Tandem NonStop (TM) and NonStop II (TM) Systems

AXCESS (TM)

Volume 2: X.25 Access Methods (X25AM)

ABSTRACT: This manual describes the X25AM method of accessing the various packet-switching networks. Also described is the X3PAD utility program, which allows interactive terminal users to directly access these networks.

PRODUCT VERSION: X25AM A04 (NonStop II systems)
X25AM E05 (NonStop systems)

OPERATING SYSTEM VERSION: GUARDIAN A04 (NonStop II systems)
GUARDIAN E05 (NonStop systems)

Tandem Computers Incorporated
19333 Vallco Parkway
Cupertino, California 95014

Part No. 82334 C00

April 1983
Printed in U.S.A.

The following are trademarks or servicemarks of Tandem Computers Incorporated:

AXCESS	BINDER	CROSSREF	DDL	DYNABUS
EDIT	ENABLE	ENCOMPASS	ENFORM	ENSCRIBE
ENVOY	EXCHANGE	EXPAND	GUARDIAN	INSPECT
NonStop	NonStop II	PATHWAY	PERUSE	TANDEM
TAL	TGAL	THL	TIL	TMF
TRANSFER	XRAY	XREF		

INFOSAT is a trademark in which both Tandem and American Satellite have rights.

HYPERchannel is a trademark of Network Systems Corporation.

IBM is a registered trademark of International Business Machines Corporation.

NEW AND CHANGED INFORMATION

This revision enhances the ITI and PTP protocol SETMODE procedures through additional Call Setup parameters. The PTP protocol is also enhanced by the addition of Fast Select support.

CONTENTS

LIST OF FIGURES

LIST OF TABLES

PREFACE

This manual describes how to use the X25AM access method to connect a Tandem NonStop or NonStop II system to a CCITT X.25 packet switching network. An outer block of X25AM parallels the protocol structure of the X.25 standard. Included in X25AM are level 1, 2, 3, and 4 protocols, of which only one (level 4) requires user interface. There are three level 4 protocols:

- Interactive Terminal Interface (ITI) protocol
- Process-to-Process (PTP) protocol
- Network (NET) protocol.

Level 1, 2, and 3 protocols are transparent to the user.

The ITI protocol allows users at remote point-to-point asynchronous terminals to communicate with a host Tandem system application via the packet switching network. These terminals interface to the network via a standard X.3 PAD (CCITT X.3/X.28/X.29 compatible). The host system interfaces to the network as a packet mode DTE. The ITI protocol supports 6510, 6520, and 6530 terminals in either the block or conversational mode, and teletype-compatible terminals in the conversational mode only.

The PTP protocol allows communication between two host systems, each functioning as a packet mode DTE. The NET protocol provides the means of connecting a local line handler on one host system to a line handler on another host system via the X.25 network.

This manual is one volume of a three-volume set of AXCESS manuals. The other two manuals are:

- Volume 1: Introduction and Communications Utility Program
- Volume 3: Device-Specific Access Methods

This manual assumes that the reader is familiar with the following documents:

- Introduction to Tandem Computer Systems
- GUARDIAN Operating System Programming Manual
- Transaction Application Language (TAL) Reference Manual

System generation (SYSGEN) information for AXCESS is provided in the following manuals:

- For the Tandem NonStop System:

  NonStop System Management Manual

- For the Tandem NonStop II System:

  NonStop II System Management Manual

SYNTAX CONVENTIONS IN THIS MANUAL

This table describes the characters and symbols used in this manual's syntax notation. For distinction, syntactical elements appear in a typeface different from that of ordinary text.

Notation	Meaning
UPPERCASE LETTERS	All keywords and reserved words appear in capital letters. If a keyword can be abbreviated, the part that can be omitted is enclosed in brackets.
lowercase letters	All variable entries supplied by the user are shown in lower-case characters.
Brackets	Square brackets ([ ]) enclose all optional syntax elements. A vertically aligned group of elements enclosed in brackets represents a list of selections from which to choose one or none.
Braces	A vertically aligned group of syntax elements enclosed in braces ({ }) represents a list of selections from which exactly one must be chosen.
Vertical Bars	A vertical bar (|) between syntax elements represents "or" in a situation in which one item is to be chosen. This usually occurs when a small number of simple elements are involved.
Ellipsis	When an ellipsis (...) immediately follows a pair of brackets or braces, the enclosed syntax can be repeated any number of times.
Punctuation	Parentheses, commas, and other punctuation symbols not described above must be entered precisely as shown. If any of the punctuation above appears enclosed in quotation marks, that character is not a syntax descriptor but a required character, and must actually be entered.

SECTION 1

GENERAL INFORMATION (X25AM)

INTRODUCTION

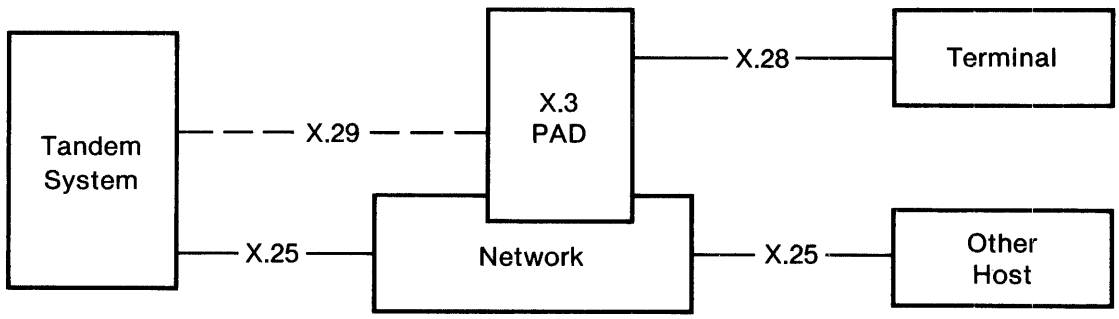
The X25AM access method allows a Tandem system (NonStop or Nonstop II) to connect to public X.25 packet switching networks such as DATANET-1, DATAPAC, DATEX-P, PSS, TELENET, TRANSPAC, TYMNET, or UNINET. The Tandem system, which interfaces to the network by means of either a bit synchronous or byte synchronous controller, provides a number of virtual circuits through which user application processes can accept calls from or initiate calls to any node within the network. Also provided by X25AM is the means of connecting a Tandem system to foreign devices that have an X.25 interface. Figure 1-1 illustrates a Tandem system interfaced to an X.25 packet switching network.

PACKET SWITCHING NETWORK

Recommended by the CCITT and accepted as the international standard for public packet switching networks, X.25 defines the formats and protocols used for communication between packet mode DTEs and DCEs. A packet mode DTE may be a host computer, a terminal, or a network interface device.

In order to implement a universal packet switching network, a common protocol is necessary among the network carrier and subscribers to the network. Recommendation X.25 defines the conventions necessary for a DTE to establish, maintain, and terminate calls with any node in the network. It also defines the protocols required to transmit user data between a DTE and another node in the network. These protocols require that user data be broken into small segments of specially formatted data packets. In addition to user data, each packet includes a header that contains control information and the address to which the packet is destined.

A packet switching network does not establish a physical end-to-end connection between nodes as does a leased line or a circuit-switched network. Data transfers in a packet switching network are performed through virtual circuits, which are logical associations between the sending and receiving nodes. Two types of virtual circuits are used: switched virtual circuits and permanent virtual circuits.



X.25 — Interface between Data Terminal Equipment (DTE) operating in the packet mode and Data Circuit-Terminating Equipment (DCE) on a Public Data Network (PDN).

X.3 — Packet Assembly/Disassembly (PAD) facility.

X.28 — DTE-to-DCE interface for a start/stop mode terminal accessing a PAD facility.

X.29 — Procedures for exchange of control information and user data between a packet mode DTE and a PAD facility.

Figure 1-1. Interrelationship of Packet-Switching Standards

A switched virtual circuit (SVC) is created, used to transfer data, and then destroyed. An SVC is created when a call request is made to the network and destroyed when the call is cleared. X25AM maps each SVC to a subdevice, which is actually a named file in the GUARDIAN file system. File system procedures such as READ and WRITE correspond to the reception and transmission of data. Certain CONTROL operations applied to the file correspond to the connection and disconnection of the circuit.

For a permanent virtual circuit (PVC), the logical connection to a DCE is always present -- it cannot be dynamically created and destroyed as can an SVC. The PVC logical connection to the DCE is created when the account is initially established with the carrier at subscription time.

X25AM binds each SVC or PVC to a subdevice; i.e., it maps the SVC or PVC to a named file in the GUARDIAN file system. The difference in operation between an SVC and a PVC is that the CONTROL operations for connecting nodes (call requests) and disconnecting nodes (call clears) are not allowed for PVC subdevices.

The Communications Utility Progran (CUP) described in Section 5, Configuration Requirements, allows you to interactively add, delete, and reconfigure subdevices; to configure permanent virtual circuits; and to bind and unbind subdevices to and from permanent virtual circuits. This program also includes utility commands that allow you to obtain detailed information about a line handler or a protocol.

Each virtual circuit (SVC or PVC) is designated by a logical channel number (LCN) carried in every packet header. The LCN, which identifies the virtual circuit, allows the network to accept interleaved streams of packets from several sources and route these packets to the proper destination.

Figure 1-2 illustrates the interleaving of packets from three subdevices across a single physical line. These subdevices are part of X25AM, and each subdevice is associated with a terminal or other subdevice through a virtual circuit. An LCN within the packet header is assiged to this temporary association, thus ensuring that the individual packets are delivered to their proper destination.

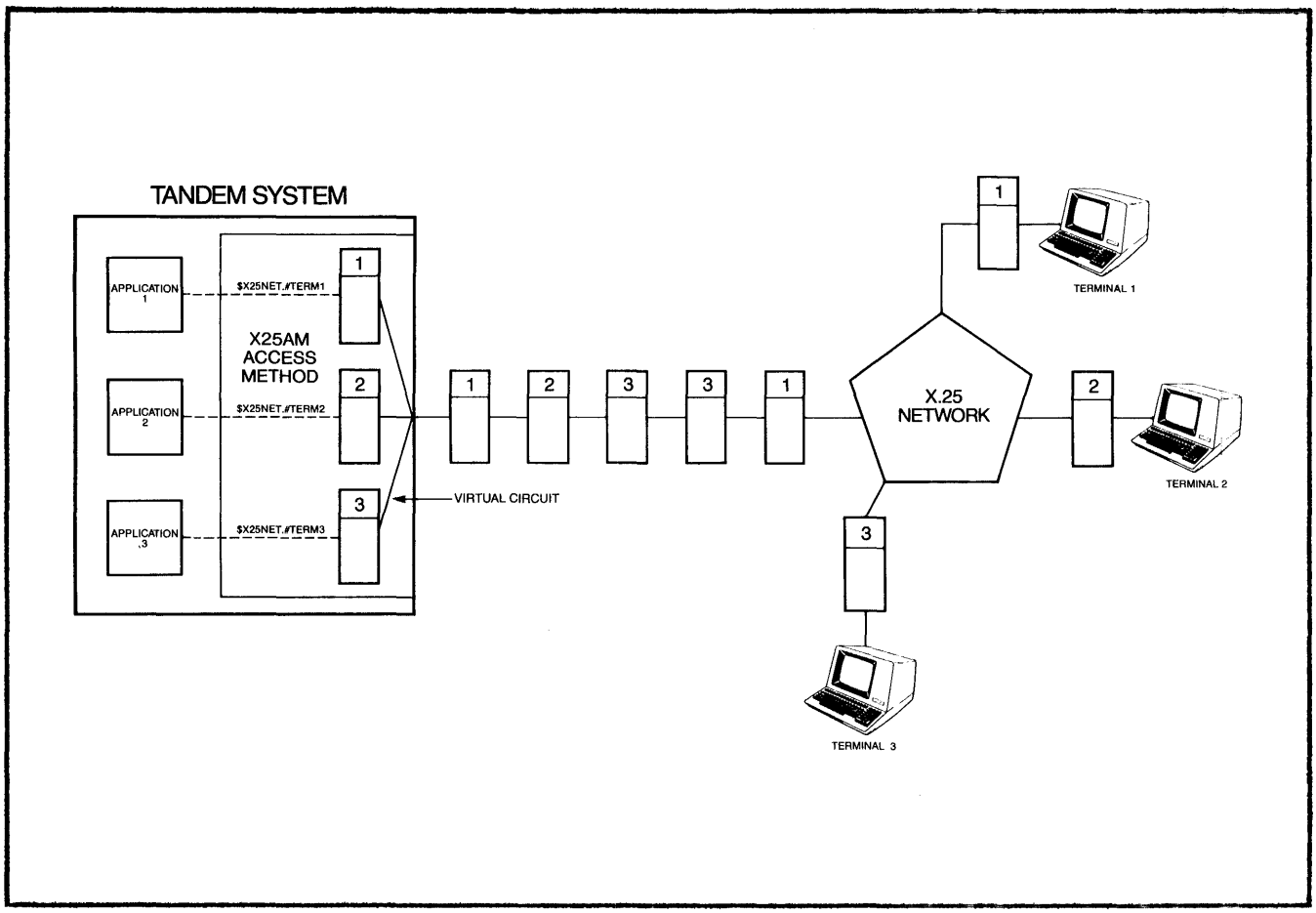


Figure 1-2. Interleaving Packets Concept

Literature containing a detailed description of a particular public packet switching network is available from the associated network carrier. Provided herein in Appendix B, Network Differences, are details of how X25AM handles the various network implementations of the X.25 standard.

FILE SYSTEM INTERFACE

Each X25AM process controls one physical communications line. Since several subdevices may share a single physical communications line, additional filename qualification allows you to define not only a given logical device (line) but also one or more subdevices sharing that line. As with any logical device, subdevices have attributes such as type, subtype, and record size. Subdevices can be dynamically added to, altered, or deleted from the line using the Communications Utility Program (CUP).

The subdevice is also an integral part of the filename. The 12-word filename consists of the device name of the physical line and the subdevice name; the subdevice name is a number sign (#) followed by up to seven alphanumeric characters. For example:

```
filename [0:3]  = "$X25NET"
         [4:7]  = "#TERM06"
         [8:11] = (blanks)
```

To start this device through the Command Interpreter, you would enter:

```
COMINT /IN $X25NET.#TERM06, OUT $X25NET.#TERM06, NOWAIT/
```

It is the subdevice that the application program opens and closes. Since subdevices are part of filenames, the file system procedures function for subdevices in the same manner as they do for other files. If an application program uses the FNAMEEXPAND and FNAMECOLLAPSE procedures, the program should run without modification.

An OPEN statement, which can succeed only after the subdevice has been defined through the Communications Utility Program (CUP), works for subdevices the same as it does for terminals except for the naming convention. An application program can determine the name to be used in an OPEN statement by either (1) calling MYTERM, (2) reading the $RECEIVE startup message, or (3) asking the user for the name. For example:

- If the program calls MYTERM, it can use the name of the home terminal in the OPEN statement.
- The names of the IN and OUT files may be identified in the $RECEIVE startup message from the Command Interpreter (COMINT). If these files are not specified, the names default to the home terminal.

- The program can ask the user to enter a file name as shown below and then call FNAMEEXPAND to convert the name into the internal format:

```
ENTER DEVICE:   $X25NET.#PTP2
```

The FNAMECOLLAPSE procedure reverses this process so that the file name may be displayed in its external form. Both the FNAMECOLLAPSE and FNAMEEXPAND procedures handle subdevices similar to the way they handle process file subnames.

The DEVICEINFO and FILEINFO procedures also handle subdevices. These procedures return the type, subtype, and record size configured for the subdevice.

APPLICATION/X25AM INTERFACE

The application interface to virtual circuits is through standard file system requests such as OPEN, READ, WRITE, WRITEREAD, CONTROL, SETMODE, and SETPARAM. After a subdevice is configured for the X.25 line, the user merely refers to the line and subdevice by name. For example,

```
$X25NET.#TERM2
```

where $X25NET is the logical device name for the synchronous line specified at SYSGEN time and #TERM2 is the subdevice name configured into the system by means of the Communications Utility Program (CUP). Figure 1-3 illustrates the X25AM/X.25 network relationship.

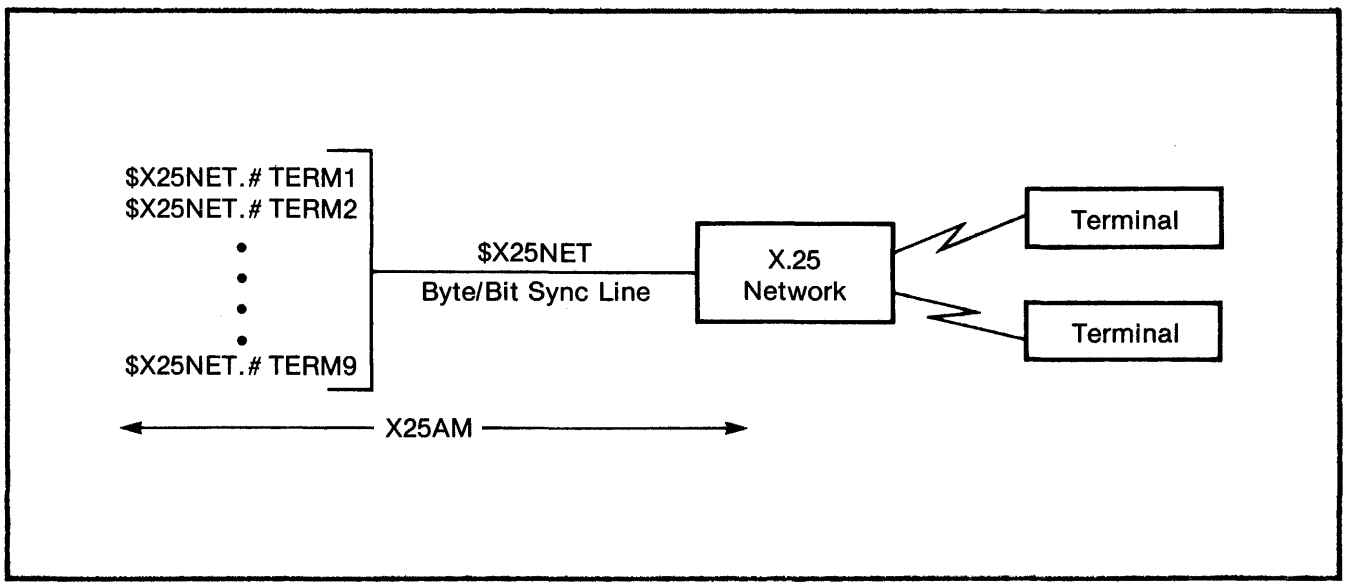


Figure 1-3. X25AM/X.25 Network Relationship

The association of a subdevice to a particular virtual circuit is made when a call is established with the X.25 network. When using the Interactive Terminal Interface (ITI) protocol described in Section 2, for example, each subdevice configured for terminal access appears as a normal conversational mode terminal to the user application. As mentioned previously, subdevices are configured into the system by means of the Communications Utility Program (CUP). The synchronous line to the X.25 network as well as the parameters established at subscription time (e.g., packet size, number of circuits, ASCII/EBCDIC characters, etc.) must be specified at SYSGEN time.

The X25AM process provides the following software:

- level 1 -- driver for the T6202 Byte Synchronous Controller and and T6203 Bit Synchronous Controller
- level 2 -- frame-level interface to the X.25 network
- level 3 -- circuit management, flow control, and multiplexing/ demultiplexing of packets between the level 2 and level 4 protocols
- level 4 -- protocols that provide the file system interface to the subdevice.

Characteristics of the various interface levels, including specific differences among the public packet switching networks, are described in Appendix A, Interface Level Characteristics.

X25AM PROCEDURAL STRUCTURE

An outer block contained in X25AM parallels the protocol structure of the X.25 standard. This outer block, called the I/O Interface, directly accesses the level 1, 2, and 3 protocols and indirectly accesses the level 4 protocols. (Refer to Figure 1-4.) Each of the boxes (except the bit/byte synchronous controller) shown in Figure 1-4 represents a process containing a calling sequence for each protocol procedure. This procedural structure provides configuration flexibility in that the user can select the proper procedures to match the controller type and the supported subdevices.

The I/O Interface handles the timer for the level 1 and 2 protocols; it also detects the loss of ownership and notifies all protocol levels which respond however necessary.

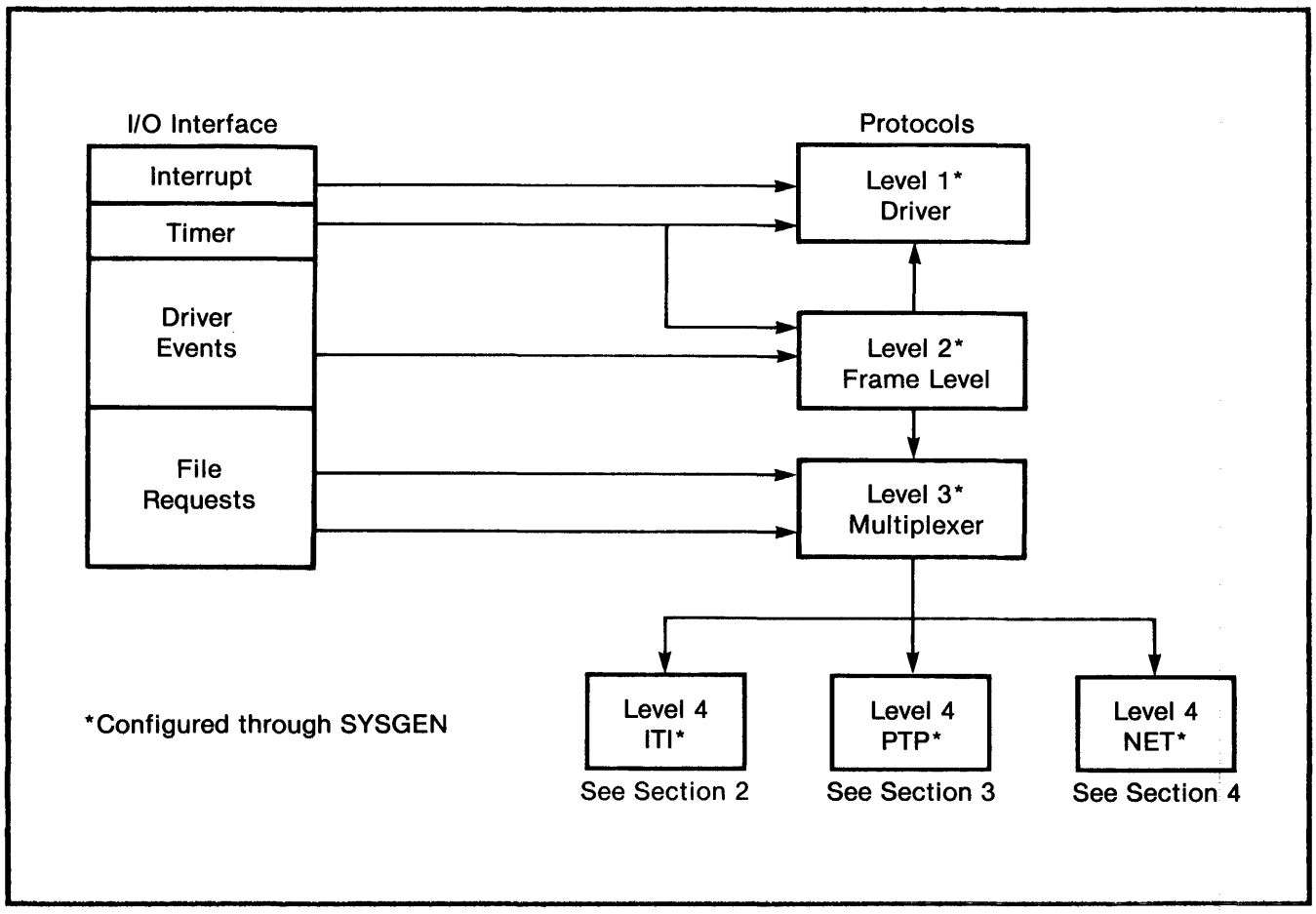


Figure 1-4. X25AM Procedural Structure

Level 1 Protocol

Level 1, the driver, provides the physical interface to the bit synchronous or byte synchronous controller. The driver handles interrupt processing and returns "events" such as READY/NOT READY or DATA IN/DATA OUT so that the I/O Interface can take the appropriate action. For example, the driver calls level 2 when a frame is to be sent or received.

Level 2 Protocol

Level 2, the frame level, provides X.25 Link Access Procedure (LAP) and Link Access Procedure Balanced (LAPB) link level management. This level contains procedures for link setup, information (data) transfer, and link disconnect.

Level 3 Protocol

Level 3, the multiplexer, serves as a multiplexer/demultiplexer that sends messages to and receives messages from the three level 4 protocols (ITI, PTP, and NET). Level 3 also handles INCOMING CALL packets

and passes them to the appropriate level 4 protocol as determined by the PORT number and the waiting subdevice.

Level 3 calls level 4 when a control packet such as CALL REQUEST or INTERRUPT must be sent, and passes all received data packets to the target level 4 protocol for processing/message assembly. Level 3 holds received DATA packets in a packet buffer if the application has not yet issued a READ request (except for the ITI 6520 block mode support).

Level 3 also calls level 4 whenever its window opens. This is done so that level 4 can provide another DATA packet from either the current outbound message or the next outbound message.

Level 4 Protocols

Level 4, which is accessed by level 3, consists of the Interactive Terminal Interface (ITI), the Process-to-Process (PTP), and the EXPAND Network (NET) protocols. (These protocols are described in Sections 2, 3, and 4, respectively. Level 4 provides the file system request queuing, performs packet assembly/disassembly, and maintains subdevice characteristics for each of the protocols.

SECTION 2

INTERACTIVE TERMINAL INTERFACE (ITI) PROTOCOL

INTRODUCTION

The ITI protocol provides the interface that allows users at remote point-to-point asynchronous terminals to communicate with a host Tandem NonStop or NonStop II system via the X.25 packet switching network. This protocol allows a terminal to communicate with an application program as if the terminal was connected directly to the computer. ITI supports the following terminals: 6510, 6520, and 6530 terminals and teletype-compatible terminals.

These terminals, which are configured in the conversational mode and may enter the block mode under application control, interface to the network through a standard X.3 PAD. The host system interfaces to the network as a packet mode DTE. (Refer to Figure 2-1.)

To use the ITI protocol, the user simply:

1. Establishes physical connection with network X.3 PAD.

2. Establishes logical connection with host by issuing a connection request command.

3. Communicates with application as if terminal is connected directly to host. If using a 6510, 6520, or 6530 terminal, user can invoke application that puts terminal in block mode.

4. Terminates current application session. Initiates another application session if necessary

5. Breaks logical connection with host (if not done by application).

6. Establishes logical connection with another host if necessary and communicates with application.

7. Breaks physical connection with X.3 PAD.

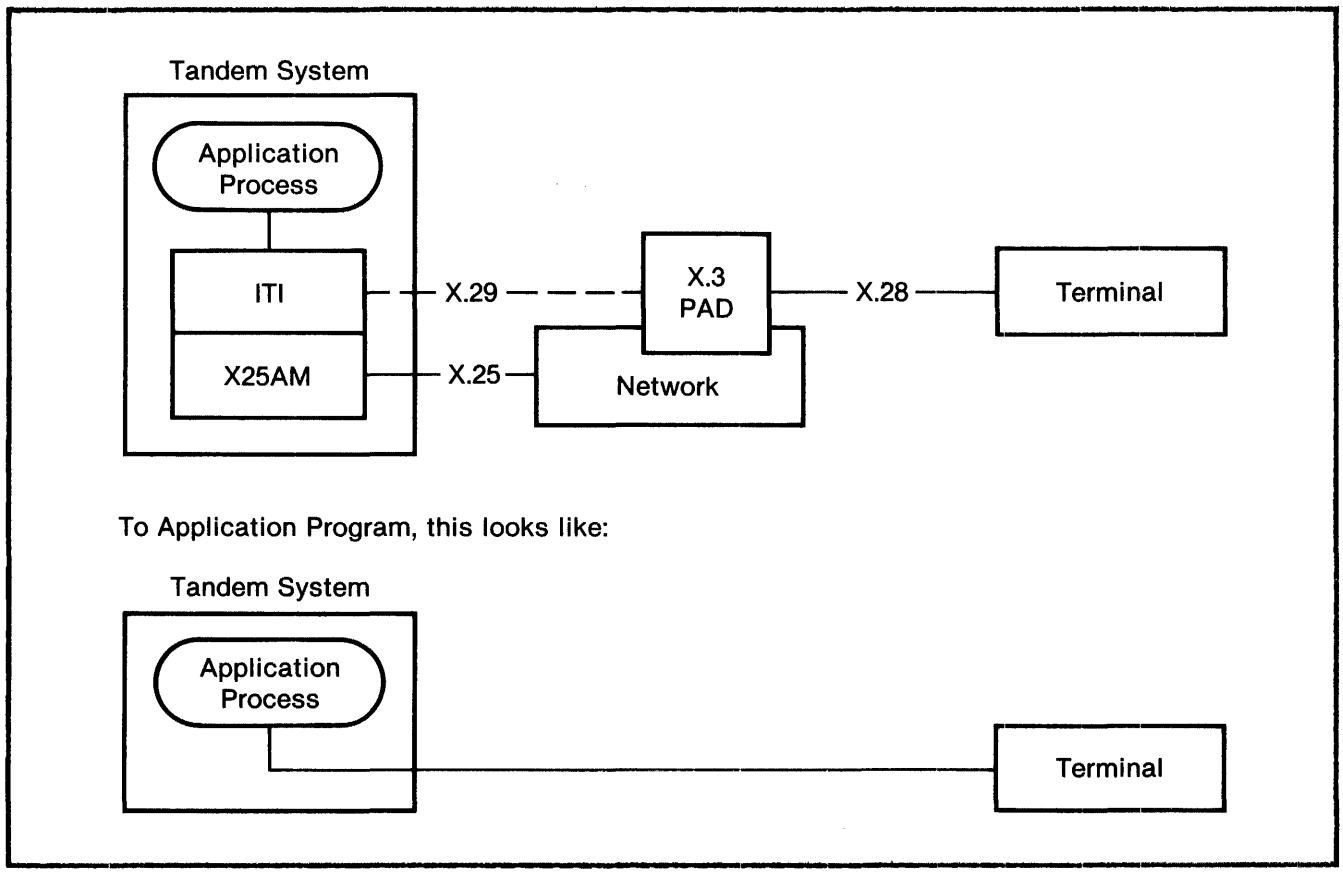


Figure 2-1. Example of ITI Protocol Use

FILE SYSTEM INTERFACE

The application, which communicates with configured X25AM subdevices through standard file system procedures, must first OPEN the subdevice and wait for connection from a remote terminal before sending data over the network. Operations at the terminal depend on the type of terminal and its current mode of operation (conversational or block mode). By means of the Communications Utility Program (CUP), the following types/subtypes can be configured for the subdevice:

Type/Subtype	Terminal Mode
(6,0)	Conversational mode terminal. Subdevice cannot be changed to block mode.
(6,1)	6510 (ADM-2) terminal. Subdevice can be changed to block mode via SETMODE procedure.
(6,2)	6520 terminal. Subdevice can be changed to block mode via SETMODE procedure.
(6,4)	6530 terminal. Subdevice can be changed to block mode via SETMODE procedure.
Others	Assumes to be type/subtype (6,0).

FILE SYSTEM PROCEDURES

Syntax and descriptions of the file system procedures supported by the ITI protocol are given in Volume 1, Introduction to AXCESS and Communications Utility Program. Because the X.25 network/PAD is designed to provide terminals with an independent interface to the host system, some of the control functions for directly-connected terminals cannot be supported by X25AM. Terminal control functions supported by X25AM are listed and described in Tables 2-1 and 2-2. Functions that are not supported by X25AM are given in Table 2-3.

Table 2-1. Supported CONTROL Operations

Operation	Terminal Mode	
	Conv./6510 Block	6520/6530 Block
1 = Terminal forms control. <parm> 0 = form feed (sends %14 in packet) 1 = vertical tab (sends %13 in packet)	(X)	Illegal
11 = Wait for modem connect. Completes when virtual circuit is established. <parm> - none	(X)	(X)
12 = Disconnect modem. Clears virtual circuit. <parm> - none	(X)	(X)

Notes:
1. (X) = operation supported for terminal mode.
2. Illegal = operation illegal for terminal mode. Operation is rejected with error 02.
3. Above operations are fully compatible with directly-connected terminals.

Table 2-2. Supported SETMODE Functions

Function	Terminal Mode	
	Conv./6510 Block	6520/6530 Block
6 = Set spacing control. <parm1>.<15> 0 = no space 1 = single space (default)	(X)	(-)
*7 = Set automatic line feed after receipt of carriage return from terminal. <parm1>.<15> 0 = off 1 = on (default) For DATAPAC, PSS, TELENET, and TYMNET networks, line feed insertion is handled locally by PAD.	(X)	(-)
*8 = Set terminal operating mode. <parm1>.<15> 0 = conversational mode (default) 1 = block mode (6510/6520/6530 only) <parm2> = error retry count (maximum of 15) (6520/6530 only) Default for error retry (LRC or response timeout) is 3.	(X)	(X)
*9 = Set interrupt characters. <parm1>.<0:7> = character 1 .<8:15> = character 2 <parm2>.<0:7> = character 3 .<8:15> = character 4 Default is back space (%10), line cancel (%30), carriage return (%15), and end-of-file (%31).	(X)	(-)
10 = Set parity checking by system. <parm1>.<15> 0 = no check (default) 1 = check parity	(X)	(-)

→

Parity is checked according to value set by SETMODE 24. If parity genaration is set to OFF by SETMODE 24, but CHECK PARITY is specified by SETMODE 10, no checking is performed. No parity checking is performed for 6520/6530 block mode data; only LRC checking is done.		
11 = Set break ownership/file access mode after BREAK key is pressed. (Also see SETPARAM 3 in Volume 1.) <parm1> 0 = disable break (default) <cpu,pin> = enable break <parm2>.<15> 0 = normal mode, any file access 1 = break mode, only break access	(X)	(X)
12 = Set terminal access mode. <parm1>.<15> 0 = normal mode, any file access 1 = break mode, only break access Set file access to terminal. <parm2> 0 = normal access 1 = break access	(X)	(X)
*20 = Set system echo mode. <parm1>.<15> 0 = off 1 = on (default)	(X)	(-)
24 = Set parity generation by system. <parm1>.<14:15> 0 = odd parity 1 = even parity 2 = none (default)	(X)	(X)

→

27 = Set system spacing mode. <parm1>.<15> 0 = postspacing (default) 1 = prespacing	(X)	(-)
*28 = Initialize to default values.	(X)	(X)
31 = Set packet mode. <parm1>.<0> 0 = ignore <parm2> 1 = <parm2> specifies leading packet size <parm2> 0 = use default packet size for transmission (default) >0 = size of first outgoing packet in each WRITE or WRITEREAD request. It must be smaller than configured packet size.	(X)	(-)
32 = Set X.25 Call Setup parameters. <parm1>.<0> 0 = do not accept charge (NOACCEPT) 1 = accept charge (ACCEPT) .<1> 0 = no request charge (NOREQUEST) 1 = request charge (REQUEST) .<2> 0 = normal outgoing call (NOPRICALL) 1 = priority outgoing call (PRICALL) .<5> 0 = send calling address 1 = don't send calling address .<6> 0 = append port number to calling address 1 = don't append port number to calling address .<8:15> = port number (PORT) <parm2>.<0> 0 = don't negotiate L3 window size 1 = negotiate L3 window size .<1> 0 = don't negotiate packet size 1 = negotiate packet size	(X)	(X)

→

.<2> 0 = don't negotiate thruput class 1 = negotiate thruput class .<8:11> = incoming thruput class .<12:15> = outgoing thruput class <last params[0]>.<3>= returns a "1" if previously accepted incoming call had requested charges. Otherwise it is zero. .<4>= returns a "1" if previously accepted call is a priority call. Otherwise it is zero.		

Notes:

1. (X) = function effective immediately when subdevice is in appropriate terminal mode.
   (-) = function accepted but has no effect on operation in this mode. It will, however, become effective when terminal changes to appropriate mode.
2. All SETMODE functions are accepted by X25AM regardless of operating mode of terminal. Some SETMODE functions, however, do not become effective until terminal is in appropriate mode.
3. Numbered SETMODE functions with asterisk (e.g., *7) indicate that Q-packet is sent because of the SETMODE.

Table 2-3. SETMODE Functions Not Supported by X25AM

Function	Action by ITI
13 = Set system read termination on ETX character.	Invalid operation
14 = Set system read termination on interrupt characters.	Invalid operation
22 = Set baud rate.	Ignored
23 = Set character size.	Invalid operation

→

38 = Set special line termination mode and character.	Ignored

Notes:
1. X25AM rejects SETMODE 13, 14, and 23 with "invalid operation" (error code 02).
2. SETMODE 22 is ignored. Transmission line speed of terminal is determined at X.3 PAD subscription time.
3. SETMODE 38 is ignored. Carriage return is always line termination character.

FILE SYSTEM ERRORS

Errors returned by file system calls are listed and described in Section 6, File System Errors and Console Messages.

APPLICATION CALL SEQUENCE

The application opens a file, accepts calls, interacts with the user, hangs up, and closes the file as follows:

1.
```
filename ':=' [ "$X25     #TERM1     " ];
CALL OPEN(filenum, filename);
```

2.
```
CALL CONTROL(filenum, 11);              ! wait for incoming call
```

3. (Optional step if application wants to determine calling DTE network address.)

```
CALL SETPARAM(filenum, 1, , , calling^dte, calling^len);
```

Application I/O

4.
```
CALL WRITE(filenum, greeting, greeting^len);

do begin
CALL WRITEREAD(filenum, promptbuf, promptlen, readlen);
        ! at beginning of prompt and processing loop
    .
    .
    .
until done;

CALL WRITE(filenum, goodbye, goodbye^len);
```

5.
```
CALL CONTROL(filenum, 12);              ! disconnect
```

6.
```
CALL CLOSE(filenum);
```

TERMINAL OPERATIONS

Terminal operations include subdevice binding and unbinding, outbound and inbound data transmission, break handling and terminal mode changes.

Subdevice Binding

The application accesses a subdevice by means of its configured name; that is:

```
$<line name>.#<subdevice name>
```

An OPEN call to a subdevice causes ITI to enable the subdevice for further file system requests.

If the subdevice is not bound to a physical terminal, the application must complete a CONTROL 11 operation (wait for modem connect) before performing read/write operations on the subdevice. ITI binds the subdevice with an outstanding "wait for modem connect" to the calling terminal if the port number associated with the subdevice matches the port number specified in the incoming call.

If the subdevice is already bound to a physical terminal, the application can issue read/write requests as soon as the subdevice is OPENed. The "wait for modem connect" function to a bound subdevice is completed immediately by ITI with no error.

All read/write requests to an unbound subdevice are rejected by ITI with "circuit is not connected" (error code 140).

ITI allows a maximum of 15 simultaneous OPENs on a subdevice. Further OPENs on the subdevice are rejected with "no file OPENs are permitted" (error code 61).

Subdevice Unbinding

A subdevice is unbound when the connection to the terminal is severed. The connection is severed when:

1. explicitly requested by terminal user
2. cleared by X.25 network or X.3 PAD because of protocol error
3. cleared by ITI because of protocol or internal procedural error
4. cleared by ITI when requested to do so by application
5. cleared by ITI when there are no more OPENs for subdevice.

For items 1, 2, 3, and 5 above, the X.25 Clear Request packet is used to sever the connection. For item 4, the PAD message "Invitation to Clear" is used if supported by the network/PAD; otherwise, the X.25

Clear Request packet is used. The PAD message "Invitation to Clear" is used only for the DATAPAC and TRANSPAC networks.

When the link is being disconnected, all read/write requests are aborted with "circuit is not connected" (error code 140). The application must complete another CONTROL 11 operation (wait for modem connect) before issuing read/write requests for the subdevice.

Outbound Data Transmission

When the application issues a WRITE or WRITEREAD request, ITI segments the write buffer into one or more data packets and inserts the appropriate framing/control characters before transmitting to the terminal. The operation on the write buffer depends on the terminal type as described in following paragraphs.

CONVERSATIONAL TERMINAL. ITI performs the following on the write buffer before sending it to the terminal:

1. Breaks message into multiple X.25 data packets; size of first packet is given by SETMODE 31. M-bit (more data) is set in all packets except last one. However, if size of first packet is smaller than normal packet size, M-bit for first packet will not be set.

2. Inserts carriage return (CR) and line feed (LF) in message if SETMODE 6 is enabled. CR/LF is inserted at front of message if SETMODE 27 is a one; otherwise it is appended at end of message.

   Note that CR/LF insertion is done only for WRITE requests. It is not done for WRITEREAD requests.

3. Sets parity depending on value of SETMODE 24.

4. Sends packets for write buffer as soon as level 3 protocol window is opened. That is, ITI does not wait for acknowledgement before sending another data packet.

6510 BLOCK MODE. Operation in this mode is the same as the conversational mode terminal except that CR/LF insertion is not done regardless of the value of SETMODE 6.

6520/6530 BLOCK MODE. ITI performs the following on the write buffer before sending it to the terminal:

1. Breaks message into multiple 256-character blocks including framing characters for each block. Leading and trailing framing characters are <STX> <sequence-number> and <ETX> <LRC-character>, respectively. An <EOT><PAD> sequence is also inserted at beginning of first block of each message. Maximum block size including framing characters is 256 characters. Hence, first block contains 250 user data characters and all subsequent blocks contain 252 user data characters.

2. Sets appropriate parity in message block and calculates its LRC.

3. Packetizes message block into multiple X.25 data packets with M-bit (more data) set in all except last data packet of each block.

4. Transmits message block to X.3 PAD and waits for acknowledgement from terminal before sending another block. If no reply is received from terminal after 25 seconds, or if NAK is obtained indicating LRC error, ITI will retransmit text block. Number of retransmissions is specifiable via SETMODE 8.

If the outbound data contains escape sequences that solicit a text reply (i.e, a message that begins with SOH or STX) from a terminal, the application should request transmission with a WRITEREAD procedure instead of a WRITE procedure. If WRITE is used, the WRITE will complete as follows:

1. If all data has been sent when text reply is received, WRITE will complete with no error. Received data is discarded.

2. If data is left to send when text reply is received, WRITE will complete with error 156 (protocol error). Received data is discarded. Application should remove escape sequences that cause text to be sent by terminal in write buffer before issuing WRITE request again.

If the outbound data does not contain escape sequences that solicit a text reply from a termimal, the application should request transmission with a WRITE procedure instead of a WRITEREAD procedure. If WRITEREAD is used, it will complete with no error when the block acknowledgement is received and the <read count> specified in WRITEREAD will be set to zero.

The five escape sequences that solicit text reply from a terminal are:

1. read cursor address (ESC a)

2. read video status (ESC ^)

3. read buffer (ESC <)

4. read with address (ESC =)

5. read with address all (ESC])

Note that ITI does not intercept or interpret the outbound application data. It only inserts the appropriate framing characters before transmission.

Inbound Data Transmission

The X.25 network PAD requires a forwarding signal before data received from the terminal is sent to the host. The forwarding signal is speci-

fied by the X.3 parameter from the host.

The forwarding signal specified by X25AM/ITI and the handling of incoming data packets by ITI depends on the operating mode of the terminal as described in following paragraphs.

CONVERSATIONAL TERMINAL. The forwarding signal requested by ITI depends on the four interrupt characters specified in SETMODE 9 by the application. If all four interrupt characters are CR (%15), the PAD will forward data received from the terminal when a carriage return (CR) is seen; otherwise it will forward the data whenever an ASCII control character (%00 to %37) is seen.

ITI performs the following on incoming data packets:

1. Assembles incoming data packets to form the read buffer to be returned to the application. If there is no read pending, ITI will hold incoming data packets until application issues a READ or WRITEREAD request.

   Parity bit (most significant bit) of character is stripped before being copied to read buffer.

2. Performs parity checking if SETMODE 10 is enabled. If parity error is detected, READ/WRITE request is completed with error 120.

3. Completes read when either the read buffer is full or an interrupt character, other than a backspace (BS) or line cancel (CAN), is encountered. If read terminates because read buffer is full, excess data (remainder of M-bit packet sequence) is discarded.

   When a data packet is received that does no have the M-bit set, the ending character in the packet is checked as follows:

   a. Strips ending LF (%12) character if it is immediately preceded by CR (%15).

   b. Terminates read (with ending character returned to application) if ending character is an interrupt character other than one of the following:

      - backspace (%10): removes character preceding backspace from read buffer and continues read operation.

      - line cancel (%30): discards data in current read buffer, sends the string "@ CR LR" to terminal, and continues read operation.

      - EOF (%31): sends the string "EOF ! CR LF" to terminal and terminates read operation with end of file condition (01). No data is returned to application.

- CR (%15): strips CR from read buffer and sends LF to terminal if SETMODE 7 is enabled and PAD has not echoed LF. Read operation is completed successfully.

6510 BLOCK MODE. Operation in this mode is the same as for a conversational mode terminal except no special handling of interrupt characters is done by ITI. Note that the interrupt character is normally a CR and is returned to the application.

6520/6530 BLOCK MODE. A file system READ request causes ITI to solicit the function key (if any) typed by the user. A file system WRITEREAD request causes ITI to obtain the device status or screen data from the terminal depending on the escape sequence specified in the application data.

ITI assembles incoming data packets to form a data block. For a 6520 terminal, a data block is 260 characters long. For a 6530 terminal, a data block is also 260 characters long except when the "PACKET BUFFERING" option is used. In this case, the data block is 256 characters long. ITI accepts the data block only if there is no LRC or sequence numbering error. No parity checking is done on the data.

The valid data block is stripped of the parity bit (most significant bit) and framing characters before being transferred to the application read buffer. Leading <SOH> and trailing <ETX> <LRC-character> are stripped if the message begins with an <SOH>. Leading <STX> <sequence number> and trailing <ETX> <LRC-character> are stripped if the message begins with an <STX>. Note that ITI does not intercept or interpret other characters in the data block.

A data block with LRC error is rejected by ITI. In this case, NAK is sent to the terminal to request retransmission of the text block. The number of retries is determined by SETMODE 8.

A file system READ or WRITEREAD request is terminated when the last data block has been received from the terminal OR when the application read buffer is full. (The excess data received is discarded.) If the READ or WRITEREAD request is aborted by the application prior to its completion, ITI transmits an EOT (End of Transmission) to reset the terminal AND discards all received data.

Note that ITI does not hold data for the application. Data received from the terminal is discarded if there is no outstanding READ or WRITEREAD request for the subdevice.

Break Handling

Break handling by X25AM/ITI is the same as the terminal process for directly-connected terminals. The ITI protocol recognizes two break-related SETMODE functions and two break-related errors. The SETMODE functions (refer to Table 2-2) are:

1. Set break ownership and file access mode after BREAK key is pressed (SETMODE 11)

2. Set terminal and file access mode (SETMODE 12).

The file errors associated with break are:

1. Only break access is permitted (error code 110)

2. Break operation aborted because of BREAK (error code 111).

In addition, the application can specify a break tag for break handling on a subdevice basis via the SETPARAM function. (Refer to Volume 1, Introduction to AXCESS and Communications Utility Program.) ITI handles incoming interrupt packets the same way that a terminal process handles break for interactive terminals. An interrupt packet is the result sent from the X.25 network when a user presses the BREAK key on the terminal.

For more details, refer to Terminals: Conversational/Page Mode section in the GUARDIAN Operating System Programming Manual (Volume 1).

Terminal Mode Changes

ITI allows the application to change the operating mode of a 6510, 6520, or 6530 terminal. To change the terminal from conversational to block mode, the application issues a SETMODE 8 function with <parm1> set to one. (Refer to Table 2-2). This causes ITI to send a "Set Block Mode" command to the terminal (if it is either a 6520 or 6530). To change the terminal from block to conversational mode, the application issues a SETMODE 8 function with <parm1> set to zero. This causes ITI to send a "Set Conversational Mode" command to the terminal (if it is either a 6520 or 6530). The terminal is assumed to be in the conversational mode at connection time. Note that ITI also sends the appropriate terminal profile to the X.3 PAD at each mode change.

The application can obtain the operational mode of the terminal (logical) by issuing a SETMODE 8 function with the <last param> parameter.

ERROR HANDLING

ITI supports all standard file system errors for device type 6. (Refer to Section 6.) In addition, the following errors are specific to this environment and warrant special attention:

1. LRC error (6520/6530 block mode only)

   ITI retransmits a text block or requests retransmission of a text block because of LRC error. ITI performs three retransmissions (default) before aborting operation with "transmission error"

(error code 173). Retry count can be changed by means of SETMODE 8 <parm2>. (Refer to Table 2-2.)

2. Response timeout error (6520/6530 block mode only)

   ITI times all transmissions to terminal except EOT (End of Transmission) and ENQ (read command). If timeout expires while waiting for terminal reply, ITI retransmits request three times (default) before aborting operation with "operation timeout" (error code 162). Response timeout value is 25 seconds and cannot be changed. Retry count, however, can be changed by means of SETMODE 8 <parm2>. (Refer to Table 2-2.)

3. Data transmission aborted by terminal (6520/6530 block mode only)

   ITI aborts read/write operation whenever it receives an EOT (End of Transmission) from terminal. There is no retry associated with this error type. ITI returns "EOT received" (error code 163) to application.

4. Network/PAD reset (all terminal types)

   ITI aborts active read/write request when "X.25 Reset" is received from network or X.3 PAD. ITI returns "circuit reset by network or PAD" (error code 122) to application.

5. Terminal reset (6520/6530 block mode only)

   Terminal can be reset (soft or hard) by entering appropriate key sequence on keyboard. A soft reset only unlocks keyboard. A hard reset, however, overwrites screen display with test pattern and changes terminal to its configured mode (assumed to be conversational). In either case, an <ENQ> <CR> sequence is sent to host.

   ITI assumes terminal is configured to its default mode whenever it receives <ENQ> <CR> sequence from terminal. ITI then executes recovery procedure as follows:

   a. If terminal is in conversational mode (logical) before reset, ITI sends "set conversational mode" command to terminal before resuming I/O operations.

   b. If terminal is in block mode (logical) before reset, ITI does the following:

      - aborts current and all queued read/write requests with "device power on" (error code 191)
      - reinitializes terminal to operate in block mode by sending "set block mode" command to terminal.

INTERFACE TO NETWORK/PAD

ITI communicates with the X.3 PAD using the CCITT X.29 protocol. The user terminal communicates with the X.3 PAD using the CCITT X.28 protocol. Operational characteristics of the PAD are controlled by a terminal profile consisting of X.3 international parameters as well as the network carrier's national parameters. This terminal profile can be modified by ITI and/or the terminal user as described in following paragraphs.

PAD Parameter Selection

The terminal user can change the operational characteristics of the PAD (assuming it is in the PAD command mode).

ITI modifies the PAD operational characteristics as follows whenever there is a switch in the terminal operating mode:

1. When connection is first established with terminal, ITI assumes terminal is in conversational mode and modifies a number of PAD parameters.

2. Whenever terminal switches to block mode, ITI modifies PAD parameters to operate in transparent mode. However, when terminal switches back to conversational mode, some of the local PAD control functions are not restored.

Table 2-4 summarizes the X.3 international parameters modifiable by ITI. Note that not all X.3 parameters are supported by all networks. Tables 2-5 through 2-8 summarize the X.3 international parameters and network-dependent national parameters supported by the appropriate network carrier. Note also that the parameters marked "Not used" imply that those parameters are not sent to the network.

Table 2-4. X.3 International Parameters Modifiable by ITI

Parameter	Parameter Value	
	Conv./6520 Block Mode	6520/6530 Block Mode
1. PAD recall by escaping from data transfer	1=enabled	0=disabled
2. Echo	0 or 1 (note 1)	0=disabled
3. Selection of data forwarding conditions	2 or 126 (note 2)	0=none
4. Idle timer	0=none	100 ms (data forwarding)
5. Ancillary device control	Not used	0=disabled
6. Suppression of network messages	Not used	0=disabled
7. Break handling	21=send interrupt packet, Indication of Break message, and flush output to terminal until host restores normal delivery	Same as Conv. Mode terminal
8. Discard output	0=restore normal data delivery to terminal	Same as Conv. Mode terminal
9. Padding after receipt of CR from terminal	Not used	0=disabled
10. Line folding	0=disabled	0=disabled
12. Flow control of PAD by terminal	Not used	0=disabled
13. Line feed (LF) insertion after receipt of CR from terminal	0 or 4 (note 3)	0=disabled
14. Padding after LF	Not used	0=disabled
15. Line editing	Not used	0=disabled

→

Notes:
1. Enabled only if SETMODE 20 is on and terminal is not in block mode.
2. Forward on CR only if all four interrupt characters are CR (=2); otherwise forward on all control characters and DEL (=126).
3. Insert LF (=4) only for conversational terminal with SETMODE 7 enabled.

Table 2-5. DATAPAC Parameters Modifiable by ITI

Parameter	Parameter Value	
	Conv./6510 Block Mode	6520/6530 Block Mode
121. Additional data forwarding conditions	0=disabled	0=disabled
122. Additional data forwarding conditions	0=disabled	0=disabled
123. Parity treatment	Not used	0=disabled
125. Output pending timer	Not used	0=disabled
126. Line feed insertion	0 or 4 (note 1)	0=disabled

Notes:
1. Parameter enabled/disabled (0/4 respectively) depending on SETMODE 7 <parm1>.
2. All X.3 parameters in Table 2-4 except 13, 14, and 15 are sent to DATAPAC in addition to above national parameters.

Table 2-6. TELENET Parameters Modifiable by ITI

Parameter	Parameter Value	
	Conv./6510 Block Mode	6520/6530 Block Mode
1. Line feed insertion	0 or 4 (note 1)	0=disabled
2. Network message display	Not used	0=disabled
10. Line feed padding	Not used	0=disabled
11. Tab padding	Not used	0=disabled
18. Vertical terminal options	Not used	0=disabled
37. Network usage display	Not used	0=disabled
41. DCE to DTE flow control	Not used	0=disabled
54. DTE to DCE flow control	Not used	0=disabled
63. Eight bit transparent	Not used	0=disabled

Notes:
1. Parameter enabled/disabled (0/4 respectively) depending on SETMODE 7 <parm1>.
2. All X.3 parameters in Table 2-4 except 13, 14, and 15 are sent to TELENET in addition to above national parameters.

Table 2-7. DATEX-P Parameters Modifiable by ITI

Parameter	Parameter Value	
	Conv./6510 Block Mode	6520/6530 Block Mode
118. Character delete	Not used	0=disabled
119. Line delete	Not used	0=disabled
120. Line display	Not used	0=disabled
121. Additional data forwarding conditions	0=disabled	0=disabled

→

122. Additional data forwarding conditions	0=disabled	0=disabled
123. Parity treatment	Not used	0=disabled
125. Output pending timer	Not used	0=disabled

Notes:
1. All X.3 parameters in Table 2-4 except 13, 14, and 15 are sent to DATEX-P in addition to above national parameters.

Table 2-8. DATANET-1/PSS/TRANSPAC/TYMNET/UNINET Parameters Modifiable by ITI

Parameter	Parameter Value	
	Conv./6510 Block Mode	6520/6530 Block Mode

Notes:
1. All X.3 parameters in Table 2-4 are sent to these networks. No network-dependent national parameters are used.

PAD Control Messages

PAD messages are used by the X.3 PAD and ITI to exchange control information. ITI uses control messages to set or read PAD parameters and to clear the circuit connection. The PAD uses control messages to return configured parameters and to indicate receipt of break from the terminal.

The following control messages are sent to the PAD:

SET
SET AND READ
INVITATION TO CLEAR (if supported by PAD)

The following control messages can be received from the PAD:

PARAMETER INDICATION
INDICATION OF BREAK
ERROR

ITI transmits and expects to receive PAD control messages in a "single" level 1 data packet with the X.25 Q-bit=1 and the M-bit=0. The receipt of an invalid PAD message, an undecodable message type, or an ERROR message causes ITI to clear the circuit.

SECTION 3

PROCESS-TO-PROCESS (PTP) PROTOCOL

INTRODUCTION

The PTP protocol is typically used where a Tandem system (NonStop or NonStop II) functions as a packet mode DTE communicating with another packet mode DTE. (Refer to Figure 3-1.) This section summarizes the file system procedures, gives an example of an application call sequence, and describes the three PTP modes of operation.

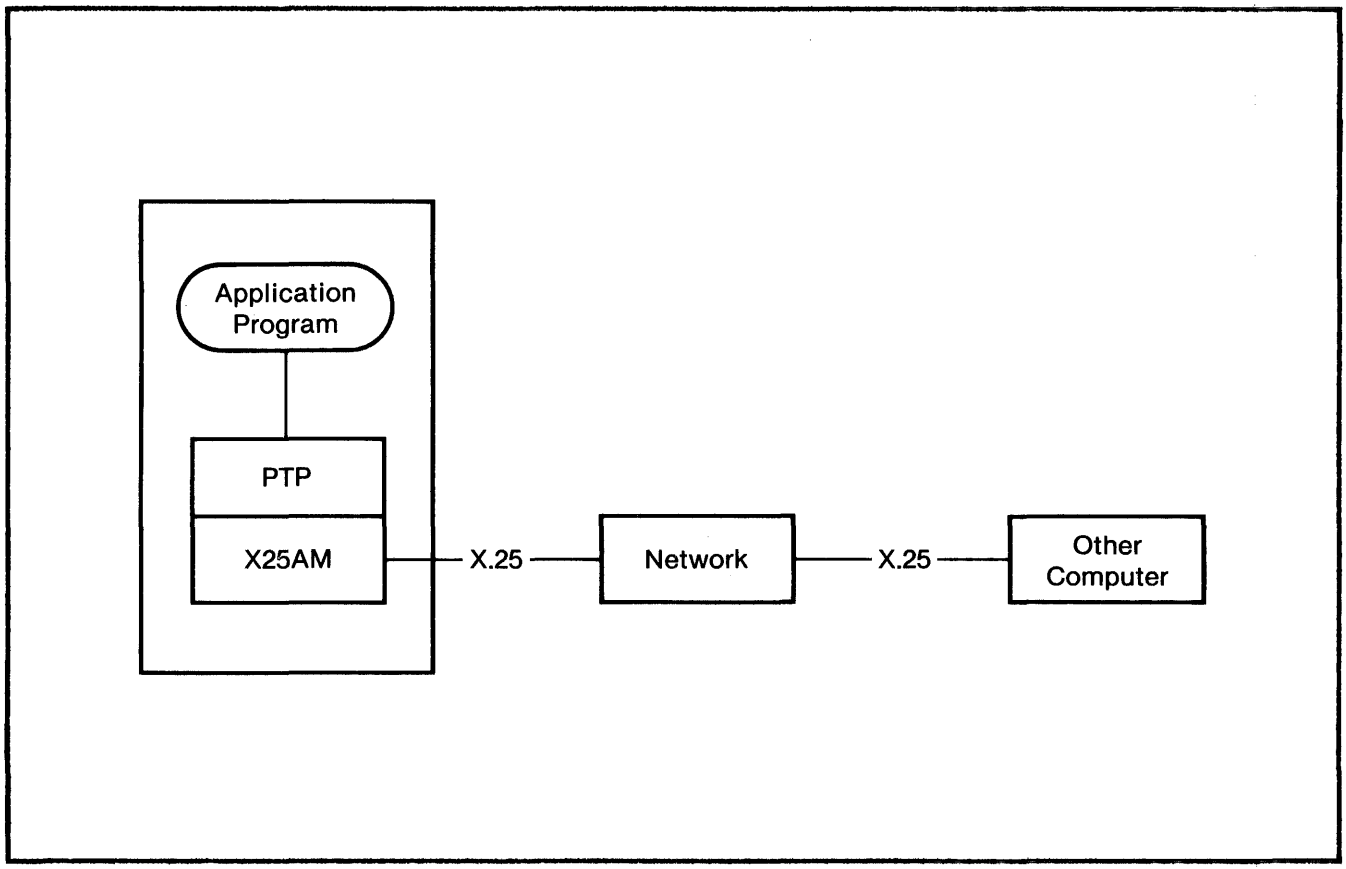


Figure 3-1. Example of PTP Protocol Use

FILE SYSTEM PROCEDURES

Syntax and descriptions of the file system procedures supported by the PTP protocol are given in Volume 1, Introduction to AXCESS and Communications Utility Program. For CONTROL and SETMODE calls, the PTP protocol supports only those operations and functions listed and described in Tables 3-1 and 3-2, respectively.

FILE SYSTEM ERRORS

Errors returned by file system calls are listed and described in Section 6, File System Errors and Console Messages.

Table 3-1. Supported CONTROL Operations

11 = Wait for modem connect. Completes when virtual circuit is established.

<parm> - none

12 = Disconnect modem. Clears virtual circuit.

<parm> - none

17 = Enable connection. Initiates call to remote DTE in X.25 network. Subdevice must have defined ADDRESS configured by CUP. Otherwise, application must call SETPARAM to specify address before using this CONTROL operation.

If circuit is established, operation completes with error = 0. Otherwise, it completes with error code 140. Error code 12, if returned, is not a fatal error; it means merely that the subdevice is already connected.

<parm> = none

Table 3-2. Supported SETMODE Functions

28 = Initialize to default values.

30 = Allow nowait I/O requests to complete in the order issued or complete in any order.

<parm1> 0 = complete in order (default setting). File system maintains one queue for all READ, WRITE, and WRITEREAD requests. As a result, all requests are completed on first-in, first-out basis regardless of order in which X25AM acts on them.

1 = complete in any order. File system completes requests in approximately same order that X25AM acts on them.

For both PTP mode 1 and mode 2 (described later), X25AM maintains a read queue and a write queue. Requests are processed on a first-in, first-out basis within each queue. When using WRITEREAD, WRITE portion is sent to write queue; when WRITE completes, READ portion is sent to read queue. After READ portion has been received to satisfy read requirement, WRITEREAD request is completed to application.

31 = Set X.25 packet mode.

<parm1>.<0> 0 = ignore <parm2>
1 = <parm2> specifies leading packet size

.<13:15> 0 = normal PTP mode (0)

PTP performs packet assembly and disassembly on application read/write buffers. For details, refer to subsequent "Operating Modes" paragraph.

1 = PTP mode 1

PTP and application communicate at packet level by using message control word (MCW) at beginning of each buffer. The MCW allows application to send/receive data and interrupt packets. For details, refer to subsequent "Operating Modes" paragraph.

→

```
                        2 = PTP mode 2

                            Same as PTP mode 1 except application
                            can send/receive more packet types.
                            For details, refer to subsequent
                            "Operating Modes" paragraph.

      <parm2>             0 = use default packet size for transmission

                        > 0 = size of first outgoing packet in each
                            WRITE or WRITEREAD request.  Must be
                            smaller than configured packet size.
                            Note that short packet transmission is
                            used only for ITI and PTP mode 0 sub-
                            devices.

32 = set X.25 Call Setup parameters.

     <parm1>.<0>      0 = do not accept charge (NOACCEPT)
                      1 = accept charge (ACCEPT)
            .<1>      0 = no request charge (NOREQUEST)
                      1 = request charge (REQUEST)
            .<2>      0 = normal outgoing call (NOPRICALL)
                      1 = priority outgoing call (PRICALL)
            .<5>      0 = send calling address
                      1 = don't send calling address
            .<6>      0 = append port number to calling address
                      1 = don't append port number
            .<8:15>     = port number (PORT)

     <parm2>.<0>      0 = don't negotiate L3 window size
                      1 = negotiate L3 window size
            .<1>      0 = don't negotiate packet size
                      1 = negotiate packet size
            .<2>      0 = don't negotiate thruput class
                      1 = negotiate thruput class
            .<8:11>     = incoming throughput class
            .<12:15>    = outgoing throughput class

     <last params[0]>.<3> = returns a "1" if previously accepted
                            incoming call has requested reverse
                            charges. Otherwise it is "0".
                       .<4> = returns a "1" if previously accepted
                            incoming call is a priority call.
                            Otherwise it is a "0".
```

APPLICATION CALL SEQUENCE

The application opens a file, accepts calls, interacts with the user, hangs up, and closes the file as follows:

```
1.  filename ':=' [ "$X25     #TERM1     " ];
    CALL OPEN(filenum, filename);

2.  CALL CONTROL(filenum, 11);              ! wait for incoming call

3.  (Optional step if application wants to determine calling DTE
    network address.)

    CALL SETPARAM(filenum, 1, , , calling^dte, calling^len);

    Application I/O

4.  CALL WRITE(filenum, greeting, greeting^len);

    do begin
    CALL WRITEREAD(filenum, promptbuf, promptlen, readlen);
            ! at beginning of prompt and processing loop
        .
        .
        .
    until done;

    CALL WRITE(filenum, goodbye, goodbye^len);

5.  CALL CONTROL(filenum, 12);              ! disconnect

6.  CALL CLOSE(filenum);
```

The application opens a PTP subdevice, sets up CALL REQUEST parameters and remote DTE address, initiates a call, does the I/O, clears the circuit, and closes the file as follows:

```
1.  filename ':=' [ "$X25     #APP2      " ];
    CALL OPEN(filenum, filename);           ! open #APP2 on $X25

2.  (Optional step if subdevice ADDR is not preset with CUP.)

    buf ':=' [ "31107030012344" ];          ! set remote DTE address
    CALL SETPARAM(filenum, 1, buf, 14);     ! DNIC 3110, area 703,
                                            ! DTE 123, port #44

3.  CALL CONTROL(filenum, 17);              ! initiate call request

    Application I/O

4.  CALL WRITE(filenum, initmsg, initmsg^len);
```

```
5.  loop
    CALL WRITEREAD(filenum, msgbuf, msglen, readlen);
                                          !for each request
        .
        .
        .
    until done;

6.  CALL WRITE(filenum, endmsg, endmsg^len);

7.  CALL CONTROL(filenum, 12);              ! disconnect

8.  CALL CLOSE(filenum);
```

OPERATING MODES

The PTP protocol has three modes of operation: normal mode, packet mode 1, and packet mode 2.

Normal Mode

In the normal mode, the PTP protocol performs packet assembly and disassembly. Write buffers are sent (disassembled) in a series of packets with the M-bit set in the packet header in all but the last packet. PTP assembles multipacket messages received (read) into the application program's read buffer. PTP does not insert carriage returns or line feeds, nor does it handle interrupt packets. Interrupt packets are automatically discarded.

Packet Modes 1 and 2

In the packet mode, the PTP protocol and the application program communicate with one another via a message control word (MCW), which is stored in the first word of each READ, WRITE, or WRITEREAD buffer. The MCW defines the associated packet type and the state of certain bits within the packet header.

In PTP mode 1, the application is allowed only to transmit/receive Data and Interrupt packets. In PTP mode 2, however, the application is allowed to transmit the following packet types: Call Request, Call Accept, Interrupt, Reset, Data, and Diagnostic. In addition, in PTP mode 2 the application is also able to receive all data packet types listed in Figure 3-3, including Fast Select.

The PTP protocol does not insert carriage returns or line feeds and returns error 179 (application buffer incorrect) to any request with and invalid MCW.

Figures 3-2 and 3-3, respectively, illustrate the MCW format for packet mode 1 and packet mode 2.

MCW Bits	Meaning*
<0:12>	Not used; set to zeros.
<13>	1 = interrupt packet.
<14>	1 = packet is PAD message (Q-bit).
<15>	1 = more data follows (M-bit).

*MCW has different values when using two-step read. Refer to "Two-Step Read" later in this section.

Figure 3-2. MCW Format (Packet Mode 1)

First byte of MCW (<0:7>) defines the following packet types* (numbers are base 10):

```
 0 - Data                              27 - Reset Request or Reset
 1 - RR (Receive Ready)                     Indication
 5 - RNR (Receive Not Ready)           31 - Reset Confirmation
11 - Call Request or Incoming          35 - Interrupt
     Call                              39 - Interrupt Confirmation
15 - Call Accept or Call              241 - Diagnostic
     Connected                        251 - Restart Request or Restart
19 - Clear Request or Clear                Indication
     Indication                       255 - Restart Confirmation
23 - Clear Confirmation
```

Second byte of MCW (<8:15>) defines state of various bits contained in heade2 of associated packet type as follows:*

```
MCW.<8:12>  = 0 (not currently used)
   .<13>    = D (end-to-end confirmation)
   .<14>    = Q (PAD message)
   .<15>    = M (more data follows)
```

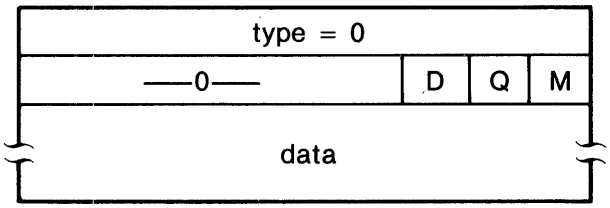
Third and remaining bytes of buffer define contents of associated packet. Formats of the various packet types are shown in Figure 3-4.

*MCW has different values when using two-step read. Refer to "Two-Step Read" later in this section.

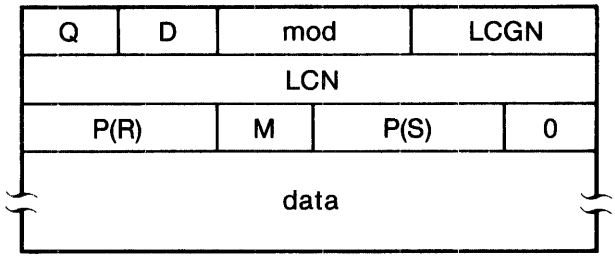
Figure 3-3. MCW Format (Packet Mode 2)

Buffer

Data Packet

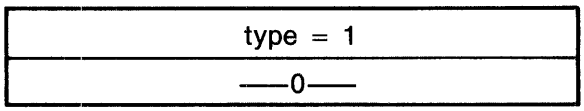


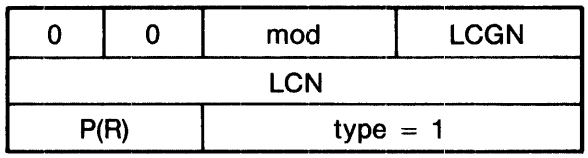
Corresponding Packet



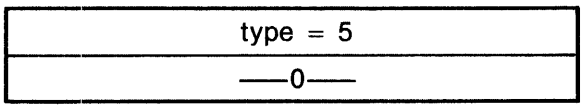
The packet format shown above is for modulo 8 packet numbering.

Receive Ready (RR)





Receive Not Ready (RNR)



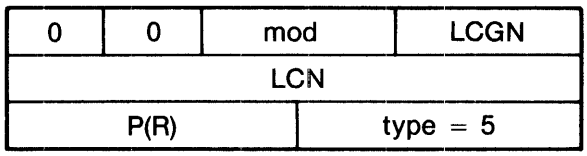


Figure 3-4. Valid Packet Types for PTP Mode 2 (Page 1 of 5)

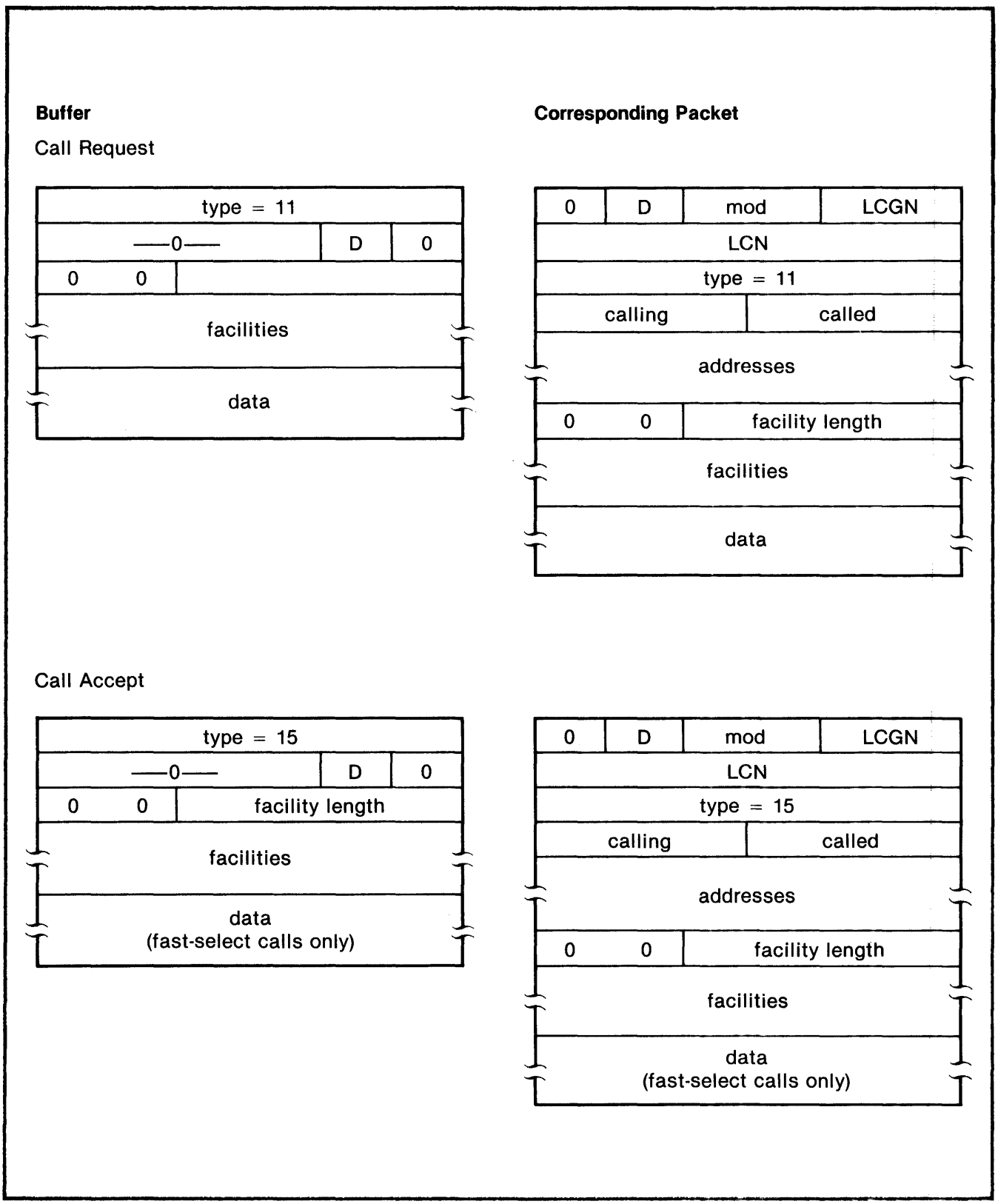


Figure 3-4. Valid Packet Types for PTP Mode 2 (Page 2 of 5)

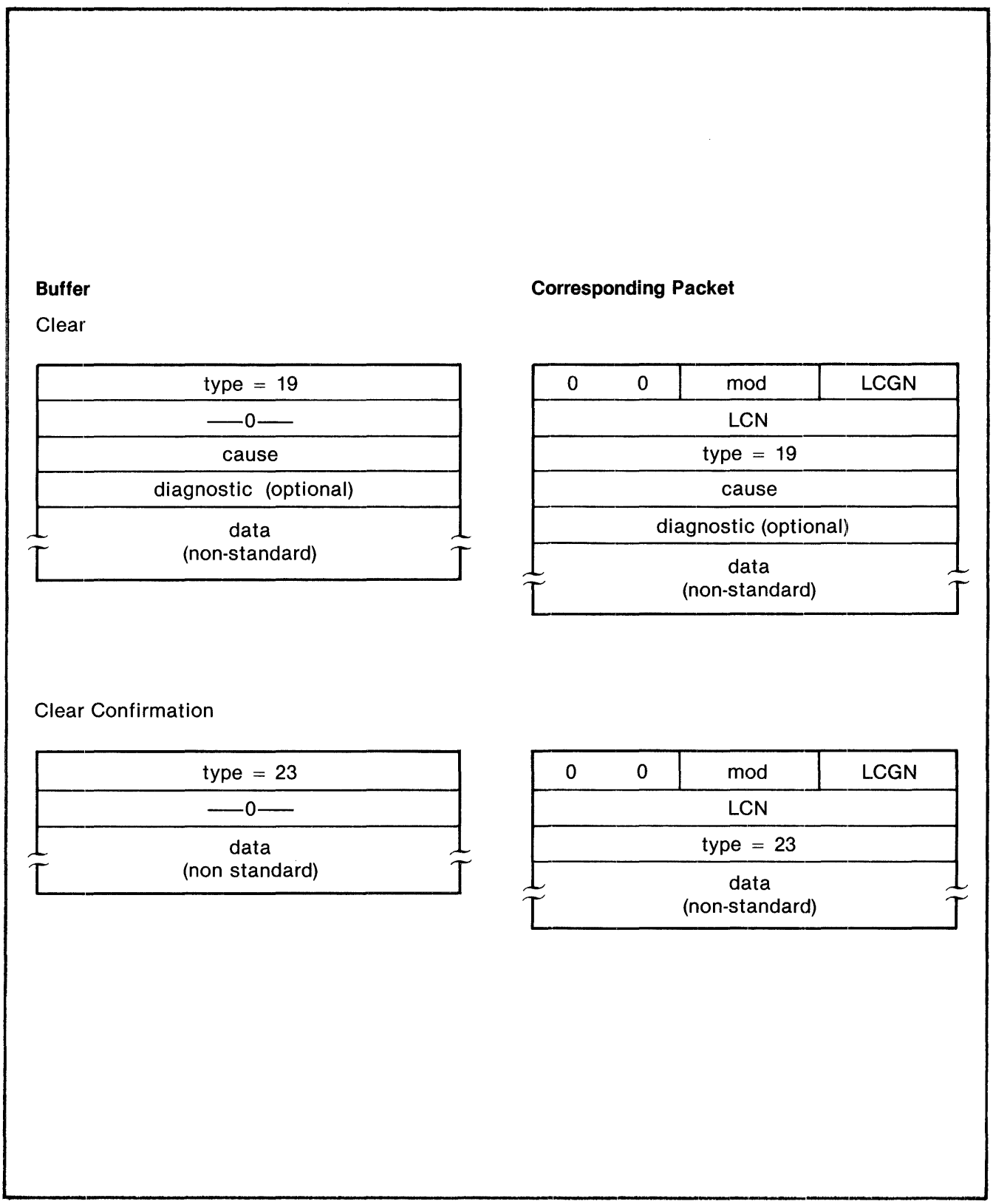


Figure 3-4. Valid Packet Types for PTP Mode 2 (Page 3 of 5)

Buffer

Reset

type = 27
——0——
cause
diagnostic (optional)

Corresponding Packet

0	0	mod	LCGN
LCN			
type = 27			
cause			
diagnostic (optional)			

Reset Confirmation

type = 31
——0——

0	0	mod	LCGN
LCN			
type = 31			

Interrupt

type = 35
——0——
data

0	0	mod	LCGN
LCN			
type = 35			
data			

Interrupt Confirmation

type = 39
——0——

0	0	mod	LCGN
LCN			
type = 39			

Figure 3-4. Valid Packet Types for PTP Mode 2 (Page 4 of 5)

Buffer

Diagnostic

type = 241
—0—
diagnostic
data

Corresponding Packet

0	0	mod	—0—
type = 241			
diagnostic			
data			
—0—			

Restart

type = 251
—0—
cause
diagnostic (optional)

0	0	mod	—0—
—0—			
type = 251			
cause			
diagnostic (optional)			

Restart Confirmation

type = 255
—0—

0	0	mod	LCGN
LCN			
type = 255			

Figure 3-4. Valid Packet Types for PTP Mode 2 (Page 5 of 5)

REQUESTER BUFFER FORMAT. The buffer format differs between packet mode 1 and packet mode 2. Packet mode 1 uses the level 3 communication services to provide level 4 services to the GUARDIAN file system user. Packet mode 1 does not provide access to the communication services of level 3.

Basically, packet mode 1 sends packets which are acknowledged by the network (not the receiving DTE) as having been delivered. Additionally, only data and interrupt packets are allowed.

Packet mode 2 provides the application program with the additional capability of (1) sending and receiving various packet types and (2) obtaining acknowledgement from the receiving application program that received the packet.

TRANSMITTING PACKETS. Packet modes 1 and 2 map each file system WRITE operation into one outgoing packet, and satisfy each file system READ operation with one incoming packet. The type and header information of the packet is mapped from (to) two bytes of information at the beginning of the WRITE (READ) buffer. A WRITE operation causes the transmission of a single packet whose type and contents are defined by the requester's buffer. (Refer to "Requester Buffer Format" described in the previous paragraph.)

As with packet mode 1, packet mode 2 also uses the first word of each READ, WRITE, or WRITEREAD buffer to store the MCW, which defines the associated packet type and the state of certain bits in the packet header.

Packet mode 2 allows users to read (incoming) and write (outgoing) portions of various packet types such as their facilities fields; user data fields; the D-bit in call establishment packets; the Cause and Diagnostic fields of Clear, Reset, and Diagnostic packets; and the Q-bit, M-bit, and D-bit in data packets.

For packet mode 2, a WRITEREAD operation functions similar to a WRITE except that the WRITEREAD does not complete until either (1) a packet is received that acknowledges the transmitted packet or (2) it is known that the transmitted packet will not be acknowledged. If the packet is acknowledged, the acknowledging packet is returned to the requester's buffer.

PACKET ACKNOWLEDGEMENT (PACKET MODE 2). Acknowledgements returned to the requester's buffer in response to the transmission of various packet types are listed and described in Table 3-3.

Table 3-3. WRITEREAD Request Acknowledgements

Packet Type	Acknowledgement
DATA	After acknowledgement, an RR or RNR packet is returned. If acknowledgement is a "piggy-backed" P(R), X25AM creates RR or RNR packet with same P(R) and returns it to requester's buffer. If circuit is reset, cleared, or restarted before DATA packet is acknowledged, RESET, CLEAR, or RESTART packet is returned to requester's buffer.
INTERRUPT	After acknowledgement, an INTERRUPT CONFIRMATION packet is returned to requester's buffer. If circuit is reset, cleared, or restarted before INTERRUPT packet is acknowledged, RESET, CLEAR, or RESTART packet is returned to requester's buffer.
RESET	After acknowledgement, a RESET CONFIRMATION packet is returned to requester's buffer. If circuit is reset, cleared, or restarted before RESET packet is acknowledged, RESET, CLEAR, or RESTART packet is returned to requester's buffer.
CALL REQUEST	After acknowledgement, a CALL ACCEPT packet is returned to requester's buffer. If circuit is cleared or restarted before CALL REQUEST packet is acknowledged, CLEAR or RESTART packet is returned to requester's buffer.
CALL ACCEPT	When a WRITEREAD causes transmission of a CALL ACCEPT packet, no data is returned to requester's buffer. Operation completes upon transmission of packet.
CLEAR	After acknowledgement, a CLEAR CONFIRMATION packet is returned to requester's buffer. However, if circuit is cleared or restarted before acknowledgement of CLEAR packet, a CLEAR or RESTART packet is returned to requester's buffer.

REQUEST QUEUING (PACKET MODES 1 & 2). Queuing of SETMODE, SETPARAM, CONTROL, READ, WRITE, and WRITEREAD requests are defined in Table 3-4.

Table 3-4. Request Queuing Execution

Request	Queuing
SETMODE	Requests may not be issued while other requests are pending. Attempt to do so will cause a return with error 27.
SETPARAM	Requests are executed in same order as issued, but before any pending CONTROL, READ, WRITE, or WRITEREAD requests.
CONTROL	Requests are executed in same order as issued, but before any pending READ, WRITE, or WRITEREAD requests.
WRITE	Requests, which are queued for execution, are executed in same order as issued.
READ	Requests, which are queued for execution, are executed in same order as issued.
WRITEREAD	Request is queued with WRITEs until its write half is executed. It is then queued with READs until its read half is executed. If WRITEREAD is issued and a READ is issued before write half of WRITEREAD is complete, READ is executed before read half of WRITEREAD.

The order in which requests complete depends on whether the subdevice is operating in the full duplex or half duplex mode. If the half duplex mode is used, requests complete in the same order in which they are issued regardless of the order in which they are executed. During full duplex operation, requests complete in the same order in which they were executed, regardless of the order in which they were issued.

The rules for the queuing of requests are independent of whether the subdevice is operating in the half duplex or full duplex mode. The operating mode affects only the file system queuing of requests executed by X25AM but not yet completed.

By default, PTP subdevices operate in the half duplex mode. Calling SETMODE (<subdevice>,30,0) selects the half duplex mode. The full duplex mode is selected by calling SETMODE (<subdevice>,30,1).

Since PTP subdevices execute requests in a full duplex manner, applications should always select the full duplex mode.

X25AM does not limit the number of request that may be queued for execution. The file system does, however, limit the number of file requests that may be queued for a device to 255. This is total number of requests that may be queued on a line for all subdevices.

If a CONTROL 11 (wait for modem connect) is attempted on a line that is already connected, an error 12 (file in use) is returned. Since no fatal error condition actually exists, normal operation will continue.

TWO-STEP READ (PACKET MODES 1 & 2). The two-step read operation allows a user process to allocate SHORTPOOL space only when needed for read operations. The two-step read consists of two file system requests.

The first request is a WRITEREAD issued with a write count of one, a read count of at least two, and an MCW with just the high-order bit turned on. This type of WRITEREAD request causes the PTP protocol to queue just the read portion of the request which, when executed, returns a one-word value. This word contains the actual byte count (plus two bytes for the MCW) of the data portion of the packet.

The second request is a READ whose byte count is the count returned by the read portion of the previously executed WRITEREAD request.

The following examples illustrate the difference between a normal read operation and a two-step read operation.

Normal Read Operation:

```
INT NETRDBUF [0:128];                ! read buffer.
    .
CALL READ (FILENUM,NETRDBUF,258)     ! read packet.
    .                                ! get space for 258 bytes.
    .
                                     ! read completes and NETRDBUF
                                     ! contains data.
```

Two-Step Read Operation:

```
INT NETRDBUF [0:128];
    .
    .
    .
NETRDBUF := %100000;                          ! Set up MCW for two-
    .                                         ! step read with the
CALL WRITEREAD (FILENUM,NETRDBUF,1,2);        ! write count = 1 and
    .                                         ! read count = 2.
    .                                         ! Read portion com-
    .                                         ! pletes, NETRDBUF
    .                                         ! contains byte count.
IF NETRDBUF.<0> AND NETRDBUF.<1:15> THEN
   CALL READ (FILENUM,NETRDBUF,NETRBUF.<1:15>) ! Read packet.
   ELSE CALL ERROR^ROUTINE                    ! Handle bad response.
                                              ! Read completes and
                                              ! NETRDBUF contains
                                              ! data from packet.
```

RECEIVING PACKETS. For packet modes 1 and 2, a READ operation is satisfied and completed when one packet has been received. The packet

type and contents are mapped into the data returned to the requester's buffer as previously described. For packet mode 2, the read phase of a WRITEREAD functions in the same manner.

INCOMING PACKET QUEUING. Note in the following discussion that all references to INTERRUPT and DATA packets are valid for mode 1 only.

To ensure that no data is lost, X25AM queues DATA, INTERRUPT, RESET, CLEAR, and RESTART packets received when no READ is pending. The maximum number of DATA packets that can be queued is equal to the level 3 window size. Only one INTERRUPT packet or one RESET packet can be queued. If the interface is reset, cleared, or restarted, all queued DATA, INTERRUPT, and RESET packets are discarded.

CALL packets are not queued. If a CALL packet is received when no READ or CONTROL 11 (wait for modem connect) is pending for a subdevice, X25AM will not route the incoming call to the subdevice.

CALL ACCEPT packets are not queued. A requester can, however, obtain the contents of a CALL ACCEPT packet by performing a WRITEREAD to initiate transmission of the CALL packet that elicited the CALL ACCEPT packet.

Only one CLEAR packet or one RESTART packet can be queued. If the interface is either cleared or restarted, a queued CLEAR packet or RESTART packet will be discarded.

CLEAR CONFIRMATION packets are not queued. A requester can obtain the contents of a CLEAR CONFIRMATION packet by performing a WRITEREAD to initiate transmission of the CLEAR packet that elicited the CLEAR CONFIRMATION packet.

DIAGNOSTIC packets are not queued. If no READ is pending when a DIAGNOSTIC packet is received, the DIAGNOSTIC packet is lost.

RECEIVED PACKET ACKNOWLEDGE (PACKET MODE 2). X25AM automatically acknowledges an incoming packet, other than a CALL packet, that has been delivered via a READ to the opener of the subdevice. DATA packets are acknowledged using level 3 flow control protocol. All other packet types are acknowledged using the appropriate confirmation packets. Since a CALL packet is not automatically acknowledged, the requester must acknowledge it by initiating the transmission of a CALL ACCEPT packet.

ERROR CONDITIONS (PACKET MODES 1 & 2). Since sending a RESTART packet will clear all calls to all subdevices on a device, this operation is not allowed. An attempt to write a restart buffer causes a return with error 179 (application buffer incorrect).

Whenever appropriate for mode 2, X25AM sends INTERRUPT CONFIRMATION, RR, RNR, REJ, RESET CONFIRMATION, CLEAR CONFIRMATION, and RESTART CONFIRMATION packets. A requester may not initiate the transmission of such packets. Any attempt to write such buffers causes a return with error 179 (application buffer incorrect).

A subdevice is always in some state such as calling, connected, disconnected, etc., but not all buffer types are allowed to be written in all states. A data buffer, for example, may not be written to a subdevice that is in a disconnected state. If an opener attempts to write a buffer whose type is invalid for the state of the subdevice, the WRITE (or WRITEREAD) will return an error 160 (request invalid for line state) and the associated packet will not be transmitted. For packet mode 1, error 140 (circuit is not connected) is returned.

Possible subdevice states and the valid buffer types for each state are listed in Table 3-5.

Table 3-5. Subdevice States and Valid Buffer Types (Packet Mode 2)

State	Valid Buffer Type
Clearing	Diagnostic
Cleared	Diagnostic
Ready	Diagnostic, Call
Calling	Diagnostic, Clear
Called	Diagnostic, Clear, Call Accept
Data Transfer	Diagnostic, Clear, Interrupt, Data
Reset	Diagnostic, Clear, Reset
Resetting	Diagnostic, Clear, Reset

HEADER BITS (PACKET MODES 1 & 2). Not all header bits may be set in a buffer of a given type. With the exception of the D-bit, Q-bit, or M-bit in a DATA packet, and the D-bit in a CALL or CALL ACCEPT packet, no other bit settings are allowed. If an opener tries to write a buffer in which a header bit is illegally set, the WRITE (WRITEREAD) returns with an error 179 (application buffer incorrect). The corresponding packet will not be transmitted.

PACKET CONTENT (PACKET MODES 1 & 2). X25AM interprets the contents of incoming and outgoing CALL REQUEST and CALL ACCEPT packets. Particular values in the facilities fields of such packets may be invalid. If invalid values are used, X25AM clears the associated circuit and the invalid incoming packet is not delivered to the requester. If the requester tries to send an invalid packet packet, X25AM clears the associated circuit and will not send the packet.

Invalid values in the facilities fields include the following:

1. For CALL packets, a packet size larger than the L3^PACKETSIZE defined at SYSGEN time.

2. For CALL ACCEPT packets, any facility that violates X.25 restrictions on facility negotiation.

PACKET SIZE (PACKET MODES 1 & 2). No WRITE or WRITEREAD buffer may be so long that is would result in the transmission of a packet longer

than the length of the longest allowed DATA packet. The maximum packet size is defined at SYSGEN time in the parameters of the X25AM macro. An attempt to write a buffer larger than the defined packet size will cause a return of error 21 (illegal count specified).

NONSTANDARD PACKET FORMATS (PACKET MODES 1 & 2). In addition to supporting all packet formats defined by CCITT Recommendation X.25 (1980), certain nonstandard extensions are supported by X25AM.

OVERSIZED PACKETS (PACKET MODES 1 and 2). A requester can transmit and receive packets longer than those defined by CCITT Recommendation X.25 (1980). However, X25AM will not transmit or receive packets that are longer than the DATA packet length specified at SYSGEN time. This is because the packets must fit into the allocated system data buffer space.

OVERSIZED PACKET TRANSMISSION (PACKET MODES 1 & 2). Packets longer than those defined by CCITT Recommendation X.25 (1980) can be transmitted by a requester if a buffer of sufficient length has been specified. Extra bytes in the requester's buffer will be mapped on a one-to-one basis into extra bytes in the transmitted packet.

If a requester attempts to transmit a packet that is longer than the length defined for a DATA packet, the WRITE (or WRITEREAD) operation will terminate immediately with an error 21 (illegal count specified).

OVERSIZED PACKET RECEPTION (PACKET MODES 1 and 2). A requester can receive a CLEAR or RESET packet that is longer than that specified by CCITT Recommendation X.25 (1980) by using a READ operation.

A requester can receive CLEAR CONFIRMATION, RESET CONFIRMATION, and INTERRUPT CONFIRMATION packets that are longer in packet mode 2 than those specified by CCITT X.25 (1980) by using a WRITEREAD operation to initiate transmission of the packet which elicits any of these confirmation packets.

If X25AM receives a packet that is longer than allowed by the defined input packet size (i.e., defined at SYSGEN time by PACKETINSIZE modifier), the level 2 frame containing the packet will be ignored at level 2. Eventually, the interface will be reset at level 2 and restarted at level 3.

FAST SELECT (PACKET MODE 2). Both incoming Fast Select packets with normal response and restricted response, as described by CCITT Recommendation X.25, are supported. Refer to Fast Select Support in Appendix A.

SECTION 4

NETWORK (NET) PROTOCOL

The NET protocol provides the means of connecting a local line handler on one Tandem system to a line handler on another Tandem system via an X.25 network. The NET protocol is used in conjunction with an EXPAND network line handler. Using the NET protocol, the two systems communicate with one another in the same manner as if they were connected by a leased line. (Refer to Figure 4-1.)

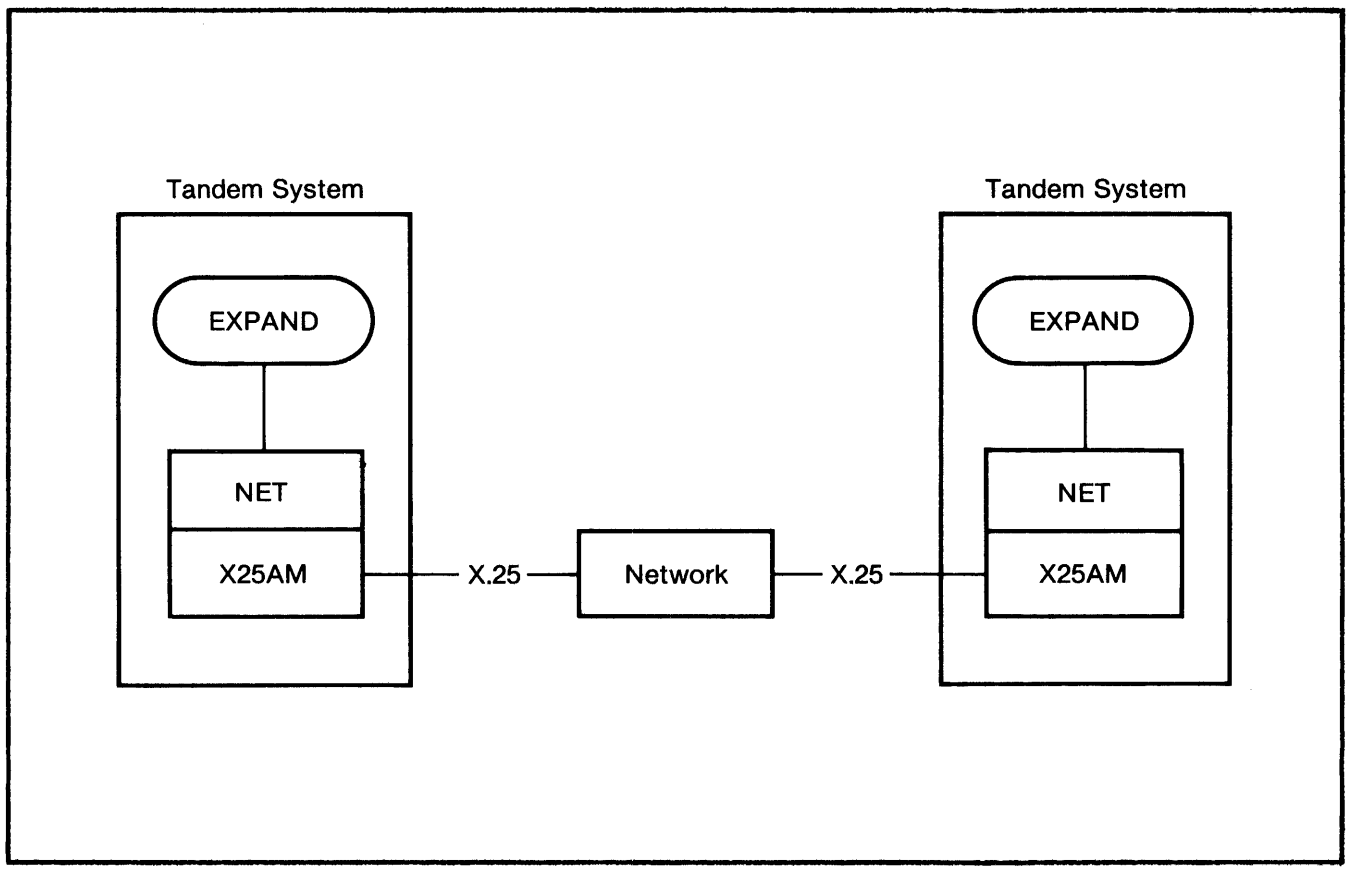


Figure 4-1. Example of NET Protocol Use

Although subdevices configured to use the NET protocol may not be accessed by application programs, several features are included in NET to allow the subdevices and the EXPAND line handler to communicate.

To use the NET protocol, the following must be defined:

1. device type and subtype
2. LDEV of EXPAND network line handler
3. unique next system number
4. X.25 network address of next system.

Details for specifying these requirements are given in Section 5, Configuration Requirements, and in the following manuals:

- For the Tandem NonStop System:

  NonStop System Management Manual

- For the Tandem NonStop II System

  NonStop II System Management Manual

SECTION 5

CONFIGURATION REQUIREMENTS

INTRODUCTION

The following three steps are required to configure a Tandem NonStop or NonStop II system for X25AM:

1. Establish an account with an X.25 network carrier.

2. Configure GUARDIAN operating system through SYSGEN to implement X25AM method and to establish number of circuits required for incoming and outgoing calls. Only one physical connection exists between the system and the X.25 DCE. This physical connection is configured as a logical device.

3. Add subdevices to this logical device using the Communications Utility Program (CUP).

ESTABLISHING ACCOUNT

The network carrier will provide the neccessary information about the X.25 packet switching network. In establishing the account, however, the carrier will request certain information. For example:

1. The number of switched virtual circuits required; i.e, the number of switched virtual circuits to be available for incoming and outgoing calls.

2. Whether ASCII or EBCDIC control characters will be used for framing on byte-synchronous interfaces. X.25AM assumes that EBCDIC code will be used, but this can be altered to ASCII through either SYSGEN or CUP.

The carrier will provide a network address, which is required by CUP for certain protocols. A network address, for example, might be the number 31102130000092. The first four digits represent the Data Network Identification (DNIC) of the network carrier and the following 10 digits are at the carrier's discretion. An example of this address in use is as follows:

```
@C 213 92
 213 92 CONNECTED          (user connects to network)
:LOGON INVENT.JAN
      .
      .                    (normal session)
      .
:LOGOFF
 213 92 DISCONNECTED       (user disconnects from network)
```

SYSGEN INFORMATION

System generation (SYSGEN) information for the Tandem NonStop and NonStop II systems is provided in the following manuals:

- NonStop System Management Manual
- NonStop II System Management Manual

COMMUNICATIONS UTILITY PROGRAM

This program (CUP) performs operations related to data communications lines, devices, and network environments. Through CUP, subdevices for X25AM may be added, deleted, or reconfigured using the ADD, DELETE, and ALTER commands, respectively. There are two additional commands (CONNECT and CLEAR) used with the NET protocol. Syntax and descriptions of these commands are provided in Volume 1.

Permanent Virtual Circuit Configuration

When an account is established between the DTE and the network, a mutually agreed upon range of consecutively numbered logical channel numbers (LCNs) is assigned for use as permanent virtual circuits (PVCs). (A logical channel is the unit of bandwidth allocated for the X.25 interface.) When the LCNs are assigned by the network carrier, X25AM must be configured accordingly using the ALTER command. For example:

```
>LINE $ACBC
>ALTER MINSVC xxx, MINPVC yyy, PVCS zzz
```

where

MINSVC is the smallest LCN assigned to SVC service. Default value is 1.

MINPVC is the smallest LCN assigned to PVC service. Default value is 0.

PVCS is the number of logical channels assigned to PVC service. Default value is 0.

If MINPVC or PVCS is altered by means of CUP so that logical channels previously assigned to PVCs are made available to SVCs, any subdevices that were bound to those logical channels are made available for SVC service.

Binding Subdevices

Any subdevice defined for an X.25 line may be bound to a PVC. The binding is performed by means of CUP and can be changed only by CUP. Either the ADD or ALTER command can be used for binding as follows:

```
>ADD #ABCD, PVC nnn

>ALTER #ABCD, PVC nnn
```

where nnn is the LCN of the PVC to which the subdevice is to be bound.

An error is returned if an attempt is made to bind a subdevice to a logical channel not assigned to PVC service, or to a logical channel to which another subdevice is already bound. In this case, an error message will be displayed but the configuration of the subdevice will remain unchanged.

Unbinding Subdevices

A subdevice can be made eligible for SVC service at any time. If a subdevice is bound to a PVC, it will be unbound. The ALTER command is used to perform unbinding as follows:

```
>ALTER #ABCD, SVC
```

Configuring Subdevices

CUP requests certain modifiers for each configuration command. The possible options for each modifier used with X.25 subdevices are as follows:

1. Required modifiers:

   PROTOCOL - defines type of protocol (ITI, PTP, or NET)

   RECSIZE - defines record size:

   ```
   ITI - 80 bytes
   PTP - application dependent
   NET - CUP sets this to PACKETSIZE from EXPAND network
   ```

   TYPE - defines subdevice type:

   ```
   ITI - (6,0) teletype compatible
         (6,1) 6510 terminal
         (6,2) 6520 terminal (point-to-point)
         (6,4) 6530 terminal (point-to-point)
   ```

PTP - (9,0) (type 9 specifies "foreign process")
NET - (63,0)

2. Optional modifiers:

[NO]ACCEPT - installation and application dependent:

ITI - ACCEPT makes it easier for terminal user
PTP - user determines use of subdevices
NET - NOACCEPT

Default for incoming call is NOACCEPT.

ADDR - used when particular subdevice must be configured to have a predetermined remote DTE address for calling OUT to that DTE. For example, NET subdevices must have proper DTE ADDR configured at both ends of EXPAND network path. ADDR is used with all X25AM subdevices.

PORT - aids in connecting incoming call to subdevice:

ITI - if 0 (default value), terminal user gets next available ITI subdevice without specifying anything special.
PTP - any mutually agreeable non-zero value
NET - any mutually agreeable non-zero value

PRICALL - to initiate call (for DATAPAC X.25 network only)

[NO]REQUEST - installation and application dependent:

ITI - not applicable since it cannot call out
PTP - application dependent
NET - if default is NOREQUEST, billing invoice reflects changes according to initiator.

3. Two modifiers apply only to NET subdevices and must be provided to establish EXPAND connection:

LHDEV - logical device number of EXPAND line handler that intends to use this subdevice to establish connection to other system.

NEXTSYST - system number of other system within EXPAND network.

Extended Format

CUP can be used to specify the use of extended format across the DTE/DCE interface at call connect time. The format of the command is:

```
>ALTER [NO] EXTFORMAT [, LINE <device>]
```

If extended format is specified, the call accept/call connect packets will contain the address length field and the facility length field even though either field may be zero. If extended format is specified but the call connect packet does not contain the length field, the call will be cleared.

Additional Parameters (For ITI Devices)

Parameters exist for specifying parity checking and parity types. The TRANSPAC network transfers all data between the host and the terminal in a transparent mode with parity added. Conversely, most other networks, specifically those available in North America such as TELENET, TYMNET, and DATAPAC, handle terminal transparency by keeping terminal-dependent handing in the X.3 PAD rather than in the host. As a result of these various methods of operation, the following ITI subdevice parameters have been included in X25AM:

```
NONE  - transmit all data characters with parity bit unchanged
EVEN  - transmit all data characters with even parity
ODD   - transmit all data characters with odd parity
CHECK - check all received data characters for correct parity as
        specified according to selection of ODD or EVEN.
```

The defaults for these parameters are NONE (instead of EVEN or ODD) and NOCHECK (rather than CHECK). Using the default parameters causes data characters to be transmitted with the parity bit set to zero. The parity bit for received data characters is ignored. The application can legitimately send and receive 8-bit data characters; however, ITI^PROTOCOL must recognize certain control characters, specifically CR (carriage return), BS (back space), EM (end of media), and CAN (cancel).

If the EVEN or ODD parameter is selected but the CHECK parameter is not, the X25AM ITI interface sets the parity bit to zero for all received data characters. If CHECK is specified, parity checking is performed and then the parity bit is set to zero.

Each of the parameters described above corresponds to existing SYSGEN modifiers.

Sample Subdevice Configuration

A sample subdevice configuration for an X.25 line is as follows:

```
>ADD #TERM1,PROTOCOL ITI,TYPE(6,1),RECSIZE 80,ACCEPT
        .
        .
        .
>ADD #TERM10, LIKE #TERM1
>ADD #TERM11, LIKE #TERM1, PORT 2
        .
        .
        .
```

```
>ADD #TERM20, LIKE #TERM11
        .
        .
        .
>ADD #TERM21, LIKE #TERM1, PORT 4
        .
        .
        .
>ADD #TERM30, LIKE #TERM21

>ADD #APP1,PROTOCOL PTP,TYPE(9,0),RECSIZE 80,PORT 44
>ADD #APP2 LIKE #APP1,PORT 55,ADDR 31102130009955
>ADD #EXPNET,PROTOCOL NET,TYPE(63,0),LHLDEV 23,NEXTSYS 3, &
            ADDR 31104150000199,PORT 99
```

TERMx specifies which terminal calling from packet network will be accepted. Defined with: ITI protocol, TYPE 6 to look like a terminal to application; RECSIZE of 80 since actual width of terminal is unknown; ACCEPT reverse charges; and default port number number of zero for first group of terminals (#TERM1 through #TERM10).

The second and third group of subdevices (#TERM11 through #TERM20 and #TERM21 through #TERM30) are assigned to ports 2 and 4, respectively. This means that on a TELENET network, for example, when a network call such as T 792 43 B or T 792 43 D is received, The B call will check port 2 for an available terminal and the D call will check port 4 for an available terminal. (The B = port 2 and D = port 4 match occurs because TELENET uses alphabet characters A-Z to represent numerals 1-26.) On TYMNET, this can be accomplished by setting up a user name that ends in at least two characters. For example, TANDEM02 could route to port 2.

The user may want to specify groups of ITI subdevices with a unique port number assigned to each group. This configuration allows the assignment of specific groups of terminals to specific applications.

In this example, terminal group #TERM1 through #TERM10 (default to port 0) could possibly have the Command Interpreter executing. Terminal group #TERM11 through #TERM20 (port 2) could be a Data Entry application. Terminal group #TERM21 through #TERM30 (port 4) could be Data Base Enquiry terminals.

#APP1 is a process-to-process port possibly used for accepting calls from foreign application attached to X25 network. Port number of 44 is used to identify this specific subdevice when processing INCOMING CALL packet. Subdevice refuses reverse charges. Foreign application attaches to this specific port by addressing it with port=44 in its CALL REQUEST packet. Applications agreed on a record size

of 80 characters. Specification of RECSIZE 80 is for information purposes and is returned by DEVICEINFO.

#APP2 is also used to communicate with foreign applications. The notable difference is that #APP2 has a predefined remote DTE address so that local application program does not have to call SETPARAM before calling out into X.25 network.

#EXPNET is an EXPAND network port for attaching one NonStop system to another through X.25 network. NEXTSYS 3 specifies EXPAND system number of system at other end of circuit.

LHLDEV identifies specific EXPAND line handler that will use this X25AM line handler.

Remote ADDR and PORT configured into each end of this X.25 circuit must accurately and completely attributes of both ends. An INCOMING CALL packet to a NET protocol subdevice is checked to see if it is from remote DTE specified by ADDR. This check provides a level of insurance for keeping unauthorized foreign system out of user's EXPAND network.

SECTION 6

FILE SYSTEM ERRORS AND CONSOLE MESSAGES

FILE SYSTEM ERRORS

Possible errors returned by file system calls are listed and described in Table 6-1. Also included are possible causes of these file system errors and suggested problem determination procedures.

Table 6-1. File System Errors

Error No.	Type	Meaning	Possible Cause/Solution
1	ITI	End of file	Terminal user has typed CTRL-Y.
2	ITI PTP	Operation not allowed	Application program logic probably in error. Make sure that attempted operation is valid for subdevice type.
12	PTP	File in use	Returned when attempting CONTROL 17 (enable connection) on a subdevice already connected. If permissible to share connection with another process, ignore error. If not, handle error as you would for a disk file.
21	ITI PTP	Illegal count specified	Usually means that given buffer is too large. Error is also returned to users of PTP subdevices operating in Packet Mode 1 or 2 when WRITE buffer size is less than two bytes.

→

33	ITI PTP	Unable to get IOPOOL space	Retry operation at intervals of 1 second. If unsuccessful, the processor containing I/O process needs more IOPOOL space.
60	ITI PTP	Device or sub-device has been DOWNed (since file was opened)	CLOSE file and then OPEN it again when operator has UPped device or subdevice.
61	ITI PTP	No file OPENs are permitted	Subdevice has reached its limit of allowable simultaneous OPENs. Retry OPEN when a process has closed subdevice.
110	ITI	Only BREAK access is per-mitted	Retry operation when owner of BREAK has handled break condi-tion.
111	ITI	Break operation aborted because of BREAK	If you are owner of BREAK, check $RECEIVE for break messages. Retry operation.
120	ITI	Parity error detected in incoming data	Ensure that parity checking is set correctly (see SETMODE). Take appropriate action to recover lost data.
122	ITI	Circuit reset by network or PAD	Transmitted or received data may have been lost. Take appropriate action to recover lost data. (See "Circuit Reset Cause and Effect" at the end of this section.)
140	ITI PTP	Circuit is not connected	Error returned if circuit has never been connected or has just been disconnected. If circuit has never been connect-ed, use CONTROL 11 or 17 to connect it. A disconnected circuit could have resulted from a CONTROL 12 by application pro-gram, by remote DTE, by packet switching network, or as result of link failure between local DTE and network.

→

			If circuit was disconnected by network or DTE, Clear Cause and Diagnostic bytes from X.25 Clear Request packet (by means of SETPARAM) will provide further information in standard CCITT format. Attempt reconnecting circuit with CONTROL 17 or abort operation. The cause of circuit disconnect and current link state can be retrieved by a SETPARAM. This will aid in determining recovery action.
156	ITI	Protocol error	For 6520/6530 block mode only. Attempted read/write operation rejected because of following invalid reply from terminal: - STX text block received for READ - SOH text block received for WRITEREAD - text block received in middle of WRITE operation Before retry, check write buffer for escape sequences that can cause invalid reply from terminal.
160	PTP	Request invalid for line state	PTP Packet Mode 2 only. Attempted operation would result in transmission of buffer, which is not allowed given current state of interface.
162	ITI	Operation time-out	No reply from terminal for more than maximum number of times.
163	ITI	EOT received	Transmission aborted by terminal.
164	ITI PTP	DISConnect received from network node and all calls have been cleared	Retry operation.

→

173	ITI	Transmission error	Excessive LRC errors in transmission between PAD and terminal.
178	ITI PTP	No address list specified	CONTROL 17 (enable connection) was attempted on subdevice for which no address was specified. Use SETPARAM to set address and retry operation.
179	ITI PTP	Application buffer incorrect	Contents of WRITE or WRITEREAD buffer not acceptable to subdevice. Error usually caused by error in logic of application program.
191	ITI	Device power on	Reset by terminal. Block mode terminal screen display destroyed and terminal changes to default conversational mode.
201	ITI PTP	Primary I/O process not accessible via message system	Retry operation when system has been repaired.
218	ITI PTP	Timeout occurred	Refer to "I/O Error 218" described later in this section.
250	ITI	System containing I/O process not accessible via EXPAND.	Retry when network has been repaired.
251	ITI	Clear	Clear by network, PAD, or ITI because of procedure errror.

OPERATOR CONSOLE MESSAGES

The following messages are logged on the operator console to record the making and breaking of the link connecting the host system to the X.25 network:

```
"nn hh:mm ddmmmyy FROM sss,cc,pp LDEV ll X25: LINE READY"
"nn hh:mm ddmmmyy FROM sss,cc,pp LDEV ll X25: LINE NOT READY;
 ERROR mmm"
"nn hh:mm ddmmmyy FROM sss,cc,pp LDEV ll X25: LINE QUALITY nnn"
"nn hh:mm ddmmmyy FROM sss,cc,pp LDEV ll X25: LCN aaa ERR bbb
 CAUSE ccc"
```

"nn hh:mm ddmmmyy FROM sss,cc,pp LDEV ll UP"
"nn hh:mm ddmmmyy FROM sss,cc,pp LDEV ll DOWN"

where

nn	=	message number
hh:mm	=	time of day (hours:minutes)
ddmmmyy	=	date (day month year)
sss	=	system
cc	=	cpu number
pp	=	pin number
ll	=	logical device number
mmm	=	file system error number
nnn	=	line quality (percentage of good messages to total messages received.
aaa	=	X.25 logical channel in error
bbb	=	file system error code as follows: 122 - LCN reset by X25AM due to cause/reason error code 201 or 202 (see "ccc" following) 140 - LCN cleared by X25AM due to cause/reason error code 203 (see "ccc" following)
ccc	=	cause/reason code of LCN error as follows: 201 - invalid P(R) received in packet 202 - invalid P(S) received in packet 203 - packet size greater than negotiated size

CONSOLE REPORTED ERRORS

A modem status error (error 140) is reported by X25AM when one of the following conditions occurs:

1. Data Set Ready signal not detected within 30 seconds after sub-device is made ready by Communications Utility Program UP command.

2. Data Set Ready, Carrier Detect, or Clear to Send signal is lost.

Item 1 above will cause the following message to be displayed on the console:

```
X25:  LINE NOT READY (000)
```

Item 2 above will cause the following message to be displayed on the console:

```
I/O ERROR
```

The controller status is contained in the first parameter of the message with the high-order bits indicating the modem status as follows:

```
status <13> = Carrier Detect
status <14> = Clear to Send
status <15> = Data Set Ready
```

Clarification of console reported error messages is provided in table 6-2.

Table 6-2. Clarification of Console Reported Error Messages

1. LDEV XX UP

   This message indicates that line (X25AM) is logically up (via PUP UP command). Message does not convey any information about link between host system and packet network. Actual state of link is described by X25 LINE READY or X25 LINE NOT READY message.

2. LDEV XX DOWN

   This message indicates that line is logically down and all activity on line is stopped. Message is usually accompanied by an "I/O Error" message indicating cause of error. Line is down when:

   a. operator issues PUP DOWN command

   b. X25AM receives DISC command from network

   c. unrecoverable modem error on transmit

   d. unrecoverable error in controller/channel/CPU (level 200 or higher error).

   Level 200 errors are "hard" errors. When this type of error is encountered, X25AM will switch controller a number of times and attempt retransmission after each switch. If retransmissions are unsuccessful, line will be brought down. The most common errors are 214 (channel timeout) and 218 (interrupt timeout), which usually indicates either modem is not powered on or is not properly connected to controller.

   After line goes down, which should occur only under severe error conditions, all connections to network are lost. Line can only be brought back up again with PUP UP command, which causes X25AM to abort (or close) all applications associated with line. This means that all applications must be restarted. This is an inconvenience but it is important that X25AM ensure the integrity and security of system because there is no guarantee that application can be reconnected to same remote user/application.

→

3. X25 LINE READY

   This message is displayed in conjunction with the LDEV XX UP message. Message indicates that line is logically up at frame level, (SABM(SARM)/UA) has been exchanged with network, and line is ready for call establishment.

4. X25 LINE NOT READY, ERROR XXX

   This message is displayed in conjuction with the LDEV XX UP message. Message indicates that line is logically up but (SABM(SARM)/UA) has not been successfully exchanged with network. Error code 000 is displayed if X25AM is unable to bring link up (at frame level) after six attempts. Line, however, remains up and X25AM will continue to send SARM/SABM at 3-second intervals.

   Other common error codes are 140 (modem error detected) and 160 (DISC received from network).

5. Error 140 returned in WRITE/READ/WRITEREAD/CONTROL

   X25AM aborts file system requests with error 140 when:

   a. unable to transmit request due to modem error

   b. subdevice has not been connected over network or was connected and then cleared.

   Application unable to distinguish between these cases.

I/O ERROR 218

This error is reported whenever the driver times out while performing a frame write to the controller. The time allowed is specified with L2TIMEOUT at SYSGEN time. Any frame must be written within the specified time or an I/O error will result. Some causes of an error 218 are:

1. Modem not physically connected to controller.

2. Incorrect type cable between controller patch panel and modem.

3. Controller not configured to match modem type (e.g., RS-232 or RS-422). For byte synchronous controller, configuration is handled by switches on board itself. For bit synchronous controller, configuration is set up via software (i.e., SYSGEN or CUP).

4. Value assigned to L2TIMEOUT at SYSGEN time is too small for the specified PACKETSIZE and modem speed.

5. Possible hardware problem due to failure to detect clock signal sent from modem to controller.

TRAP HALTS

Certain errors will cause the operating system to suspend application and system processing in the associated processor. When such an error occurs, an error number is displayed in the display register. Table 6-3 lists the halt codes, reporting level of software, and the reason for the halt.

Table 6-3. Trap Halts

HALT Code		Reported By	Reason
NonStop I	NonStop II		
%000500	%047500	Level 3	State tables invalid.
%000501	%047501	Level 3	Bad free list.
%000502	%047502	Level 3	Bad frame list.
%000503	%047503	Level 2	Call to nonexistent level 4 protocol.
%000504	%047504	Level 2	CKTCB points to invalid subdevice.
%000505	%047505	Level 2	State tables invalid.
%000506	%047506	Level 2/4	Invalid state transition.
%000507	%047507		Insufficient dedicated buffer space.

CIRCUIT RESET CAUSE AND EFFECT

Circuit reset is initiated by X25AM whenever one of the following events occurs (see error 122 in Table 6-1):

1. A reject (REJ) packet is received.

2. A DATA packet with an invalid, or out of sequence, P(S) is received.

3. A backup process takeover occurs due to a primary CPU failure. This ensures that both ends of call regain synchronization.

The X.25 network generates a circuit reset if it detects any condition that could result in a lost or out-of-sequence DATA packet. Some such conditions are:

1. network congestion

2. network mode failure

3. existing, but undetected, FCS errors

4. system hardware or software errors.

Circuit resets initiated by the network or by the host system generally result in the following conditions:

1. Any data packet or packets in either data stream are lost.

2. An I/O request pending for application is completed with error 122 (possible data lost). Error 122 generated regardless of whether or not any data packets exist in data stream. Error notification occurs when (a) RESET CONFIRMATION packet is received in response to Tandem RESET REQUEST packet, or (b) when RESET INDICATION packet is received from network.

3. If no I/O REQUEST PENDING exists for application when circuit reset occurs, application will not become aware of possible loss of data. Since user application may not have a chance to reissue its I/O request on an ownership change to backup process before circuit reset occurs, error 122 might not be issued to application even though data might have been lost. (Refer to "Line Reset" information for TRANSPAC network in Appendix B.)

SECTION 7

X.3 PACKET ASSEMBLER/DISASSEMBLER

INTRODUCTION

The X.3 packet assembler/disassembler (X3PAD), which is based on CCITT Recommendation X.3, enables an interactive terminal user on a host system to use an X.25 network via the X25AM access method. Possible uses of the X3PAD include data base inquiries, accessing network mail services, and simple conversational sessions between terminals.

This section describes the PAD functions, gives an overview of the PAD operation, lists and describes the terminal commands, and describes the interface to a remote DTE.

X3PAD PROGRAM FUNCTION

The primary function of the X3PAD program is to pass data between a terminal and a network. Buffering and formatting functions are performed by the X3PAD and can be controlled either by commands from the terminal or by control packets from the network. The X3PAD can also be used to pass data from GUARDIAN files to the network or pass data from the network to GUARDIAN files.

Before you can use the X3PAD, the following requirements must be met:

1. X.25 network line connection must exist
2. X25AM subdevice must be configured for appropriate level 4 protocol
3. address of remote DTE must be available.

Section 5, Configuration Requirements, provides details on how to meet these requirements.

Figure 7-1 illustrates the use of the X3PAD on a Tandem system. The file system facilities are used by the X3PAD for accessing an interactive terminal. Various file types are accessed for file transfers, for hard copy of an output stream, and for an alternate input stream.

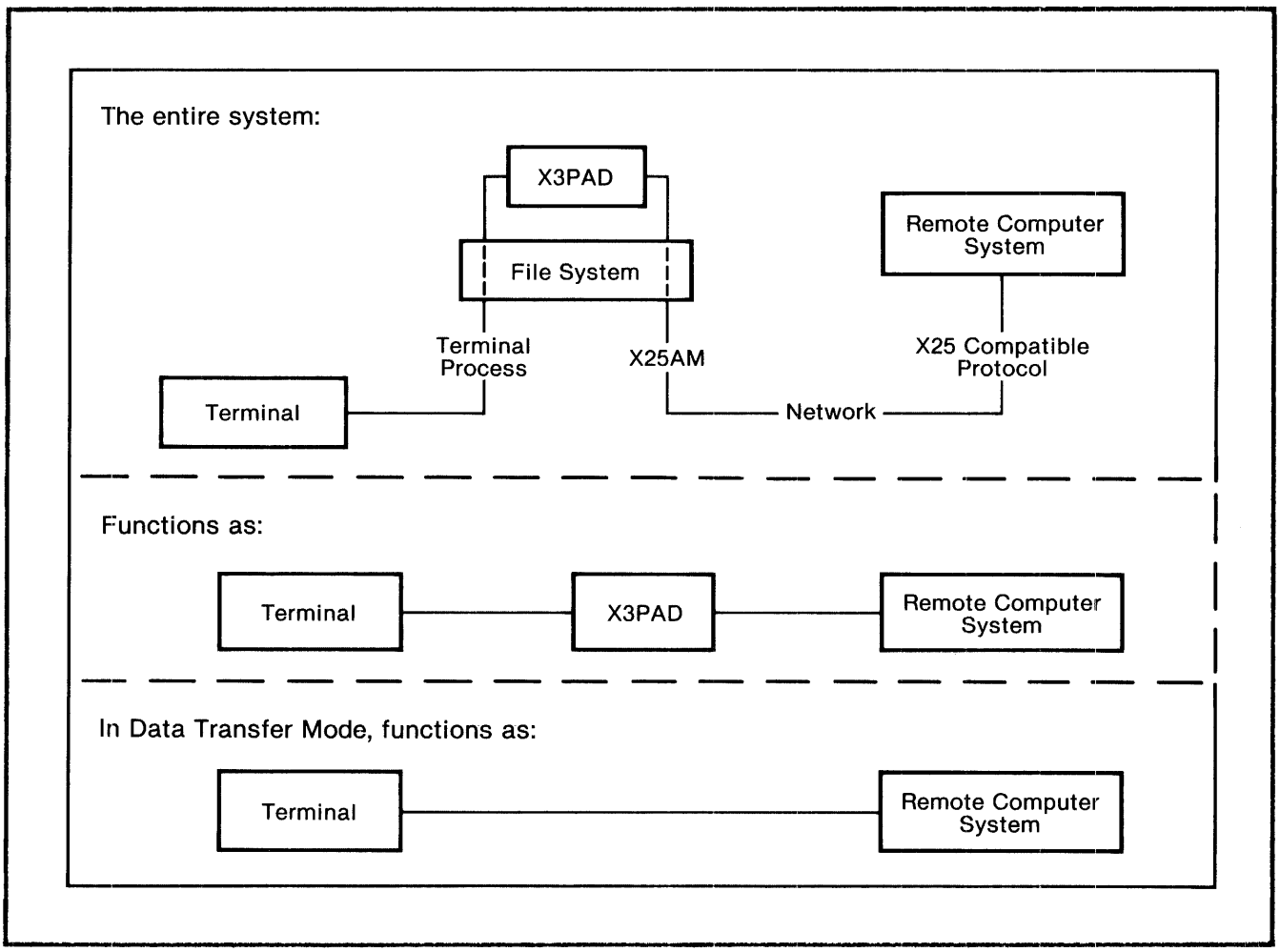


Figure 7-1. X3PAD Functional Diagram

The X3PAD can write an image of a terminal's output stream to a file. This capability allows saving a hard copy of messages received via a network mail service such as TELEMAIL on the TELENET network.

SAMPLE X3PAD SESSION

The following example illustrates an X3PAD session. This example is presented here merely to give a general idea of how the X3PAD is used. A more detailed example is given later in this section.

```
:X3PAD $X25.#PTP                              !Run X3PAD program
} set echo on, e$iting on, linefold 80        !Set PAD parameters
} call 123-456-78911                          !Call remote system
CALL CLEARED. THAT NUMBER NOT IN SERVICE.     !Call failed
} call 123-456-78910                          !Change address and retry
CALL ESTABLISHED                              !Call established
```

At this point the user's terminal functions as though connected to the remote host system. Normally, the user logs on, performs processing, and then logs off as follows:

```
WELCOME TO THE VWXYZ TIMESHARING SYSTEM.
PLEASE LOG ON: abcdef                          !Log on remote system
PASSWORD: jklmno
GOOD AFTERNOON, abcdef
     .                                         !Transfer information
     .                                              .
     .                                              .
LOGOFF                                         !Log off remote system
CALL CLEARED BY CALLED PARTY.
ADDRESS 12345678910                            !Results from clearing
54 PACKETS SENT                                !call
400 PACKETS RECEIVED                                .
0:20:33 ELAPSED TIME                                .
} exit                                         !Halt X3PAD program
:
```

OPERATIONAL OVERVIEW

When the X3PAD is invoked using the Command Interpreter, the syntax of the command is:

```
X3PAD [ /[ IN  <terminal>     ],.../] <netfile> [ ; <command> ] ...
         [ OUT <receive file> ]
```

The execution of this command causes the X3PAD to read the terminal commands and data from the IN <file>. The initial input from the IN <file> consists of terminal commands.

The X3PAD writes command prompts, diagnostic information, and data received from the network to OUT <file>. If OUT <file> is blank, the information is discarded.

The <netfile> parameter must be the name of the X25AM subdevice. This subdevice must be configured for the PTP protocol of X25AM. The X3PAD accesses the network to which the subdevice is connected. (Note that references to "the network" appearing in descriptions of the X3PAD operation may be read as references to <netfile>.

The command parameter must be a valid X3PAD command. Commands are executed as though they appeared in the IN <file>. When execution is completed, the X3PAD reads terminal commands and/or data from the IN <file>.

The X3PAD opens its input file, which is normally a terminal, and also opens an X25AM subdevice. A network call is established by either a CALL command in the IN <file> or some entity in the network calling the host system.

Data is read from the network via X25AM and also from the input file. Characters read from the input file are assembled into packets, which are then written to the network. Packets read from the network are disassembled into their constituent characters and then written to the X3PAD output file (which is usually the same terminal). The assembly and disassembly of packets is a function of X25AM.

Some packets read from the network may be interpreted by the X3PAD as PAD commands. These commands are used to set the parameters used for packet assembly and disassembly. PAD command packets are identified by the value of the Q-bit at the beginning of the packet as follows:

Q-Bit Value	Meaning
0	PAD Command Packet
1	Data Packet

Data and commands contained in the input file are distinguished in a different manner by the X3PAD, which is always in one of two modes: data transfer mode or command mode. The X3PAD switches modes in response to certain character sequences or terminal commands and, as a result, the input file can contain commands as well as data.

In the data transfer mode, characters from the input file are assembled into packets and transmitted across the network. In the command mode, characters in the input file are processed as X3PAD commands.

The X3PAD can be configured so that it changes from the data transfer mode to the command mode when a particular ESCAPE character is read from the input file. This character may be specified by the ESCAPE operational parameter of the SET command described later in this section.

Pressing the BREAK key may also cause the X3PAD to switch from the data transfer mode to the command mode, depending on the setting of the BREAK operational parameter.

The network can also cause the X3PAD to switch modes. For example, when an established call is cleared by the network, the X3PAD enters the command mode. Note, however, the the X3PAD does not enter the data transfer mode in response to the establishment of a call by the network.

The current X3PAD mode affects only the interpretation of characters read from the terminal. Data packets read from the network are always disassembled and written to the terminal regardless of the current X3PAD mode. Similarly, command packets read from the network are always interpreted and obeyed regardless of the current X3PAD mode.

TERMINAL COMMANDS

The terminal commands used in X3PAD include most of the functions defined by CCITT Recommandation X.28. These functions include establishings calls, clearing calls, and controlling the buffering and formatting of transmitted data. In addition to these X.28 functions, certain Tandem system standard functions such as FC, HELP, and EXIT are also available. Table 7-1 summarizes the terminal commands, each of which is described later in this section.

Table 7-1. Terminal Command Summary

Command	Purpose
BREAK	Generate interrupt.
CALL [<address>]	Establish call and enter data transfer mode.
CLEAR	Clear current call.
ENV [LOG / OUT / RECEIVE / VOLUME] , ...	Display status of LOG, OUT, or RECEIVE file, or default volume or subvolume.
EXIT	Clear current call and halt.
FC	Edit and reexecute previous command.
HELP [<command> / <symbol>]	Display syntax and synopsis of terminal commands.
LOG TO <file>	Make copy of terminal output.
LOG STOP	Terminate copying terminal output to LOG file.
OBEY <file>	Read terminal commands from file.
OUT <file>	Redirect terminal output.
RECEIVE TO <file>	Make copy of data received from network.
RECEIVE STOP	Terminate writing to RECEIVE file.
RESUME	Enter data transfer mode.
SEND <file>	Send contents of file over network. If file is not specified, SEND command functions same as RESUME command.

→

SET { <param> <value> } , ...	Set X3PAD operational parameters.
SHOW [<param> , ...] | ALL	Display X3PAD operational parameters.
STATUS	Display status of current call.
SYSTEM [<system>]	Set to default system name.
VOLUME { $<volume> | [$<volume>.] <subvol> }	Set default for volume and/or subvolume.

DIAGNOSTIC MESSAGES

There are two types of diagnostic messages: (1) notifications of successful completion of commands and (2) detected errors due to incorrect command syntax or incomplete operations due to malfunctions in the hardware or software. Syntax errors are flagged and explanatory messages are written to the output file. Messages specific to X3PAD commands are described in this section under the applicable command.

The current call can be cleared at any time, with or without intervention by the terminal user. When a call is cleared, a status indication is written to the output file. The status indication may be one of the following:

CALL CLEARED {
- .
- : LOCAL PROCEDURE ERROR.
- : REMOTE PROCEDURE ERROR.
- . THAT NUMBER IS NOT IN SERVICE.
- . THAT NUMBER IS BUSY. TRY AGAIN LATER.
- . THAT NUMBER WILL NOT ACCEPT REVERSE-CHARGE CALLS.
- . THAT NUMBER WILL NOT ACCEPT "FAST SELECT" CALLS.
- . NETWORK IS BUSY. TRY AGAIN LATER.
- . NETWORK WILL NOT ALLOW YOU TO CALL THAT NUMBER.
- . A FACILITY WHICH WAS REQUESTED IS NOT AVAILABLE.
- BY THE CALLED PARTY.
- BY THE CALLED PARTY, USING 'INVITATION TO CLEAR'.

}

In addition to the information given by the call clearing indication, bytes of the diagnostic data may also be received from the network. This information is written to the output file in the form:

```
DIAGNOSTIC OCTET #n = xxx
```

where

#n is a sequential octet number and xxx is a decimal integer in the range of 0-255.

X3PAD USE EXAMPLE

A typical use of the X3PAD is accessing a network mail service such as TELEMAIL on the TELENET network. An example of such use follows. In this example, the X3PAD $X25.#PTP command initializes the X3PAD program. (Note that a PTP device must exist.)

All terminal output to the network follows the } prompt and requests for input such as "User name?", "Password?", and "Command?". All other information is from the network (TELENET in this case).

```
:X3PAD $X25.#PTP
}call 123456789012
CALL ESTABLISHED

User name?  Tandem
Password?  A

Welcome to TELEMAIL!  Your last access was Monday, May 1, 1982
1:22 PM

CHECK these bulletin boards:
  TELENET

No.   Delivered      From           Subject                    Lines

 1    1 Apr 13:24    TANDEM         SEMINAR IN LOS ANGELES     3

Command?  help
```

When the system issues the "Command?" prompt, it is waiting for instructions from you. The TELENET commands are as follows:

ANSWER	DELETE	FORWARD	PASSKEYS	SCAN
BYE	DIRECTORY	INSERT	PURGE	SEND
CANCEL	DISPLAY	LIST	READ	TRANSFER
CHECK	EDIT	MEMBERS	REMOVE	UNPURGE
COMPOSE	EXIT	MODIFY	REPEAT	
COPY	FILE	NUMBER	SAVE	

You can obtain detailed information on any of these commands by entering a question mark followed by the command name. For example:

```
Command?  ? SCAN


** Control Characters **

Control H - backspace
Control X - line delete
```

```
Control R - replay current line
Control S - stop printing
Control Q - resume printing
Break     - usually returns you to a point where you can
            reissue instructions
```

A period on a line by itself (followed by a carriage return) indicates the end of text. Use this to get out of the COMPOSE and INSERT modes.

```
}status
ADDRESS 123456789012
17 PACKETS SENT.
37 PACKETS RECEIVED.
0:02:59 ELAPSED TIME.

Command? read 1
Posted:  Thu Apr 1, 1982   1:24 PM PST          Msg: SGCR-1257-3447
From:    TANDEM
To:      TANDEM
Subj:    SEMINAR IN LOS ANGELES

PLEASE ATTEND THE TANDEM SEMINAR IN LOS ANGELES ON THURSDAY
APRIL 1, 1982.  IT WILL START AT 9 AM AND BE HELD AT THE
BONAVENTURE HOTEL.

Command?  logoff

This TELEMAIL session is now complete.
CALL CLEARED.  CLEARING CAUSE = 0.
DIAGNOSTIC OCTET #1 = 0.
ADDRESS 123456789012
13 PACKETS SENT.
29 PACKETS RECEIVED.
0:05:35 ELAPSED TIME.
```

COMMAND DESCRIPTIONS

Syntax and descriptions of the terminal commands listed in Table 7-1 are presented on following pages.

BREAK Command

This command, which generates an interrupt and is functionally identical to pressing the BREAK key while in the data transfer mode, is of the following form:

```
BREAK
```

Various effects can be caused according to the setting of the BREAK operational parameter of the SET command. Pressing the BREAK key while the X3PAD is waiting for input causes the Command Interpreter to gain control of the terminal. Pressing the BREAK key while the X3PAD is executing a command causes the X3PAD to abort that command and prompt for input. Pressing the BREAK key while in the data transfer mode causes the X3PAD to retain control and act according to the setting of the BREAK operational parameter(s).

CALL Command

This command, which establishes a call to the remote DTE, is of the following form:

```
CALL [ <address> ]
```

This command establishes a call to the remote DTE specified by the X3PAD operational parameter ADDRESS. (Refer to the SET command.) The <address> parameter specifies a network user and is similar to a telephone number used for call establishment. When the remote DTE initiates a call, ADDRESS is set to that DTEs address. ADDRESS may be set by the terminal user only when no call is in progress. If the current value of ADDRESS is undefined, the X3PAD writes an error message to the output file and remains in the command mode.

After the CALL command has been successfully executed, the X3PAD enters the data transfer mode and data entered from the terminal is assembled into packets and sent to the remote DTE.

The effect of "CALL nnn" is identical to the effect of:

```
SET ADDRESS nnn; CALL
```

Refer also to the descriptions of the CLEAR, RESUME, and STATUS commands.

Messages:

```
CALL ESTABLISHED
<clear indication>
ERROR:  CALL ALREADY IN PROGRESS : <call status>
ERROR:  ADDRESS IS UNDEFINED
```

Refer to the STATUS command for the <call status> syntax. Refer to described under the STATUS command. For the <clear indication> syntax, refer to Diagnostic Messages presented earlier in this section.

CLEAR Command

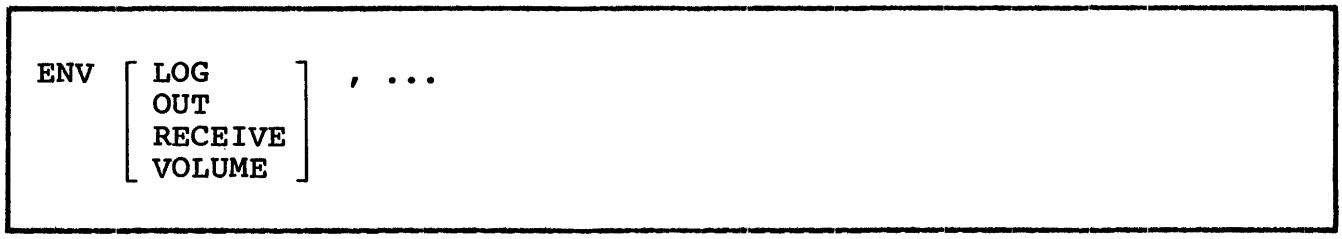
This command, which clears the current call, is of the following form:

```
CLEAR
```

The CLEAR command is analogous to hanging up the phone. The X3PAD signals the successful clearing of a call with a CALL CLEARED status indication. (Refer also to the CALL command.) If no call is in progress, the X3PAD writes the diagnostic message "THERE IS NO CALL IN PROGRESS" to the output file.

ENV Command

This command displays the status of the LOG, OUT, or RECEIVE file, or the default volume and subvolume. The ENV command is of the following form:



EXIT Command

This command, which clears the current call and halts the program, is of the following form:

```
EXIT
```

If the EXIT command is detected in the input file, the call currently in progress is cleared and the X3PAD halts.

An EXIT command detected in an OBEY file has a stronger effect than does an end-of-file, which causes the X3PAD to resume taking input from the previous input file. EXIT causes the X3PAD to halt immediately.

FC Command (Fix Command)

This command, which allows you to edit and reexecute the previous terminal command, is of the following form:

```
FC
```

Executing the FC command causes the display of the previous line of terminal commands followed by a prompt at the beginning of the next line. Input from the terminal may then be used as positional editing information in a manner similar to the FIX command in the EDIT program. The terminal user may edit the previous line and then indicate satisfaction with the edited form by pressing the RETURN key to execute the edited command(s) Alternatively the user may abort the FC command by typing "//". In this case, the line is restored to its original (unedited) form and no further action is taken.

HELP Command

This command displays the syntax of the indicated terminal command. A synopsis of the meaning of each command is displayed along with the syntax of the command. The HELP command is of the following form:

```
HELP [ <command> ]
     [ <symbol>  ]

where

  <command>

    is the name of the command.

  <symbol>

    can be any syntactic entity in the X3PAD terminal command
    language.
```

If no <symbol> is specified, <command> is assumed.

LOG Command

This command, which makes a copy (record) of the X3PAD session, is of the following form:

```
LOG | TO <file> |
    | STOP      |

where

  <file> := [ [ [ \<system>. ] $<volume>. ] <subvol>. ] <filename>

    may be any type of GUARDIAN file.
```

If LOG TO <file> is specified and the command is successfully executed, the X3PAD begins writing a copy of the data appearing on the output file to <file>. (Data written to the LOG <file> is filtered. Refer to Appendix B.) Any existing LOG file is closed and the message I HAVE CLOSED THE LOG FILE <file> is displayed. If the new file cannot opened, the old file remains open and the following messages are displayed:

```
ERROR: I CANNOT OPEN <file> (ERROR n)
THERE IS NO CURRENT LOG FILE
```

If <file> does not exist, and EDIT file is created. If <file> already contains data, it is not erased. Instead, the new data is added to the end of the file and a message is written to the output file.

If the X3PAD already has a log file open and a LOG command is detected that either (1) specifies a <file> or (2) uses the STOP keyword, the old log file is closed. In this case, a message is written to both the output file and the new log file.

If an unrecoverable error occurs while writing to a LOG file, the file is closed and a message is written to the output file.

The LOG command has no effect on data written to the current RECEIVE or OUT files. (Refer to the OUT and RECEIVE commands.)

A LOG STOP command terminates the write operation.

Messages:

```
ERROR: I CANNOT OPEN <file>.
I HAVE CLOSED THE LOG FILE <file>.
I HAVE CREATED AN EDIT FILE NAMED <file>.
THE CURRENT LOG FILE IS <file>.
THERE IS NO CURRENT LOG FILE.
```

OBEY Command

This command, which reads terminal commands from a file, is of the following form:

```
OBEY <file>

where

  <file> := [ [ [ \<system>. ] $<volume>. ] <subvol>. ] <filename>

    may be any type of GUARDIAN file.
```

The X3PAD takes input from <file> until an EXIT command or the end of <file> is detected, at which point the file is closed. The X3PAD then resumes taking input from the input file that contains the OBEY command.

The <file> may contain a CALL, RESUME, or SEND command followed by data. If the command is followed by data, the X3PAD enters the data transfer mode and the data is sent over the network. If the X3PAD is in the data transfer mode when the end of <file> is detected, it does not return to the command mode after <file> is closed.

During the execution of an OBEY command by the X3PAD, information is written to the output file so that the output appears as though the characters in <file> were read from the original input file.

No more than four OBEY files may be open at the same time.

Refer also the the SEND command.

Messages:

ERROR: I CANNOT OPEN <file>.
ERROR: YOU MAY NOT NEST MORE THAN 4 OBEY FILES.

OUT Command

This command, which redirects the terminal output, is of the following form:

```
OUT [ <file> ]

where

  <file> may any type of GUARDIAN file.
```

When the OUT command is successfully executed, the current file is closed and the subsequent output is written to <file>. If <file> does not exist, an EDIT file is created.

If the X3PAD detects an OUT <file> command and the file is successfully opened, a message naming the old output file is written to the new output file. (Data written to the new output file is filtered. Refer to Appendix B.) If this operation is also successful, a message naming the new output file is written to the old output file and the old output file is closed.

The OUT command has no effect on the writing of data to the file specified in a previous or subsequent RECEIVE or LOG command.

If an unrecoverable error occurs while writing to an OUT file, the X3PAD halts.

Messages:

ERROR: I CANNOT OPEN <file>.
I HAVE CLOSED THE OUTPUT FILE <file>.
I HAVE CREATED AN EDIT FILE NAMED <file>.
I WILL DIRECT SUBSEQUENT OUTPUT TO <file>.

RECEIVE Command

This command, which copies data from the network to a file, is of the following form:

```
RECEIVE  { TO <file> }
         { STOP      }

where

  <file> may be any type of GUARDIAN file.
```

When RECEIVE TO <file> is specified and the command is successfully executed, the X3PAD begins writing a copy of data read from the network to <file>. Each data packet read from the network is written to <file> in one WRITE operation.

If <file> does not exist, an EDIT file is created and a message is written to the output file. If a new RECEIVE file is specified, any existent old file is closed and the message I HAVE CLOSED THE RECEIVE FILE is displayed. If the new file cannot be opened, the old file remains open and the following messages are displayed.

```
ERROR: I CANNOT OPEN <file> (ERROR n)
THERE IS NO CURRENT RECEIVE FILE
```

If the <file> already contains data, execution of the RECEIVE command does not erase it. Instead, the new data is added to the end of the file.

If the X3PAD is writing to a RECEIVE file and a RECEIVE command is encountered, a message is written to the output file.

If an unrecoverable error occurs while writing to a RECEIVE file, the file is closed and a message is written to the output file.

A RECEIVE STOP command terminates the copying of data to the <file>.

Note: If <file> is an EDIT file, it is possible to write control characters to <file> that are unacceptable to EDIT. If the data packets read from the network were sent by an X3PAD using the SEND command, the record structure of the original file may be lost. (Refer to the SEND command.)

Messages:

I CANNOT OPEN <file>.
I HAVE CLOSED THE RECEIVE FILE <file>.
I HAVE CREATED A FILE NAMED <file>.
THERE IS NO CURRENT RECEIVE FILE.

RESUME Command

This command, which causes the X3PAD to enter the data transfer mode, is of the following form:

```
RESUME
```

If no call is in progress, a message is displayed and the X3PAD remains in the command mode. If a call is in progress, the X3PAD enters the data transfer mode and subsequent input from the terminal is assembled into packets and written to the network.

Messages:

THERE IS NO CALL IN PROGRESS.

SEND Command

This command, which sends a file over the network, is of the following form:

```
SEND [ <file> ]

where

  <file> := [ [ [ \<system>. ] $<volume> ] <subvol> ] <filename>

    may be any type of GUARDIAN file.
```

If <file> is specified, the X3PAD opens that file, reads data from it, and writes the data to the network. When the end of <file> is reached, the file is closed and the X3PAD resumes executing commands from the input file. If <file> is omitted, the SEND command has the same effect as the RESUME command.

While the X3PAD is executing a SEND command, it writes to the output file in such a way that the output looks much like it would if the characters in <file> had been read from the original file.

Note: The SEND command might not preserve the record structure of <file>. If <file> contains records longer than the network packet size, each record will be fragmented into multiple packets and, depending on the data and interpretation by the receiver, record boundaries might be lost.

If the send file contains empty records (i.e., length = 0), they are not transmitted. As a result, the receive file will not be identical to the send file.

Messages:

ERROR: I CANNOT OPEN <file>.
ERROR: THERE IS NO CALL IN PROGRESS.

SET Command

This command, which sets the X3PAD operational parameters, is of the following form:

```
SET { <param> <value> } , ...

where

  <param> and <value> are defined below.
```

<param>	<value>
ADDRESS	<address>
BREAK	{ <break action> / "(" <break action> , ...")" / NONE } <break action> is { DISCARD / ESCAPE / INTERRUPT / SEND }
CHARKILL	<character>
CRDELAY	{ <integer> | NONE }
DISCARD	{ ON | OFF }
ECHO	{ ON | OFF }
EDITING	{ ON | OFF }
ESCAPE	{ <character> | NONE }
LFDELAY	{ <integer> | 0 }
LFINSERT	{ <data stream> / "(" <data stream> , ...")" / NONE } <data stream> is { ECHO / SEND / RECEIVE }

→

LINEFOLD	{ <integer> | NONE }
LINEKILL	<character>
NOTIFY	{ ON | OFF }
PROMPT	{ ON <character> | OFF }
SEND	{ CR | ETX | NONE }
SPEED	<number> BPS
<character>	can be any graphic character enclosed in quotation marks or an ASCII mnemonic for a nongraphic character. A quotation mark itself is indicated by four consecutive quotation marks.
<integer>	is a decimal integer within the range of 0-255.
<number>	is a decimal rational number.
<address>	is a sequence of decimal digits interspersed with hyphens. (The hyphens and their placements are insignificant; their purpose is merely to enhance legibility.)

Operational Parameters:

CCITT Recommendation X.3 defines the parameters used by a PAD to control an asynchronous terminal. The PAD is a necessary element for support of asynchronous terminals. The X.3 parameters, also called interactive terminal interface (ITI) parameters, can be changed by the terminal user using the SET command or by the remote DTE via X.29.

When a user connects to a public data network, the PAD parameters are initialized to the standard values for the particular terminal type. The terminal is in the command mode at initialization time and, as a result, may set or read the PAD parameters.

When a call has been established, the terminal enters the data transfer mode. To return to the command mode, the user must enter the appropriate command or control character sequence.

The X3PAD operational parameters are listed and described in Table 7-2. The parameters defined by CCITT Recommendation X.3 are listed and described in Appendix E.

Table 7-2. X3PAD Operational Parameters

Parameter	Function
ADDRESS	Contains address of network user at other end of current call. ADDRESS is used for call establishment. (Refer to CALL command.) When remote DTE initiates a call, ADDRESS is set to that DTE's address. ADDRESS may be set by terminal user only when no call is in progress. Default value depends on X25AM subdevice.
BREAK	If terminal user presses BREAK key while X3PAD is in data transfer mode, any specified combination of the following actions is performed by X3PAD: DISCARD: Set DISCARD operational parameter ON. (This action approximates common convention for handling interrupts. Remote DTE will set DISCARD OFF when it receives "Indication of Break" packet.) ESCAPE: Enter terminal command mode. INTERRUPT: Send interrupt package to remote DTE. SEND: Send "Indication of Break" packet to remote DTE. Default values are INTERRUPT, SEND, and DISCARD.
CHARKILL	If EDITING is ON and X3PAD reads specified ASCII <character> from input file, the last character is deleted from editing buffer. Default value is BS (backspace).
CRDELAY	<integer> NUL characters are written to output file file after each carriage return. Purpose of CRDELAY is to allow sufficient time for terminal printer to move its print head. Default value is zero.
DISCARD	If ON, data packets received from remote DTE are discarded. If OFF, packets are written to terminal. Primary use of this parameter is to discard packets after a BREAK.

→

	Default is OFF.
ECHO	If ON, each character read from input file while in data transfer mode is echoed to output file. If OFF, echoing is suppressed. Characters read from input file while in command mode are always echoed to output file. Default is ON.
EDITING	If ON, bytes read from input file while in data transfer mode may be edited before they are assembled into packets. If OFF, edit feature is disabled and bytes read from input file are immediately assembled into packets. OFF also implies that CHARKILL and LINEKILL characters are to be treated as normal data characters. If SEND character is read from input file, or if EDITING is OFF, all characters in editing buffer are assembled into packets. Default in ON. (EDITING is always ON when X3PAD is in command mode.)
ESCAPE	If specified <character> is entered while X3PAD is in data transfer mode, it will enter command mode. Selecting NONE option indicates no such <character> has been specified and, as a result, no particular key can be pressed to cause X3PAD to enter terminal command mode. Permitted values for <character> are DLE (normally (CTRL-P) and ASCII graphic (printable) characters. Default is DLE.
LFDELAY	<integer> NUL characters are written to output file after each line feed (LF) to allow sufficient time for terminal's printer to move its carriage or tractors. Default value is 0. Valid range is 0-7.
LFINSERT	Using LFINSERT, X3PAD inserts LF character following every CR in any combination of the following character streams: →

	ECHO: Characters read from terminal and echoed to terminal. (ECHO must be ON.) SEND: Characters read from terminal assembled into packets and sent across network. RECEIVE: Characters received from network and written to output file. Default is ECHO.
LINEFOLD	Causes a terminal's output line to be folded if line exceeds <integer> characters during a single WRITE operation. First <integer> characters are written on one operation; next <integer> characters are written on next operation. Valid <integer> range is 1-255 characters/line. Selecting NONE option causes all lines to be written to output file in unfolded format. Default is NONE.
LINEKILL	If EDITING is ON, reads specified <character> from input file; all characters currently in editing buffer are discarded. Default is CANCEL.
NOTIFY	If ON, status indication is written to terminal whenever change occurs in status of current call. For example, if NOTIFY is ON and remote DTE changes value of operational parameter, a status indication is written to terminal. If OFF, any indication of change in status is suppressed. Default is ON.
PROMPT	If ON, a prompt <character> is written to terminal when X3PAD expects to read terminal command. If no <character> is specified, previously chosen prompt character is used. If OFF, prompt is suppressed. Because remote DTE can turn off prompt, it is recommended that PROMPT parameter be left ON. →

	Default is ON with <character> = } .
SEND	Causes X3PAD to assemble characters from terminal until a full packet is created or until a specified <character> is entered. In either case, assembled characters are then written to network. Normally, terminal user presses <character> to indicate end of multicharacter message. Default is CR (carriage return).
SPEED	Indicates transmission speed (bits per second) of medium used by X3PAD to communicate with terminal. Value depends on terminal type. This value can be shown (see SHOW command) but it cannot be set.

Messages:

ERROR: SPEED CANNOT BE SET.
ERROR: ESCAPE CANNOT BE SET TO <character>.
ERROR: ADDRESS CANNOT BE SET WHILE A CALL IS IN PROGRESS.

SHOW Command

This command, which displays the X3PAD parameters, is of the following form:

```
SHOW [ <param> , ... ]
```

This command causes the selected parameters and their value to be written to the output file. The syntax of the <param> output is described under the SET command. If no <param> is specified, the values of all operational parameters are displayed one per line. Refer also to the descriptions of the SET and STATUS commands.

STATUS Command

This command, which displays the status of the current call, is of the following form:

```
STATUS
```

The status of the current call is written to the output file. If no call in in progress, the clearing indication of the most recent call is written to the output file. Possible responses to execution of this command are as follows:

```
THERE IS NO CALL IN PROGRESS.
<clear indication>

CALL IN PROGRESS
<call status>
```

The syntax of <call status> is:

```
ADDRESS <address>
<integer> PACKETS SENT.
<integer> PACKETS RECEIVED.
<integer>:<integer>:<integer> ELAPSED TIME.
```

The syntax of <address> is described under the SHOW command.

SYSTEM Command

This command, which sets the default system name, is of the following form:

```
SYSTEM [ \<system> ]

where

  <system> is a GUARDIAN system name.
```

This command sets the default system name, which is used to resolve ambiguous file names. The initial setting of this default is the name of the system in which the X3PAD runs.

If SYSTEM is specified without a parameter, the initial default setting is reinstated. Refer also the the descriptions of the OBEY, RECEIVE, and SEND commands.

VOLUME

This command, which sets the default volume or subvolume, is of the following form:

```
VOLUME   |   $<volume>                    |
         | [ $<volume>. ] <subvolume>     |
```

where

<volume> is a GUARDIAN volume name.

<subvolume> is a GUARDIAN subvolume name.

Execution of this command sets the default values for resolving ambiguous file names. The initial setting of these defaults in taken from the Command Interpreter startup message. Refer also to the descriptions of the OBEY, RECEIVE, and SEND commands.

INTERFACE TO REMOTE DTE

The X3PAD supports CCITT Recommendation X.29, which defines a protocol for the exchange of data, control information, and interrupts between a PAD and a DTE via an X.25 virtual circuit. The exhanged information consists of data packets and interrupt packages.

CCITT Recommendation X.29 is supported by the X3PAD except under the following conditions:

1. X3PAD does not consider arrival of SET, READ, or SET and READ as a data forwarding conditon. Recommendation X.29 requires PAD to forward any partial buffer it might have assembled before acting on a SET, READ, or SET and READ message.

2. X3PAD ignores national parameters in parameter field of SET, READ, or SET^AND^READ messages.

DATA PACKETS

Two types of data packets can be transferred between the PAD and the remote DTE:

1. user sequences containing information to or from terminal

2. PAD messages containing control or status information interpreted or generated by PAD.

The type of data packet is indicated by the Q-bit contained in the beginning of each packet.

Syntax and semantics of PAD messages are defined by CCITT Recommendation X.29. The syntax is specified as bytes of machine-readable information.

Control messages transferred from the remote DTE to the PAD are:

```
SET           [ <param><value> ]...[ 0 0 [ <netparam><value> ]...]

READ          [ <param>   0    ]...[ 0 0 [ <netparam>   0    ]...]

SET^AND^READ  [                ]...[ 0 0 [ <netparam><value> ]...]

INVITATION^TO^CLEAR
```

Status messages transferred from the PAD to the remote DTE are:

```
STATUS [ <param><value> ]...[ 0 0 [ <netparam><value> ]...]

ERROR { NO^MESSAGE^CODE       } <message code>

      { INVALID^MESSAGE^CODE }
```

```
        { INVALID^PARAMETERS    }

        { NOT^BYTE^MULTIPLE     }

  BREAK^INDICATION [ 8 1 ]
```

Variables in these messages are defined as follows:

<param> refers to a PAD operational parameter whose syntax and semantics are defined by CCITT Recommendation X.3 and described in Appendix E. The implementation of X3PAD is described under the SET command and in Table 7-2.

<netparam> refers to a PAD operational parameter, the syntax and semantics of which are defined by each packet switching network.

<value> is the value assigned to a <param> or <netparam>.

<message code> is set to the code of the message that caused the error. For NO^MESSAGE^CODE, <message code> is undefined.

Encoding of these messages is defined by CCITT Recommendation X.29. Each syntactic entity in the machine-readable syntax is encoded as one byte.

INTERRUPT PACKETS

An interrupt packet contains just one byte. Interrupt packets have priority in that they are delivered more quickly than data packets. An interrupt packet is sent by the PAD to the remote DTE if the terminal sends a BREAK operational parameter and the INTERRUPT option has been executed.

If a PAD receives an interrupt packet from a remote DTE, the PAD acknowledges receipt of the packet but performs no other action.

APPENDIX A

INTERFACE LEVEL CHARACTERISTICS

INTRODUCTION

Described in this appendix are specific parameters and interface details required for connecting a Tandem NonStop or NonStop II system to a public X.25 packet switching network.

LEVEL 1

The boundaries of frames can be marked using either BSC or HDLC techniques. BSC framing can be used only with the Tandem T6202 Byte Synchronous Controller. HDLC framing can be used only with the Tandem T6203 Bit Synchronous Controller.

If BSC framing is used, then either ASCII or EBCDIC control characters may be used. The conrol character set may be selected (1) at SYSGEN time using the ASCII or EBCDIC modifier or (2) at run time using the CUP ALTER MODE command. Control characters in the two sets have the following hexadecimal values:

	SYN	STX	ETX	DLE
ASCII:	X"16"	X"02"	X"83"	X"10"
*EBCDIC:	X"32"	X"02"	X"03"	X"10"

*EBCDIC must be used when connecting to TELENET.

Note that if ASCII control characters are used, the Tandem system will transmit them with odd parity and must receive them with odd parity. Odd parity means that the most significant bit of each 8-bit control character (whose value is unspecified by ASCII) has a value such that there is an odd number of "1" bits in the entire character.

The choice of the control character set is meaningless if HDLC framing is used, since HDLC does not use a control character scheme for marking frame boundaries.

LEVEL 2

The following definitions and restrictions apply to frame level support:

- single link, modulo 8 only
- LAP or LAPB. LAPB is symmetric version of LAP specified in Recommendation X.25 (1979)
- default K (maximum frames outstanding) value is 4, with a maximum value of 7
- default T1 (transmission timer) value is 3 seconds
- default N2 (maximum number of transmissions) value is 10
- default Frame Address field is for DTE with DCE addressing also selectable
- default idle frame to be sent is response RR frame. Command RR frame with poll bit set can be used for LAPB, in which case no I-frame is transmitted until command RR frame is responded to. Command RR forces network for a response, hence it is better for X25AM to ascertain the link status.
- default idle timeout is 15 seconds. X25AM transmits idle RR frames (command or response frame) when no idle I-frames are being sent. If specified idle timeout is zero, no idle frame will be sent.

LEVEL 3

By default, logical channels are numbered from 1 through n (0 through n-1 for TRANSPAC network), where n is defined via the CIRCUITS modifier at SYSGEN time. The maximum number of logical channels is 255 and Group 0 support only. The range of logical channel numbers (LCNs) may be modified with the CUP ALTER command; the number of LCNs may not be modified. When operating as a DTE, CALL requests are assigned to the next available LCN (from the top down). When operating as a DCE, CALL requests are assigned to the next available LCN (from the bottom up).

The search for the next LCN for an outgoing packet is performed circularly, beginning at the next LCN in use following the last LCN over which a packet was sent. The next LCN is the previous LCN plus one, modulo the number of circuits. Data packets are sequenced modulo 8. Modulo 128 is not used.

The default packet size is 128 and the default window size is 2. Both of these values can be changed at SYSGEN time. It is recommended that the window size be set to at least 2.

The CALL REQUEST packet facilities include the following:

- Reverse charging can be requested for outgoing calls according to options specified by operator or application process. Similarly, reverse charging can be accepted for incoming calls if option is specified by operator or application process.
- For TELENET connections, throughput class on incoming calls can be recognized, and national parameters for packet size/window size are interpreted.
- Throughput class on outgoing calls can be set via CUP attribute or application call to SETMODE.
- Packet size or window size can be varied on a per-call basis but only by PTP subdevices operating in Mode 2.
- For DATAPAC connections, parameter 1 can be set to 2 to indicate a priority call (PRICALL=1). This limits packet size to 128 bytes, which places restriction on gateway between TELENET and DATAPAC.
- No national facilities are sent or interpreted.
- Fast Select is supported for PTP subdevices operating in Mode 2 only. (Refer to Fast Select Support.)

The 4-byte Protocol ID within the user data field is set to zero for outgoing calls and is ignored for incoming calls.

Reject packages are not supported and will not be sent.

LEVEL 4 (ITI PROTOCOL ONLY)

The X.3 international parameters plus the network-dependent national parameters modifiable by the ITI protocol are listed in Section 2.

FAST SELECT SUPPORT

Both incoming Fast Select packets with normal response and restricted response as described in CCITT Recommendation X.25 are supported.

An incoming Fast Select packet is passed only to an application communicating with a PTP mode 2 subdevice that has a pending READ operation and is not yet connected to a circuit. That is, the subdevice is waiting for a call.

An incoming Fast Select packet may have a larger packet size than a normal data packet size. The maximum Fast Select packet size is specified at SYSGEN time by the PACKETINSIZE modifier. For Call User data of 128 octets, a packet size of 208 octets is recommended: 16 for address fields, 64 for facilities, and 128 for Call User data.

Note that outgoing Fast Select packets are still limited to the normal size.

APPENDIX B

NETWORK DIFFERENCES

INTRODUCTION

Each of the public packet switching networks supported by X25AM varies to some extent from CCITT Recommendation X.25. This appendix describes those variations and the means used within X25AM to resolve possible operational problems. Discussed are the DATAPAC, DATEX-P, PSS, TELENET, TRANSPAC, TYMNET, and UNINET networks. Differences between the various networks on how they handle the call setup and DTE addresses are described in Appendix C.

DATAPAC NETWORK

Addressing

This network uses three forms of DTE addresses: national, gateway, and international.

National addresses, which refer to DTEs within the DATAPAC network, always consist of eight digits. The first digit may or may not be a one. These addresses do not contain a port number. Instead, the port number is specified by the first character in the call "user data" field. The character must be within the range of letters A-Z which corresponds to the numerals 1-26.

Gateway addresses, which refer to DTEs outside the DATAPAC network, always consist of eight digits. The first digit may or may not be a one. The last four digits must be zeros. When using the gateway address format, the call "user data" field must contain the address of the called DTE. This address is encoded as ASCII decimal digits.

International addresses consist of 5 to 15 digits. The first digit must be a one followed by a 4-digit Data Network Identification Code (DNIC), which in turn is followed by up to 10 digits. International addresses may refer to DTEs within the DATAPAC network, whose DNIC is 3020, or to DTEs in other networks.

Priority Call Option

If the PRICALL option is enabled for the calling subdevice, call option 1 is set to a value of 2 in the CALL REQUEST packet. If the option is enabled, outgoing data packets are limited to a length of 128 bytes. The PRICALL option must be enabled when placing international calls.

ITI/PAD Parameters

National parameter 126 is set to a value of 4 for line feed (LF) insertion following a carriage return (CR) to the terminal.

DATEX-P NETWORK

Addressing

This network uses four addressing formats. In all cases, a DTE address consists of a number of digits that identify a DTE followed by a number of digits that identify a port within the DTE. The four addressing formats are as follows:

Type	DTE Digits	Port # Digits
1	8	3
2	9	2
3	10	1
4	11	0

PSS NETWORK

This network does not recognize an Invitation-to-Clear packet from the host. Consequently, a CLEAR REQUEST must always be sent.

TELENET NETWORK

ITI/PAD Parameters

National parameter 1 is set to a value of four for line feed (LF) insertion following a carriage return (CR) to the terminal.

TELENET TIPs (Including Gateway from DATAPAC)

Using ITI protocol, a received SET PARAMETERS packet setting national parameter 14 to a value of one (Send Break) is interpreted as the equivalent of receiving an Indication-of-Break packet.

Circuit Clearing

Some TELENET TIPs do not recognize an Invitation-to-Clear packet from the host. Consequently, a CLEAR REQUEST must always be sent.

TRANSPAC NETWORK

Circuit Numbering

The LCNs range from 0 through CIRCUITS-1 rather than from 1 through CIRCUITS.

Addressing

The DTE addresses in INCOMING CALL and CALL REQUEST packets can be of variable length. Outgoing CALL REQUEST packets must contain only the subdevice PORT # in the calling DTE's address field.

Line Feed Insertion

The ITI protocol automatic line feed feature is controlled by SETMODE function 7. The default setting for this feature is LFTERM. The packet network's PAD supplies the line feed (LF) when a carriage return (CR) is received from the terminal. The TRANSPAC network does not support this feature and, as a result, X25AM defaults to LFSYS when connected to the network. LFSYS means that the host system supplies the line feed when a carriage return is received.

Line Reset

Any line reset that occurs is considered as a valid reason for indicating a packet level restart (which clears any calls). This action eliminates the possibility of lost or duplicate packets, which could result in a line failure/recovery situation. X25AM must perform a line reset following an owner change for the byte synchronous/bit synchronous controller. This situation is a disadvantage to Tandem TRANSPAC network users, but it is not a problem other than the time required for retransmissions.

TYMNET NETWORK

This network does not recognize an Invitation-to-Clear packet from the host. Consequently, a CLEAR REQUEST must always be sent.

UNINET NETWORK

Addressing

The DTE addressing format is identical to TELENET.

ITI/PAD Parameters

These parameters are identical to the X.3 international parameters.

PURE X.25 OPERATION

When no network identifier is specifed at SYSGEN time or by CUP, X25AM operates with the following characteristics:

- LCNs range from 1 through CIRCUITS
- handling of DTE addresses for INCOMING CALL and CALL REQUEST packets:
  - accepts variable length addresses
  - inserts entire remote DTE address into calling DTE's address field of outgoing CALL REQUEST packet

The LFSYS default is used for ITI subdevices because the packet network's PAD cannot be assumed to supply a line feed (LF) upon receipt of a carriage return (CR) from a terminal.

APPENDIX C

CALL SETUP AND DTE ADDRESS HANDLING

INTRODUCTION

This appendix describes the call setup parameters, the processing sequence for accepting an incoming call, the interpretation of the called DTE address in incoming and outgoing CALL REQUEST packets, and the construction of the called and calling DTE address in outgoing CALL REQUEST packets.

CALL SETUP PARAMETERS

Protocols within X25AM may initiate or receive calls, and there are several parameters available through CUP and the SETMODE file system procedure that determine who (the initiator or the recipient) will pay for the calls.

For faster service, the DATAPAC network offers a parameter for priority calls (PRICALL). This parameter, which restricts the packet size to 128 bytes, is set set through CUP or SETMODE function 32. DATAPAC requires a priority call for all international calls.

The use of port numbers allows incoming calls to specify a particular subdevice. An incoming CALL REQUEST packet may or may not contain a port number. Each subdevice has a port number assigned to it by CUP or by SETMODE function 32. Thus, an incoming call can be connected to a subdevice whose port number is specified in the CALL REQUEST packet.

Multiple subdevices may have the same port number. Incoming calls specifying this port number can be routed to any of these subdevices. This creates a "rotary" effect in which a group of subdevices are interchangeable at call setup time. Incoming calls are not shared among the members of such a rotary on a round-robin basis. Instead, each incoming call is routed to the first member of the rotary that will accept it. The first member is the first in the order of subdevices listed by the CUP SHOW command.

If a port number is not contained in an incoming CALL REQUEST packet, it is treated as though it specified port zero. When a subdevice is created, it is assigned a default port number of zero.

ACCEPTING INCOMING CALLS

When a CALL REQUEST packet is received, X25AM performs the following sequence of operations:

1. CALL REQUEST packet and its call parameters are checked for validity. If packet is invalid, call is cleared.

2. Subdevices are searched in order in which they are displayed by CUP SHOW command. Search ends when all subdevices have been examined or when a subdevice is found that satisfies the following conditions:

   a. Port number contained in CALL REQUEST packet matches port number of subdevice.

   b. Subdevice is waiting for call. Application must have opened subdevice and issued a CONTROL 11 (Wait for Modem Connect) request.

   c. If incoming call requests reverse charging, subdevice must be configured to accept reverse charging.

   Call is cleared if no such subdevice is found.

3. If preceding conditions are satisfied, INCOMING CALL packet is given to level 4 protocol handling selected subdevice. Level 4 protocol then completes CONTROL request.

   If a NET subdevice is receiving a call, the CALLING DTE ADDRESS contained in CALL REQUEST packet is compared to address configured for subdevice. If addresses do not match, call is not accepted. This check ensures that only the proper DTEs are connected in a given EXPAND network.

4. If level 4 protocol accepts call, the circuit and subdevice are connected until cleared. Clearing may result from CONTROL 12 (Disconnect Modem) or from CUP CLEAR or DOWN command.

CALLED DTE ADDRESS (INCOMING CALL PACKETS)

A CALL REQUEST packet may optionally contain a port number, the format and location of which depends on the particular network. X25AM extracts the port number using the following algorithm:

Define NAL	number of digits in NETADDR (length of address on network/device)
Define CALLED^ADDRESS Address	array containing digits of Called DTE Address field of CALL REQUEST packet. CALLED^ADDRESS [1] is first digit of this field.
Define CAL	number of digits in CALLED^ADDRESS.
Define USER^DATA	array containing bytes of Call "User Data" field of CALL REQUEST packet. USER^DATA [1] is first byte of this field.

The algorithm is as follows:

```
CALLED^PORT := 0;                    !This is the default. !
```

For DATANET-1 network:

```
IF $NUMERIC (USERDATA [1])
THEN CALLED^PORT := USERDATA [1] - "0";
```

For DATAPAC network:

```
IF CAL = 8
AND CALLED^ADDRESS [1] <> 1
THEN    !  DATAPAC addressing convention  !
        IF $ALPHA (USERDATA [1])
        THEN CALLED^PORT := USERDATA [1] - "A" + 1;
ELSE    !  international addressing convention  !
        IF CAL > NAL AND (CAL-NAL) <= 3
        THEN CALLED^PORT := CALLED^ADDRESS [NAL+1:CAL];
```

For DATEX-P network:

```
N := CALLED^ADDRESS [7] + 7;
IF N > 7 AND N < 12
THEN CALLED^PORT := CALLED^ADDRESS [NAL-CAL+1:CAL]
```

For TELENET, TYMNET, PSS, and pure X.25 networks:

```
IF CAL > NAL AND (CAL-NAL) <= 3
THEN CALLED^PORT := CALLED^ADDRESS [NAL+1:CAL]
```

For TRANSPAC network:

```
IF CAL > 0 AND CAL <= 3
THEN CALLED^PORT := CALLED^ADDRESS [1:CAL]
```

For UNINET network:

```
IF $ALPHA (USERDATA [1])
THEN CALLED^PORT := USERDATA [1] - "A" + 1;
```

CALLED DTE ADDRESS (OUTGOING CALL REQUEST PACKETS)

Interpretation

After an incoming call is successfully routed to a subdevice, the Calling DTE Address field of the CALL REQUEST packet is copied into the remote DTE address field of that subdevice. X25AM does not interpret this field. The opener of the subdevice can retrieve this address using the SETPARAM procedure.

If the call is cleared and the opener of the subdevice subsequently initiates a call, the same remote DTE address is used to construct the Called DTE Address in the outgoing CALL REQUEST packet. The net effect of such a sequence of operations is to call back to the originator of the previous call.

Construction for DATANET-1 Network

If the called address consists of at least seven digits, then the first seven digits of the called address are copied into the Called DTE Address field of the CALL REQUEST packet. The remaining digits are converted into a single ASCII digit (0, 1, ... 9).

If the called address consists of less than seven digits, then only the called address is copied into the Called DTE Address field of the CALL REQUEST packet.

Construction for DATAPAC Network

If the first digit of the called address is a "1", or if the called address is less than eight digits long, the international addressing convention is used. The called address is copies into the Called DTE Address field of the CALL REQUEST packet without modification.

If the called address consists of from 8 to 11 digits, and the first digit is is not a "1", the DATAPAC national addressing convention is used. The first eight digits of the called address are copied into the Called DTE Address field of the CALL REQUEST packet. The remaining digits are converted into a single ASCII character (e.g., "1" is converted to "A", "2" is converted to "B", "3" is converted to "C",

etc.) This character is inserted as the first byte of the User Data field of the CALL REQUEST packet.

If the called address consists of at least 12 digits and the first digit is not a "1", the DATAPAC gateway addressing convention is used. The first four digits of the called address are copied into the first four digits of the Called DTE Address field of the CALL REQUEST packet. The next four digits are set to zeros. The remaining digits are converted to ASCII decimal format and inserted in the beginning of the User Data field of the CALL REQUEST packet.

Construction for UNINET Network

If the called address consists of at least eight digits, then the first eight digits of the called address are copied into the Called DTE Address field of the CALL REQUEST packet. The remaining digits are converted into a single ASCII character, which is inserted as the first byte of the User Data field of the CALL REQUEST packet.

If the called address consists of less than eight digits, then only the called address is copied into the Called DTE Address field of the CALL REQUEST packet.

Construction for All Other Networks

The called address is copied into the Called DTE Address field of the CALL REQUEST packet.

CALLING DTE ADDRESS (OUTGOING CALL REQUEST PACKETS)

Construction for DATANET-1 Network

The Calling DTE Address field contains the NETADDR as configured by the Communications Utility Program (CUP).

Construction for DATAPAC Network

If the first digit of NETADDR is not a "1" (a national address), NETADDR is copied into the Calling Address field.

If the first digit of NETADDR is a "1" (an international address), NETADDR is concatenated with the subdevice's port number and copied into the Calling DTE Address field. The port number is encoded as two BCD digits.

Construction for DATEX-P Network

The Calling DTE Address field is subdivided into several subfields. All but the last subfield is copied from NETADDR. The last subfield (subaddress) contains the port number of the calling subdevice. If the subaddress field is not large enough to contain the port number, it is set to zeros.

Construction for TRANSPAC Network

The Calling DTE Address field contains only the port number of the calling subdevice. The port number is encoded as two BCD digits.

Construction for UNINET Network

The Calling DTE Address field contains the NETADDR as configured by the Communications Utility Program (CUP).

Construction for All Other Networks

The Calling DTE Address field contains the NETADDR, configured by CUP, concatenated with the port number of the calling subdevice. The port number is encoded as two BCD digits.

APPENDIX D

X25AM DATA SPACE REQUIREMENTS

DEDICATED BUFFER SPACE CALCULATION

X25AM requires dedicated buffer space for level 2 data transfers. The size of the dedicated buffer can be determined as follows:

```
L = ((K + R) * ((PACKETSIZE/2) + 5)) + 5
```

where

L = buffer length in words

K = level 2 window size

R = number of read frames desired. Set to 2 for 9600 baud or slower. For faster lines or when using the bit synchronous controller, set to 3 or 4 to keep no-frame buffer count under control. Refer to CUP STATS command in Volume 1 for details.

Example: PACKETSIZE=128, 2 read buffers, L2WINDOW=4

```
L = (4 + 2) * 69 + 5 => 419 words
```

BUFFER REQUIREMENTS

X25AM requires packet buffers for inbound data buffering. The number of packet buffers required is a function of the number of virtual circuits actually in use, and the line parameters PACKETSIZE and L3WINDOW. Buffer requirements for NonStop and NonStop II systems are defined in following paragraphs.

NonStop System Buffers

Packet buffers for the virtual circuits are allocated when the virtual circuit is established, and released when the circuit is cleared. Each packet buffer is approximately L3WINDOW * PACKETSIZE in size, and is

allocated from IOPOOL. Packet buffer space is maintained by the primary line handler.

Message buffer space is required for each read or write request. Incoming data is first read into a packet buffer and then assembled in the read message buffer before being returned to the user. Outbound data is moved directly from the write message buffer into the frame buffer.

When the access method is initialized, LONGPOOL space is obtained for the circuit control blocks. The size of this LONGPOOL space is:

number of words = # of circuits + 16 * (# of circuits + 1)

In addition, LONGPOOL space is also allocated for the level 3/4 timer list at initialization. The size of this LONGPOOL space is:

number of words = # of circuits + 1

NonStop II System Buffers

Packet buffers for the virtual circuits are allocated when the virtual circuit is established, and released when the circuit is cleared. Each packet buffer is approximately L3WINDOW * PACKETSIZE in size and is allocated from the LOCALPOOL.

Message buffer space is required for each read or write request. Incoming data is first read into a packet buffer and then assembled in the read message buffer before being returned to the user. Outbound data is moved directly from the write message buffer into the frame buffer.

SYSTEM DATA SPACE REQUIREMENTS

Data space requirements for NonStop and NonStop II systems are defined in following paragraphs.

NonStop System Data Space

The types of NonStop system data space required by X25AM are shown in Table D-1.

Table D-1. NonStop System Data Space Requirements

SYSTEM RESIDENT	LONGPOOL	IOPOOL	CBSPACE
PDT PDT EXTENSION DEDICATED FRAME BUFFERS X25AM STACK	CKT^TABLE CKT CB's L3/L4 TIMEOUTS	PACKET BUFFERS IO BUFFERS	SUBDEV CB's OCBTABLE

NonStop II System Data Space

The types of NonStop II system data space required by X25AM are shown in Table D-2.

Table D-2. NonStop II System Data Space Requirements

SYSTEM RESIDENT	LOCALPOOL
DEDICATED FRAME BUFFERS	X25AM STACK CKT^TABLE SUBDEV^TABLE CKT CB's SUBDEV CB's OCBTABLE L3/L4 TIMERLIST PACKET BUFFERS

MEMORY REQUIREMENTS

Memory requirements (in words) for NonStop and NonStop II systems are given in following paragraphs.

NonStop System Memory

System Data:

PDT	= 13	
PDT EXTENSION	= 159	
DEDICATED FRAME BUFFERS	= see "Dedicated Buffer Space Calculation"	
CKT^TABLE	= NCKTS	From LONGPOOL
CKT CB's	= 16 * (NCKTS + 1)	From LONGPOOL
SUBDEV CB's	= 62 * NSDEV	From CB space
OCBTABLE	= 7 * NOPENS + 3	From CB space
I/O buffers for each active circuit or subdevice waiting for call	= (PACKETSIZE/2 + 3)	From IOPOOL

where

NCKTS	= number of circuits defined at SYSGEN time
NSDEV	= number of subdevices configured via CUP
NOPENS	= number of opens
PACKETSIZE	= data packet size defined at SYSGEN time

NonStop II System Memory

System Data (from system space):

DEDICATED FRAME BUFFERS	= see "Dedicated Buffer Space Calculation"

User Data (from X25AM LOCALPOOL space):

PDT/PDT EXTENSION	= 146
CCB	= 262
STACK	= 600
CKT^TABLE	= NCKTS + 1
SUBDEV^TABLE	= NCKTS + 1
CKT CB's	= 16 * (NCKTS + 1)
SUBDEV CB's	= 62 * NSDEV

```
OCBTABLE            = 7 * # of opens + 3

L3/L4 TIMERLIST     = NCKTS + 1

PACKET BUFFERS      = ((PACKETSIZE + 6)/2 * L3WINDOW + 6) * N
```

where

```
NCKTS      = number of circuits defined at SYSGEN time
NSDEV      = number of subdevices configured via CUP
PACKETSIZE = data packet size defined at SYSGEN time
L3WINDOW   = L3 window defined at SYSGEN time
N          = number of circuits connected or subdevices waiting
             for call
```

Note 1: The dynamic control blocks and buffers described above are obtained from the LOCALPOOL space defined by the SYSGEN parameters LOCALPOOLPAGES and MAXLOCALAREA.

Note 2: The "Dedicated Buffers" are obtained from system data space defined by the SYSGEN parameter LINEBUFFERSIZE.

SAMPLE CALCULATIONS

Given: PACKETSIZE = 128, L3WINDOW = 2, CIRCUITS = 50
Assume: 128-byte READs on each circuit, one per subdevice

LONGPOOL

CKT^TABLE	16	= 16
CKT CB's	16 * (50 + 1)	= 816

CB Space

SUBDEVCB's	62 * 50	= 3100
Subdevice OCBTBL's	(7 * 1 + 3) * 50	= 500

IOPOOL

Read Buffers	64 * 50	= 3200
Packet Buffers	((128/2 + 3) * 2) * 50	= 6700

APPENDIX E

X.3 OPERATIONAL PARAMETERS

Operational parameters of CCITT Recommendation X.3 are given in Table E-1. The implementation of these parameters are presented in Table 7-2. Note that Table E-1 also provides the terminal command name and the default of each X.3 parameter as it is implemented in the X3PAD.

Table E-1. CCITT X.3 Operational Parameters

Parameter	Function
1	Escape from data transfer mode to command mode. 0 = do not permit escape. 1 = permit escape on DLE or graphic character. Values of 32 to 126 specify character used to initiate escape. X3PAD = ESCAPE. Default is escape on DLE.
2	Echo characters from terminal. 0 = PAD will not echo characters. 1 = PAD will echo characters. X3PAD = ECHO. Default is ON (echo enabled).
3	Data Forwarding signal, which causes buffer to be sent to remote DTE when specified character is detected. Following is list of characters that can be used to indicate data forwarding:

→

	0 = no data forwarding character. 1 = alphanumeric A-Z, a-z, or 0-9. 2 = CR (carriage return). 4 = ESC, BEL, ENQ, or ACK. 8 = DEL, CAN, or DC2. 16 = ETX or EOT. 32 = HT, LF, VT, or FF. 64 = any character not listed above. Valid combinations are: 0, 2, 6 (2+4), 18 (2+16), and 126 (2+4+8+16+32+64). X3PAD = SEND. Default is CR.
4	Idle timer value used with data forwarding on timeout. 0 = no timeout value. n = 1-255 in increments of 50 milliseconds. X3PAD = not used.
5	Ancillary device control (flow control of terminal). 0 = XON and XOFF not used. 1 = XON and XOFF used. X3PAD = not used.
6	PAD service signals. These are PAD-generated signals to terminal: 0 = suppress all PAD-generated signals. 1 = generate messages other than prompt (NOTIFY). 4 = generate prompt (PROMPT). X3PAD = NOTIFY. Default is ON (all signals are accepted).
7	Procedure to follow after receiving BREAK signal from terminal: 0 = do nothing. 1 = send Interrupt packet to host. 2 = send Reset packet to host. 4 = send Indication of Break PAD message 8 = escape to X.28 command mode. 16 = discard output. →

	X3PAD = BREAK. Defaults are: INTERRUPT (1), SEND (4), and DISCARD (16).
8	Discard output (used to indicate whether data sent to terminal is being flushed by PAD): 0 = data being delivered normally. 1 = data being flushed (discarded). X3PAD = DISCARD. Default is OFF (deliver data).
9	Carriage return padding (number of pad characters sent to terminal after a carriage return to allow time for carriage to physically return to beginning of line): 0 = no padding characters. n = 1-7 pad characters. X3PAD = CRDELAY. Default is 0 (no padding).
10	Line folding (number of characters PAD should allow for each print line to terminal): 0 = no line folding. n = 1-255 characters allowed per line. X3PAD = LINEFOLD. Default is NONE (no line folding).
11	Terminal speed. For informational purposes only and cannot be modified by terminal user or PAD. Values from 1 through 18 indicate speeds from 110 bps to 64 kbps. X3PAD = SPEED. Default is terminal dependent.
12	Flow control of PAD by terminal via XON and XOFF: 0 = XON and XOFF disabled. 1 = XON and XOFF enabled. X3PAD = not used.
13	Line feed insertion (indicates action of PAD when a carriage return is received from terminal or remote DTE): →

	0 = no line feed insertion. 1 = insert line feed after each carriage return to terminal. 2 = insert line feed after each carriage return from terminal. 4 = insert line feed after each carriage return sent as echo to terminal. Valid combinations are: 0, 1, 4, 5 (1+4), 6 (2+4), and 7 (1+2+4). X3PAD = LFDELAY. Default is ECHO (echo to terminal).
14	Line feed padding (indicates number of pads transmitted after line feed is transmitted to terminal): 0 = no line feed padding. n = 1-7 pads. X3PAD = LFDELAY. Delay is 0 (no padding).
15	Editing (allows changing of characters while in data transfer mode): 0 = no editing 1 = editing permitted; idle timer is disaable. User should use caution when selecting data forwarding characters. X3PAD = EDITING. Default is ON (editing permitted).
16	Delete character (ASCII character used to signal character delete): n = 0-127 (any ASCII character). X3PAD = CHARKILL. Default is BS (backspace).
17	Line delete (used to cause all characters in editing buffer to be deleted): n = 0-127 (any ASCII character). X3PAD = LINEKILL. Default character is CANCEL.

→

18	Line display character (used to request a display of editing buffer): n = 0-127. X3PAD = not used.
The National Parameter Separator is defined by CCITT Recommendation X.29 as a signal to use the national ITI parameters. The parameter number is 0 and the value is 0. This parameter is used in a manner similar to that of the National Facilities Marker, and it indicates PDN-specific PAD parameters.	

APPENDIX F

DATA FILTERING

The X3PAD filters data streams written to output and log files. Each file is filtered independently. Descriptions of these files are given in Section 7 under the OUT and LOG terminal commands.

If the output or log file is an interactive terminal, the X3PAD writes all data to the file in an unfiltered (unchanged) form. If the file is not a terminal, however, the X3PAD applies a filtering algorithm to the data. This algorithm assumes that the data consists of ASCII characters (one per byte). The data is interpreted so that each line of output corresponds to one line of unfiltered output to a printing terminal.

The ASCII formatting characters BS (backspace), HT (horizontal tab), LF (line feed), VT (vertical tab), FF (form feed), and CR (carriage return) are translated as shown in Table F-1. Control characters other than those listed are discarded.

Table F-1. ASCII Formatting Characters

Character	Function
BS	Move cursor left one position.
HT	Move cursor right one position.
VT	Move cursor down one position.
FF	Move cursor down one position
LF	Move cursor down one position
CR	Move cursor left to beginning of current line.

If the file accepts SETMODE and CONTROL operations for forms control, the operations are performed by the X3PAD as specified. If the file does not accept SETMODE and CONTROL operations for forms control, the X3PAD further filters the character stream so that each WRITE operation to the file corresponds to what would appear on one line of printed output. If multiple graphic characters (other than a space character) are written to the same position of a line, only the first character appears in the filtered output.

Note that the filtering algorithm might not produce perfect images on some types of terminals.

The RECEIVE terminal command described in Section 7 should be used to make an unfiltered copy of data read from the network.

INDEX

YOUR COMMENTS PLEASE

Tandem NonStop™ & NonStop II™ Systems
AXCESS™—Volume 2: X.25 Access Methods (X25AM)
82334 C00

Tandem welcomes your comments on the quality and usefulness of its publications. Does this publication serve your needs? If not, how could we improve it? If you have specific comments, please give the page numbers with your suggestions.

This comment sheet is not intended as an order form. Please order Tandem publications from your local Sales office.

FROM:

Name _____ Date _____

Company _____

Address _____

City/State _____ Zip _____

FC

NO POSTAGE
NECESSARY
IF MAILED
IN THE
UNITED STATES

BUSINESS REPLY MAIL
FIRST CLASS PERMIT NO. 482 CUPERTINO, CA, U.S.A.

POSTAGE WILL BE PAID BY ADDRESSEE

Tandem Computers Incorporated
19333 Vallco Parkway
Cupertino, CA 95014-9990

Attn: Manager—Technical Communications

FC

TAPE

TAPE