receiving any acknowledgment. If you write more packets than the specified window size, the **write** system call blocks the request until the window is opened.

Window size is used to specify the number of data packets which can have a pending acknowledgment.

Default window size is the maximum number of data packets that can be outstanding unless the window size is negotiated during call establishment. In this case, the negotiated window size is in effect for the entire period of time that the virtual circuit is active.

Specifying the Default window size option allows you to select the default window size which will be used either to send data packets or to flow control data packets as they are received. The number you enter must be between 1 and 7 or, when extended packet sequence numbering is specified, between 1 and 127.

To specify the Default window size option on the Packet Level Parameters Menu, complete the following steps:

1. Press **Tab** until the highlighted cursor is positioned over the edit field located to the right of the Default window size option.

2. Type a number between 1 and 7 or between 1 and 127.

3. Press **Tab** to position the cursor over the next edit field,

   or

   press **Exit** to return to the Configure X.25 Network Gateway Menu. (The appropriate key name is displayed at the bottom of the menu.)

The default for the Default window size option is 2.

MAXIMUM WINDOW SIZE

The Maximum window size option represents the maximum window size that can be negotiated during call establishment. Without negotiation, the default window size becomes the maximum.

Specifying the Maximum window size option allows you to select the maximum window size that can be negotiated during call establishment. You must specify a number between 1 and 7 or, when extended sequence number is used, between 1 and 127.

To specify the Maximum window size option on the Packet Level Parameters Menu, complete the following steps:

1. Press **Tab** until the highlighted cursor is positioned over the edit field located to the right of the Maximum window size option.

2. Type a number between 1 and 7 or between 1 and 127.

3. Press **Tab** to position the cursor over the next edit field,

   or

   press **Exit** to return to the Configure X.25 Network Gateway Menu. (The appropriate key name is displayed at the bottom of the menu.)

The default for the Maximum window size option is equal to that of the Default window size option.

T20 (RESTART REQUEST) TIMER

T20 is the timeout period for Restart Request packets. If the Restart Confirmation packet is not received within the specified period of time, a Restart Request packet is retransmitted.

Specifying the T20 timer function allows you to choose an appropriate timeout period for Restart Request packet response.

To specify the T20 timer option on the Packet Level Parameters Menu, complete the following steps:

1. Press **Tab** until the highlighted cursor is positioned over the edit field located to the right of the T20 timer option.

2. Type the appropriate number of seconds.

3. Press **Tab** to position the cursor over the next edit field,

   or

   press **Exit** to return to the Configure X.25 Network Gateway Menu. (The appropriate key name is displayed at the bottom of the menu.)

The default for the T20 timer option is 180 seconds.

T21 (CALL REQUEST) TIMER

T21 is the timeout period for a Call Request packet. If the Call Confirmation packet is not received within the specified time, a Clear Request packet is sent to clear the call.

Specifying the T21 timer option allows you to designate an appropriate timeout period for a Call Request packet response. To specify the T21 timer option on the Packet Level Parameters Menu, complete the following steps:

1. Press **Tab** until the highlighted cursor is positioned over the edit field located to the right of the T21 timer option.

2. Type the appropriate number of seconds.

3. Press **Tab** to position the cursor over the next edit field,

   or

   press **Exit** to return to the Configure X.25 Network Gateway Menu. (The appropriate key name is displayed at the bottom of the menu.)

The default for the T21 timer option is 200 seconds.

T22 (RESET REQUEST) TIMER

T22 is the timeout period for a Reset Request packet. If the Reset Confirmation packet is not received within the specified period of time, a Reset Request packet is retransmitted.

Specifying the T22 timer option allows you to choose an appropriate timeout period for a Reset Request packet response.

To specify the T22 timer option on the Packet Level Parameters Menu, complete the following steps:

1. Press **Tab** until the highlighted cursor is positioned over the edit field located to the right of the T22 timer option.

2. Type the appropriate number of seconds.

3. Press **Tab** to position the cursor over the next edit field,

   or

press **Exit** to return to the Configure X.25 Network Gateway Menu. (The appropriate key name is displayed at the bottom on the menu.)

The default for the T22 timer option is 180 seconds.

T23 (CLEAR REQUEST) TIMER

T23 is the timeout period for a Clear Request packet. It is started when the Clear Request packet is sent out. It is cancelled when a Clear Confirmation packet is received. If the Clear Confirmation packet is not received within the specified period of time, a Clear Request packet is retransmitted.

Specifying the T23 timer option allows you to choose an appropriate timeout period for a Clear Request packet response.

To specify the T23 timer option on the Packet Level Parameters Menu, complete the following steps:

1. Press **Tab** until the highlighted cursor is positioned over the edit field located to the right of the T23 timer option.

2. Type the appropriate number of seconds.

3. Press **Tab** to position the cursor over the next edit field,

   or

press **Exit** to return to the Configure X.25 Network Gateway Menu. (The appropriate key name is displayed at the bottom of the menu.)

The default for the T23 timer option is 180 seconds.

RESET REQUEST (NUMBER OF RETRIES)

The Reset Request counter specifies the maximum number of times that a Reset Request packet is retransmitted on a switched virtual circuit. When the retransmission count has reached the specified limit, this particular virtual circuit is cleared.

Specifying the Reset request option allows you to choose an appropriate counter for the retransmission of Reset Request packets.

To specify the Reset request option on the Packet Level Parameters Menu, complete the following steps:

1. Press **Tab** until the highlighted cursor is positioned over the edit field located to the right of the Reset request option.

2. Type the appropriate number of retries.

3. Press **Tab** to position the cursor over the next edit field,

   or

   press **Exit** to return to the Configure X.25 Network Gateway Menu. (The appropriate key name is displayed at the bottom of the menu.)

The default for the Reset request option is 5 times.

CLEAR REQUEST (NUMBER OF RETRIES)

The Clear Request counter specifies the maximum number of times that a Clear Request packet is retransmitted on a switched virtual circuit. When the retransmission count has reached the specified limit, this particular virtual circuit is cleared.

Specifying the Clear request option allows you to choose an appropriate counter for the retransmission of Clear Request packets.

To specify the Clear request option on the Packet Level Parameters Menu, complete the following steps:

1. Press **Tab** until the highlighted cursor is positioned over the edit field located to the right of the Clear request option.

2. Type the appropriate number of retries.

3. Press **Tab** to position the cursor over the next menu field,

   or

   press **Exit** to return to the Configure X.25 Network Gateway Menu. (The appropriate key name is displayed at the bottom of the menu.)

The default for the Clear request option is 5 times.

CONFIGURATION OF VC'S

> **NOTE**
>
> In the following sections, DTE refers to your system and DCE refers to the network to which your system is connected.

Logical channels are a means to support multiple connections on the same physical medium. All virtual circuits are either permanent virtual circuits (PVCs) or switched virtual circuits (SVCs).

A permanent virtual circuit is always ready to send or receive X.25 data packets. A switched virtual circuit requires a call setup. Data transfer begins only after the call is connected. In the case of a PVC, the application must select a logical channel to use (PVCs are located on specific LCNs). In the case of a switched virtual circuit, a free channel is assigned from a preconfigured range by the software during the call setup. Switched virtual circuits can also be restricted to incoming only or outgoing only. Most switched circuits are two-way virtual circuits.

A virtual circuit uses a logical channel, which is characterized by the *logical channel group number* (LCGN) and the logical channel number (LCN). The LCGN is a number from 0-15. The LCN is a number from 0 to 255. Allocation of LCNs must be agreed upon with the network (DCE) to which you are connected.

The range for each type of virtual circuit can not overlap. The convention for logical channel allocation in numeric order is as follows:

- Permanent virtual circuits
- Incoming-only virtual circuits
- Two-way virtual circuits
- Outgoing-only virtual circuits

LCGN is a network related parameter. For example, a specific network may require usage of a specific range of LCGNs for each LC type.

Incoming-only circuits can be used only for incoming calls. In this case, outgoing calls are not allowed.

Outgoing-only circuits can be used only for outgoing calls. In this case, incoming calls are not allowed.

Two-way circuits can be used to receive incoming calls and place outgoing calls. Two-way circuits are the most commonly used VC type.

Designated rules must be followed when assigning the LCN. LCN for incoming-only virtual circuits is assigned by the DCE. LCN for outgoing-only virtual circuits is assigned by DTE. There are two special rules that govern the assignment of LCN for a two-way virtual circuit. They are as follows:

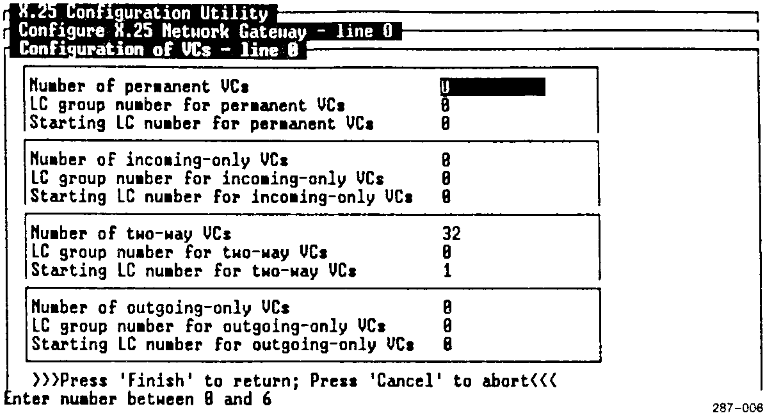
- incoming calls starting with the lowest free channel number (assigned by DCE) within the range
- outgoing calls starting with the highest free channel number (assigned by DTE) within the range

Rules that govern the LCN assignment for a two-way virtual circuit avoid line contention.

Specifying the Configuration of VCs option allows you to configure the following virtual circuit parameters:

- number of permanent VCs
- LC group number for permanent VCs (0 to 15)
- starting LC group number for permanent VCs (0 to 255)
- number of incoming-only VCs
- LC group number for incoming-only VCs (0 to 15)
- starting LC group number for incoming-only VCs (0 to 255)
- number of two-way VCs
- LC group number for two-way VCs (0 to 15)
- starting LC group number for two-way VCs (0 to 255)
- number of outgoing-only VCs
- LC group number for outgoing-only VCs (0 to 15)
- starting LC number for outgoing-only VCs (0 to 255)

After selecting the Configuration of VCs option on the Packet Level Parameters Menu, as shown in Figure 3-8, the Configuration of VCs Menu (shown in Figure 3-9) is displayed on your terminal screen.



```
X.25 Configuration Utility
Configure X.25 Network Gateway - line 0
Configuration of VCs - line 0

Number of permanent VCs                      0
LC group number for permanent VCs            0
Starting LC number for permanent VCs         0

Number of incoming-only VCs                  0
LC group number for incoming-only VCs        0
Starting LC number for incoming-only VCs     0

Number of two-way VCs                        32
LC group number for two-way VCs              0
Starting LC number for two-way VCs           1

Number of outgoing-only VCs                  0
LC group number for outgoing-only VCs        0
Starting LC number for outgoing-only VCs     0

 >>>Press 'Finish' to return; Press 'Cancel' to abort<<<
Enter number between 0 and 6
```

287-006

Figure 3-9. Configuration of VCs Menu

To specify any of the options displayed on the Configuration of VCs Menu, complete the following steps:

1. Press **Tab** until the highlighted cursor is positioned over the edit field located to the right of the appropriate option field.

2. Type the appropriate number.

3. Press **Tab** to position the cursor over the next edit field,

   or

press **Exit** to return to the Configure X.25 Network Gateway Menu. (The appropriate key name is displayed at the bottom of the menu.)

NOTE

The total number of logical channels configured cannot exceed the maximum aggregate number of VCs supported for the Gateway (see the Release Notice for the CTIX X.25 Network Gateway for additional information regarding maximum aggregate VCs). However, the actual number of allocated virtual circuits must not exceed the number of virtual circuits to which you have subscribed (plus 1 if LCN 0 is reserved).

NUMBER OF PERMANENT VC'S

Permanent virtual circuits are virtual circuits that exist continuously. Call establishment is not necessary, and data transfer can occur at any time.

Specifying the Number of permanent VCs option allows you to allocate the desired number of PVCs. Refer to the bottom of the menu for the maximum number of PVCs that can be configured.

To specify the Number of permanent VCs option on the Configuration of VCs Menu, complete the following steps:

1. Press **Tab** until the highlighted cursor is positioned over the edit field located to the right of the Number of permanent VCs option.

2. Type the appropriate number in the edit field.

3. Press **Tab** to position the cursor over the next edit field,

   or

   press **Exit** to return to the Configure X.25 Network Gateway Menu. (The appropriate key name is displayed at the bottom of the menu.)

The default for the Number of permanent VCs option is 0 to indicate no allocation of PVCs.

> **NOTE**
>
> The total number of configured logical channels, can not exceed the maximum aggregate number of VCs supported for the Gateway including the RESTART channels. (See the Release Notice for the CTIX X.25 Network Gateway for additional information regarding maximum aggregate VCs).

LC GROUP NUMBER FOR PERMANENT VC'S

Specifying the LC group number for permanent VCs function allows you to select the logical channel group for permanent virtual circuits. You must specify a number from 1 to 15.

To specify the LC group number for permanent VCs option on the Configuration of VCs Menu, complete the following steps:

1. Press **Tab** until the highlighted cursor is positioned over the edit field located to the right of the LC group number for permanent VCs option.

2. Type the appropriate number in the edit field.

3. Press **Tab** to position the cursor over the next edit field,

   or

   press **Exit** to return to the Configure X.25 Network Gateway Menu. (The appropriate key name is displayed at the bottom of the menu.)

The default for the LC group number for permanent VCs option is different for each network.

STARTING LC GROUP NUMBER FOR PERMANENT VC'S

Specifying the Starting LC number for permanent VCs function allows you to select the starting logical channel number for permanent virtual circuits. You must specify a number between 0 and 255.

To specify the Starting LC number for permanent VCs option on the Configuration of VCs Menu, complete the following steps:

1. Press **Tab** until the highlighted cursor is positioned over the edit field located to the right of the Starting LC number for permanent VCs option.

2. Type the appropriate number in the edit field.

3. Press **Tab** to position the cursor over the next edit field,

   or

   press **Exit** to return to the Configure X.25 Network Gateway Menu. (The appropriate key name is displayed at the bottom of the menu.)

The default for the Starting LC number for permanent VCs is 0.

NUMBER OF INCOMING-ONLY VC'S

Incoming-only VCs are switched virtual circuits that originate from a remote DTE. Specifying the Number of incoming-only VCs function allows you to allocate a number of virtual circuits for incoming-only calls. You must specify a number between 0 and a number not to exceed the maximum aggregate number of VCs.

To specify the Number of incoming-only VCs option on the Configuration of VCs Menu, complete the following steps:

1. Press **Tab** until the highlighted cursor is positioned over the edit field located to the right of the Number of incoming-only VCs option.

2. Type the appropriate number in the edit field.

3. Press **Tab** to position the cursor over the next edit field,

   or

press **Exit** to return to the Configure X.25 Network Gateway Menu. (The appropriate key name is displayed at the bottom of the menu.)

The default for the Number of incoming-only VCs option is 0 to indicate no allocation of incoming-only virtual circuits.

LC GROUP NUMBER FOR INCOMING-ONLY VC'S

Specifying the LC group number for incoming-only VCs function allows you to select the logical channel group for incoming-only virtual circuits. You must specify a number between 0 and 15.

To specify the LC group number for incoming-only VCs option on the Configuration of VCs Menu, complete the following steps:

1. Press **Tab** until the highlighted cursor is positioned over the edit field located to the right of the LC group number for incoming-only VCs option.

2. Type the appropriate number in the edit field.

3. Press **Tab** to position the cursor over the next edit field,

   or

   press **Exit** to return to the Configure X.25 Network Gateway Menu. (The appropriate key name is displayed at the bottom of the menu.)

The default for the LC group number for incoming-only VCs is 0.

NUMBER OF TWO-WAY VC'S

Two-way virtual circuits are switched virtual circuits originating from either the local DTE or the remote DCE.

Specifying the Number of two-way VCs function allows you to allocate virtual circuits for two-way calls. You must specify a number between 1 and a number not to exceed the maximum aggregate number of VCs allowed.

To specify the Number of two-way VCs option on the Configuration of VCs Menu, complete the following steps:

1. Press **Tab** until the highlighted cursor is positioned over the edit field located to the right of the Number of two-way VCs option.

2. Type the appropriate number in the edit field.

3. Press **Tab** to position the cursor over the next edit field,

   or

   press **Exit** to return to the Configure X.25 Network Gateway Menu. (The appropriate key name is displayed at the bottom of the menu.)

There is no default for the Number of two-way VCs option.

LC GROUP NUMBER FOR TWO-WAY VC'S

Specifying the LC group number for two-way VCs function allows you to select the logical channel group for two-way virtual circuits. You must specify a number between 0 and 15.

To specify the LC group number for two-way VCs option on the Configuration of VCs Menu, complete the following steps:

1. Press **Tab** until the highlighted cursor is positioned over the edit field located to the right of the LC group number for two-way VCs option.

2. Type the appropriate number in the edit field.

3. Press **Tab** to position the cursor over the next edit field,

   or

   press **Exit** to return to the Configure X.25 Network Gateway Menu. (The appropriate key name is displayed at the bottom of the menu.)

The default for the LC group number for two-way VCs option is different for each network.

STARTING LC NUMBER FOR TWO-WAY VC'S

Specifying the Starting LC number for two-way VCs function allows you to select the starting logical channel number for two-way virtual circuits. You must specify a number between 0 and 255.

To specify the Starting LC number for two-way VCs option on the Configuration of VCs Menu, complete the following steps:

1. Press **Tab** until the highlighted cursor is positioned over the edit field located to the right of the Starting LC number for two-way VCs option.

2. Type the appropriate number in the edit field.

3. Press **Tab** to position the cursor over the next edit field,

   or

   press **Exit** to return to the Configure X.25 Network Gateway Menu. (The appropriate key name is displayed at the bottom of the menu.)

The default for the Starting LC number for two-way VCs is different for each network.

NUMBER OF OUTGOING-ONLY VC'S

Outgoing-only VCs are switched virtual circuits that originate from the local DTE. Specifying the Number of outgoing-only VCs function allows you to allocate virtual circuits for outgoing-only calls. You must specify a number between 0 and a number not to exceed the maximum aggregate number of supported VCs.

To specify the Number of outgoing-only VCs option on the Configuration of VCs Menu, complete the following steps:

1. Press **Tab** until the highlighted cursor is positioned over the edit field located to the right of the Number of outgoing-only VCs option.

2. Type the appropriate number in the edit field.

3. Press **Tab** to position the cursor over the next edit field,

   or

   press **Exit** to return to the Configure X.25 Network Gateway Menu. (The appropriate key name is displayed at the bottom of the menu.)

The default for the Number of outgoing-only VCs function is different for each network.

> NOTE
>
> The total number of permanent, incoming, two-way, and outgoing virtual circuits can not exceed the maximum aggregate number of VCs supported by the Gateway, including the RESTART channels. (See the Release Notice for the CTIX X.25 Network Gateway for additional information about maximum aggregate VCs).

LC GROUP NUMBER FOR OUTGOING-ONLY VC'S

Specifying the LC group number for outgoing-only VCs function allows you to select the logical channel group for outgoing-only virtual circuits. You must specify a number between 0 and 15.

To specify the LC group number for outgoing-only VCs option on the Configuration of VCs Menu, complete the following steps:

1. Press **Tab** until the highlighted cursor is positioned over the edit field located to the right of the LC group number for outgoing-only VCs option.

2. Type the appropriate number in the edit field.

3. Press **Tab** to position the cursor over the next edit field,

   or

   press **Exit** to return to the Configure X.25 Network Gateway Menu. (The appropriate key name is displayed at the bottom of the menu.)

The default for the LC group number for outgoing-only VCs option is different for each network.

STARTING LC NUMBER FOR OUTGOING-ONLY VC'S

Specifying the Starting LC number for outgoing-only VCs function allows you to select the starting logical channel for outgoing-only virtual circuits. You must enter a number between 0 and 255.

To specify the Starting LC number for outgoing-only VCs option on the Configuration of VCs Menu, complete the following steps:

1. Press **Tab** until the highlighted cursor is positioned over the edit field located to the right of the Starting LC number for outgoing-only VCs option.

2. Type the appropriate number in the edit field.

3. Press **Tab** to position the cursor over the next edit field,

   or

   press **Exit** to return to the Configure X.25 Network Gateway Menu. (The appropriate key name is displayed at the bottom of the menu.)

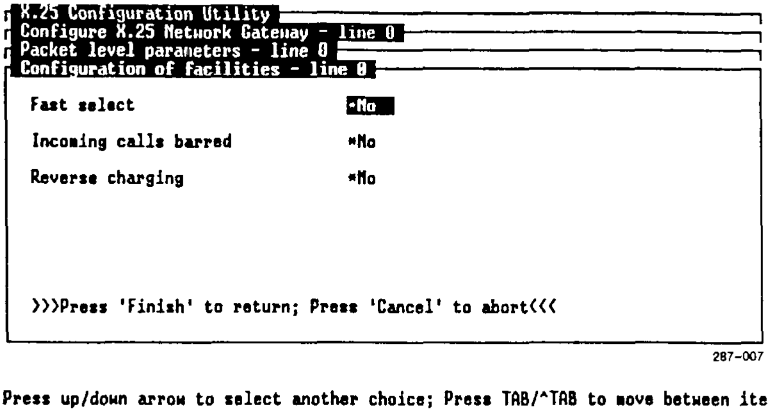
The default for the Starting LC number for outgoing-only VCs option is the channel number of the last two-way virtual circuit plus 1.

CONFIGURATION OF FACILITIES

Selecting the Configuration of facilities option on the Packet Level Parameters Menu allows you to configure the following functions:

- Fast select
- Incoming calls barred
- Reverse charging

After selecting the Configuration of facilities option from the Packet Level Parameters Menu, the Configuration of Facilities Menu (shown in Figure 3-10) is displayed on your screen.



```
X.25 Configuration Utility
Configure X.25 Network Gateway - line 0
Packet level parameters - line 0
Configuration of facilities - line 0

 Fast select                         *No

 Incoming calls barred               *No

 Reverse charging                    *No




 >>>Press 'Finish' to return; Press 'Cancel' to abort<<<
                                                                 287-007
Press up/down arrow to select another choice; Press TAB/^TAB to move between ite
```

Figure 3-10. Configuration of Facilities Menu

FAST SELECT

Specifying the Fast select option allows you to embed user data in the user data field of the Call Request, Call Accept, or Clear Request packets. It is a convenient way to use PDN services when user data is less than 128 bytes in length.

If you do not choose to subscribe to the Fast select option, incoming calls requesting the fast select facility are not presented to the local system.

To specify the Fast select option on the Configuration of Facilities Menu, complete the following steps:

1. Press **Tab** until the highlighted cursor is positioned over the menu field located to the right of the Fast select option.

2. Press the **Up** or **Down** cursor keys until the desired value (**yes** or **no**) appears in the menu field.

3. Press **Tab** to position the cursor over the next menu field,

   or

   press **Exit** to return to the Packet Level Parameters Menu. (The appropriate key name is displayed at the bottom of the menu.)

The default for the Fast select option is **no** (do not subscribe to the Fast select option).

INCOMING CALLS BARRED

Barring incoming calls is a method of restricting the usage of the X.25 line. For example, you may specify the Incoming calls barred option as an extra security measure against unwanted system access. Specifying the Incoming calls barred function allows you to decide whether to bar incoming calls on the X.25 line.

To specify the Incoming calls barred option on the Configuration of Facilities Menu, complete the following steps:

1. Press **Tab** until the highlighted cursor is positioned over the menu field located to the right of the Incoming calls barred option.

2. Press the **Up** or **Down** cursor keys until the desired value (**yes** or **no**) appears in the menu field.

3. Press **Tab** to position the cursor over the next menu field,

   or

   press **Exit** to return to the Packet Level Parameters Menu. (The appropriate key name is displayed at the bottom of the menu.)

The default for the Incoming calls barred option is **no** (do not bar incoming calls).

REVERSE CHARGING

Specifying the Reverse charging option determines whether a collect call will be accepted. If the Reverse charging option is specified, incoming call requests reverse charging to the application. When the Reverse charging option is not specified, incoming calls requesting reverse charging are cleared and not presented to the application.

To specify the Reverse charging option on the Configuration of Facilities Menu, complete the following steps:

1. Press **Tab** until the highlighted cursor is positioned over the menu field located to the right of the Reverse charging option.

2. Press the **Up** or **Down** cursor keys until the desired value (**yes** or **no**) appears in the menu field.

3. Press **Tab** to position the cursor over the next menu field,

   or

press **Exit** to return to the Packet Level Parameters Menu. (The appropriate key name is displayed at the bottom of the menu.)

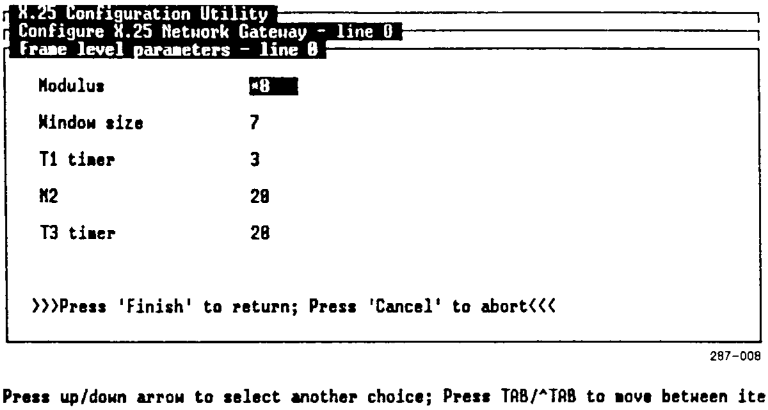
The default for the Reverse charging option is **no** (collect calls are not accepted).

FRAME LEVEL PARAMETERS

Specifying the Frame level parameters option on the Configure X.25 Network Gateway Menu, allows you to configure the following frame level-related parameters for the X.25 line:

- o Modulus (8 or 128)
- o Window Size (1-7 or 1-127)
- o T1 (retransmission) Timer
- o N2 (maximum number of retries)
- o T3 (SABM) Timer

After selecting the Frame level parameters option from the Configure X.25 Network Gateway Menu, the Frame Level Parameters Menu (shown in Figure 3-11) appears on your terminal screen.



```
X.25 Configuration Utility
Configure X.25 Network Gateway - line 0
Frame level parameters - line 0

  Modulus              *8

  Window size          7

  T1 timer             3

  N2                   20

  T3 timer             20


  >>>Press 'Finish' to return; Press 'Cancel' to abort<<<

Press up/down arrow to select another choice; Press TAB/^TAB to move between ite
```

287-008

Figure 3-11. Frame Level Parameters Menu

To specify the options located on the Frame Level Parameters Menu, complete the following steps:

1. Press **Tab** until the highlighted cursor is positioned over the menu field or the edit field located to the right of the selected menu option.

2. Press the **Up** or **Down** cursor keys until the desired value appears in the menu field,

   or

   type the appropriate value in the edit field.

3. Press **Tab** to position the cursor over the next menu or edit field,

   or

   press **Exit** to return to the Configure X.25 Network Gateway Menu. (The appropriate key name is displayed at the bottom of the menu.)

MODULUS

Modulus is used to number the X.25 frames. It begins with 0 and increments by 1 every time a frame is transmitted or received. It goes up to the modulus minus 1, cycles back to 0, and begins again. Transmitting and receiving sides are handled separately.

Specifying the <u>Modulus</u> function allows you to select X.25 frame level sequence numbering modulus of 8 (for normal sequence numbering) or 128 (for extended sequence numbering).

To specify the Modulus option on the Frame Level Parameters Menu, complete the following steps:

1. Press **Tab** until the highlighted cursor is positioned over the menu field located to the right of the Modulus option.

2. Press the **Up** or **Down** cursor keys until the desired value (**8** or **128**) appears in the menu field.

3. Press **Tab** to position the cursor over the next edit field,

   or

   press **Exit** to return to the Configure X.25 Network Gateway Menu. (The appropriate key name is displayed at the bottom of the menu.)

The default for the Modulus option is 8.

WINDOW SIZE

The window size is the maximum number of outstanding frames without receipt of an acknowledgment.

Frame level window size has the same meaning and effect as the packet level window size parameter.

Specifying the Window size option allows you to select the window size that is used either to send frames or to flow control frames upon receipt. You must specify a number between 1 and 7 or, if extended sequence numbering is specified, between 1 and 127.

To specify the Window size option on the Frame Level Parameters Menu, complete the following steps:

1. Press **Tab** until the highlighted cursor is positioned over the edit field located to the right of the Window size option.

2. Type the appropriate number in the edit field.

3. Press **Tab** to position the cursor over the next edit field,

   or

   press **Exit** to return to the Configure X.25 Network Gateway Menu. (The appropriate key name is displayed at the bottom of the menu.)

The default for the Window size option is 7.

T1 (RETRANSMISSION) TIMER

T1 is the timeout period for frame retransmission. If an acknowledgment is not received within the specified period of time, either the supervisory frames or all unacknowledged data frames are retransmitted.

Specifying the T1 timer function allows you to select an appropriate timeout period for frame retransmission.

To specify the T1 timer option on the Frame Level Parameters Menu, complete the following steps:

1. Press **Tab** until the highlighted cursor is positioned over the edit field located to the right of the T1 timer option.

2. Type the appropriate number of seconds.

3. Press **Tab** to position the cursor over the next edit field,

   or

   press **Exit** to return to the Configure X.25 Network Gateway Menu. (The appropriate key name is displayed at the bottom of the menu.)

The default for the T1 timer option is different for each network.

N2 (MAXIMUM NUMBER OF RETRIES)

N2 is the maximum number of frame retransmissions allowed. Recovery action must be taken when the value specified in N2 is reached.

Specifying the N2 option allows you to choose an appropriate retransmission count for frame retransmission.

To specify the N2 option on the Frame Level Parameters Menu, complete the following steps:

1. Press **Tab** until the highlighted cursor is positioned over the edit field located to the right of the N2 option.

2. Type the appropriate number of retries.

3. Press **Tab** to position the cursor over the next edit field,

   or

   press **Exit** to return to the Configure X.25 Network Gateway Menu. (The appropriate key name is displayed at the bottom of the menu.)

The default for the N2 option is 20 times.

T3 (SABM) TIMER

T3 is the frame retransmission timeout period to use after SABM has been transmitted N2 times. This procedure ensures constant polling.

Specifying the T3 timer option allows you to select an appropriate timeout period for SABM after the SABM retransmission count has reached the value specified in N2.

To specify the T3 timer option on the Frame Level Parameters Menu, complete the following steps:

1. Press **Tab** until the highlighted cursor is positioned over the edit field located to the right of the T3 timer option.

2. Type the appropriate number of seconds.

3. Press **Tab** to position the cursor over the next menu field,

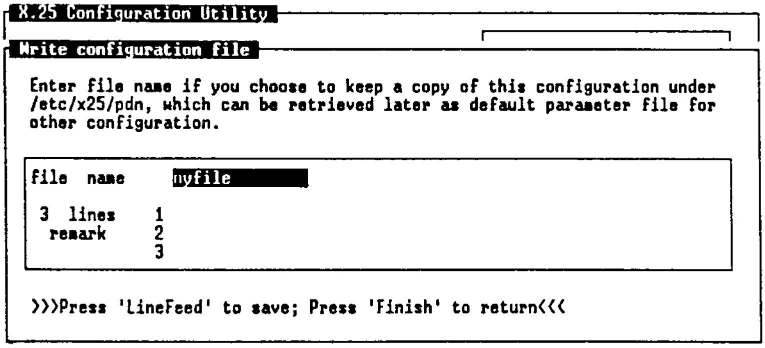
   or

press **Exit** to return to the Configure X.25 Network Gateway Menu. (The appropriate key name is displayed at the bottom of the menu.)

The default for the T3 timer option is 20 seconds.

WRITE CONFIGURATION FILE

After properly configuring the Configure X.25 Network Gateway option parameters on the X.25 Configuration Utility Main Menu, the Write Configuration File Menu (shown in Figure 3-12) appears on your screen.



X.25 Configuration Utility

Write configuration file

Enter file name if you choose to keep a copy of this configuration under /etc/x25/pdn, which can be retrieved later as default parameter file for other configuration.

```
file  name      myfile

 3  lines     1
  remark      2
              3
```

>>>Press 'LineFeed' to save; Press 'Finish' to return<<<

287-009

Figure 3-12. Write Configuration File Menu

To specify the file name and 3 lines remark options, complete the following steps:

1. Press **Tab** until the highlighted cursor is positioned over the file name option edit field.

2. Type the appropriate file name.

3. If you choose to add remarks, press **Tab** until the highlighted cursor is positioned over the 3 lines remark field.

4. Type up to three lines of information.

5. Press **Enter** to save the new configuration. (The appropriate key name is displayed at the bottom of the menu.)

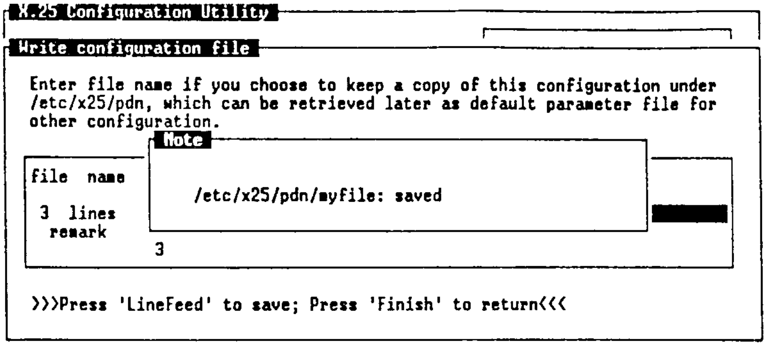
   or

   Press **Exit** to return to the X.25 Configuration Main Menu.

If you do not specify a file name, the configuration parameters that you specify are automatically saved by the system.

If you press **Cancel** or **Exit**, the configuration parameters that you specified are lost.

Figure 3-13 shows a sample Write Configuration File Menu.



```
Enter file name if you choose to keep a copy of this configuration under
/etc/x25/pdn, which can be retrieved later as default parameter file for
other configuration.

file  name
              /etc/x25/pdn/myfile: saved
 3  lines
  remark
              3

>>>Press 'LineFeed' to save; Press 'Finish' to return<<<
```

287-010

Figure 3-13. Write Configuration File Sample Menu

START X.25 NETWORK GATEWAY

The Start X.25 Network Gateway option is shown on the X.25 Configuration Utility Main Menu (see Figure 3-1. When selected, the Start X.25 Network Gateway option starts all of the configured X.25 lines.

SHUT DOWN X.25 NETWORK GATEWAY

The Shut down X.25 Network Gateway option is located on the X.25 Configuration Utility Main Menu (see in Figure 3-1). Specifying the Shut down X.25 Network Gateway option deallocates all of the X.25 lines.

RECONFIGURE X.25 LINE

The Reconfigure X.25 line option is located on the X.25 Configuration Utility Main Menu (see Figure 3-1). Specifying the Reconfigure X.25 line option allows you to dynamically reconfigure, shut down, or restart a single X.25 line without affecting the operation of other X.25 lines.

STATUS OF X.25 NETWORK GATEWAY

The Status of X.25 Network Gateway option is located on the X.25 Configuration Utility Main Menu (see Figure 3-1).

To select the Status of X.25 Network Gateway option on the X.25 Configuration Utility Main Menu and to display the Network Gateway status, complete the following steps:

1. Press **Tab** until the highlighted cursor is positioned over the Status of X.25 Network Gateway option.

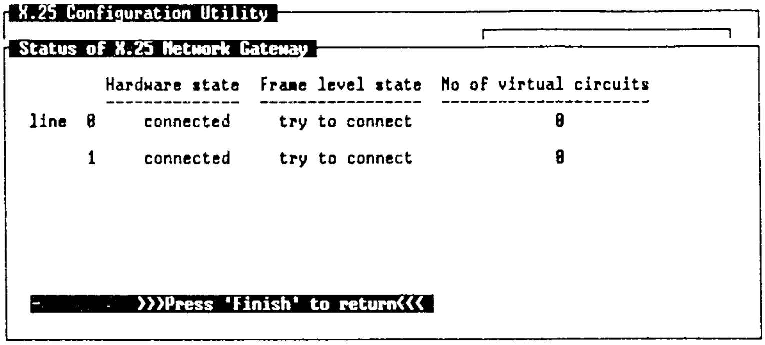
2. Press **Enter**. (The appropriate key name is displayed at the bottom of the menu.)

When the status of all X.25 lines is requested, the Status of X.25 Network Gateway Menu (shown in Figure 3-14) appears on your terminal screen.In this case, the Status of X.25 Network Gateway option displays the following information for all of the X.25 lines:

- hardware state

- frame level state

- number of permanent and switched virtual circuits



```
X.25 Configuration Utility

Status of X.25 Network Gateway

            Hardware state  Frame level state  No of virtual circuits
            --------------  -----------------  ----------------------
line  0        connected      try to connect             0

      1        connected      try to connect             0


                  >>>Press 'Finish' to return<<<
```

287-011

Figure 3-14. Status of X.25 Network Gateway Menu: All X.25 Line Statuses Request

The hardware can be in one of the following states:

- o disconnected
- o no carrier
- o connected

Frame level can be in one of the following states:

- o disconnect
- o trying to connect
- o data transfer

The number of virtual circuits includes both permanent and active switched virtual circuits.

To select the Status of X.25 Network Gateway option on the X.25 Configuration Utility Main Menu and to display the status of a single (specified) line, complete the following steps:

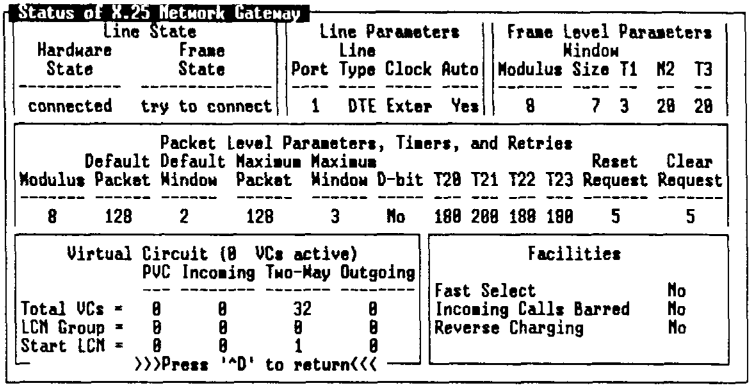
1. Press **Tab** until the highlighted cursor is positioned over the Status line number option.

2. Type the appropriate line number.

2. Press **Enter**. (The appropriate key name is displayed at the bottom of the menu.)

When the status of a single X.25 line is requested, the Status of X.25 Network Gateway Menu (shown in Figure 3-15) appears on your terminal screen. The Status of X.25 Network Gateway menu displays, in table format, the information described above. In addition, it displays the following information:

- line state
- line parameters
- packet level parameters
- frame level parameters

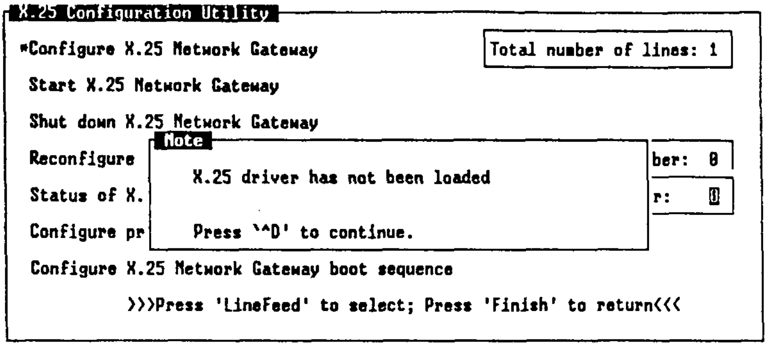


```
Status of X.25 Network Gateway
   Line State              ||   Line Parameters     || Frame Level Parameters
 Hardware      Frame       ||         Line          ||         Window
  State        State       || Port Type Clock Auto  ||Modulus Size T1  N2  T3
----------- -------------- || ---- ---- ----- ----  ||------- ---- --  --  --
 connected  try to connect ||  1   DTE  Exter  Yes  ||   8       7 3   20  20

               Packet Level Parameters, Timers, and Retries
        Default Default Maximum Maximum                       Reset   Clear
Modulus Packet  Window  Packet  Window D-bit T20 T21 T22 T23 Request Request
------- ------  ------  ------  ------ ----- --- --- --- --- ------- -------
   8     128      2      128      3     No   180 200 180 180    5       5

      Virtual Circuit (0  VCs active)       |              Facilities
              PVC Incoming Two-Way Outgoing |
              --- -------- ------- -------- | Fast Select                No
Total VCs =    0     0        32      0     | Incoming Calls Barred      No
LCN Group =    0     0        0       0     | Reverse Charging           No
Start LCN =    0     0        1       0     |
           >>>Press '^D' to return<<<
```

287-012

Figure 3-15. Status of X.25 Network Gateway Menu: Single X.25 Line Status Requested

If the specified line is not configured, the message shown in Figure 3-16 appears.



```
X.25 Configuration Utility

*Configure X.25 Network Gateway                  Total number of lines: 1

 Start X.25 Network Gateway

 Shut down X.25 Network Gateway
              Note
 Reconfigure                                                 ber:  0

 Status of X.    X.25 driver has not been loaded             r:    0

 Configure pr    Press '^D' to continue.

 Configure X.25 Network Gateway boot sequence

        >>>Press 'LineFeed' to select; Press 'Finish' to return<<<
```

287-013

Figure 3-16. X.25 Status Error Message

CONFIGURE PROC_TABLE

The Configure proc_table option is located on the X.25 Configuration Utility Main Menu (see Figure 3-1).

The **proc_table** consists of entries that determine how incoming calls are answered, based on the X.121 subaddress. Each entry is a line of fields with the following format:

sub-address auto-answer X25-or-VTI pathname args

where:

sub-address — is the last two digits of X.121 address.

auto-answer — indicates whether x25d or the application will answer the incoming call.

X25-or-VTI — indicates whether the application interfaces to the X.25 packet mode directly or to the virtual terminal interface. (The CTIX X.25 Terminal/Host Adapter is required for the VTI option.)

pathname — is the absolute pathname of the application to invoke upon an incoming call on the specified sub-address.

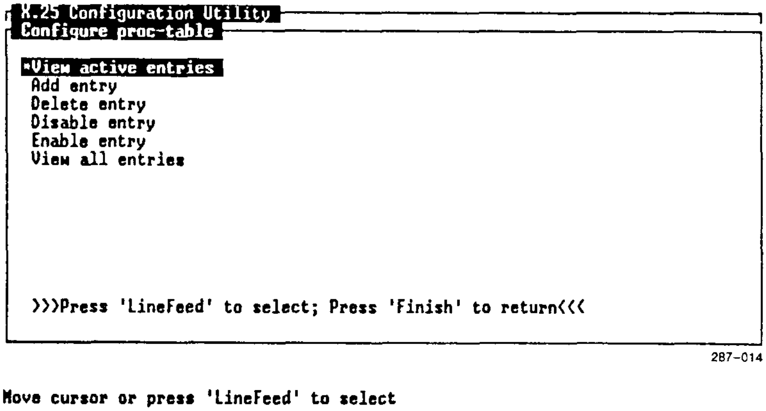
args — are up to 12 argument strings passed to the application, in argv or argc, during execution.

The Configure proc_table option allows you to configure the **proc_table** through use of the following options:

- View Active Entries
- Add Entry
- Delete Entry
- Disable Entry
- Enable Entry
- View all Entries

> **NOTE**
>
> If you press **Cancel** while configuring from the X.25 Configuration Utility Main Menu, none of **proc_table** entries are saved. If you choose to begin the configuration process again, press **Cancel**, and begin to reconfigure.

Figure 3-17 shows the Configure proc_table Menu.



```
X.25 Configuration Utility
Configure proc-table

*View active entries
 Add entry
 Delete entry
 Disable entry
 Enable entry
 View all entries

 >>>Press 'LineFeed' to select; Press 'Finish' to return<<<
                                                        287-014
Move cursor or press 'LineFeed' to select
```

Figure 3-17. Configure proc_table Menu

VIEW ACTIVE ENTRIES

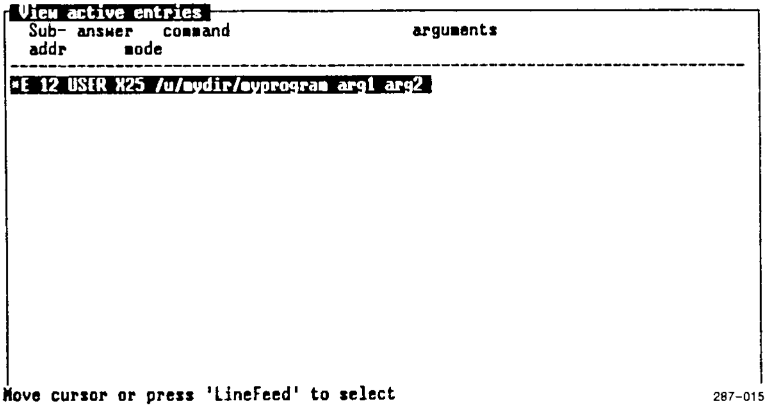
The View active entries function allows you to view all active entries in the **proc_table**. No modification is allowed.

To select the View active entries option from the Configure proc_table Menu, complete the following steps:

1. Press the **Up** or **Down** cursor keys until the highlighted cursor is positioned over the View active entries option on the Configure proc_table Menu.

2. Press **Enter**. (The appropriate key name is displayed at the bottom of the menu.)

Figure 3-18 shows the View Active Entries Menu.



```
 View active entries
  Sub- answer   command                         arguments
  addr      mode
-------------------------------------------------------------------------------
*E 12 USER X25 /u/mydir/myprogram arg1 arg2

Move cursor or press 'LineFeed' to select
```

287-015

Figure 3-18. View Active Entries Menu

ADD ENTRY

To select the Add entry option from the Configure proc_table Menu, complete the following steps:

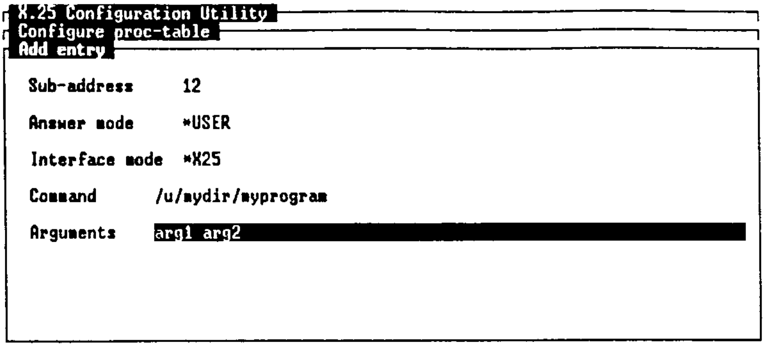
1. Press the **Up** or **Down** cursor keys until the highlighted cursor is positioned over the Add entry option.

2. Press **Enter**. (The appropriate key name is displayed at the bottom of the menu.)

The following information is displayed on the Add Entry Menu, as shown in Figure 3-19:

- Sub-address
- Answer mode
- Interface mode
- Command
- Arguments

The Sub-address, Answer Mode, Interface Mode, and Command options must be completed on the Add Entry Menu.



```
X.25 Configuration Utility
Configure proc-table
Add entry

Sub-address     12

Answer mode     *USER

Interface mode  *X25

Command         /u/mydir/myprogram

Arguments       arg1 arg2
```

287-016

Figure 3-19. Add Entry Menu

SUB-ADDRESS

To select the Sub-Address option from the Add Entry Menu, complete the following steps:

1. Press **Tab** until the highlighted cursor is positioned over the edit field located to the right of the Sub-address option.

2. Type the appropriate number.

3. Press **Tab** to move to the next field (the Answer Mode option).

Do not press **Cancel** or **Exit** until you have completed all of the items on the Add Entry Menu.

ANSWER MODE

To select the Answer mode option from the Add Entry Menu, complete the following steps:

1. Press **Tab** until the highlighted cursor is positioned over the menu field located to the right of the Answer mode option.

2. Press the **Up** or **Down** cursor keys until the appropriate value is displayed in the menu field.

3. Press **Tab** to move to the next field (Interface Mode option field).

Do not press **Cancel** or **Exit** until you have completed all of the items on the Add Entry Menu.

INTERFACE MODE

To select the Interface mode option from the Add Entry Menu, complete the following steps:

1. Press **Tab** key until the highlighted cursor is positioned over the menu field located to the right of the Interface mode option.

2. Press the **Up** or **Down** cursor keys until the appropriate value is displayed in the menu field.

3. Press **Tab** to move to the next field (Command option field).

Do not press **Cancel** or **Exit** until you have completed all of the items on the Add Entry Menu.

COMMAND

To select the Command option from the Add Entry Menu, complete the following steps:

1. Press **Tab** until the highlighted cursor is positioned over the edit field located to the right of the Command option.

2. Type the full pathname.

3. Press **Tab** to move to the next field (Arguments option field),

   or

press **Enter** to save the new configuration,

or

press **Exit** to return to the X.25 Configuration Main Menu.

ARGUMENTS

To select the Arguments option from the Add Entry Menu, complete the following steps:

1. Press **Tab** until the highlighted cursor is positioned over the edit field located to the right of the Arguments option.

2. Enter the parameters you want to have passed to the specified command.

3. Press **Enter** to save the new configuration,

   or

   press **Exit** to return to the X.25 Configuration Main Menu. (The appropriate key name is displayed at the bottom of the menu.)

DELETE ENTRY

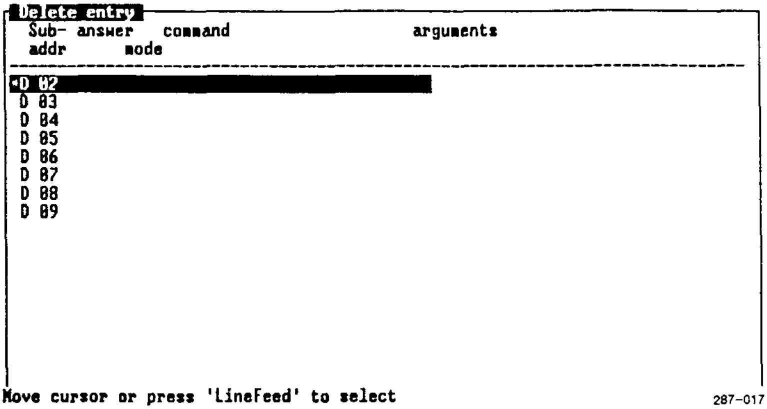
The Delete entry function allows you to delete the sub-address entry in the **proc_table.**

To specify the Delete entry option on the Configure proc_table Menu, complete the following steps:

1. Press the **Up** or **Down** cursor keys until the highlighted cursor is positioned over the Delete entry option.

2. Press **Enter.** (The appropriate key name is displayed at the bottom of the menu.)

The Delete Entry Menu appears as shown in Figure 3-20.



```
 Delete entry
  Sub- answer   command                        arguments
  addr      mode
-----------------------------------------------------------------------
*D 02
 D 03
 D 04
 D 05
 D 06
 D 07
 D 08
 D 09

Move cursor or press 'LineFeed' to select
```

287-017

Figure 3-20. Delete Entry Menu

To use the Delete entry function, complete the following steps:

1. Press the **Up** or **Down** cursor keys until the highlighted cursor is positioned over the sub-address you want to delete.

2. Press **Enter**. (The appropriate key name is displayed at the bottom of the menu.)

3. Repeat Steps 1 and 2 as often as necessary.

4. Press **Exit**. (The appropriate key name is displayed at the bottom of the menu.)

DISABLE ENTRY

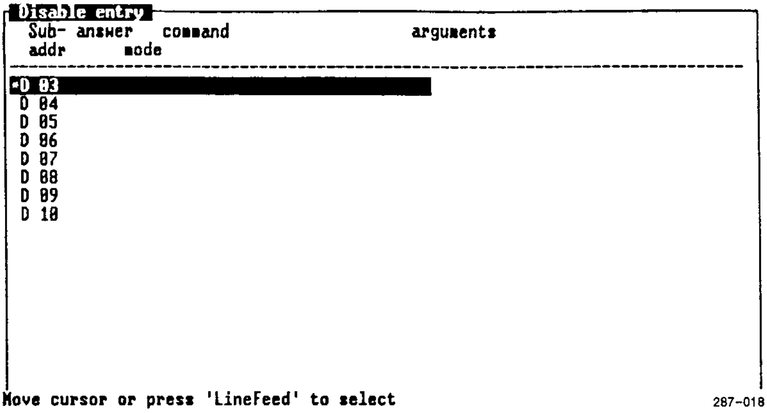
The Disable entry function allows you to disable the sub-address entry in the **proc_table**.

To select the Disable entry from the Configure proc_table Menu, complete the following steps:

1. Press the **Up** or **Down** cursor keys until the highlighted cursor is positioned over the Disable entry option on the Configure proc_table Menu.

3. Press **Enter**. (The appropriate key name is displayed at the bottom of the menu.)

The Disable Entry Menu appears as shown in Figure 3-21.



```
 Disable entry
  Sub- answer   command                       arguments
  addr     mode
-------------------------------------------------------------------------
 D 03
 D 04
 D 05
 D 06
 D 07
 D 08
 D 09
 D 10

Move cursor or press 'LineFeed' to select
```

Figure 3-21. Disable Entry Menu

To use the Disable entry function, complete the following steps:

1. Press the **Up** or **Down** cursor keys until the highlighted cursor is positioned over the subaddress you want to disable.

2. Press **Enter.** (The appropriate key name is displayed at the bottom of the menu.)

3. Repeat Steps 1 and 2 as often as necessary.

4. Press **Exit.** (The appropriate key name is displayed at the bottom of the menu.)

ENABLE ENTRY

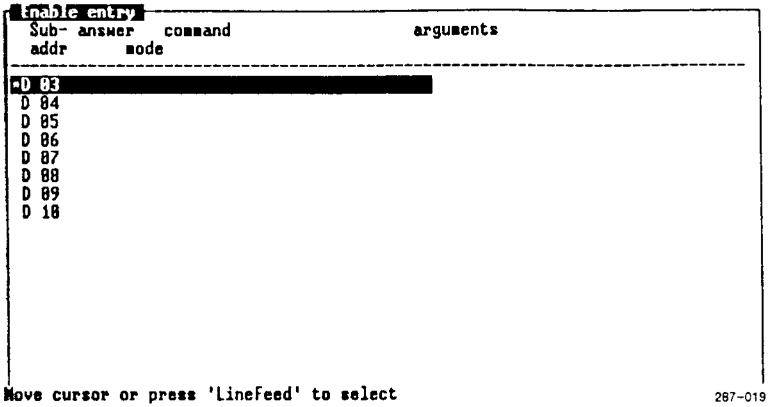
The Enable Entry function allows you to enable the sub-address in the **proc_table**.

To specify the Enable entry option on the Configure proc_table Menu, complete the following steps:

1. Press the **Up** or **Down** cursor keys until the highlighted cursor is positioned over the Enable entry option on the Configure proc_table Menu.

3. Press **Enter**. (The appropriate key name is displayed at the bottom the menu.)

The Enable Entry Menu appears as shown in Figure 3-22.



```
Enable entry
 Sub- answer   command                        arguments
 addr      mode
-----------------------------------------------------------------------
*D 03
 D 04
 D 05
 D 06
 D 07
 D 08
 D 09
 D 10

Move cursor or press 'LineFeed' to select
```

287-019

Figure 3-22. Enable Entry Menu

To use the Enable entry function, complete the following steps:

1. Press the **Up** or **Down** cursor keys until the highlighted cursor is positioned over the sub-address you want to enable.

2. Press **Enter**. (The appropriate key name is displayed at the bottom of the menu.)

3. Repeat Steps 1 and 2 as often as necessary until you have finished enabling sub-addresses.

4. Press **Exit**. (The appropriate key name is displayed at the bottom of the menu.)

VIEW ALL ENTRIES

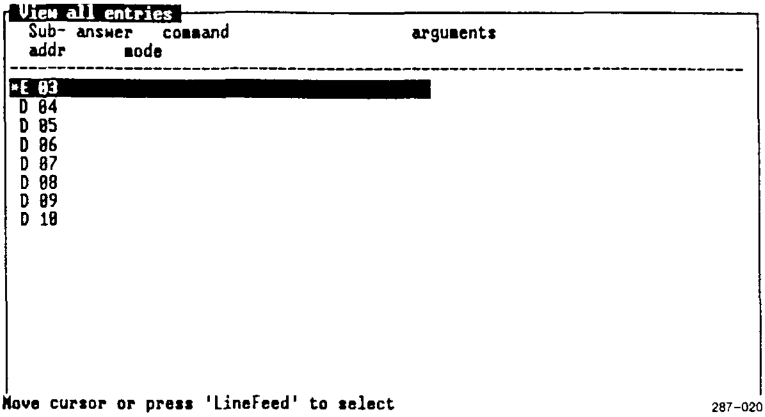
The View all entries function allows you to view all of the enabled and disabled entries in the **proc_table**. No modification is allowed.

To specify the View all entries option on the Configure proc_table Menu, complete the following steps:

1. Press the **Up** or **Down** cursor keys until the highlighted cursor is positioned over the View all entries option on the Configure proc_table Menu.

3. Press **Enter**. (The appropriate key name is displayed at the bottom of the menu.)

The View all Entries Menu appears as shown in Figure 3-23.



```
View all entries
 Sub- answer  command                    arguments
 addr      mode
-------------------------------------------------------------------------------
*E 03
 D 04
 D 05
 D 06
 D 07
 D 08
 D 09
 D 10
Move cursor or press 'LineFeed' to select
```

287-020

Figure 3-23. View all Entries Menu

CONFIGURE X.25 NETWORK GATEWAY BOOT SEQUENCE

The Configure X.25 Network Gateway boot sequence option modifies the **/etc/rc** file to bring up the X.25 Network Gateway automatically during the system boot.

To select the Configure X.25 Network Gateway boot sequence option, complete the following steps:

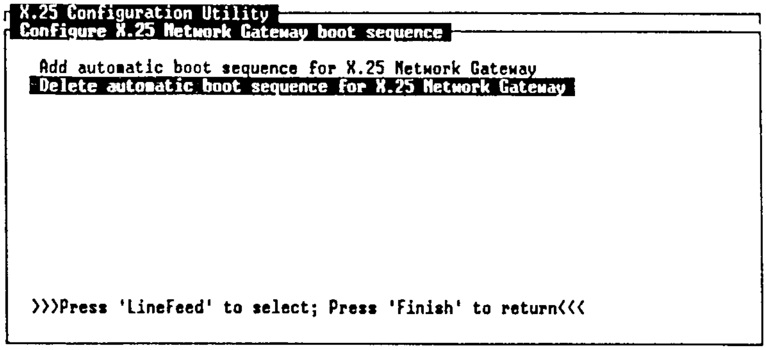
1. Display the X.25 Configuration Utility Main Menu (see Figure 3-1).

2. Press **Tab** until the highlighted cursor is positioned over the menu area.

3. Press the **Up** or **Down** cursor keys until the highlighted cursor is positioned over the Configure X.25 Network Gateway boot sequence option.

4. Press **Enter**. (The appropriate key name is displayed at the bottom of the menu.)

The Configure X.25 Network Gateway Boot Sequence Menu is displayed as shown in Figure 3-24.



```
X.25 Configuration Utility
Configure X.25 Network Gateway boot sequence

  Add automatic boot sequence for X.25 Network Gateway
  Delete automatic boot sequence for X.25 Network Gateway

>>>Press 'LineFeed' to select; Press 'Finish' to return<<<
```

287-021

Figure 3-24. Configure X.25 Network Gateway Boot Sequence Menu

To specify the Configure X.25 Network Gateway boot sequence option, complete the following steps:

1. Press the **Up** or **Down** cursor keys until the highlighted cursor is positioned over the configuration option of your choice on the menu.

2. Press **Enter.** (The appropriate key name is displayed at the bottom of the menu.)

4 PROGRAMMATIC INTERFACE

Chapter 4 describes the Programmatic Interface of the Gateway. This chapter contains the following information:

- a complete description of the Programmatic Interface and the five basic system calls
- instructions for establishing a virtual circuit
- an in-depth description of data transfer modes
- a sample program

OVERVIEW

The Programmatic Interface resides as a loadable character device driver in the CTIX kernel. The following sections provide in-depth discussions of the functions supported by the Gateway.

The Programmatic Interface uses the **open**, **close**, **read**, **write** and **ioctl** basic system calls.

USING THE DRIVER COMMANDS

This section describes the following procedures:

- how to establish a virtual circuit (VC)
- how to transfer data on an established VC
- how to close an established VC

TERMINOLOGY

close

close is one of the five basic system calls used by the Programmatic Interface. The **close** function closes a CTIX X.25 special file, or LUN. The LUN is deallocated, and the associated virtual circuit is cleared.

Data Mode

Data mode is one of two Gateway modes of operation in which the CTIX application program transfers only pure data. When operating in data mode, the Gateway performs the segmenting and packetizing (assembly/ disassembly) of data.

ioctl

ioctl is one of the five basic system calls used by the Programmatic Interface. Many **ioctl** functions are defined by CTIX. Additional **ioctl** functions are supported by the Gateway to provide various mechanisms to control parameters and inquire about the status of the Gateway.

open

open is one of the five basic system calls used by the Programmatic Interface. The **open** function opens a CTIX X.25 special file, or LUN. It allocates a LUN for communication over the X.25 network. The **open** function does not initiate packet transmission.

Packet Mode

Packet mode is one of two Gateway modes of operation in which the CTIX application programs may specify X.25 packet transfer over the Gateway.

Permanent Virtual Circuit

A permanent virtual circuit (PVC) is a permanent logical connection between two DTEs. A PVC does not require normal call establishment or clearing procedures.

read

read is one of the five basic system calls used by the Programmatic Interface. The **read** function receives data buffers (packets) that were previously received by the Gateway on the X.25 line.

Virtual Circuit

A virtual circuit (VC) is a logical association between two DTEs.

write

write is one of the five basic system calls used by the Programmatic Interface. The **write** function presents data buffers (or packets) to the Gateway for subsequent transmission to the X.25 line.

ESTABLISHING A VIRTUAL CIRCUIT

Virtual circuits are established in one of the following ways:

- by transmitting a Call Request packet and waiting for the Call Connected packet

  or

- by receiving an Incoming Call and transmitting a Call Accepted packet

A permanent virtual circuit (PVC) is one form of virtual circuit. A PVC does not require transmission or reception of the Call Request and Call Accepted packets to establish a virtual circuit for data transfer. See Chapter 5, "Permanent Virtual Circuits," for additional information.

Outgoing Call Request

An LUN must be opened before data exchange can occur. Opening a LUN is accomplished by issuing an **open** system call accompanied by an available LUN. Unless you issue the IOX25LUN command, you are responsible for locating a free LUN. An error code is returned if the application requests an **open** on a LUN that is in use. You can also specify that the X.25 driver choose an available LUN (see "Programmatic Interface ioctl Formats" later in this chapter for additional information). The **open** system call returns a LUN descriptor that is used by the application along with the **read**, **write**, **open**, **close**, and **ioctl** system calls.

The application then specifies which line and adapter to use by issuing an IOX25MODE **ioctl** command. If you do not specify the line and adapter, X.25 uses the default values (Default line = 0; Adapter = X.25).

The next step in establishing a virtual circuit is to issue an IOX25OPN **ioctl** command. IOX25OPN creates a path through the driver to the link in the Programmatic Interface. The logical channel number (LCN) is issued as an argument to IOX25OPN and must be specified equal to -1 for outgoing calls.

At this point, the LUN is available for writing the Call Request packet. The application formulates the necessary information in the Call Request packet being issued to the remote station. The **write** system call is then used to transmit the Call Request packet to X.25 for transmission on the line.

The virtual circuit is established as soon as the Call Connected packet is received. You must use the **read** system call to receive the Call Confirmation packet. Once the Call Request packet is sent, one of the following situations occurs:

- A Call Connected packet is received if the call is successful.

  or

- A Clear packet is received if the remote rejects the Call Request packet or if the X.25 driver determines that the packet is improperly formatted.

Complete one of the following procedures if a Clear packet is received on the **read** request:

- issue a new Outgoing Call Request packet on the same LUN

  or

- use the **close** system call to close the LUN

When a Call Accept packet is received on the **read**, the LUN is ready for data exchange in packet mode. If you choose, data exchange can also occur in data mode. To convert a LUN from packet mode to data mode, issue the IOX25DATA **ioctl** command. To switch the LUN back from data mode to packet mode, issue the IOX25PKT **ioctl** command.

Incoming Calls

Incoming Calls are considered to be an asynchronous event and, as such, are addressed in the following manner.

When an Incoming Call is received, X.25 sends a signal (SIGX25) to the x25d daemon process. x25d reads the Incoming Call packet, the only packet type ever read on the SERVER LUN. The Incoming Call packet has the X.25 line number encoded into the packet type field. x25d passes the line number and LCN in the arguments to the application, which is selected by the sub-address in the Call Request packet.

Next, the application opens an available LUN. The available LUN is located with the IOX25LUN **ioctl** command or by requesting an available LUN from X.25. It issues the IOX25MODE and IOX25OPN **ioctl** commands as it would do for an Outgoing Call.

However, in the case of an Incoming Call, the LCN specified in the IOX25OPN **ioctl** must be the one on which the Incoming Call packet was received.

Both the line number and the LCN for the application's IOX25MODE and IOX25OPN **ioctl** commands are obtained from the Incoming Call packet.

Once the **open** and consecutive **ioctl** commands are successful, the application must issue a **read** system call. Issuing the **read** system call enables the application itself to receive the Incoming Call packet. At this point, the application verifies whether the remote host is requesting a facility that is either not supported or not acceptable. When this situation occurs, the application rejects the Incoming Call Request packet.

To reject the Incoming Call packet, the application performs one of the following procedures:

- o writes a Clear packet, posts a **read** to wait for the Clear Confirmation, and closes the LUN

  or

- o closes the LUN immediately.

This prompts X.25 to send the Clear packet automatically. An exit from the program also closes the LUN.

Accepting the Incoming Call Request packet is done by sending a Call Accept packet with the **write** system call. When this procedure is completed, the LUN is ready for data transfer in packet mode. You can switch the LUN to data mode by issuing the IOX25DATA **ioctl** command.

Permanent Virtual Circuits

A permanent virtual circuit (PVC) is a virtual circuit that is mutually agreed upon between the network and the system administrator. A PVC is considered to be in data transfer state upon the line being physically and logically established. The Gateway is configured with a specified number of LUNs, which are reserved for use as PVCs. Applications cannot transmit Call Request packets or receive Incoming Call packets on PVCs.

To use the PVC, complete the following steps:

1. Open a LUN reserved for PVC use. (Your system administrator can provide you with a list of available PVCs.)

2. Issue the IOX25MODE **ioctl** command to specify the appropriate line and adapter numbers.

3. Issue the IOX25OPN **ioctl** command to actually open the LUN for data transfer.

Use an LCN number of (-1) in the IOX25OPN **ioctl** argument, as the Programmatic Interface knows which virtual circuit is associated with the LUN.

If the open sequence was successful, the LUN is opened for data transfer in packet mode. The application can transfer data on the PVC as it would a switched virtual circuit opened for data transfer.

Resetting a PVC can be accomplished in one of the following ways:

- Send a Reset packet on the LUN, and then close the LUN.

  or

- Close the LUN, forcing X.25 to automatically transmit a Reset packet.

Because the X.25 protocol does not allow transmission of a Clear packet on a PVC, the Programmatic Interface converts any Clear Request packet into a Reset packet.

DATA TRANSFER

Packet Mode

Packet mode operation allows you limited access to all X.25 features. The packet mode format of read and write packets is identical to that described in Data Communication Networks Services and Facilities: Terminal Equipment and Interfaces. All CTIX applications running in conjunction with the Gateway must comply with these requirements.

When reading a data packet, the M bit, D bit, and Q bit must be set in the same way they were received from the network. When writing a data packet, the M bit, D bit, and Q bit must be set in the packet header according to your specific Gateway-to-network requirements.

The Programmatic Interface does not notify the application when a data packet is received. Therefore, it is good practice to always have a pending read on any open LUN in the data exchange state, or use the IOX25READ **ioctl** command to poll for data packets.

The application is not allowed to write a data packet larger than the packet size used on the virtual circuit. If the application attempts to write an oversize data packet, an error is returned and the data packet is not transmitted.

Data Mode

Data mode operation allows an application to perform data exchange without regard to packet size and packet headers. When the transmitted packet size is greater than the specified maximum packet size, the driver automatically segments the packet into several packets with the M bit (more bit) set.

Packets that are received with the M bit set are combined and available to one large **read**. Reads performed in the.data mode return when one of the following conditions exists:

- The data packet is received with the M bit cleared. It then becomes the last packet of the sequence.

- The requested data length was reached. In this case, the remaining received data is available on the next **read**.

- The virtual circuit is cleared or reset. In this case, a read error is returned to you.

CAUTION

The following restrictions apply when transmitting data in the data mode:

- The Q bit is not supported. Packets received with the Q bit set are processed in the same way as a regular data packet. Applications cannot send a packet with the Q bit set in data mode.

- Packets that are received with the D bit set are processed the same as regular data packets. However, these packets are processed by X.25. The application cannot send packets with the D bit set.

Clear, Reset, and Interrupt packets are processed in a specific way and are discussed in the following sections of this chapter.

X.25 Adapter

When using the X.25 adapter, data is transferred to and from the X.25 network exactly as it is written and read. There are two requirements pertaining to the size of buffers read:

- buffer size < received packet size

  If the buffer size is less than the received packet size (or less than the total length of a series of packets with the M bit set and the last one with the M bit clear), the **read** is totally satisfied. Only the requested amount of data is transferred to you for the **read**. The rest of the data remains until the next **read**.

- buffer size > received packet size

  In this case, the **read** is satisfied immediately, but only the number of bytes received are transferred to you. You should not attach significance to the fact that fewer bytes were returned than were asked for.

The following information pertains to buffers written using the **write** system call:

- buffer size < maximum packet size

  In this case, data is sent immediately in one packet.

- buffer size > maximum packet size

  In this case, data is sent in several packets. The first n-1 packets are all of maximum packet size length and have the M bit set. The last packet may be of any size and must have the M bit clear. The M bit indicates that the packets are part of a multipacket sequence and should be recombined when they arrive on the receiving end.

X.29 Adapter

The main difference between the X.25 and X.29 adapters is that the X.29 adapter handles X.29 messages received from the remote DTE and reflects X.3 parameter changes in the **termio** structure. TCSETA and TCGETA **ioctl** commands may be used in the X.29 adapter. The same rules apply for buffer sizes as described above in "X.25 Adapter," with the following constraint:

When receiving data in data mode, data is transferred to you only up to the first end-of-line character, or up to the end of the received packet or packet sequence (with M bit). Subsequent **reads** return more data on a line-by-line basis. The end-of-line character is defined in the initialization table.

In certain modes, the LF character is mapped into a CR character on data send, and the CR character is mapped into an LF character on data received.

Use of the D Bit

Reception of data packets with the D bit set is transparent to the Gateway. The D bit is transmitted to the application in the packet header only when data is transferred in packet mode.

The D bit is transparently transferred. It is the application's responsibility to implement a protocol for handling the D bit end-to-end acknowledgment of data packets.

Sending data packets with the D bit set can occur only when data is transmitted in packet mode. To transmit such a packet, the D bit must be set in the Call Request and Call Confirmation packets when the virtual circuit is established.

When the application sends a data packet with the D bit set, one byte of data (used as a type of serialization or local identification) must be provided in the packet header. This identification byte is returned to the application in a Data Confirmation packet once it is acknowledged by the remote host. This Confirmation packet is formatted in the same way as an Interrupt packet, but with a packet type value of 10 (0x0A).

The identification byte provided in the data packet is returned in the first data byte of the Confirmation packet.

> **NOTE**
>
> The CCITT X.25 recommendations do not support a Data Confirmation packet. The Data Confirmation packet was introduced into the Gateway to advise applications of the D bit confirmation. A Data Confirmation packet can never be transmitted on the X.25 line. The identification byte provided in the packet header must be a nonzero value.

CLEAR PACKET PROCESSING

Overview

Clear Request packets can be sent by the application to clear the associated virtual circuit. An application is prompted to clear the virtual circuit for one or more of the following reasons:

- o to reject an incoming call
- o to abort a transaction with the remote host when an error is detected by the application
- o to terminate a transaction with the remote host

A Clear Indication packet might be received as an answer to an Outgoing Call Request under one or more of the following conditions:

- o The remote host rejected the call.
- o The remote host is busy and cannot be reached by the network.
- o The specified address is either illegal or does not exist.
- o The network detected an incorrect or nonsupported facility field.
- o The network did not answer the call within the specified time (T21 Timer).
- o X.25 locally rejected the Outgoing Call Request for one or more of the following reasons:
    - The LUN was not opened.
    - No virtual circuit was available.
    - There was an incorrect address or facility field.
    - The virtual circuit was already established.

A Clear packet may also be received any time during data transfer when the remote host wants to terminate the transaction or when a procedure error is detected either by the remote host, the network, or by the local X.25 Gateway.

When a clear packet is received, the Programmatic Interface automatically responds with a Clear Confirmation packet. Applications are not responsible for writing Clear Confirmation packets.

When a Clear packet is transmitted or received, it releases the virtual circuit. The application can then use the LUN again to place a new Outgoing Call Request or to receive a new Incoming Call Request on the same virtual circuit. However, it is recommended that you close the LUN, or exit, each time a Clear packet is received. This procedure ensures that the application does not make an error in reusing the LUN and that the LUN does not remain open while not in use.

Applications can clear a virtual circuit in a practical and efficient way by exiting or closing the LUN. The Programmatic Interface automatically transmits a Clear Request packet upon the close. This automatic Clear Request packet transmission occurs when the LUN is in either packet or data mode.

Clear packets are not flow controlled. When an application transmits a Clear packet, X.25 automatically inserts the Clear packet in front of any buffered packets. In this case, some data packets may be lost. The application should be aware of this and provide a method of synchronization to avoid subsequent loss of data occurring at the end of a transaction.

NOTE

A successful **write** of a data packet indicates that the X.25 driver has accepted the data. However, it does not imply that the remote host has received it.

Reception of Clear Packets in Packet Mode

When operating the Gateway in packet mode, the following actions are taken whenever a Clear packet is received:

- If there is a pending **write**, the **write** returns with an error status. Data related to this particular **write** is lost.

- If there is a pending **read**, the Clear packet is returned to the **read**.

Reception of Clear Packets in Data Mode

When operating the Gateway in data mode, Clear packets cannot be sent directly to the application because **read** commands in data mode are used only to read data. As a result, when receiving Clear packets in data mode, **read** and **write** commands are addressed in one of the following ways:

- If there is a pending, **write**, it returns with an error status and the data is lost.

- If there is a pending **read**, it returns with an error. All of the data received and buffered by X.25 is lost.

- If there is no pending **read**, a SIGBAND signal is sent to the application. The application then performs a **read** to retrieve the Clear packet. The X.25 driver may be prompted to send a SIGBAND signal when it receives one of the following packet types:

Interrupt Confirmation packet
Interrupt packet
Clear packet
Reset packet

For this reason, the application must perform a **read** to retrieve the precise cause of the SIGBAND.

In either case, a SIGBAND signal is queued to the application by the Gateway. The application may use an IOX25STR **ioctl** command (status request) to retrieve the cause of the SIGBAND. A Clear cause message is then returned in the argument portion of this **ioctl**. The application program may also switch to packet mode and read the received packet causing the SIGBAND.

When a Clear packet is received, the LUN is automatically switched to packet mode. If the Clear packet includes user data or is a Clear packet with a Fast Select facility field, the application must perform a **read** to retrieve the data. If the **read** is not performed, the LUN must be closed.

Transmitting Clear Packets

Clear packets are transmitted by an application to terminate a transaction with the remote host.

> **NOTE**
>
> An LUN must be operating in packet mode to transmit a Clear packet.

If the application wants to transmit a Clear packet while operating in data mode, the application must first switch the LUN to packet mode. This is accomplished by issuing the IOX25PKT **ioctl** command before transmitting the Clear packet.

If the application transmits the Clear Request packet, no Clear Confirmation packet is sent to the application.

RESET PACKET PROCESSING

Reset packets are processed by X.25 in much the same fashion as Clear packets, with the following exception. After a Reset packet is received, the virtual circuit is not released and can be used to resume the current transaction. However, it is the responsibility of the application program to implement a protocol that performs an appropriate function on the Reset.

As stated above, transmitting and receiving Reset packets is the same as that for Clear packets. Refer to "Clear Packet Processing," for additional information.

INTERRUPT PACKET PROCESSING

Interrupt packets, as is the case with Reset and Clear packets, are not subject to normal packet-level flow control. Interrupt packets have an immediate effect and bypass any flow-controlled packets in the network. An Interrupt packet provides a convenient way of sending one byte of information, as expedited data, to a remote host.

Reception of Interrupt Packets in Packet Mode

If there is a pending **read** on the LUN, it returns immediately with the Interrupt packet. Because the Programmatic Interface transmits Interrupt Confirmation packets automatically, the application is not required to write Interrupt Confirmation packets. The Interrupt Confirmation is generated by the Gateway when the Interrupt packet is read by the user.

Reception of Interrupt Packets in Data Mode

When the Gateway receives an Interrupt packet on a LUN operating in data mode, a SIGBAND signal is sent to the application. The application then retrieves the cause of the SIGBAND by issuing an IOX25STR **ioctl**. This procedure returns an "interrupt received" event with the interrupt information byte contained in the Interrupt packet. The application is not required to send an Interrupt Confirmation packet, as the Gateway does this automatically. The Interrupt Confirmation is generated by X.25 when the Interrupt packet is read by the application.

Transmission of an Interrupt Packet in Packet Mode

To transmit an Interrupt packet on a LUN operating in packet mode, the application must properly format the Interrupt packet and then write the packet on the LUN. Interrupt packets may include one byte of data plus a packet header. When the Gateway receives the Interrupt Confirmation packet from the remote host, the following actions occur:

1. If there is a pending **read** on the LUN, it returns with the Interrupt Confirmation packet.

2. If no pending **read** exists on the LUN, a SIGBAND signal is sent to the application. The application must then perform a **read** to retrieve the Interrupt Confirmation packet.

Sending an Interrupt packet has no effect on data transmission.

Transmission of an Interrupt Packet in Data Mode

Interrupt packets cannot be directly written to X.25 on a LUN operating in data mode (a **write** is reserved for Data packets in data mode). Therefore, the application has one of the following options when sending an Interrupt packet:

- The LUN can be switched to packet mode by issuing the IOX25PKT **ioctl** system call. Once in packet mode, the Interrupt packet can be written.

- The IOX25IT **ioctl** command can be issued to prompt the Programmatic Interface to formulate and transmit the Interrupt packet. It is not necessary to switch the LUN to packet mode, as in the previous option. The LUN remains in data mode.

Upon reception of the Interrupt Confirmation packet from the remote host, the Programmatic Interface sends a SIGBAND signal to the application. The IOX25STR **ioctl** system call must be issued to retrieve the cause of the SIGBAND. The returned event is an "Interrupt Confirmation Received."

Sending an Interrupt packet in data mode has no effect on data transmission.

SYSTEM CALLS

The Programmatic Interface uses the **open**, **close**, **read**, **write**, and **ioctl** system calls. The following sections describe each of the system calls. Figure 4-1 shows the relationship between the Gateway Programmatic Interface and the system calls.

There are numerous **ioctl** functions supported by the Gateway. See Chapter 5, "Programmatic Interface ioctl Formats," for additional information.

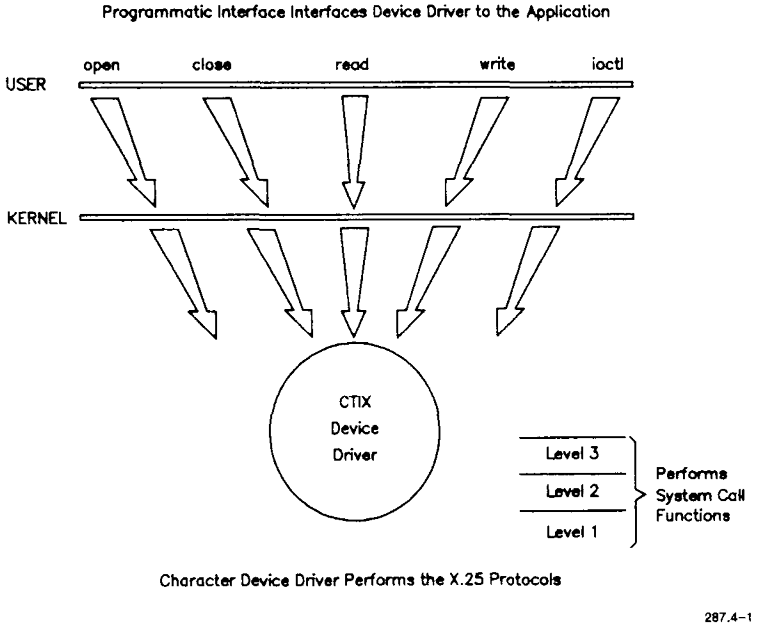

Programmatic Interface Interfaces Device Driver to the Application
open
close
read
write
ioctl
USER
KERNEL
CTIX Device Driver
Level 3
Level 2
Level 1
Performs System Call Functions
Character Device Driver Performs the X.25 Protocols
287.4-1

Figure 4-1. Relationship Between the Programmatic Interface and the Device Driver

SYNOPSIS

```
int close (fd)
int fd; /* file descriptor returns by open call*/
```

RETURN VALUE

The return value is 0 if the attempt was successful, -1 if an error is received.

DESCRIPTION

The **close** command deallocates a LUN that was previously used by an application program. The **close** command is automatically issued if the application program exits. This prompts transient buffers to be freed, and a Clear Request transmission is issued to the DCE if the virtual circuit is still active. Once the LUN's variables are reset and the buffers are deallocated, it is free for use by another application program.

SYNOPSIS

```
int ioctl(fd, command, arg)
int fd, command, *arg;
```

where:

fds — is the file descriptor returned from a successful open request. In the X.25 environment, fds is associated with a properly opened LUN and is the handle to the virtual circuit.

command — is the command type to be used for the **ioctl** request (the **ioctl** code).

arg — is an argument to an array of 4 bytes (long or integer, depending on the host CPU) used to exchange complimentary information between the Gateway and X.25. These four bytes are used either to pass information to X.25 or to return information from X.25.

RETURN VALUE

The return value is 0 if successful or, if an error is received, -1.

DESCRIPTION

The **ioctl** command is primarily used to control the device driver of the Gateway. This is accomplished by issuing a set of defined commands. Each command has an argument (a pointer to an integer, or a structure), which may be used for either passing parameters to the X.25 Gateway, passing responses from the X.25 Gateway, or both. The function of each **ioctl** command is briefly described below:

IOX25LUN — IOX25LUN is issued only on the SERVER LUN. It allocates a free LUN to the application program as the first step in establishing a virtual circuit.

IOX25OPN — IOX25OPN selects the logical channel number (LCN) on which virtual circuit establishment is attempted.

IOX25STR — IOX25STR aids the application process (Gateway operating in data mode) in determining the type of packet received when the SIGBAND signal is issued.

IOX25DATA — IOX25DATA switches the LUN into data mode.

IOX25PKT — IOX25PKT switches the LUN into packet mode.

IOX25LST

IOX25LST extracts status information from the X.25 line and returns it to the application. The IOX25LST command can be issued only on the SERVER LUN.

IOX25VST

IOX25VST returns the virtual circuit status on any USER LUN to the application. The IOX25VST command is issued on the G2 LUN.

IOX25SHUT

IOX25SHUT is issued on the SERVER LUN to inhibit further call establishment attempts.

IOX25MODE

IOX25MODE is used for selecting the desired physical line and adapter type. The IOX25MODE command is issued only on the SERVER LUN.

IOX25NET

IOX25NET is issued on the USER LUN and returns the network identification code for the line on which the virtual circuit is operational. The IO code is used by the administrator during configuration of the Gateway.

IOX25READ

IOX25READ allows polling the Gateway for pending packets available to be read. If IOX25READ is used in a polling loop, make sure that it is issued no more than once every five seconds. This avoids an extensive load on the CTIX system.

IOX25WRITE IOX25WRITE allows the application to determine if a **write** operation is blocked. Do not issue the **write** command more than once every five seconds.

IOX25DOWN IOX25DOWN allows the application to bring the link down a specified X.25 line. This procedure should be accomplished only by an administrative application.

IOX25UP IOX25UP allows an application to logically start an X.25 line. This procedure should be accomplished only by an administrative application.

IOX25START IOX25START allows an application to enable the modem handshake interface. This procedure should be accomplished only by an administrative application.

IOX25STOP IOX25STOP allows an application to disable the modem handshake interface. This procedure should be accomplished only by an administrative application.

IOX25GETINIT IOX25GETINIT retrieves a dynamic segment of the initialization table for a specified physical line from the X.25 software to the application.

SYNOPSIS

```
int open(name, mode)
char *name;     /* Path name for the LUN */
int mode;       /* Read-write mode */
```

RETURN VALUE

The return value is a file descriptor or, if an error is received, -1.

DESCRIPTION

The **open** command initializes the state and variables for the LUN specified in the argument to the **open** command. With the exception of the SERVER LUN, a LUN can be opened only once, and then it must be closed. This restriction prevents multiple application programs from attempting to open and use the same LUN (or reserves a LUN exclusively for an application). Upon completion of a successful **open** command, the application program references the file descriptor of the LUN.

The SERVER LUN can be opened multiple times. This allows multiple application programs to issue the **open** command to the SERVER LUN. Only **ioctl** commands can be issued from any application program on the SERVER LUN.

Therefore, an application program opens the following two LUNS:

- The SERVER LUN is opened for **ioctls** pertaining to general services (common to all VCs).

- The USER LUN is opened for data communication.

An attempt to issue a **close**, **read**, **write** or **ioctl** command to a LUN that is not currently open returns -1. The errno prompts an ENXIO error return. The **open** command is strictly internal to the X.25 driver environment and does not prompt any packet or frame level transmission.

SYNOPSIS

```
int read(fd, buffer, nbytes
int fd;         /* file descriptor returned by open
                command */
char *buffer;
unsigned int nbytes;
```

RETURN VALUE

The return value is -1 if an error is received or the actual number of bytes read.

DESCRIPTION

When the **read** command is issued while the Gateway is in packet mode, Data packets are copied to the read buffers of the application program. The **read** command blocks until the Gateway receives the data on the channel. If avoiding the blocking of the function call is necessary, an IOX25READ ioctl command is issued to check the availability of the data.

If the Gateway is operating in data mode, only the data portion of the Data packets are copied to the application program's buffer when a **read** command is successfully issued. There are no restrictions on the buffer size of the read buffer from the application. If the M bit is set in the Data packets, the data fields are concantenated in the application's buffer and are returned only when the buffer is full or when a packet with the M bit set to 0 was received.

SYNOPSIS

```
int write(fd, buffer, nbytes)
int fd;   /* file descriptor is returned by open
          command */
char *buffer;
unsigned int nbytes;
```

RETURN VALUE

The return value is the actual number of bytes written or, if an error is received, -1.

DESCRIPTION

The **write** command is issued to pass packets or data buffers to the Gateway for subsequent transmission on the line. The **write** command always blocks. It returns only when the packet is acknowledged. Issue IOX25WRITE to determine whether the **write** command will be blocked (whether a flow-control condition exists).

When the Gateway is operating in data mode, the write buffer may be of any size and may contain any type of data. If the buffer exceeds the maximum packet size (for example, 128), the Gateway segments the data and sends each 128-byte segment as a Data packet with the M bit set. The last segment in the buffer is always transmitted with the M bit set to 0.

NOTE

When operating in data mode, only Data packets are sent with each occurrence of the **write** command. The packets holding the M bit always consist of the maximum allowable bytes (confirmation default or negotiated number).

SIGNALS

The loadable character device driver sends signals to the application program to indicate the exceptional conditions of the Gateway. Values of the following signals are located in **/usr/include/sys/x25.h**.

SIGBAND

The SIGBAND signal is sent by the driver to the application when one of the following packet types is received:

- o Interrupt packet
- o Confirmation packet
- o Clear packet
- o Reset packet

The application can issue a **read** command (in packet mode) or an IOX25STR (in data mode) to determine the exact cause of the SIGBAND.

SIGX25

The SIGX25 signal is issued upon the arrival of an Incoming Call. This is used by x25d to detect the presence of an incoming call.

SIGXCDON

SIGXCDON is issued to signal a physical line connection.

SIGXCDOFF

SIGXCDOFF is issued to signal a physical line disconnection.

SIGXL2OFF

SIGXL2OFF is issued to signal a logical link disconnection.

SIGXL2ON

SIGXL2ON is issued to signal a logical link connection.

SIGDAT

SIGDAT is issued to signal receipt of a Data packet on the line.

HOW TO WRITE AN APPLICATION PROGRAM USING X.25

The following program listings are two simple test programs. One is the transmitter, and the other is the receiver.

The transmitter initiates an Outgoing Call and then enters a loop where it transmits and receives Data packets indefinitely.

The receiver must be spawned on an Incoming Call (for example, from the "transmitter") and loops back any packet received. A **proc_table** entry (no auto-answer and X.25 mode) must be created for the receiver before starting this test.

As shown in the sample listings, there is little difference between the two types of progams. Many of the setup procedures can be accomplished by using the functions of the X.25 Library.

SAMPLE TRANSMITTER PROGRAM

```
/*
This program initiates a Call_Request and then
continuously sends and receives data.
*/

#include <fcntl.h>
#include <stdio.h>
#include <signal.h>
#include <errno.h>

#define SIGBAND        26
                 /* Signals an unexpected event */
#define IOX25LUN       (('X'<<8)|1)
#define IOX25STR       (('X'<<8)|3)
#define IOX25OPN       (('X'<<8)|2)
#define IOX25MODE      (('X'<<8)|10)
#define CALL_REQUEST                0x0b
#define INCOMING_CALL               CALL_REQUEST
#define CALL_ACCEPT                 0x0f
#define CALL_CONNECTED              CALL_ACCEPT
#define CLEAR_REQUEST               0x13
#define CLEAR_INDICATION            CLEAR_REQUEST
#define CLEAR_CONFIRMATION          0x17
```

```
#define INTERRUPT                 0x23
#define INTER_CONFIRMATION        0x27
#define RESET_REQUEST             0x1b
#define RESET_INDICATION          RESET_REQUEST
#define RESET_CONFIRMATION        0x1f
#define REST_REQUEST              0xfb
#define REST_INDICATION           RESTART_REQUEST
#define REST_CONFIRMATION         0xff
#define DATA_PACKET               0x00
#define NBYTE 131

int fdev;         /* device file descriptor     */
int fserv;        /* server file descriptor     */
int mode;         /* packet mode = 0, data = 1  */
char lunpath[11]; /* x25 device pathname        */

char call_req[] =        /* Call_Request packet */
{0x00,0x00,CALL_REQUEST,0x0e,0x31,0x10,0x40,0x80,
 0x00,0x00,0x02,0x00};

char data_pack[] =               /* data packet */
{00,00,00,"This is a Data packet"};

sigband()        /*sigband server for data mode */
{
long status;
status = ioctl(fserv, IOX25STR, 0);
                  /* get packet type received   */
printf("sigband status is %08x ",status);
exit(0);
}

main()  /* START OF SENDER  */
{

char path[20], buffer[NBYTE];
int loopcount, nbyte, arg, cc, minor, lc_numb;
int line = 0;
int lcn = -1;                 /* outgoing call    */

if((fserv = open("/dev/x25000",O_RDWR)) == -1)
                              /* open serverlun   */
   {
   printf("Sender: /dev/x25000 open failed.");
   exit(0);
   }

if(ioctl(fserv,IOX25LUN,&minor) == -1)
                               /* get a free LUN  */
```

SAMPLE RECEIVER PROGRAM

```
/*
This program accepts a Call_Request and then
continuously sends and receives data.
*/

#include <fcntl.h>
#include <stdio.h>
#include <signal.h>
#include <errno.h>

#define SIGBAND        26
                 /* Signals an unexpected event */
#define IOX25LUN        (('X'<<8)|1)
#define IOX25STR        (('X'<<8)|3)
#define IOX25OPN        (('X'<<8)|2)
#define IOX25MODE       (('X'<<8)|10)
#define CALL_REQUEST            0x0b
#define INCOMING_CALL           CALL_REQUEST
#define CALL_ACCEPT             0x0f
#define CALL_CONNECTED          CALL_ACCEPT
#define CLEAR_REQUEST           0x13
#define CLEAR_INDICATION        CLEAR_REQUEST
#define CLEAR_CONFIRMATION      0x17
#define INTERRUPT               0x23
#define INTER_CONFIRMATION      0x27
#define RESET_REQUEST           0x1b
#define RESET_INDICATION        RESET_REQUEST
#define RESET_CONFIRMATION      0x1f
#define REST_REQUEST            0xfb
#define REST_INDICATION         RESTART_REQUEST
#define REST_CONFIRMATION       0xff
#define DATA_PACKET             0x00
#define NBYTE 131

int fdev;             /* device file descriptor */
int fserv;            /* server file descriptor */
FILE *flog;                 /* log file pointer */
int mode;          /* packet mode = 0, data = 1 */
char lunpath[50];        /* x25 device pathname */

char call_acc[] =       /* Call_Accepted packet */
{0x00,0x00,CALL_ACCEPT};

sigband()         /*sigband server for data mode */
{
fprintf(flog,"receiver: sigband received");
fclose(flog);
exit(-1);
}
```

```
main(argc,argv)            /* START OF RECEIVER */

int argc;
char *argv[];
{

char path[80], buffer[NBYTE];
int nbyte, i, line, minor, lcn;
long arg;

signal(SIGBAND,sigband);       /* enable signal */

line = atoi(argv[argc-2]);
       /* 2nd parameter, passed by x25 software */
lcn = atoi(argv[argc-1]);
       /* 3rd parameter, passed by x25 software */

flog = fopen(argv[1],"w");
fprintf(flog,"arg log file: %s\n",argv[1]);
fprintf(flog,"arg line no. = %d, arg lcn =
%d\n",line,lcn);

if((fserv = open("/dev/x25000",O_RDWR)) == -1)
                              /* open serverlun */
   {
   fprintf(flog,"receiver: /dev/x25000 open
failed.\n");
   exit(-1);
   }

fprintf(flog,"fserv = %d\n",fserv);

if(ioctl(fserv,IOX25LUN,&minor) == -1) /* get a
free LUN  */
   {
   fprintf(flog,"receiver: IOX25LUN errno is %d
\n",errno);
   exit(-1);
   }

minor >>= 24; /* convert LUN into minor number  */
minor &= 0x0ff;
if(minor == 0x0ff)
   {
   fprintf(flog,"receiver: no LUN available\n");
   exit(-1);
   }
```

```
sprintf(lunpath,"/dev/x25%03x",minor);
  /* convert minor # into path for special file */
fprintf(flog,"lunpath = %s\n",lunpath);

if((fdev = open(lunpath,O_RDWR)) == -1)
                                     /* open user lun */
   {
   fprintf(flog,"receiver: userlun %s open
failed\n",lunpath);
   fclose(flog);
   exit(-1);
   }

fprintf(flog,"userlun fdev = %d\n",fdev);

if(ioctl(fdev,IOX25MODE,&line) < 0)
                            /* select line # and mode */
   {
   fprintf(flog,"receiver: IOX25MODE errno %d
\n",errno);
   fclose(flog);
   exit(-1);
   }

fprintf(flog,"ioctl(%d,IOX25MODE,%d)\n",fdev,
line);

if(ioctl(fdev,IOX25OPN,&lcn) < 0)
                            /* setup lcn for incoming */
   {
   fprintf(flog,"receiver: IOX25OPN errno
%d\n ",errno);
   fclose(flog);
   exit(-1);
   }

fprintf(flog,"ioctl(%d,IOX25OPN,%d)\n",fdev,lcn);

if((nbyte = read(fdev,buffer,NBYTE)) == 0)
                              /* expect Incoming_Call */
   {
   fprintf(flog,"receiver: Incoming_Call not
received\n");
   fclose(flog);
   exit(-1);
   }

for(i = 0;i < nbyte;i++)
   fprintf(flog,"%02x ", (0x000000ff &
buffer[i]));
```

```
if(buffer[2] != INCOMING_CALL)
   {
   fprintf(flog,"Incoming_Call error\n");
   fclose(flog);
   exit(-1);
   }

fprintf(flog,"\nreceived an Incoming_Call
packet\n");

/* Establish VC by acknowledging Incoming_Call */

if(write(fdev,call_acc,sizeof(call_acc)) < 0)
   {
   fprintf(flog,"Call_Accepted errno %d\n",errno);
   fclose(flog);
   exit(-1);
   }
fprintf(flog,"sent Call_Accepted packet\n");

/* VC established.
Echo all received data packets */

while(1)
   {
   nbyte = read(fdev,buffer,NBYTE);
   fprintf(flog,"\nPacket content received:");
   for(i = 0;i < nbyte;i++)
        fprintf(flog,"%02x ",(0x000000ff &
buffer[i]));
   fprintf(flog,"\nPacket type received: ");
   switch(buffer[2])
        {
        case CLEAR_INDICATION:
        fprintf(flog,"Clear_Indication");
        fclose(flog);
        exit(-1);
        break;

        case INTERRUPT:
        /* gateway sent confirmation, just echo
           received INTERRUPT */
        fprintf(flog,"Interrupt Data");
        break;

        case INTER_CONFIRMATION:
        case RESET_CONFIRMATION:
        /* suppress transmission,
           should never occur */
        nbyte = 0;
        break;
        case RESET_INDICATION:
        fprintf(flog,"Reset_Indication");
        break;

        default:
        fprintf(flog,"Data Packet assumed");
        }

        if(write(fdev,buffer,nbyte) != nbyte)
             {
             fprintf(flog,"write errno
%d\n",errno);
             fclose(flog);
             exit(1);
             }
        fprintf(flog,"\nTransmitting packet
content: ");
        for(i = 0;i < nbyte;i++)
             fprintf(flog,"%02x ",(0x000000ff &
buffer[i]));
   }                                /* end of while */

}  /* END OF RECEIVER  */
```

5 PROGRAMMATIC INTERFACE IOCTL FORMATS

Chapter 5 provides a description of the Programmatic Interface **ioctl** formats.

OVERVIEW

The Programmatic Interface supports two different types of **ioctl** requests, depending upon the amount of information that must be transferred in the **ioctl** system call. Most **ioctl** requests use only four bytes of information, and the syntax is as follows:

```
ret = ioctl(fd, command, arg);
```

where:

fd — is the file descriptor returned from a successful open request. In the X.25 environment, fd is associated with a properly opened LUN and is the handle to the virtual circuit.

command — is the command type to be used for the **ioctl** request (the **ioctl** code).

arg — is a pointer to an integer (an array of 4 bytes, depending on the host CPU) used to exchange complementary information between the application and the driver.

ret is the return value from the Gateway. A value of 0 is returned if the **ioctl** is successful. If any other value is returned, this indicates an error. In this case, the content of the integer pointed to by arg is nonsignificant, and the error code is returned from errno.

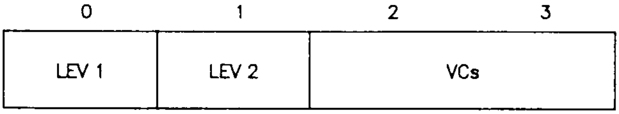
DESCRIPTION

The IOX25GETINIT command retrieves a dynamic segment of an initialization table for a specified physical line from the X.25 process for an application. This allows you to analyze the parameters currently in effect on the line being queried.

If the line number in this structure is set to (-1), the status segment of the initialization table is returned.

The IOX25GETINIT **ioctl** may be used at any point after normal startup of the X.25 process. It is particularly useful in obtaining an initialization table for a line that has failed an IOX25SNDINIT **ioctl**.

The format of the table to be transferred is that of **/usr/include/sys/initab.h**. After successful completion, the **xio_data** area of the **xio_struct** structure is filled with one byte 0xff on, one byte line number and the requested initialization table. A definition of xio_struct structure is located in **/usr/include/sys/x25.h**. Figure 5-1 shows the **xio_data** structure.



0	1	2	3
LEV 1	LEV 2	VCs	

287.5-1

Figure 5-1. xio_data Structure

IOX25LST

SYNOPSIS

```
ioctl(fd, IOX25LST, &arg);
int fd;         /* file descriptor to SERVER LUN */
int arg;        /* int for returning free LUN */
```

where:

fd — is the file descriptor to the SERVER LUN.

IOX25LST — is defined as (('X'<<8) | 6).

arg — is a pointer to an integer containing the specified line number for which the status is to be checked. (No initialization is necessary.)

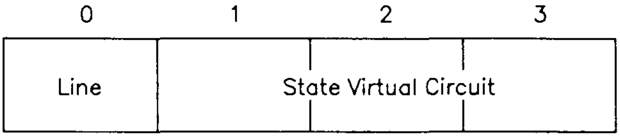
RETURN VALUE

If the **ioctl** is successful, 0 is returned. The return value from X.25 (found in the integer pointed to by arg) is a valid LUN number; -1 indicates that no LUN is available.

DESCRIPTION

The IOX25LST command can be issued only on the SERVER LUN.

The format of the integer pointed to by the argument when the IOX25LST returns successfully is shown in Figure 5-2.



287.5-2

Figure 5-2. IOX25LST Return Argument Format

where:

LEV1 is the current state of the hardware (level 1). The value is 0xff if disconnected, (0) if the connection is requested but the carrier is not yet up, (1) if level 1 is connected.

LEV2 is the current state of the frame level (level 2). The value is 0xff if in "disconnect" state, (0) if the frame level is currently trying to establish frame level, and (1) if the frame level is in data transfer state.

VCs is the number of active virtual circuits found on the line. This number includes both permanent and switched virtual circuits.

IOX25MODE

SYNOPSIS

```
ioctl(fd, IOX25MODE, &arg);
int fd;         /* file descriptor to SERVER LUN */
int arg;        /* int for returning free LUN */
```

where:

fd — is the file descriptor to the SERVER LUN.

IOX25MODE — is defined as (('X'<<8) | 10).

arg — is a pointer to an integer containing the requested adapter and line number as ((adapter<<8) | line). (No initialization is necessary.)

RETURN VALUE

If the **ioctl** is successful, 0 is returned. The return value from X.25 (found in the integer pointed to by arg) is a valid LUN number; -1 indicates that no LUN is available.

DESCRIPTION

The IOX25MODE command initializes the adapter and line number associated with a LUN.

NOTE

The IOX25MODE command can be issued only after the **open** system call.

The IOX25MODE command is used as the second step of the three-step open sequence. Issuing the IOX25MODE command ensures correct line and adapter selections. This command should be issued in cases where the initialization table has been modified without notification to all system users.

The possible values for the adapter number are as follows:

A_X25 requests the standard X.25 adapter. The value of A_X25 is 0.

A_X29 requests the X.29 adapter for TTY emulation. The value of A_X29 is 1.

SYNOPSIS

```
ioctl(fd, IOX25NET, &arg);
int fd;         /* file descriptor to USER LUN */
int arg;        /* int for returning free LUN */
```

where:

fd — is the file descriptor to the USER LUN.

IOX25NET — is defined as (('X'<<8) | 13).

arg — is a pointer to an integer. (No initialization is necessary.)

RETURN VALUE

If the **ioctl** is successful, 0 is returned. The return value from X.25 (found in the integer pointed to by arg) is a valid LUN number; -1 indicates that no LUN is available.

DESCRIPTION

The IOX25NET command, when issued on an opened LUN, returns the network identification code for the line on which the LUN is operating. It returns the network ID that was entered by the system administrator during Gateway configuration. The administrator is responsible for entering a valid code for the specific networks to which your system subscribes. It allows the application to distinguish different PDNs configured on different X.25 lines.

IOX25OPN

SYNOPSIS

```
ioctl(fd, IOX25OPN, &arg);
int fd;         /* file descriptor to SERVER LUN */
int arg;        /* int for returning free LUN */
```

where:

fd — is the file descriptor to the SERVER LUN.

IOX25OPN — is defined as (('X'<<8) | 2).

arg — is a pointer to an integer with a valid logical channel number (LCN). Possible values range from 0 to 4095 and may also be -1. (No initialization is necessary.)

RETURN VALUE

If the **ioctl** is successful, 0 is returned. The return value from X.25 (found in the integer pointed to by arg) is a valid LUN number; -1 indicates that no LUN is available.

DESCRIPTION

Issuing the IOX25OPN command is the last step required in opening a USER LUN. It allows the application to transmit outgoing calls, receive incoming calls, and exchange data. The IOX25OPN command directs X.25 to assign a LCN to a LUN. If the application opens a LUN to place an outgoing call or opens a LUN associated with a permanent virtual circuit, the LCN value must be -1.

A LCN value of -1 allows X.25 to choose a free LCN when placing an Outgoing Call Request or to make the physical connection to the Permanent Virtual Circuit (PVC) connected to the opened LUN.

In response to an incoming call, the application must pull the valid LCN from the Incoming Call packet as read on the SERVER LUN. In applications where x25d and the application are distinct, x25d is responsible for communicating the line number and LCN of the Incoming Call packet to the application.

IOX25PKT

SYNOPSIS

```
ioctl(fd, IOX25PKT, &arg);
int fd;         /* file descriptor to SERVER LUN */
int arg;        /* int for returning free LUN */
```

where:

fd — is the file descriptor to the SERVER LUN.

IOX25PKT — is defined as (('X'<<8) | 5).

arg — is a pointer to an integer. (No initialization is necessary.) The value of the addressed integer is changed by a return value from X.25. If the **ioctl** command is successful, the value returned is equal to the mode of the LUN before the IOX25PKT command was issued. Possible return values from X.25 are 0 (packet mode) and 1 (data mode).

RETURN VALUE

If the **ioctl** is successful, 0 is returned. The return value from X.25 (found in the integer pointed to by arg) is a valid LUN number; -1 indicates that no LUN is available.

DESCRIPTION

The IOX25PKT command is used to switch a LUN to packet mode. Do not issue this command if a **read** or **write** is pending on the LUN or if the LUN is not in the data transfer state.

When a LUN is operating in packet mode, the application must provide the 3-byte packet overhead required by the X.25 protocol and keep within the limit on the maximum Data packet size.

IOX25READ

SYNOPSIS

```
ioctl(fd, IOX25READ, &arg);
int fd;         /* file descriptor to SERVER LUN */
int arg;        /* int for returning free LUN */
```

where:

fd — is the file descriptor to the SERVER LUN.

IOX25READ — is defined as (('X'<<8) | 14).

arg — is a pointer to an integer. (No initialization is necessary.)

RETURN VALUE

If the **ioctl** is successful, 0 is returned. The return value from X.25 (found in the integer pointed to by arg) is a valid LUN number; -1 indicates that no LUN is available.

DESCRIPTION

The IOX25READ command allows an application to determine if there is data available to be read. Since the **read** system call blocks, the IOX25READ command facilitates applications to perform a **read** only when it will not block.

On return, the integer addressed by arg contains the number of packets currently held by the X.25 driver.

IOX25READ (continued)

IOX25READ should not be issued more than once every 5 seconds to avoid excessive system loading.

IOX25SHUT

SYNOPSIS

```
ioctl(fd, IOX25SHUT, &arg);
int fd;         /* file descriptor to SERVER LUN */
int arg;        /* int for returning free LUN */
```

where:

fd — is the file descriptor to the SERVER LUN.

IOX25SHUT — is defined as (('X'<<8) | 9).

arg — is a pointer to an integer containing the line number to shut down. (No initialization is necessary.)

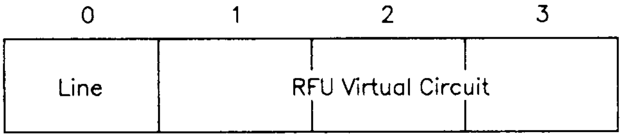
RETURN VALUE

If the **ioctl** is successful, 0 is returned. The return value from X.25 (found in the integer pointed to by arg) is a valid LUN number; -1 indicates that no LUN is available.

DESCRIPTION

The IOX25SHUT command can be issued only on the SERVER LUN. Its purpose is to prepare for a graceful shutdown before reloading. After issuing this **ioctl**, virtual circuit establishment is no longer allowed on the line, but any active virtual circuits are not affected.

The IOX25SHUT command returns two parameters upon successful completion in the integer pointed to by arg. The format of the returned value is shown in Figure 5-3.



287.5-3

Figure 5-3. IOX25SHUT Return Value Format

where:

Line is the line number as found in the integer pointed to by arg.

VC is the number of active virtual circuits at the time the IOX25SHUT request was issued.

NOTE

IOX25SHUT should be used only by administrative utilities.

IOX25START

SYNOPSIS

```
ioctl(fd, IOX25START, &arg);
int fd;         /* file descriptor to SERVER LUN */
int arg;        /* int for returning free LUN */
```

where:

fd — is the file descriptor to the SERVER LUN.

IOX25START is defined as (('X'<<8) | 19).

arg — argument is a pointer to an integer containing the line number to start. (No initialization is necessary.)

RETURN VALUE

If the **ioctl** is successful, 0 is returned. The return value from X.25 (found in the integer pointed to by arg) is a valid LUN number; -1 indicates that no LUN is available.

DESCRIPTION

The IOX25START command allows an application to physically start the Level 1 hardware of the communications board. It can be issued only on the SERVER LUN.

The IOX25START command directs Level 1 to turn on Request To Send (RTS) and Data Terminal Ready (DTR).

The IOX25START return value, in the integer pointed to by the arg, is 1 if the operation is completed and 0 if the operation is pending, that is, if the carrier is not yet up.

NOTE

The IOX25START command should be issued only by administrative utilities.

IOX25STOP

SYNOPSIS

```
ioctl(fd, IOX25STOP, &arg);
int fd;         /* file descriptor to SERVER LUN */
int arg;        /* int for returning free LUN */
```

where:

fd — is the file descriptor to the SERVER LUN.

IOX25STOP — is defined as (('X'<<8) | 16).

arg — is a pointer to an integer containing the line number to stop. (No initialization is necessary.)

RETURN VALUE

If the **ioctl** is successful, 0 is returned. The return value from X.25 (found in the integer pointed to by arg) is a valid LUN number; -1 indicates that no LUN is available.

DESCRIPTION

The IOX25STOP command allows an application to physically turn off the modem signals on the specified line. It can be issued only on the SERVER LUN.

The IOX25STOP command directs Level 1 to turn off Request To Send (RTS) and Data Terminal Ready (DTR).

The IOX25STOP return value, in the integer pointed to by arg, is 1 if the operation is completed.

IOX25STR

SYNOPSIS

```
ioctl(fd, IOX25STR, &arg);
int fd;         /* file descriptor to SERVER LUN */
int arg;        /* int for returning free LUN */
```

where:

fd — is the file descriptor to the SERVER LUN.

IOX25LUN — is defined as (('X'<<8) | 3).

arg — is a pointer to an integer. (No initialization is necessary.) In this integer, X.25 returns the status of the LUN as a result of the special condition.

RETURN VALUE

If the **ioctl** is successful, 0 is returned. The return value from X.25 (found in the integer pointed to by arg) is a valid LUN number; -1 indicates that no LUN is available.

DESCRIPTION

The IOX25STR command may be used by an application in data mode to determine an exceptional condition when it receives a SIGBAND signal. The SIGBAND signal is sent to the application upon reception of a Clear, Reset, Interrupt, Interrupt Confirmation, or special data confirmation (D-bit) packet when the Gateway is operating in data mode.

IOX25STR values are listed below.

Packet	4-Byte Answer Values 0	 1	 2	 3
Clear	0x13	Cause field	Diagnostic field	0, if standard clear, 1 if data to read
Reset	0x1B	Cause field	Diagnostic field	Not used
Interrupt	0x23	Interrupt data	Flag	Not used
Interrupt	0x27	Not used	Not used	Not used
Data	0x0a	Confirmation data	Not used	Not used
No event	0x00	Not used	Not used	Not used

NOTE

When a Clear packet is received on a LUN operating in data mode, the LUN is automatically returned to packet mode by X.25. If byte 3 of the **ioctl** argument is equal to (1), it indicates that the Clear packet includes some user data. The application is responsible for issuing a **read** call on the LUN to retrieve the data.

IOX25UP

SYNOPSIS

```
ioctl(fd, IOX25UP, &arg);
int fd;        /* file descriptor to SERVER LUN */
int arg;       /* int for returning free LUN */
```

where:

fd — is the file descriptor to the SERVER LUN.

IOX25UP — is defined as (('X'<<8) | 18).

arg — is a pointer to an integer containing the line number to be brought up. (No initialization is necessary.)

RETURN VALUE

If the **ioctl** is successful, 0 is returned. The return value from X.25 (found in the integer pointed to by arg) is a valid LUN number; -1 indicates that no LUN is available.

DESCRIPTION

The IOX25UP command allows the application to logically start an X.25 line. It directs Level 2 (Frame) to start transmitting SABMs.

The IOX25UP command is issued only on a SERVER LUN.

The IOX25UP command return value, in the integer pointed to by arg, is 1 if the operation is completed and 0 if the operation is pending, that is, if the frame level could not yet establish the line.

NOTE

The IOX25UP command should be issued only by administrative utilities.

IOX25VST

SYNOPSIS

```
ioctl(fd, IOX25VST, &arg);
int fd;        /* file descriptor to G2 LUN */
int arg;       /* int for returning free LUN */
```

where:

fd — is the file descriptor to the G2 LUN.

IOX25VST — is defined as (('X'<<8) | 7).

arg — is a pointer to an integer containing the LUN number for which the status is to be checked. The LUN number may be 0 if the IOX25VST command is issued on the USER LUN. (No initialization is necessary.)

RETURN VALUE

If the **ioctl** is successful, 0 is returned. The return value from X.25 (found in the integer pointed to by arg) is a valid LUN number; -1 indicates that no LUN is available.

DESCRIPTION

The IOX25VST command is used to obtain the status of a specific virtual circuit. There is a one-to-one relationship between LUNs and VCs that allows the LUN number argument to be 0 when issued on the USER LUN. This also allows the IOX25VST command to be issued on the SERVERLUN to find the state of any given VC/LUN.

The return value from the IOX25LST command contains 3 different values when it is successful. The format of the integer pointed to by the argument is shown in Figure 5-4.

The IOX25VST command can be issued on the SERVER LUN to request the status of the VC/LUN specified in the argument. If the IOX25VST is issued on a USER LUN, the value of the integer pointed to can be 0.

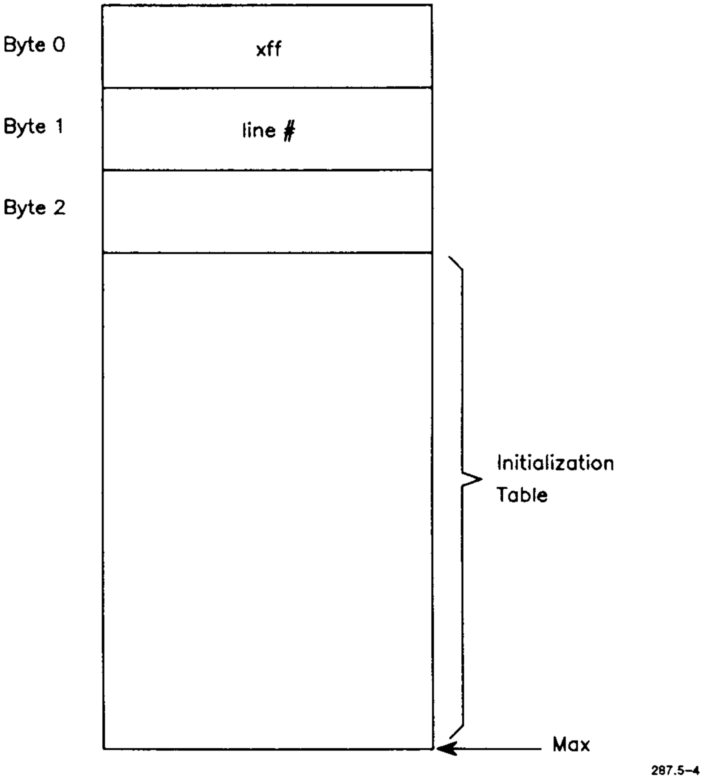


Figure 5-4. IOX25VST Return Value Format

where:

Line — is the line number this virtual circuit (VC) is associated with.

State — is the state of the VC. Possible values are (0) for disconnected or (1) for connected.

IOX25VST (continued)

VC is the logical channel number (LCN) associated with this VC/LUN.

IOX25WRITE

SYNOPSIS

```
ioctl(fd, IOX25WRITE, &arg);
int fd;         /* file descriptor to SERVER LUN */
int arg;        /* int for returning free LUN */
```

where:

fd — is the file descriptor to the SERVER LUN.

IOX25WRITE is defined as (('X'<<8) | 15).

arg — is a pointer to an integer. (No initialization is necessary.)

RETURN VALUE

If the **ioctl** is successful, 0 is returned. The return value from X.25 (found in the integer pointed to by arg) is a valid LUN number; -1 indicates that no LUN is available.

DESCRIPTION

The IOX25WRITE command allows an application to determine if a **write** operation will block. Since the **write** system call blocks, the IOX25WRITE command allows you to determine whether it will be blocked.

On return, the integer addressed by arg contains the number of open positions currently held by X.25 in the outgoing window of Level 3.

6 TROUBLESHOOTING

Chapter 6 provides a complete list of device driver, x25 Library, and x25d error codes and messages defined by the Gateway.

ERROR MESSAGES

Device Driver and Library Error Codes

The error codes listed in this section are returned by the Gateway's device driver in the following global variable:

errno	(defined in the CTIX library, **libc.a**)

It is available to the application program by including the **/usr/include/errno.h** file in the program.

The error codes returned by the X.25 Library are in the following global variables:

x25errno	(error code internal to the x25 Library)
xoserrno	(CTIX error code returned by the device driver)

The following are values for errno and oserrno:

Error Code	Possible Causes
ENXIO	Trying to download the initialization table more than once. LUN number greater than the specified maximum. Initialization error; port may be in use.
EL3HLT	Trying to access the Gateway without downloading the initialization table.
EIO	The Gateway did not successfully complete the request due to any of the following errors: - LUN is already open. An attempt was made to open an existing LUN. - LUN number out of range. The LUN number must be within the range of minor devices (see **/dev/x25**). - Bad adapter number. Only X.25 or X.29 (values 0 or 1, respectively) is allowed. - Bad line number. The line number requested does not exist or is not configured. - LUN is closed. An attempt was made to **read**, **write**, or issue an **ioctl** on a LUN that was already closed. - Outstanding **read** on LUN. IOX25PKT or IOX25DATA was issued when a **read** call was pending.

- Outstanding **write** on LUN. IOX25PKT or IOX25DATA was issued when a **write** call was pending.

- A **write** is done when the LUN is not in **write** state.

- No buffer pool for writing.

- No buffer for writing.

- Bad **ioctl** type. An **ioctl** command was issued that is not defined or recognized by the driver.

- Bad command from driver. A command was issued that is not defined or recognized by the driver.

- SERVER LUN is not open. **Read**, **write**, **ioctl**, or **close** was performed on this SERVER LUN when the SERVER LUN was not open.

- Link level is down. When an operation that required the operational link level is done, the link level is down.

- Invalid byte count on packet **read**.

- Invalid byte count on packet **write**.

- Link level already up. Attempting to bring up link level when it is already up.

- Modem I/F off.

- Modem I/F already on.

- **Write** invalid on this LUN (SERVER LUN).
- Bad **ioctl** code for this LUN. An **ioctl** that is only allowed on the SERVER LUN.
- **Read** invalid on this LUN (INIT LUN).
- Operation already in progress.
- Illegal DTE address format.
- Illegal packet type for **write**.
- Virtual circuit is cleared. Attempting to send data on a cleared VC.

The following are values for x25errno:

Function Call	**Value**	**Error Code Description**
x25open	1	Can not open the SERVER LUN.
	2	Can not get the free LUN.
	3	Can not close the SERVER LUN.
	4	Can not get the free LUN.
	5	Can not open the USER LUN.
	6	Can not define adapter and line number.
	7	Can not define the associated LCN number.

Function Call	Value	Error Code Description
x25call	8	Bad address - longer than 15 digits.
	9	Bad address - non-numeric characters in the address.
	10	Error in **write**.
	11	Error in **read**.
	12	CLEAR received instead of CALL ACCEPT.
	13	UNKNOWN TYPE received instead of CALL ACCEPT.
	14	Can not set the DATA MODE.
x25accept	15	Error in **read**.
	16	The packet **read** was not an Incoming Call packet.
	17	Error in **write**.
	18	Can not set the DATA MODE.

Figure 6-1 shows an example of how error codes can be read by the application using the library to analyze the failure of the library call:

```
extern   int   oserrno,x25errno ;
main()   /* Application Program - Pseudocode */
{
  .
  .
  .
  .
  if (lib_call_error)
  {
  printf("X25 library call failed.  Error %d-%d
     \n",x25errno,oserrno);
  }
  .
  .
  .
  .
}
```

Figure 6-1. Library Call Failure Example

x25d ERROR CODES

All error codes logged by x25d in the **x25.log** file are in message form rather than numeric form.

An error that is returned because of inability to accept the incoming call displays the error codes from the x25 Library in the following format:

Incoming Call on Line <line_no> LCN <lcn> cleared due to error <x-y>

where:

x = x25errno
y = oserrno

APPENDIX A: CONCEPTS OF X.25

OVERVIEW

Before attempting to use the CTIX X.25 Network Gateway, it is helpful if you have some understanding of the X.25 protocol.

This appendix is not a tutorial. It is intended to complement the International Telephone and Telegraph Consulative Committee's (CCITT) recommendations for X.25. This appendix is also designed to clarify some of the terminology used throughout this manual.

X.25 is a layered protocol specification by the CCITT for peer-to-peer connection-oriented data communication over bit-synchronous lines. Figure A-1 shows an example of peer-to-peer communication.

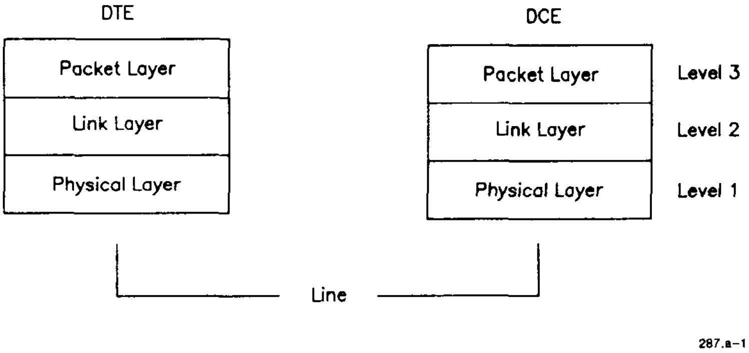


Figure A-1. Peer-to-Peer Communication

X.25 consists of the following three layers:

- Physical Layer
- Frame Layer
- Packet Layer

Each layer independently performs its own protocol with independent flow control. (See Figure A-1 for the layer designations.) The protocols of all three layers are handled entirely within the Gateway.

The phrase peer-to-peer indicates that each layer of the protocol logically communicates only with the peer of its own layer. For example, the local Packet Layer communicates only with its remote peer Packet Layer, the local Frame Layer communicates only with its remote peer Frame Layer, and so on. Of course, the actual flow goes up through the Physical Layer to the Frame Layer and on to the Packet Layer. For example, the DTE Packet Layer communicates with its remote peer by transmitting data and control information down through the Frame and Physical layers, over the physical line, and back up through the DCE Physical and Frame Layers until it reaches the Packet Layer. Figure A-1 also shows the communication path.

The Packet and Frame Layers each transmit their own data and control information to their respective peers independent of one another. Layer independence is accomplished by adding new header information as data penetrates another layer toward the physical line and stripping the header information as data penetrates another layer away from the physical line. Figure A-2 shows the protocol headers.

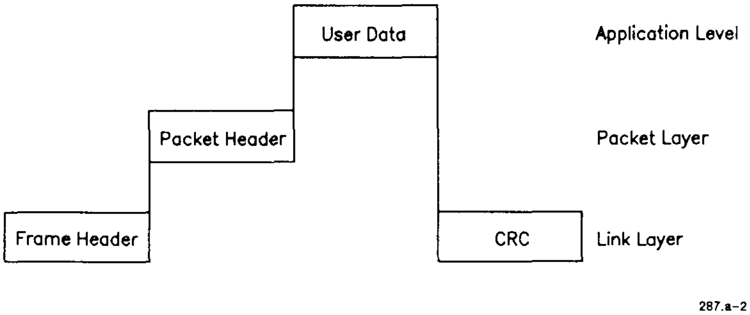


Figure A-2. Protocol Headers

Figure A-2 illustrates user data being transferred over X.25. The following steps describe the sequence of events:

1. The Packet header is added, and the Data Packet is transferred to the Link Layer.

2. Once the Data Packet reaches the Link Layer, the Frame header and Frame trailer are added.

Any information received from an upper layer is treated transparently. For example, the Frame Layer encapsulates any information from the Packet Layer as pure data, even though this information may consist only of control information and no user data. Control information from the Link Layer does not contain any Packet header information. Header and trailer usage is described further in the following sections.

Note that peer-to-peer communication is defined only between the DTE and the DCE. Routing and relaying, for example, through a public data network (PDN) is not within the scope of the X.25 recommendations. Therefore, the X.25 protocols do not provide end-to-end significance between the local DTE and the remote DTE. The Packet protocol, however, provides an option for a simple end-to-end delivery/confirmation protocol for user data. This protocol option must be implemented entirely at the user application level.

THE PHYSICAL LAYER

The Physical Layer handles the transmission and reception of data to and from the medium, in the case of the Gateway, a copper wire. The Physical Layer's responsibility also includes synchronizing a received bitstream into bytes and operating the handshake protocol of the modem.

An X.25 line is always a point-to-point connection. The two sides of the connection are designated as Data Circuit Equipment (DCE) and the Data Terminal Equipment (DTE). Each layer has its own DCE/DTE definition, but generally the DCE represents the network end of the connection, and the DTE is the subscriber end of the network (in this case, your S/MT). At the Physical Layer, the DCE is the modem that translates the digital patterns of the DTE into analog signals suitable for transmission over long distances over telephone lines. Today, connections to X.25 public data networks (PDNs) are required to be permanently connected telephone lines or leased lines.

THE FRAME LAYER

The Frame Layer (Level 2) protocol performs the function that sustains the point-to-point link between the DTE and the DCE. There is no multiplexing/demultiplexing of data streams or segmentation/blocking of data at the Frame Layer.

The Frame Layer uses the Link Access Procedure Balanced (LAPB) protocol, which is a subset of the High Level Data Link Control (HDLC) protocol. To some degree, LAPB may be categorized as a connection-oriented protocol. In other words, there are various initial steps involved in the transmission and reception of control information that must be performed before data can be transferred. The connection and disconnection phase may also be viewed as a well-defined synchronization point for error recovery. This action is also referred to as Link Reset and is the only Frame Layer action that has significance to the local Packet Layer; for example, a Frame Layer reset is synchronized with a Packet Layer restart.

The transmission unit of the Frame Layer is a frame. Each frame consists of a 2-byte Frame Header, an optional Information field, and a 16-bit Cyclic Redundancy Check (CRC). The CRC field is added for detection of bit errors that may have been introduced during transmission over poor-quality lines.

The 2-byte Frame Header is divided into an Address field and a Frame Control field. The Address field serves to designate a Frame as a command or a response. It also distinguishes commands and responses between the DTE and the DCE.

The Control field is encoded for the following Frame types:

- Unnumbered Frames
- Supervisory Frames
- Information Frames

Each one of these Frame types fulfills a specific protocol purpose as described in the following sections.

UNNUMBERED FRAMES

Unnumbered frames, sometimes referred to as U-Frames, are exchanged between the peer Frame Layers to connect and disconnect the link. During error recovery, U-Frames are used for signaling irrecoverable errors and for resetting the link (disconnect-connect sequence).

SUPERVISORY FRAMES

Supervisory Frames, sometimes referred as S-Frames, are used to perform data flow control on the link and to control the integrity of the received Information Frames. The S-Frames also carry acknowledgments for received Information Frames.

INFORMATION FRAMES

Information Frames, sometimes referred to as I-Frames, carry the packets to and from the Packet Layer. The packet information is in the Information field of the I-Frame and is treated transparently as data by the Frame Layer.A sequence number for sequentially numbering all I-Frames being sent and a sequence number that may be used for carrying an acknowledgment of received I-Frames is encoded into the Control field.

The Frame Layer protocol operates with a transmit and receive window. For example, a specified number of I-Frames corresponding to the window size may be transmitted before an acknowledgment is required, and some number of I-Frames may be received before an acknowledgment is sent.

If the integrity of the I-Frame sequence is lost by corrupted transmission, the protocol automatically attempts to restore the integrity by retransmitting the I-Frames that are in error. If this procedure is unsuccessful, built-in retries and timeouts may eventually lead to recovery. If all fails, a Link Reset is performed.

THE PACKET LAYER

The Packet Layer (Level 3) protocol is quite similar to the Frame Layer protocol, except that it multiplexes several user data streams into one link. The protocol is performed independently for each data stream.

The multiplexing channels are called logical channels. Each logical channel is individually numbered. A maximum of 4,097 Logical Channels may exist between a DTE and a DCE. In general, the number of logical channels is less than 100. The range and types of logical channels must be agreed upon between the DTE and the DCE. The following is a list of possible logical channel types available from a PDN:

- o incoming
- o outgoing
- o two-way

The logical channels exist only between the DTE and the DCE and may be used as soon as the packet Layers exchange the Restart packet and the Restart Confirmation packet. Restart is synchronized with the Frame Layer connection establishment and effects a reset at the Packet Layer when issued during normal data transfer. When a Link Reset is issued, there is a high probability of either data loss or duplication.

Although the X.25 recommendation is universally accepted, one major difference among PDNs is in the use of logical channel number 0. In the United States, for example, logical channel number 0 is allocated for the exchange of Restart Request and Restart Confirmation packets; whereas on some European PDNs, Restarts may occur on any logical channel.

The transmission unit of the Packet Layer is called a packet. There are packet types similar to those of the Frame Layer for acknowledgment, flow control, and error recovery. Packets used in connection establishment are different because they carry the address of the destination host. The Packet header has a field for the logical channel number (LCN) and the Packet type.

When your application program wants to send data over the network, a connection must first be established to the remote DTE. This procedure is accomplished by transmitting a Call Request packet to the DCE. The DCE must then return a Call Connected packet indicating that the remote DTE accepted the call. Prior to sending the call, the DTE must choose a free logical channel on which tosend the call. This procedure is accomplished automatically in most implementations. The receiving DTE receives the call on a particular logical channel number (chosen by the DCE), and it must respond on the same logical channel number.

After the Call Request and Call Connected packets are exchanged by the DTE and DCE, a virtual circuit (VC) exists between the local and remote DTE. The VC continues to use the same logical channel number as long as it exists. The VC exists until a Clear Request or Clear Indication is sent by either the DTE or the DCE. When this occurs, the logical channel is freed up and may be used again in establishing a new virtual circuit. The logical channel number (LCN) being used by the local DTE/DCE may not necessarily be the same as the logical channel used by the remote DTE/DCE in the same virtual circuit.

When a virtual circuit is established, the DTE needs only to reference the LCN for sending Data packets to the remote DTE. Each virtual circuit uses one logical channel, and there may be as many virtual circuits as there are available logical channels.

APPENDIX B: PACKET MODE DATA FORMATS

Appendix B describes the packet mode data formats. This appendix contains the following information:

- a description of packet mode
- a description of the Call Request, Call Accept, Data, Reset Request, Interrupt, and Clear Request packet formats

PACKET MODE

The following information describes the format of data exchanged with the X.25 software when the Gateway is operating in packet mode. Each buffer of data has a 3-byte header field. Some header fields are managed by the X.25 software, and it is not necessary for you to supply the information. In the following sections, the available packet types and their formats are discussed.

As stated above, it is not necessary for you to supply all the X.25 fields in the packet header. When operating in packet mode, X.25 manages the following header fields:

- General Format Identifier
- Logical Channel Group Number
- Logical Channel Number
- Packet Sequence Numbers

You are responsible for managing the following fields:

- Q bit
- D bit
- M bit
- Packet Type Identifier

In the following descriptions of data formats, each bit is described as it should be written by an application. When a packet is read from X.25, the undefined bits (bits that are without a specific explanation in the description) may be set to any value and are ignored by the application.

The following section describes the packet header format. The packet header resembles the X.25 packet header for modulo 8. The format used here does not change when modulo 128 is used. Figure B-1 shows the packet format.

	7	6	5	4	3	2	1	0
Byte 0	Q	D	X	X	X	X	X	X
Byte 1	lcn							
Byte 2	Packet_type							

287.b-2

Figure B-1. Packet Format

CALL REQUEST PACKET

The Call Request packet is used to place an outgoing call and can be written only when no virtual circuit is set up or in progress on a LUN. The format is the same for an Incoming Call packet. Figure B-2 shows the Call Request Packet format.

0	D	0	0	0	0	0	0
0	0	0	0	0	0	0	0
0	0	0	0	1	0	1	1
Calling				Called			
Called Address							
Calling Address							
Fac_len							
Fac_field							
Call_data							

287.b-2

Figure B-2. Call Request Packet Format

D_bit — D_bit is set to (1) to verify the possible use of the D bit.

lcn — lcn is the logical channel number agreed upon between the DTE and DCE for a particular virtual circuit number. A range of numbers is assigned when you subscribe to the PDN.

Packet_type — Packet_type is specified as 0x0b.

Addr_len — Addr_len is 4 bits (upper) of calling address length and 4 bits (lower) of called address length. Each address length is within the range of 0 to 4.

Addr_field — Addr_field consists of the calling and called addresses. Addresses are BCD coded (4 bits per digit); if there is an odd number of digits, the last byte is padded with four 0 bits. (See CCITT X.25 Recommendations, section 6.2.1.3, for format information.)

Fac_len — Fac_len is the length of the facility field. Fac_len is expressed in bytes and must be specified within the range of 0 to 63. Fac_len is a mandatory field and must be present even if there is a 0 value in the field.

Fac_field — Fac_field represents the optional facilities field. (See CCITT X.25 Recommendations, section 6.2.1.5 and Chapter 7, for additional details).

Call_data — Call_data represents call user data. Its value can be 16 bytes unless Fast Select is selected. (See CCITT X.25 Recommendations, section 6.2.1.6, for format and usage.)

Reading a Call Request packet from the SERVER LUN is used only by x25d. In this case, the packet type field is replaced by the line number on which the incoming call arrived. Only Call Request packets are read from the SERVER LUN, so the packet type field is not required.

CALL ACCEPT PACKET

The Call Accept packet is used to accept an incoming call and to negotiate the flow control parameters present in the facilities field. The Call Accept packet can be written only on a LUN that is opened to accept an incoming call (LCN is different from 0xffff). For some networks (for example, Transpac), addr_len and addr_field are optional in Call Confirm packets. The format is the same for confirmation of an outgoing call.

Figure B-3 shows the format for a Call Accept Packet.

0	D	0	0	0	0	0	0
0	0	0	0	0	0	0	0
0	0	0	0	1	1	1	1
Calling				Called			
Called Address							
Calling Address							
Fac_length							
Fac_field							

287.b-3

Figure B-3. Call Accept Packet Format

D_bit D_bit is set to 1 to check the possible use of the D bit.

lcn lcn is the logical channel number agreed upon between the DTE and DCE for a particular virtual circuit number. A range of numbers is assigned when you subscribe to the PDN.

Packet_type Packet_type is specified 0x0f.

Addr_len Addr_len is 4 bits (upper) of calling address length and 4 bits (lower) of called address length.

Each address length is within the range of 0 to 15.

Addr_field

Addr_field consists of the calling and called addresses. Addresses are BCD coded (4 bits per digit); if there is an odd number of digits, the last byte is padded with four 0 bits. (See CCITT X.25 Recommendations, section 6.2.1.3, for format information.)

Fac_len

Fac_len is the length of the optional facility field. Fac_len is expressed in bytes and must be specified within the range of 0-63. (See CCITT X.25 Recommendations, section 6.2.2.4, for additional information.)

Fac_field

Fac_field represents the optional facilities field. (See CCITT X.25 Recommendations, section 1.2.2.5, for additional details.) The facilities present here are the negotiated parameters in response to the ones present in the Call Request packet.

Call_data

Call_data represents call user data. Its value is 0 to 128 bytes long in the case of Fast Select. (See CCITT X.25 Recommendations, section 6.2.1.6, for format and usage.) If Fast Select is not specified, Call_data can be up to 16 bytes. Call_data is usually not present.

DATA PACKETS

Data packets are used to exchange data sent or received on a LUN.

Figure B-4 shows the Data packet format.

<table>
<tr><td>Q</td><td>D</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td></tr>
<tr><td colspan="8">Conf _ data</td></tr>
<tr><td>0</td><td>0</td><td>0</td><td>M</td><td>0</td><td>0</td><td>0</td><td>0</td></tr>
<tr><td colspan="8">Data _ field</td></tr>
</table>

287.b-4

Figure B-4. Data Packet Format

Q bit — The Q bit is used and has one of the following values: 0 for normal data or 1 for an X.29 message.

D bit: — The D bit is used and has one of the following values: 0 for normal or 1 for end-to-end confirmation requested.

Conf_data — One byte of data may be received in the Conf_data field when the D bit (end-to-end confirmation) field was used to transmit a packet. This byte is returned in a special packet type. In this way, you know that the data sent in this packet is confirmed by the remote DTE and not just locally confirmed to the network DCE. Otherwise, the field is lcn.

M bit

The M bit is used and has one of the following values: 0 for normal (no additional packets) or 1 for part of a multipacket sequence (additional packets coming).

Packet_type

Packet_type is specified as 0x00. If the M bit is set, specify the Packet_type as 0x10.

Data_field

The maximum length of the Data_field should correspond to the default value set for the specified network. The maximum length can be negotiated to be equal to or less than the maximum values.

When the D bit (end-to-end confirmation) is used, you can place a byte of data in the Conf_data field. This byte is returned in a special packet type. In this way, you know that the data sent in this packet is confirmed by the remote DTE and not just locally confirmed to the network DCE. Using the Data_field slows down the transfer of data and is not recommended for routine use.

RESET REQUEST PACKET

A Reset packet is sent to you when the X.25 software detects the possibility of data loss. Data loss can be caused by a network or an X.25 problem. The most usual cause is a mismatch of window size at the packet level when window size is nonnegotiable. When it is received by you, the Reset packet has already been confirmed automatically by the software. Therefore, it is not necessary to confirm it from the user level.

Figure B-5 shows the format for a Reset Request packet.

0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0
0	0	0	1	1	0	1	1
Cause _ code							
Diag _ code							

287.b-5

Figure B-5. Reset Request Packet Format

Figure B-6 shows the format for a Reset Confirmation packet.

0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0
0	0	0	1	1	1	1	1

287.b-6

Figure B-6. Reset Confirmation Packet Format

You can initiate a reset procedure by sending a Reset Request packet.

Packet_type — Packet_type should be set as follows: 0x1b (Reset Request).

Cause_code — Cause_code indicates the cause of the reset. (See CCITT X.25 Recommendations, section 6.5.3.1, for coding and a more detailed explanation of this field.)

Diag_code — Diag_code provides additional information on the reset. (See CCITT X.25 Recommendations, section 6.5.3.2, for additional information.)

The cause and diagnostic fields are optional; however, both must be either present or absent.

INTERRUPT PACKETS

There are three types of packets that fall under the interrupt class; they are:

- o Interrupt Request
- o Data Confirmation
- o Interrupt Confirmation

The Interrupt Request and Data Confirmation packets each have 1 byte of data associated with them.

Figure B-7 shows the Interrupt Request packet format.

0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0
0	0	1	0	0	0	1	1
Data _ byte							

287.b-7

Figure B-7. Interrupt Request Packet Format

Figure B-8 shows the Data Confirmation packet format.

0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0
0	0	0	0	1	0	1	0
Data _ byte							

287.b-8

Figure B-8. Data Confirmation Packet Format

Figure B-9 shows the Interrupt Confirmation Packet format.

0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0
0	0	1	0	0	1	1	1

287.b-9

Figure B-9. Interrupt Confirmation Packet Format

Data_byte — Data_byte is 1 byte of data associated with Interrupt Request and Data packets.

The Interrupt Request packet may be sent or received and indicates an end-to-end interrupt condition. If you have received an Interrupt Request packet, it is not necessary to confirm it, as the X.25 software does this automatically. Therefore, the Interrupt Confirmation packet can only be received.

A Data Confirmation packet is received when a data packet that was sent using D bit = 1 is confirmed by the remote DTE. The data byte is returned to indicate which data packet is being confirmed. The Data Confirmation packet is not a standard CCITT X.25 Recommendations packet. D bit confirmation is indicated with the end-to-end significance of packet sequence numbers. Since you have no access to these numbers, a special packet type is provided to indicate the confirmation.

CLEAR REQUEST PACKET

The Clear Request packet is used to clear a virtual circuit that is set up or received by the application to indicate that the remote DTE or the network has cleared the call. Like the Reset packet, the X.25 software takes care of confirming the Clear Request packet.

Figure B-10 shows the Clear Request packet format

0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0
0	0	0	1	0	0	1	1
Cause _ code							
Diag _ code							

287.b-10

Figure B-10. Clear Request Packet Format

Figure B-11 shows the Clear Confirmation packet format.

0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0
0	0	0	1	0	1	1	1

287.b-11

Figure B-11. Clear Confirmation Packet Format

Packet_type — Packet_type should be set as follows: 0x13 (Clear Request).

Cause_code — Cause_code indicates the cause of the clear. (See CCITT X.25 Recommendations, section 6.2.3.1, for coding and further explanation of the cause_code field.)

Diag_code — Diag_code provides additional information on the clear. (See CCITT X.25 Recommendations, section 6.2.3.2, for further information regarding this field.)

If a Fast Select call is being cleared, there is an opportunity to send data with the Clear packet. In this case, there are additional fields after the mandatory cause and diagnostic fields. (See CCITT X.25 Recommendations, section 6.8.2.3, for additional information about Call Clearing with the Fast Select facility.) The additional optional fields include the following:

Addr_len — Addr_len is 4 bits (upper) of calling address length, and 4 bits (lower) of called address length. Each address length is within the range of 0 to 15.

Addr_field

Addr_field consists of the calling and called addresses. Addresses are BCD coded (4 bits per digit); if there is an odd number of digits, the last byte is padded with four 0 bits. (See CCITT X.25 Recommendations, section 6.2.1.3, for format information.)

Fac_len

Fac_len is the length of the facility field. Fac_len is expressed in bytes and must be specified within the range of 0 to 63. Fac_len is a mandatory field and must be present even if there is a 0 value in the field.

Fac_field

Fac_field represents the optional facilities field. (See CCITT X.25 Recommendations, sections 6.2.1.5 and Chapter 7, for additional details.) The Fast Select facility must be selected.

Clr_data:

Clr_data is data to be transmitted to the remote DTE.

APPDNDIX C: X.25 IOCTL SUMMARY

Appendix C lists the G2 LUN **ioctl** codes, SERVER LUN **ioctl** codes, and the USER LUN **ioctl** codes.

G2 LUN IOCTL CODES

ioctl	Description
IOX25VST	Get status information for a virtual circuit.
IOX25SHUT	Disable further call processing.
IOX25START	Establish the physical layer.
IOX25STOP	Shut down the physical layer.
IOX25UP	Start Level 2 (Frame).
IOX25DOWN	Stop Level 2 (Frame).
IOX25GETINIT	Read the current initialization table.

SERVER LUN IOCTL CODES

ioctl	Description
IOX25LUN	Get the number of an available LUN.
IOX25LST	Get line status information.
IOX25SHUT	Disable further call processing (Level 2).

USER LUN IOCTL CODES

ioctl	Description
IOX25MODE	Initializes line number and adapter (open).
IOX25OPN	Actual X.25 open of a LUN. Specifies LCN.
IOX25DATA	Switch to data mode.
IOX25PKT	Switch to packet mode.
IOX25STR	Status: Retrieve an unexpected event.
IOX25NET	Retrieve Network Identification.
IOX25READ	Test if data is available for read.
IOX25WRITE	Test whether nonblocking write is possible.

APPENDIX D: KEYBOARD TABLES

This appendix lists the preconfigured terminal types supported with this release of the CTIX X.25 Network Gateway.

Since the CTIX X.25 Network Gateway supports several terminal types (and, therefore, several different keyboards), this guide refers to keys by virtual key names. For example, where this manual reads **Enter**, you press **Go** or **Linefeed**, depending on the type of terminal you use.

This appendix provides key tables for the standard preconfigured terminals. Use the appropriate key table to determine the actual keys on your keyboard that match the virtual key names given in this guide.

If you use a nonstandard terminal that is not configured with special keys, you can type key sequences to perform many functions. Refer to the "Generic Keyboard" discussion, later in this appendix, for a list of key sequences to type for the virtual keys.

PRECONFIGURED TERMINAL TYPES

This release of the CTIX X.25 Network Gateway supports the following preconfigured terminals:

PT and GT (with RS-422 and RS-232-C
connections)
Wyse 85
DEC VT-100 compatibles
DEC VT-102
DEC VT-220
Link (S/T2)
Freedom 100

GENERIC KEYBOARD

The following table translates virtual key names into keystroke sequences for a generic (dumb) terminal keyboard.

WHEN YOU SEE THIS KEY NAME	TYPE THIS KEY SEQUENCE
Back	Escape bw
BackSpace	BackSpace
BackTab	Escape Tab
Begin	Escape bg
Break	Break
Cancel	Control-X
Clear	Escape ce
ClearLine	Escape ci
Close	Escape cl
Command	Control-C
Copy	Escape cp
Create	Escape cr
Delete	Escape dl
Delete Character	Delete
Down	Escape dn
End	Escape en
Enter	LineFeed
Exit	Control-D
Escape	ESC
Find	Escape fi
Forward	Escape fw
F1	Escape 1
F2	Escape 2
F3	Escape 3
F4	Escape 4
F5	Escape 5
F6	Escape 6

F7	Escape 7
F8	Escape 8
F9	Escape 9
F10	Escape 0
Help	Escape ?
Home	Escape hm
InputMode	Escape im
Mark	Escape mk
Message	Escape ms
Move	Escape mv
Next	Escape nx
Open	Escape op
Options	Escape ot
Page	Escape pg
PF1	Escape f1
PF2	Escape f2
PF3	Escape f3
PF4	Escape f4
PF5	Escape f5
PF6	Escape f6
PF7	Escape f7
PF8	Escape f8
PF9	Escape f9
PF10	Escape f0
PF11	Escape f-
PF12	Escape f=
Previous	Escape pv
Print	Escape pr
Redo	Escape ro
Ref	Escape re
Replace	Escape rp
Restart	Escape rs
Resume	Escape rm
Return	Return
Save	Escape sa

Scroll Up	Escape ru
Scroll Down	Escape dn
Select	Escape sl
Shift-Back	Escape BW
Shift-Beginning	Escape BG
Shift-Cancel	Escape CX
Shift-ClearLine	Escape CI
Shift-Command	Escape CM
Shift-Copy	Escape RP
Shift-Create	Escape CR
Shift-Delete Character	Escape DC
Shift-End	Escape EN
Shift-Exit	Escape EX
Shift-Find	Escape FI
Shift-Forward	Escape FW
Shift-Help	Escape HL
Shift-Home	Escape HM
Shift-InputMode	Escape NJ
Shift-Message	Escape MS
Shift-Move	Escape MV
Shift-Next	Escape NX
Shift-Options	Escape OT
Shift-Page	Escape PG
Shift-Previous	Escape PV
Shift-Print	Escape PR
Shift-Redo	Escape RO
Shift-Replace	Escape RP
Shift-Resume	Escape RM
Shift-Save	Escape SV
Shift-Undo	Escape UD
Shift-F1	Escape !
Shift-F2	Escape @

Shift-F3	Escape #
Shift-F4	Escape $
Shift-F5	Escape %
Shift-F6	Escape Control
Shift-F7	Escape &
Shift-F8	Escape *
Shift-F9	Escape (
Shift-F10	Escape)
Undo	Escape ud
Up	Escape up

PT/GT KEYBOARD

The following table translates the virtual key names used in this guide to actual keys on the PT/GT terminal keyboard.

WHEN YOU SEE THIS KEY NAME	TYPE THIS KEY SEQUENCE
Back	Left Arrow
BackTab	Control-Tab
Begin	Control-Up Arrow
Break	Cancel
Cancel	Cancel
Clear	Control-F7
ClearLine	Shift-Cancel
Close	Control-F6
Command	Control-C
Copy	Copy
Delete	Control-Delete
Delete Character	Delete
Down	Down Arrow
End	Control-Down Arrow
Enter	Go
Exit	Finish
Escape	Control-E
Forward	Right Arrow
F1	F1
F2	F2
F3	F3
F4	F4
F5	F5
F6	F6
F7	F7
F8	F8
F9	F9

F10	F10
Help	Help
Home	Shift-Up Arrow
InputMode	Overtype
Mark	Mark
Move	Move
Next	Shift-Right Arrow
Page	Next Page
Previous	Shift-Left Arrow
Print	Control-F4
Redo	Control-F2
Replace	Control-F3
Save	Control-F1
Scroll Up	Scroll Down
Scroll Down	Scroll Up
Shift-Back	Control-Left Arrow
Shift-Copy	Shift-Copy
Shift-Delete Character	Shift-Delete
Shift-Exit	Shift Finish
Shift-Forward	Control-Right Arrow
Shift-Help	Shift-Help
Shift-Home	Shift-Down Arrow
Shift-Move	Shift-Move
Shift-Page	Prev Page
Shift-F1	Shift-F1
Shift-F2	Shift-F2
Shift-F3	Shift-F3
Shift-F4	Shift-F4
Shift-F5	Shift-F5
Shift-F6	Shift-F6
Shift-F7	Shift-F7
Shift-F8	Shift-F8
Shift-F9	Shift-F9
Shift-F10	Shift-F10
Up	Up Arrow

VT-100 KEYBOARD

The following table translates the virtual key names used in this guide to actual keys on the VT-100 terminal keyboard or to keystroke sequences. Note that the + sign, as used below, indicates a second instance of a key mapping for a particular virtual key name: **+F1** is a second **F1** key.)

WHEN YOU SEE THIS KEY NAME	TYPE THIS KEY SEQUENCE
Back	Left Arrow
BackSpace	BackSpace
BackTab	Escape Tab
Begin	Escape bg
Break	Break
Cancel	Control-X
Clear	Escape ce
ClearLine	Escape ci
Close	Escape cl
Command	Control-C
Copy	Escape cp
Create	Escape cr
Delete	Escape dl
Delete Character	Delete
Down	Down Arrow
End	Escape en
Enter	LineFeed
Exit	Control-D
Escape	ESC
Forward	Right Arrow
F1	Escape 1
F2	Escape 2
F3	Escape 3
F4	Escape 4
F5	Escape 5

F6	Escape 6
F7	Escape 7
F8	Escape 8
F9	Escape 9
F10	Escape 0
+F1	Pf1
+F2	Pf2
+F3	Pf3
+F4	Pf4
Help	Escape ?
Home	Escape hm
InputMode	Escape im
Mark	Escape mk
Message	Escape ms
Move	Escape mv
Next	Escape nx
Open	Escape op
Options	Escape ot
Page	Escape pg
PF1	Escape f1
PF2	Escape f2
PF3	Escape f3
PF4	Escape f4
PF5	Escape f5
PF6	Escape f6
PF7	Escape f7
PF8	Escape f8
PF9	Escape f9
PF10	Escape f0
PF11	Escape f-
PF12	Escape f=
Previous	Escape pv
Print	Escape pr
Redo	Escape ro
Ref	Escape re

Replace	Escape rp
Restart	Escape rs
Resume	Escape rm
Return	Return
Save	Escape sa
Scroll Up	Escape ru
Scroll Down	Escape dn
Select	Escape sl
Shift-Back	Escape BW
Shift-Beginning	Escape BG
Shift-Cancel	Escape CX
Shift-ClearLine	Escape CI
Shift-Command	Escape CM
Shift-Copy	Escape RP
Shift-Create	Escape CR
Shift-Delete Character	Escape DC
Shift-End	Escape EN
Shift-Exit	Escape EX
Shift-Find	Escape FI
Shift-Forward	Escape FW
Shift-Help	Escape HL
Shift-Home	Escape HM
Shift-InputMode	Escape NJ
Shift-Message	Escape MS
Shift-Move	Escape MV
Shift-Next	Escape NX
Shift-Options	Escape OT
Shift-Page	Escape PG
Shift-Previous	Escape PV
Shift-Print	Escape PR
Shift-Redo	Escape RO
Shift-Replace	Escape RP

Shift-Resume	Escape RM
Shift-Save	Escape SV
Shift-Undo	Escape UD
Shift-F1	Escape !
Shift-F2	Escape @
Shift-F3	Escape #
Shift-F4	Escape $
Shift-F5	Escape %
Shift-F6	Escape Control
Shift-F7	Escape &
Shift-F8	Escape *
Shift-F9	Escape (
Shift-F10	Escape)
Undo	Escape ud
Up	Up Arrow

Scroll Up	Escape ru
Scroll Down	Escape dn
Select	Escape sl
Shift-Back	Escape BW
Shift-Beginning	Escape BG
Shift-Cancel	Escape CX
Shift-ClearLine	Escape CI
Shift-Command	Escape CM
Shift-Copy	Escape RP
Shift-Create	Escape CR
Shift-Delete Character	Escape DC
Shift-End	Escape EN
Shift-Exit	Escape EX
Shift-Find	Escape FI
Shift-Forward	Escape FW
Shift-Help	Escape HL
Shift-Home	Escape HM
Shift-InputMode	Escape NJ
Shift-Message	Escape MS
Shift-Move	Escape MV
Shift-Next	Escape NX
Shift-Options	Escape OT
Shift-Page	Escape PG
Shift-Previous	Escape PV
Shift-Print	Escape PR
Shift-Redo	Escape RO
Shift-Replace	Escape RP
Shift-Resume	Escape RM
Shift-Save	Escape SV
Shift-Undo	Escape UD
Shift-F1	Escape !
Shift-F2	Escape @

Shift-F3	Escape #
Shift-F4	Escape $
Shift-F5	Escape %
Shift-F6	Escape Control
Shift-F7	Escape &
Shift-F8	Escape *
Shift-F9	Escape (
Shift-F10	Escape)
Undo	Escape ud
Up	Escape up

WYSE-85 KEYBOARD

The following table translates the virtual key names used in this guide to actual keys on the Wyse-85 keyboard or to keystroke sequences to be typed from the Wyse-85 keyboard.

WHEN YOU SEE THIS KEY NAME	TYPE THIS KEY SEQUENCE
Back	<X]
BackSpace	BackSpace
BackTab	Escape Tab
Begin	Escape bg
Break	Break
Cancel	Control-X
Clear	Escape ce
ClearLine	Escape ci
Close	Escape cl
Command	Control-C
Copy	Escape cp
Create	Escape cr
Delete	Remove
Delete Character	Shift-<X]
Down	Escape dn
End	Escape en
Enter	Do
Exit	Control-D
Escape	ESC
Find	Find
Forward	Escape fw
F1	F6
F2	F7
F3	F8
F4	F9
F5	F10

F6	F17
F7	F18
F8	F19
F9	F20
F10	Escape 0
Help	Help
Home	Home
InputMode	Insert-Here
Mark	Select
Message	Escape ms
Move	Escape mv
Next	Escape nx
Open	Escape op
Options	Escape ot
Page	Escape pg
PF1	Escape f1
PF2	Escape f2
PF3	Escape f3
PF4	Escape f4
PF5	Escape f5
PF6	Escape f6
PF7	Escape f7
PF8	Escape f8
PF9	Escape f9
PF10	Escape f0
PF11	Escape f-
PF12	Escape f=
Previous	Escape pv
Print	Escape pr
Redo	Escape ro
Ref	Escape re
Replace	Escape rp
Restart	Escape rs
Resume	Escape rm
Return	Return

LINK (S/T2) KEYBOARD

The following table translates the virtual key names used in this guide to actual keys on the LINK (S/T2) keyboard or to keystroke sequences. Note that the + sign, as used below, indicates a second instance of a key mapping for a particular virtual key name: **+Cancel** is a second **Cancel** key.)

WHEN YOU SEE	TYPE THIS
THIS KEY NAME	KEY SEQUENCE
Back	Left
BackSpace	
BackTab	Control Tab
Begin	Control Up
Break	Control]
Cancel	Control-X
+Cancel	F20
Clear	Control F7
ClearLine	Shift F20
Close	Control F6
Command	Control-C
+Command	F13
Copy	F11
Create	***
Delete	Remove
Delete Character	<X
Down	Down
End	Control-Down
Enter	Do
Exit	Control-D
+Exit	F14
Escape	ESC
Find	
Forward	Right

F1	F1
F2	F2
F3	F3
F4	F4
F5	F5
F6	F6
F7	F7
F8	F8
F9	F9
F10	F10
Help	Help
Home	Shift-Up
InputMode	Insert-Here
Mark	Select
Message	
Move	F12
Next	Shift-Right
Open	Control-F5
Options	
Page	Next Page
Previous	Shift-Left
Print	F19
Redo	Control-F2
Ref	
Replace	Control-F3
Restart	
Resume	
Return	
Save	Control-F1
Scroll Up	Prev Screen
Scroll Down	Next Screen
Shift-Back	Control-Left
Shift-Beginning	
Shift-Cancel	

Shift-ClearLine	
Shift-Command	
Shift-Copy	Shift-F11
Shift-Create	
Shift-Delete Character	Control-<X
Shift-End	
Shift-Exit	Shift-F14
Shift-Find	
Shift-Forward	Control-Right
Shift-Help	Shift-Help
Shift-Home	Shift-Down
Shift-InputMode	
Shift-Message	
Shift-Move	Shift-F12
Shift-Next	
Shift-Options	
Shift-Page	Prev Page
Shift-Previous	
Shift-Print	
Shift-Redo	
Shift-Replace	
Shift-Resume	
Shift-Save	
Shift-Undo	
Shift-F1	Shift-F1
Shift-F2	Shift-F2
Shift-F3	Shift-F3
Shift-F4	Shift-F4
Shift-F5	Shift-F5
Shift-F6	Shift-F6
Shift-F7	Shift-F7
Shift-F8	Shift-F8
Shift-F9	Shift-F9

Shift-F10	Shift-F10
Undo	Control-U
Up	Up

GLOSSARY

alphanumeric. An alphanumeric string is made up of letters (alphabetic) and numbers (numeric).

application program. An application program is a computer program that performs a data processing task rather than a control function. In CTIX it also denotes a program running in user space rather than in the kernel; that is, the application is being swapped.

ASCII (American Standard Code for Information Interchange). ASCII is a control and graphic character set consisting of 7-bit coded characters (8 bits including parity check), used for information interchange between data communications systems.

ASCII terminal. An ASCII terminal (CRT) is a terminal consistent with the American Standard Code for Information Interchange (ASCII), commonly used for serial data transmission.

binary digit (bit). A bit is a unit of information that designates one of two possible states, represented by either 1 or 0.

bit. See **binary digit.**

bit rate. Bit rate is the rate at which bits (binary digits) are transmitted over a communications path, normally expressed in bits per second (bps). The bit rate is not to be confused with the data signaling rate (baud), which measures the rate of signal changes being transmitted.

bit stream. Bit stream refers to a continuous series of bits being transmitted on a transmission line.

block. A block is a set of contiguous bits and/or bytes that make up a definable quantity of information.

blocking. Blocking is a procedure that indicates the detention of an unsatisfied request until such time as the request can be satisfied. For example, a Read request will block until a Data Packet is received and placed into the Read buffer.

bps. Bits per second. Bps is the basic unit of data communications transmission rate measurement, generally referring to the number of information bits transmitted per second.

buffer. A buffer is a storage area for a block of data.

byte. A byte is a consecutive sequence of bits operated upon as a unit.

call (virtual call). In X.25 a call is the establishing of a virtual circuit by transmitting a Call Request and receiving a Call Connected.

Call Accept packet. The Call Accept packet is used to acknowledge an incoming call and to negotiate the flow control parameters present in the Facilities field.

call collision. A call collision is a conflict that occurs at a DTE/DCE interface when both sides transmit a Call Request simultaneously.

Call Request packet. A Call Request packet is used to place an outgoing call and can be transmitted only when there is no virtual circuit set up or currently in progress on a LCN.

carrier. A carrier is a continuous wave that is modulated by an information-bearing signal. In this case, it is the Data Carrier Detect signal on the modem.

CCITT (International Telephone and Telegraph Consultative Committee). CCTI is an international organization, including telecommunications carriers, that is responsible for developing telecommunications standards by making recommendations.

channel. A channel is a data communications path.

circuit switching network (CSN). A circuit switching network is a type of data communications network where a physical and exclusive line is maintained between two communicating devices for the call duration. An all-digital, circuit switching network is often referred to as an X.21 network.

clear packet processing. Clear packet processing is generated by an application to clear the associated virtual circuit.

Clear Request packet. A Clear Request packet is transmitted to clear a virtual circuit that is set up, or if received by an application, indicates that the remote DTE or the network has cleared the call.

clock. Clock is the entity for the bit-rate timing on the physical line. The clock signal can be generated either internally by the DTE or externally by the DCE modem.

close. **close** is one of the five basic system calls in the Programmatic Interface. The **close** system call deallocates a LUN previously being used by an application program.

coaxial cable. A coaxial cable is a two-conductor wire with longitudinal axes coincident. The cable has a shield against noise around a signal-carrying conductor.

communications. Communications is the transmission of information between two points of origin and reception.

communications line. A communications line is (1) any physical link, such as a wire or a telephone circuit, that connects one or more remote terminals to a communications control unit or connects one communications control unit to another; (2) the portion of a circuit external to the apparatus, consisting of the conductors connecting a telegraph or telephone set to the exchange or connecting two exchanges; (3) the group of conductors on the same overhead routing or in the same cable.

communications line controller. A communications line controller is a hardware unit that performs line control functions with the modem.

conditioning. Conditioning is a method of tuning a communications line to a particular modem. Conditioning is performed by the common carrier to reduce the likelihood of line transmission errors.

configuration. Configuration is a group of machines, devices, or programs that constitute a data processing system.

data. In relation to X.25, data is a collection of bits to which meaning is assigned by higher protocols in order to convey information to users.

data communications. Data communications is the transfer of data between a data source and a data sink using one or more data links according to a designated protocol.

data link. A data link is (1) the physical means of connecting one location to another for the purpose of transmitting and receiving data; (2) an assembly of those parts of two data terminal equipments (DTEs) that defines protocol, along with its interconnecting data circuit.

D bit (delivery confirmation bit). Used in the X.25 protocol, the setting of the D bit in Data packets indicates whether delivery acknowledgment of the packet is required from the local DCE or from the remote DTE. It therefore allows the choice between local and end-to-end acknowledgment.

DCE (data circuit-terminating equipment). The network side of the user-to-network interface.

dedicated line. A dedicated line is a communications line that is leased from a common carrier and used for communications between two points. It is also called a leased or private line.

default packet size. The default packet size is the size of packet to use unless another size is negotiated. The value is configurable.

default window size. The default window size (both transmit and receive) is to be used by the DTE, unless another size is negotiated.

device. A device is a terminal, printer, disk, tape, or other input/output medium that can be attached to a system. A device can be a physical unit or a logical unit.

device driver. A device driver is a program that controls a specified device. In CTIX, a device driver resides in the kernel.

device numbers. Device numbers are maintained for use in referencing devices. See **major** and **minor device numbers.**

dial line. A dial line is a communications line that is dialed, as opposed to dedicated or leased.

digital. The term digital is used to describe devices and communications systems that operate using discontinuous signals that change from one state to another in discrete steps.

DTE (data terminal equipment). DTE is the device at the user's side of a user-to-network interface. In the case of X.25, this is the computer system (packet mode DTE).

duplex. The term duplex is used to describe communications circuits or operations that permit simultaneous two-way messages or information to be passed between two points.

facilities. Facilities are optional services offered by a packet switching network's administration and requested by the user either at the time of subscription for network access or negotiated at the time a call is made. Facilities are encoded fields in the Call Request packet and Call Accept packet.

Fast Select. Fast Select is an optional packet switching network facility by which user data may be transmitted as part of the control packet that establishes or terminates a virtual connection.

file. A file is a set of related records treated as a unit. A CTIX file is a collection of data records that may be different from one another and may or may not have any relationship to one another.

flag. A flag is an interframe fill character (value 7E in hexadecimal).

flow control. Flow control is a procedure for controlling the rate of transfer of packets (Level 3 flow control) and frames (Level 2 flow control) between two nominated points in a network, usually the DTE and the DCE.

format. Format refers to the structure of a message or data such that specific controls or data can be identified by their position during processing.

frame. A frame is a sequence of bits generated by the Level 2 protocol of X.25 that consists of an address field, a control field, a frame check sequence and (in information frames only) an information field containing data.

frame check sequence (FCS). A frame check sequence is a sequence of bits generated by X.25 at Level 2 that forms part of the frame and guarantees the integrity of the frame's content.

full duplex (FDX). Full duplex refers to the capability of transmitting data in two directions at one time. The term is also used to define a four-wire circuit.

gateway. The gateway is the loadable character device driver comprising the X.25 protocol state machines. It establishes a point of connection between two dissimilar systems, for example, between your system and a public data network.

half duplex (HDX). Half duplex refers to a communications line consisting of two wires or to a protocol capable of transmitting in only one direction at a time.

Incoming Call. An incoming call is a call request packet when it is being received, in other words, when it is transmitted by the remote end as a Call Request packet.

Information frame (I-frame). An Information frame is a frame that contains an information field and, therefore, carries user data. The data consists of Level 3 packets.

interface. An interface defines the rules by which interaction occurs between two different systems or processes.

invoke. To invoke is to activate a procedure at one of its entry points.

ioctl. **ioctl** is a CTIX system call designated for controlling a driver, as opposed to the **read** and **write** calls.

I/O (Input/Output). I/O is the process of moving information between a central processing unit and peripheral devices. It may also refer to the particular peripheral hardware that is used.

ISO (International Standards Organization). The ISO is an international federation of national standards organizations involved in developing international standards, including communication standards.

layer. A layer is a defined, functional separation of the X.25 protocols. The X.25 protocol consists of three layers: physical, frame, and packet.

leased line. A leased line is a communications line for voice and/or data leased from a communications carrier. It is also known as a private line or a dedicated line.

Library. A library is a collection of objects which may be linked into an application program.

line control. Line control is a control program used to perform data communications functions over network lines. It consists of both handshaking and line control functions that move the data between transmission and receiving stations.

line noise. Line noise is noise that originates in a transmission line.

logical channel. A logical channel is a bidirectional logical association between a DTE and a DCE. A channel is apparent (and its identification significant) only at the respective local DTE/DCE interface.

logical channel number (LCN). Logical channel numbers are numbered by a plan that allows the DTE/DCE interface to recognize specific LCs. The numbers must coincide on the DTE/DCE interface.

logical unit number (LUN). The logical unit number is the equivalent of the Gateway's minor devices.

major device number. A major device number selects the appropriate driver to handle a protocol discipline (for example, X.25, SNA, etc.).

maximum packet size. The maximum packet size may be negotiated with the Gateway. The specific value is configurable.

maximum window size. The maximum window size may be negotiated with the Gateway. The specific value is configurable.

M bit (more data bit). Setting the M bit in a Data packet indicates that at least one or more Data packets are required to complete a message of contiguous data.

minor device number. A minor device number denotes a subdevice of a driver.

modem. Modem is a contraction for modulator/demodulator unit. The unit converts digital information from a terminal or computer port into an analog carrier signal to be transmitted over an analog line.

network. A network, within the context of this manual, refers to a configuration of data processing products, such as processors, controllers, PDNs, and terminals, established and operated by users for the purpose of processing data and exchanging information.

network gateway. A network gateway is a software product that provides a transport service allowing application subsystems to communicate with or utilize a PDN.

network gateway driver. A network gateway driver is a program, utility, or procedure that directs the operation of a peripheral device under CPU control.

network ID. Public data networks (PDNs) are identified by a DNIC (data network identification code). It is generally the first four digits of the network address.

node. A node is a terminal computer or peripheral device that provides a switching or terminating point in a network.

open. **open** is a CTIX system call that allocates and initializes a driver (subdevice), in this context, a LUN.

packet. A packet is a block of data with a fixed length; it is the unit of information exchanged by X.25 at Level 3. There are Data packets and various control packets. A packet type is identified by the encoding of its header.

parameter. A parameter is (1) a variable that is given a constant value for a specified application, which may denote the application; (2) a name in a procedure used to refer to any argument passed to that procedure.

PDN (public data network). A PDN is a data communications network whose services are available to any user willing to pay for them. Most PDNs use packet switching techniques, but some use circuit switching.

physical unit (PU). A physical unit (PU) is the entity responsible for controlling a node's resources.

point-to-point. Point-to-point refers to a communications line connected directly from one point to another point, as opposed to multipoint lines.

port. A port is that part of a data processor dedicated to a single data channel for receiving data from or transmitting data to one or more remote external devices.

Proc_table. The **proc_table** is a file in the **/usr/spool/x25** directory that defines user applications to be executed on incoming calls on a specific subaddress.

protocol. A protocol is the specification of the rules whereby data communications, over a data link, are performed in terms of the particular transmission code, transmission mode, and control and recovery procedures.

PSN (packet switching network). A PSN is any data communications network using packet switching techniques whereby data is broken up into packets at the source interface and disassembled back into a data stream at the destination interface. A public PSN offers the service to any paying customer.

PVC (permanent virtual circuit). A PVC is a permanent logical association between two physically separate DTEs that does not require call set-up or clearing procedures.

Q Bit (qualified bit). When set in DATA packets the Q bit signifies that the packet's user data is an X.29 message.

read. **read** is a CTIX system call that to reads received data from a device driver.

remote host. A remote host is any computer with which communications are to take place. More than one remote host computer can be designated in a data communications network.

Reset Request packet. A Reset Request packet is used to reset the sequence numbers at the packet layer, which generally causes retransmission of data.

SABM (Set Asynchronous Balance Mode). The SABM unnumbered command is used to place the addressed DTE or DCE in the asynchronous balanced mode (ABM) information transfer phase.

SERVER LUN. The SERVER LUN is a special LUN allocated for providing services to x25d and applications.

sub-address. A sub-address generally consists of digits in the network address. Not all PDNs support a sub-address. The Gateway treats the last two digits in the address as the subaddress.

SVC (switched virtual circuit). An SVC is a temporary logical association between two physically separate DTEs that exists only for the duration of the data transfer. Call setup and call clearing procedures are required with an SVC.

Tl timer. Tl timer represents the time between frame retransmissions.

T3 timer. T3 timer represents the SABM retransmission after the number of N2 retransmissions.

T20 timer. T20 timer represents the time between restart packet retransmissions.

T21 timer. T21 timer represents the time between the transmission of Clear Request to a nonanswered call request.

T22 timer. T22 timer represents the time between Reset Request packet retransmissions.

T23 timer. T3 timer represents the time between Clear request retransmissions.

virtual call. In X.25 a virtual call is the communication of two DTEs using a virtual circuit identified by logical channel identifiers at the respective DTE/DCE interfaces. See **DTE** and **DCE**.

virtual circuit (VC). A virtual circuit is a logical association between two physically separate DTEs.

window. A window specifies the maximum number of outstanding (unacknowledged) packets or frames permitted between a DTE and DCE when local acknowledgment is used, and between a DTE and a remote DTE when end-to-end acknowledgment is used.

write. **write** is a CTIX system call generally used for transmitting user data through a driver.

X.25. X.25 defines the interface between a DTE and a DCE for packet-mode operation on a public data network (PDN).

X.25 Adapter. The X.25 adapter is a mode of operation in the Gateway designed for controlling and interfacing to the packet layer of the driver.

X.25 packet layer. The packet layer is the uppermost layer in the X.25 recommendations.

X.29. X.29 defines the interface for the exchange of control information and user data between a packet-mode DTE and a remote Packet Assembly/Disassembly (PAD) facility, over a packet switching network.

X.29 Adapter. The X.29 Adapter is a mode of operation in the Gateway designed for transparent use of the Gateway when the gateway is operating in data mode.

INDEX

Page numbers in **boldface** indicate the principal discussion of the topic.