UNISYS

BTOS X.25 Gateway

Operations and Programming Guide

Relative to Release Level 6.0

March 1987

Priced Item

1208055

UNISYS

BTOS X.25 Gateway

Operations and Programming Guide

Relative to Release Level 6.0 — March 1987

Priced Item — 1208055

Correspondence regarding this publication should be forwarded, using the Product Improvement Card at the back of this manual, or remarks may be addressed directly to Unisys Corporation, Corporate Product Information East, Building C, Township Line and Union Meeting Road, Blue Ball, PA 19424 U S America.

About This Manual

Purpose

This is a reference guide for programmers using the X.25 Network Gateway product. It presents a history of public data networks, describes how the X.25 gateway fits within this environment, and provides detailed instructions on how to use the BTOS X.25 Network Gateway software product.

Audience

This manual is intended for use by advanced communications systems analysts and application programmers.

Organization

Section 1, "Overview," provides a general introduction to the X.25 Network Gateway product, including a brief description of access levels and information on the current release level.

Section 2, "Communications Network Concepts," discusses switching schemes, the standard interface to packet-switched public data networks, logical channels, and virtual circuits.

Section 3, "Installation," presents step-by-step instructions on installing the X.25 Network Gateway on BTOS systems, including systems using the XE 520 master.

Section 4, "Packet Access Method," details the use of the packet access method, its network interaction, and each of its operations.

Section 5, "Sequential Access Method," details the use of the sequential access method, its network interaction, and each of its operations.

Section 6, "Multimode Terminal Program X.25 Communications," details the use of the Multimode Terminal Program, its network interaction, and each of its operations.

Section 7, "X.25 Status Program," provides reproductions of video screens containing X.25 status information and an explanation of each item on the screen.

Four appendixes, a glossary, and an index are included.

Related Product Information

BTOS Reference Manual, Volumes 1 and 2

BTOS Multimode Terminal Program (MTP) Programming Reference Manual

BTOS Multimode Terminal Program (MTP) Operations Guide

Contents

Illustrations

Tables

Overview

The X.25 Network Gateway is a software product that enables a BTOS system to operate on packet-switching public data networks (PDNs) that support CCITT Recommendation X.25 (1980 standard). CCITT stands for Comite Consultatif Internationale de Telegraphique et Telephonique (International Telegraph and Telephone Consultive Committee). CCITT Recommendation X.25 defines the standard interface between packet-switched PDNs and user devices operating in the packet-switched mode. See Section 2 for more information on CCITT recommendations and Appendix D for reference information on CCITT.

This section presents the features of X.25 Network Gateway, including access methods, memory use, and hardware and software dependency, and it details the capabilities of the current release level.

The X.25 Network Gateway consists of the following six programs: (See Figure 1-1, which shows how a local user interacts with a public data network via the X.25 system.)

1 *X.25 Network Gateway System Service* This system service implements the X.25 protocol for the BTOS system, and must be installed on the BTOS workstation before any other X.25 network components can be used.

   This system service supports the X.25 packet access method, which consists of a set of procedural interfaces. These interfaces enable BTOS user programs to send and receive individual X.25 data and control packets and to directly monitor the establishment of X.25 connections. The packet access method is used by other X.25 Network Gateway components and is available to the communications programmer for use in custom designed X.25-based software.

   The X.25 Network Gateway System Service is accessible through packet access method requests issued from an application program.

2 *X.25 Status Monitor* Use this program to display the status of specified X.25 connections.

3 *X.25 Byte Streams* This provides device-independent access to X.25 through the sequential access method. The X.25 byte stream provides the BTOS application programmer with tools for transmitting and receiving data over PDNs without requiring extensive knowledge of the X.25 protocol.

4 *X.25 Byte Stream Configuration File Editor* Use this file editor at your workstation to create configuration files for X.25 byte streams. These configuration files define X.25 communications options for a particular X.25 byte stream.

5 *Multimode Terminal Program (MTP)* Use this program to communicate with host computers or other workstations over a PDN.

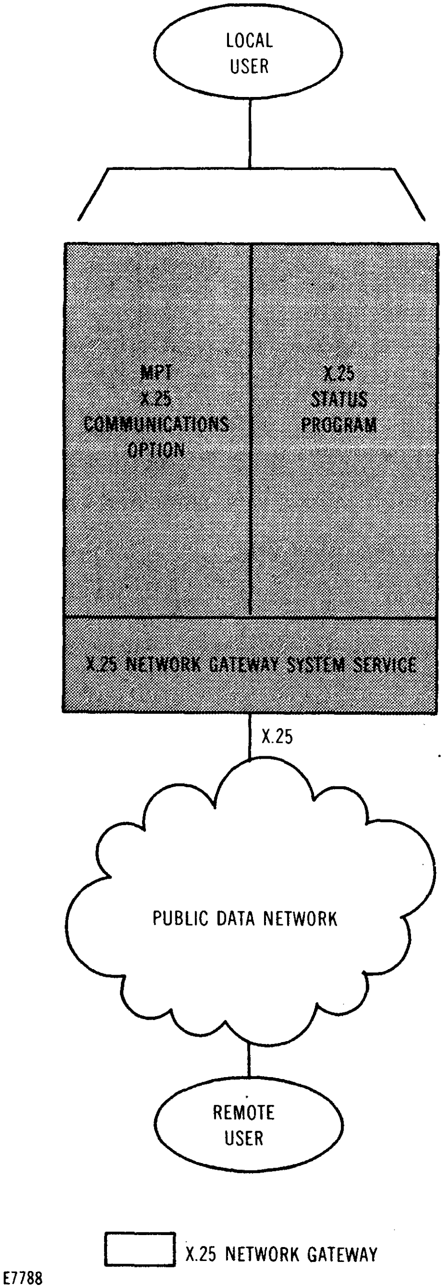
6 *X.25 Deinstall Utility* Use this utility to remove the gateway from memory without rebooting. Deinstall is not designed for use on XE 520 masters or on the Intelligent Data Communications (IDS) module.

Access Levels

The X.25 Network Gateway provides three levels of access to a PDN:

- Packet access method (supported by the X.25 Network Gateway system service)
- Sequential access method
- Multimode Terminal Program

Figure 1-1 **X.25 Network Gateway Overview**



Packet Access Method (PAM)

The packet access method (PAM) enables your program to send and receive individual X.25 control and data packets and to directly monitor the establishment of X.25 connections. This level allows the communications system programmer to design sophisticated X.25-based applications such as interfaces to other computer networks, server programs, and packet assembly/disassembly (PAD) facilities. To design such applications, the system programmer must have extensive knowledge of X.25 (see the discussion of the CCITT Recommendation X.25 in Section 2). It also helps the system programmer to know the upper layers of the International Standards Organization (ISO) Open System Interconnection (OSI).

Following are packet access method operations and their functions:

- *Call establishment and clearing operations* begin and end calls over virtual circuits.
- *Data transfer operations* transfer data over an established virtual circuit.
- *Status operation* monitors the status of the X.25 Network Gateway system service.

Packet access method operations are served by the X.25 Network Gateway system service, which is the BTOS implementation of CCITT Recommendation X.25 packet and link level protocols.

See Section 4 for more information on the packet access method.

Sequential Access Method (SAM)

The Sequential Access Method (SAM) enables you to send data to other computer systems over a PDN without the sophisticated techniques required by the packet access method. This level of access is appropriate for the applications programmer, because detailed knowledge of the CCITT Recommendation X.25 is not required.

See Section 5 for more details on the sequential access method.

Multimode Terminal Program (MTP)

The Multimode Terminal Program (MTP) is designed for the novice communications user. MTP allows a workstation to appear as an intelligent terminal to a host computer on a PDN through CCITT Recommendation X.25.

For a basic introduction to MTP, refer to the *BTOS Multimode Terminal Program (MTP) Operations Guide*, and for additional information, refer to the *BTOS Multimode Terminal Program (MTP) Programming Reference Manual*.

For more information on the Multimode Terminal Program, see Section 6.

Memory Use

The X.25 Network Gateway system service requires approximately 60 K-bytes of memory plus 750 bytes for each virtual circuit.

The amount of memory required for each circuit is based on installation parameters and is calculated as follows (the * indicates multiplication):

Number of Bytes = (Number of Circuits * 176) + [(2 * Number of Circuits + 8) * Default window size * (Max packet size + 12)]

Using the default installation parameters for default window size (2) and max packet size (128), approximately 750 bytes of memory are required per circuit.

The maximum size of the memory used for buffers and control structures is about 42 K-bytes.

114 K-bytes are required for 64 VCs using a packet size of 128 bytes.

The X.25 Status Monitor program requires about 19 K-bytes at the workstation where it is run.

The X.25 Byte Streams program requires about 4 K-bytes and a 1 K-byte buffer per open byte stream (which your program supplies) at the workstation where a program using X.25 Byte Streams is run.

The X.25 Byte Stream Configuration File Editor program requires about 6 K-bytes at the workstation where it is run.

MTP requires 73 K-bytes and available memory (up to 64 K-bytes) for its display memory at the workstation where it is run.

The X.25 Deinstall Utility requires 20 K-bytes at the workstation where it is run.

Required Files

Individual X.25 Network Gateway programs require only some of the files that make up this product. If you are using only a few X.25 Network Gateway programs, the unused programs do not need to remain on your hard disk. (The X.25 Network Gateway System Service must be installed before any other X.25 program can be used.) Required files for each program are as follows:

Utilities:

- X.25 Network Gateway System Service
  X25.Run
- X.25 Status Monitor
  X25Mon.Run
- X.25 Byte Streams Configuration File Editor
  X25Config.Run
- X.25 Deinstall Utility
  X25Din.Run

Libraries:

- X.25 Sequential Access Method
  X25Sam.Lib
- X.25 Packet Access Method
  RqLablX25.Lib

An X.25 byte stream accesses a default configuration file, [Sys]<sys>X25Config.sys, if no configuration file is specified when the byte stream is opened. You should create your own configuration files with the X.25 Byte Stream Configuration File Editor program. Type the command **Configure X.25** to create configuration files.

Software and Hardware Requirements

BTOS X.25 Network Gateway release level 6.0 runs on the XE 520 with MS8 operating system and B 26, B 27, B 28, and B 38 master, standalone, and cluster systems using BTOS operating system release level 7.0 or 8.0.

The X.25 Network Gateway uses the onboard serial channels of the BTOS workstation. The gateway can use any port on the four-port data communications (DCX) module. It also runs on the Intelligent Data Communications Slice (IDS) module.

For a master or cluster workstation without either the DCX or IDS modules, no special processor hardware is required to run the gateway. The gateway supports use on XE 520 cluster processor channels 1-2 and on terminal processor channels 1-4.

Your BTOS workstation should be connected to a network by an RS–232 cable with a synchronous modem, at speeds of 2400 to 9600 baud (19,200 baud supported on B 28 and B 38, up to 64,000 baud on port B of an IDS module with an external RS-232 to V.35 converter).

Refer to your PDN requirement specification for information on the precise equipment necessary and whether it is available from the PDN. For example, a PDN may specify that a connection between a workstation and a Public Data Network (PDN) for X.25 communications consist of a leased telephone line with synchronous modems. The spec would also indicate whether the PDN supplies this equipment or whether you have to obtain it elsewhere.

Support of CCITT Recommendations

The X.25 Network Gateway supports 1980 CCITT Recommendations X.21bis and X.25. Refer to the discussion of network interaction in Sections 4, 5, and 6 for more information on how the protocols defined in these recommendations are supported.

Files on the Product Disk

[B20X56]<SYS>CrashDump.Sys
[B20X56]<SYS>FdSys.version
[B20X56]<SYS>Sys.cmds
[B20X56]<SYS>MTE-Ini
[B20X56]<SYS>FileHeaders.Sys
[B20X56]<SYS>mfd.sys
[B20X56]<SYS>SysImage.Sys
[B20X56]<SYS>Install.Sub
[B20X56]<SYS>XEInstall.Sub
[B20X56]<SYS>X25Sam.Lib
[B20X56]<SYS>RqLablX25.Lib
[B20X56]<SYS>MTP-Hlp
[B20X56]<SYS>Log.Sys
[B20X56]<SYS>BadBlk.Sys
[B20X56]<SYS>X25Config.sys
[B20X56]<SYS>Request.F.Sys
[B20X56]<SYS>Request.F.Version
[B20X56]<SYS>VersionNumber.Run
[B20X56]<SYS>MRequest.F.Message
[B20X56]<SYS>MRequest.F.Sys
[B20X56]<SYS>rqInstall.Sub
[B20X56]<SYS>rqCopy.Sub
[B20X56]<SYS>Request.F.Message
[B20X56]<SYS>MRequest.F.Version
[B20X56]<Unisys>X25.Run
[B20X56]<Unisys>X25Din.Run
[B20X56]<Unisys>MTP.Run
[B20X56]<Unisys>X25Mon.Run
[B20X56]<Unisys>X25Config.Run

Supported PDNs

Release level 6.0 is tested for support on:

Telenet
Transpac
PSS
Datex-P (Austria and Germany)
Tymnet
Uninet
RCA Cylix
Datanet-1

New Features of X.25 Gateway

- Deinstallation capability permits removal of the gateway without rebooting the workstation.
- Maximum number of virtual circuits is expanded to 64, provided a sufficient amount of memory is available.
- Subaddressing is supported, including subaddressing for Transpac users.
- Access of the gateway is possible over a B-NET connection from a user-written application and from the status monitor.
- A new parameter allows user applications to specify whether they wish to wait for a packet acknowledgment from the DCE when calling WriteX25PacketNet.
- Up to 19.2K baud rate on IDS, B 28, and B 38 in certain configurations.
- Up to 64K baud rate on channel B of the IDS module with external RS-232 to V.35 converter.
- The gateway now supports packet sizes of 16, 32, 64, 128, 256, 512, and 1,024 bytes (128 remains as the default).
- The gateway now supports Context Manager.
- The gateway now supports the four-port data communications (DCX) module.
- The gateway now supports the Intelligent Data Communications (IDS) module.

Migration and Coexistence

- The 6.0 release replaces the 5.0 release.
- Several new network requests have been added, and custom applications should migrate to the new requests.
- All previous requests are still supported.
- Release 6.0 of the Status Monitor program provides a networked request to the gateway; therefore, release 6.0 Status Monitor cannot be used with release 5.0 of the gateway.

- The gateway now requires loadable requests.
- B 21 and B 22 workstations are not supported by this release.
- MTP users cannot use B-NET, optional facilities, or Permanent Virtual Circuits (PVC).

Communications Network Concepts

•

This section explains the communications concepts you need to know to use the X.25 network. The following topics are explained:

- Switching
- CCITT Recommendation X.25
- CCITT Recommendations X.3, X.28, and X.29
- Logical Channels and Virtual Circuits

Switching

Switching, which defines the communications path between parties, is a basic element in all communications systems. Two schemes are used in switching:

- *Circuit switching* preallocates the path for the duration of the communication.
- *Message switching* continually reallocates the path while the communication is in progress. This is known as *dynamic allocation.*

Circuit and Message Switching

Telephone, Telex, and TWX systems are preallocating schemes and use circuit switching. This method uses switching points (for example, telephone switchboards), which set up a connection between two parties until one party terminates the connection. Information transferred along this connection is conveyed without delay because all switching decisions were performed when the communication was initially established.

Mail and message systems are examples of dynamic allocation; information from many senders shares the same paths. A decision is made at each switching point along the route where to send the information next. The switching point sends the information closer to its intended destination. Delays in information transfer occur while switching decisions are made and implemented.

Historically, circuit switching was the preferred method because of few delays in data transfer, thus allowing interactive communication. However, circuit switching is inefficient because a connection is reserved between two parties even if information is not actually being transferred. Because most communications have gaps in transmission where information is not being exchanged, a circuit-switched call often causes under-utilization of the communications system.

Message switching allows consistently high utilization of a communications path by sharing (multiplexing) the path among a number of users and filling gaps in one user's communication with information from another's communication. However, message switching has traditionally required time-consuming switching decisions at the switching points. Further, someone transferring very long messages can dominate a message-switched system, thereby causing excessive delays for other users.

Packet Switching

Packet switching is a type of message switching in which messages are divided into small segments, or packets. These packets are transferred individually by a message-switched method and reassembled at their destination. Very fast message-switching points using high-speed digital computers can move packets along a dynamically allocated route at high speeds. Packet switching offers high system use of message switching without long switching delays. It also eliminates the conditions that allow one user to monopolize the system.

CCITT Recommendation X.25

As packet switching became more popular in computer communications during the 1970s, a number of corporations began offering public data communications services over packet-switched public data networks (PDNs). Because there was no common standard for internal operation, PDNs differed in design and operation. The resulting variety of user interface requirements made it apparent that a standard interface allowing communication with all PDNs was required for user equipment.

In 1976, the CCITT Sixth Plenary Assembly adopted Recommendation X.25, the standard device-independent interface between packet-switched PDNs and user devices operating in the packet-switched mode. In February 1980, a revised Recommendation X.25 was issued. BTOS systems use this revised standard for the implementation of X.25 as defined in this manual (see Appendix D).

CCITT Recommendation X.25 defines a protocol to connect a user's terminal (data terminal equipment, or DTE) to a PDN. The protocol requires that CCITT Recommendation X.25 support be resident in both the DTE and the data circuit-terminating equipment (DCE). X.25 Network Gateway is an example of required DTE-resident X.25 software. (Although some PDNs use CCITT Recommendation X.25 for their internal protocol, only a PDN's external behavior, from the perspective of the user, must match the CCITT Recommendation X.25 protocol).

Protocol Levels

CCITT Recommendation X.25 specifies three protocol levels:

- Physical level defines the electrical or modem interface between the data terminal equipment (DTE) and data circuit-terminating equipment (DCE), based on Recommendation X.21 or X.21bis (RS232).
- Link level defines the link access procedure for transferring packets accurately between the DTE and DCE. The link level consists of procedures for link set up, data transfer, and link disconnect. The protocol used is balanced link-access procedure (LAPB), a symmetric version of the high-level data link control (HDLC) protocol.
- Packet level defines procedures for call establishment, data transfer, flow control, error recovery, and call clearing between two communicating DTEs. Using the packet-level protocol, up to 4,096 logical channels can be multiplexed over a single link.

X.25 Multiplexing

The multiplexing capability of the CCITT Recommendation X.25 packet level allows the single physical link between the DTE and the DCE to be shared by multiple users. Data is transferred over logical channels, known as *virtual circuits*. Although the logical channels share the same physical link to the DCE, from the point of view of the parties communicating with each other, these logical channels appear to be private communications links.

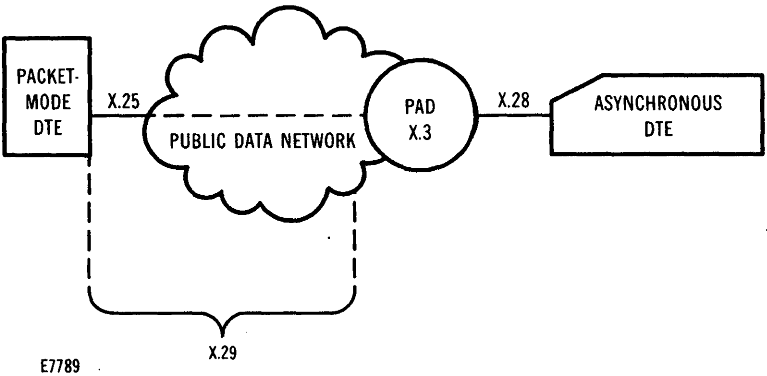
Intermixing of data from various logical channels onto the physical link is handled transparently to users by the X.25 packet level. Each transferred packet is marked with a logical channel number. This number enables the X.25 packet level to match packets and logical channels.

Up to 4,096 logical channels are supported by the CCITT Recommendation X.25 packet level (which can be limited by the PDN and by this gateway); any number of logical channels can be active simultaneously. Link use is allocated sequentially according to the number of the virtual circuit so no logical channel can monopolize the physical link. This random multiplexing technique allows maximum use of the connection between the DTE and the DCE, independent of the number of users and the amount of data transferred.

CCITT Recommendations X.3, X.28, and X.29

CCITT Recommendations X.3, X.28, and X.29 define the protocols for providing an interface between nonpacket-mode asynchronous (stop/start) terminals and a PDN (see Figure 2-1). These three procedures are informally referred to as the Interactive Terminal Interface (ITI) standards. Special software is required for these protocols to process the data from asynchronous (dumb) terminals into a format compatible with CCITT Recommendation X.25 (which defines the protocols for interfacing a packet-mode DTE with a PDN). This software is called a Packet Assembly/ Disassembly (PAD) facility. From the PDN, a PAD is viewed as a packet-mode DTE.

Figure 2-1 **Public Data Network Protocols**



Recommendation X.3 defines the parameters that specify an asynchronous terminal's characteristics to a PAD.

Recommendation X.28 defines the procedure for communicating these parameters between an asynchronous terminal and a PAD.

Recommendation X.29 defines the procedure for communicating between a PAD and a true packet-mode DTE over the PDN. This procedure allows a packet-mode DTE to use Recommendation X.3 parameters in a manner similar to that of an asynchronous terminal.

Logical Channels and Virtual Circuits

A logical channel is a data stream multiplexed over a physical link. X.25 networks use logical channels so that different users can share the same physical link. Logical channel numbers have only local significance; a logical channel number at one end of the connection may be different than the number at the other end.

A virtual circuit is a two-party connection that uses a logical channel. Once a virtual circuit is established over the network, packets do not need to carry a network destination address. Instead, they carry logical channel identification numbers from which the network derives a network destination address.

CCITT Recomendation X.25 defines 4,096 logical channels for each DTE-DCE physical link. These logical channels are divided into 16 logical channel groups, which forms the first part of the identification number. The logical channels within each group are then numbered from 0 to 255 (see Figure 2-2), which forms the second part. The number of channels actually supported depends on the gateway employed and on the PDN you're using.

By convention, logical channel 0 of logical channel group 0 is reserved for protocol operations. The remaining logical channels are divided into virtual circuit ranges at installation: two-way (both incoming and outgoing), incoming, outgoing, and permanent.

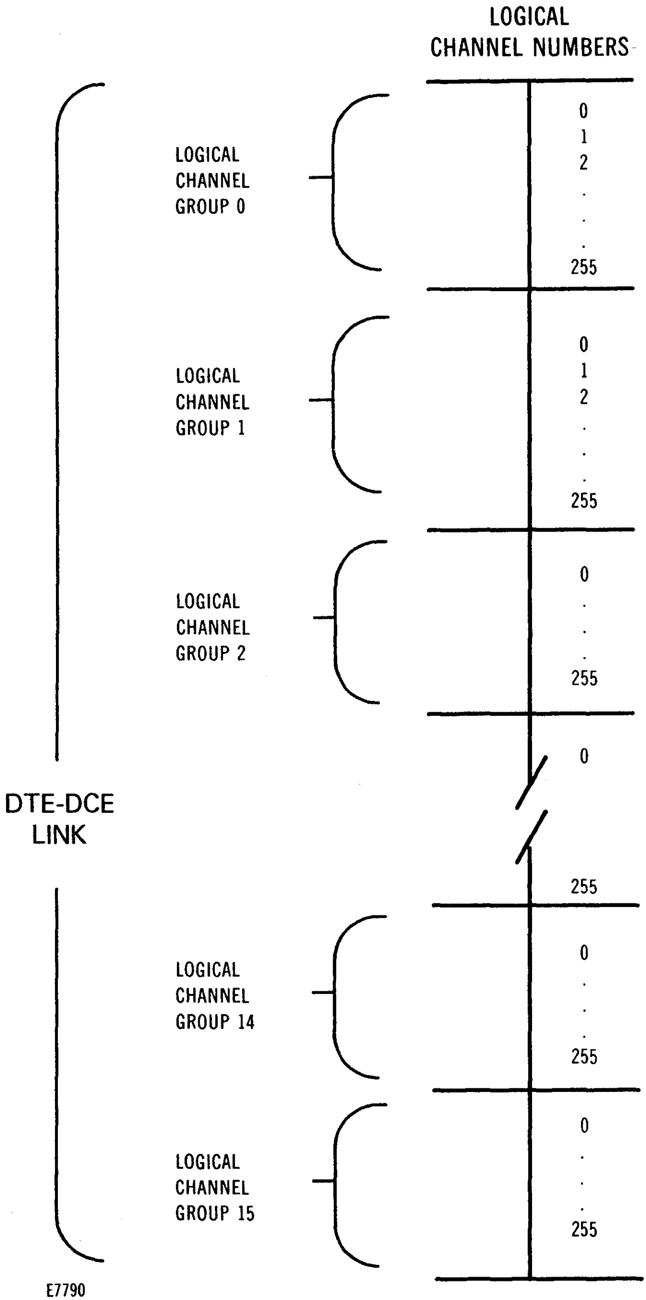
With virtual calls, the logical channel identification number is assigned when the virtual call is established. With permanent virtual circuits, the logical channel identification number is determined when you lease the facility.

The ordering of these virtual circuits on the DTE-DCE link is:

1. Permanent, which must begin with logical channel 1 of logical channel group 0
2. Incoming
3. Two-way
4. Outgoing

Only virtual circuit 0 and one other virtual circuit need to be present. Although virtual circuit ranges do not have to be contiguous, the logical channel numbers within each virtual circuit range must be contiguous.

Figure 2-2 **Logical Channels on a DTE-DCE Link**



1208055

Virtual Circuit Ordering

Figures 2-3, 2-4, and 2-5 are examples of virtual circuit ordering. In Figure 2-3, 19 virtual circuits are divided among four virtual circuit ranges; all are within logical channel group 0. These virtual circuits are ordered as follows:

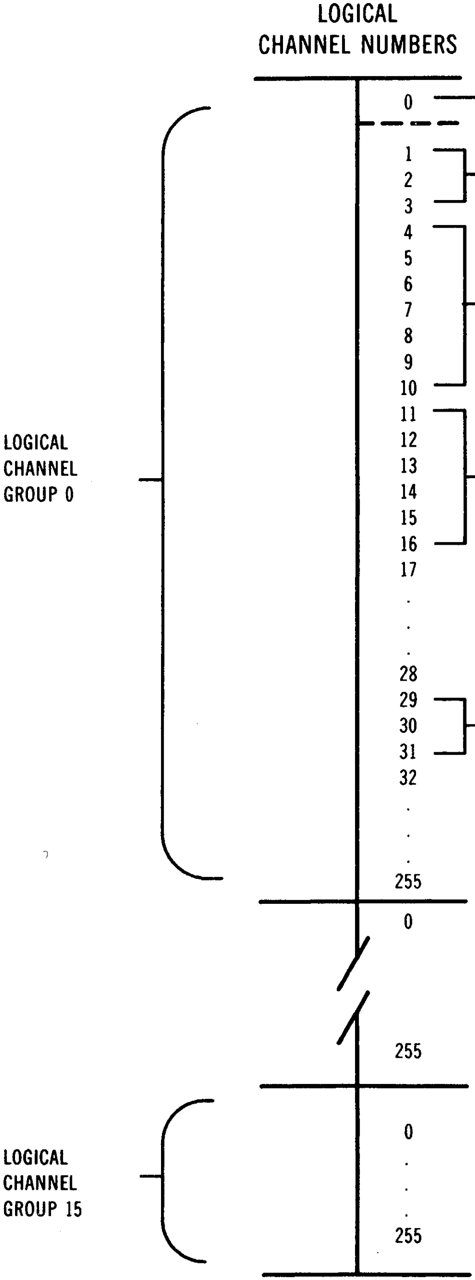
- Three permanent virtual circuits, beginning with logical channel 1.
- Seven incoming virtual circuits, beginning with logical channel 4.
- Six two-way virtual circuits, beginning with logical channel 11.
- Three outgoing virtual circuits, beginning with logical channel 29.

In Figure 2-4, 32 virtual circuits are two-way virtual circuits, beginning with logical channel 240 in logical channel group 12 and continuing to logical channel 15 in logical channel group 13.

In Figure 2-5, ten virtual circuits are divided between two ranges, each range in a different logical channel group. These virtual circuits are ordered as follows:

- Four permanent virtual circuits, beginning with logical channel 1 in logical channel group 0.
- Six two-way virtual circuits, beginning with logical channel 11 in logical channel group 2.

Figure 2-3 Example of Virtual Circuit Definition: Four Ranges Within One Logical Channel Group



E7791

1208055

Figure 2-4 **Example of Virtual Circuit Definition: One Range Spanning Two Logical Channel Groups**

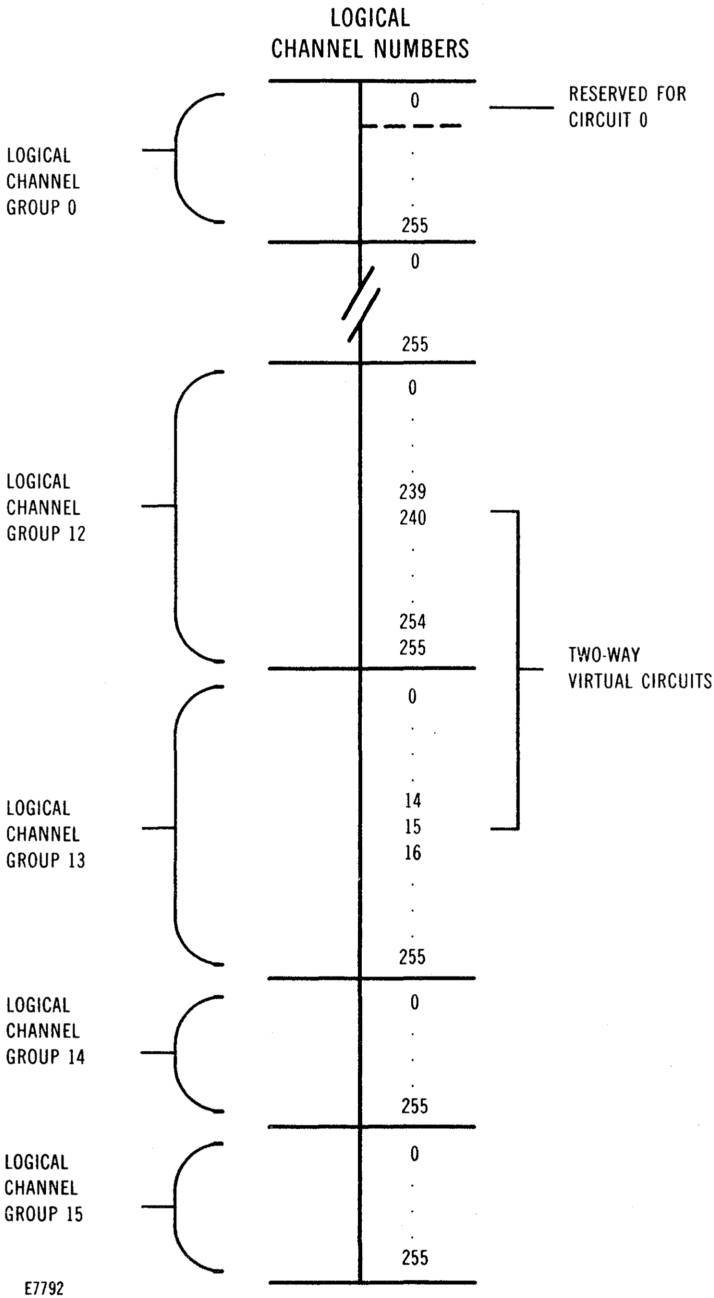
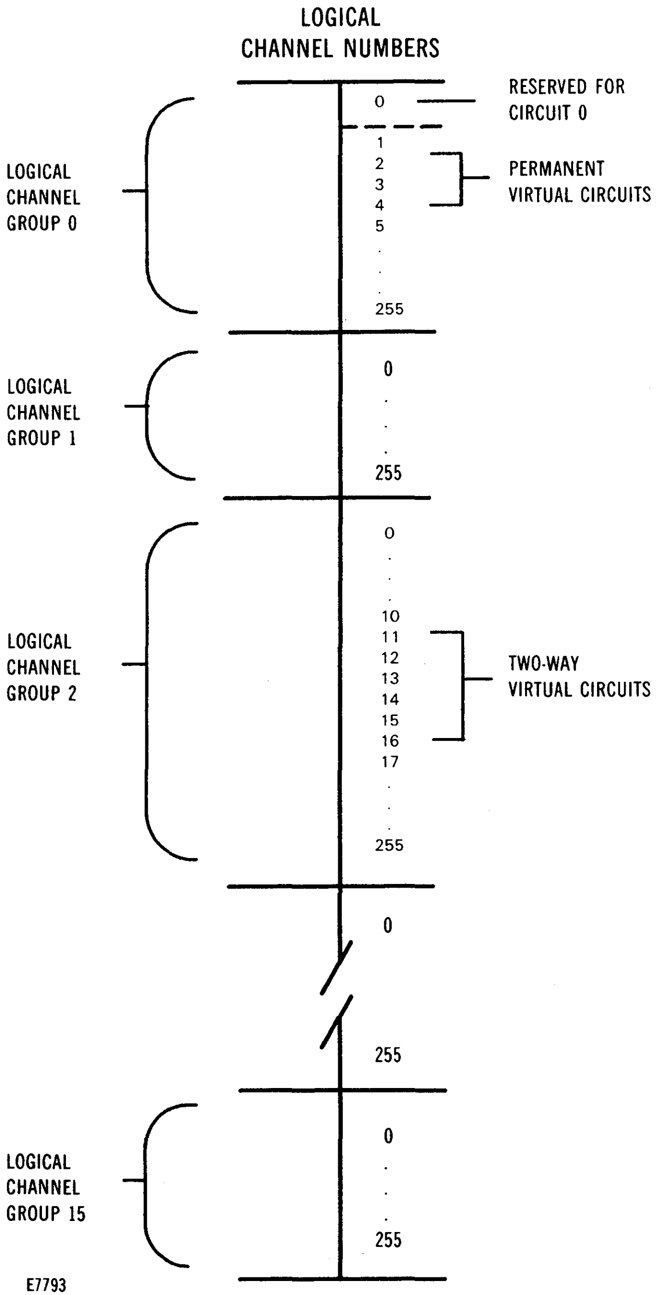


Figure 2-5 **Example of Virtual Circuit Definition: Two Ranges, Each in a Different Logical Channel Group**



1208055

Installation

This section explains how to install the X:25 Network Gateway on BTOS systems, including the XE 520. This section assumes that the person installing the program knows how to load software applications and knows how to operate a BTOS workstation.

BTOS Master, Cluster, and Standalone Workstations with Hard Disk Drives

1 Log on to your system.
2 Set the path to [d0]<sys>.
3 If you have a master workstation, power down all cluster workstations.
4 Insert the product disk into floppy disk drive [f0]. (Do not press RESET.)
5 At the Command line, type **Software Installation** and press GO.
6 Follow the prompts that appear on the screen to complete the installation.
7 When a message informs you that the installation is complete, remove the product disk and save it as an archive.
8 Reboot your workstation.
9 Turn on the cluster workstations.

Go to "Parameters and Response Options" later in this section to continue using the gateway.

BTOS Standalone Workstations with Dual Floppy Disk Drives

1 Make a copy of the X.25 Network Gateway product disk and save the original as an archive. (Subsequent references to the product disk assume you are working with the copy.)
2 Insert disk 1 of the dual floppy standalone operating system disks into floppy disk drive [f0]. Boot the system. Leave disk 1 in drive [f0].

3 Insert the product disk in floppy disk drive [f1].

4 Copy the Exec.run file from [f0]<sys> to [f1]<sys>.

5 Copy the Request.F.sys file from [f1]<sys> to [f0]<sys>.

6 Reboot the system (leaving the modified operating system disk in drive [f0]).

Note: Copying Exec.run from the operating system to the product disk is required only the first time the product disk is used. For subsequent uses of the gateway, insert the product disk into floppy disk drive [f0].

Go to "Parameters and Response Options" later in this section to continue using the gateway.

BTOS Systems with IDS Modules

See the *BTOS Intelligent Data Communications Module (IDMSS) System Software Opeartions Guide* for information on running the gateway on the IDS module.

XE 520 Master Systems

1 Log on to the system with user *ADMIN*.

Note: The following steps must be completed for each "User" command file on the XE520 that requires access to the commands.

2 Set the path to [!sys]<sys>.

3 Insert the product disk into floppy disk drive [f0]. (Do not press RESET.)

4 At the Command line, type **XESoftware Installation** and press GO.

5 Follow the prompts that appear on the screen to complete the installation.

6 When a message informs you that the installation is complete, remove the product disk and save it as an archive.

To edit JCL files, see "Using JCL Files on XE 520 Systems" before you reboot your system.

7 Reboot your workstation.

8 Turn on the cluster workstations.

Go to "Parameters and Response Options" later in this section to continue using the gateway.

Using JCL Files on XE 520 Systems

You can configure any XE 520 master to load the gateway during system initialization by modifying the appropriate JCL file.

Depending on whether your XE 520 contains a cluster or terminal processor board (or both), use the Editor to edit the following cluster processor (CP) or terminal processor (TP) file:

[Sys]<Sys>InitCp*nn*.jcl (where *nn* is the CP number)
[Sys]<Sys>InitTp*nn*.jcl (where *nn* is the TP number)

and add the following line:

$Run [!Sys]<Sys>X25.Run, (supply installation parameters)

The user-specified parameters are separated by commas and are:

[Net Address]
[Number of VCs]
[LC group no. for two-way VCs]
[Starting LCn for two-way VCs]
[Number of incoming-only VCs]
[LC group no. for incoming-only VCs]
[Starting LCn for incoming-only VCs]
[Number of outgoing-only VCs]
[LC group no. for outgoing-only VCs]
[Starting LCn for outgoing-only VCs]
[Number of permanent VCs]
[Channel]
[Modulus]
[Max packet size]
[Default packet size]
[Default window size]
[PDN Type]
[Wait for DCE Ack]

See "Parameters and Response Options" later in this section for descriptions of these parameters.

Note: The allowable values for [Channel] are 1 or 2 for an XE 520 cluster processor and 1 through 4 for a terminal processor.

To issue the default values, type the following JCL command in Cpnn.jcl (or Tpnn.jcl):

```
$Run [!Sys]<Sys>X25.Run
```

After saving the file and leaving the Editor, edit Cpnn.cnf (or Tpnn.cnf for a terminal processor). Delete any line of text that inappropriately defines your chosen channel as asynchronous.

To assign the following:

- Gateway network address 1234567890
- 24 virtual circuits, with the two-way group number of 2, the starting LC number of 1, and 16 incoming virtual circuits in group 1 starting at LC number 1
- Channel B of an XE terminal processor
- Maximum packet size of 256 K-bytes
- Transpac as the PDN type
- Instruct the gateway not to wait for DCE acknowledgments

Type these assignments on a JCL command line in the order that they appear in the BTOS Executive Command form, separating each parameter with a comma. A comma marks the space where you want the gateway to use the default values:

```
$Run [!Sys]<Sys>X25.Run,1234567890,24,2,1,16,1,1,,,,,B,,256,,,3,N
```

In this example, you would next edit Tpnn.cnf. Delete any line of text that inappropriately defines your chosen channel as asynchronous.

Installation via Command Line Interpreter

You can install the gateway on the XE 520 via the Command Line Interpreter (CLI) of a BTOS workstation. This procedure is similar to installation via JCL file; it involves a modification of the appropriate file: either Cp*nn*.cnf (where *nn* is the CP number) or Tp*nn*.cnf (where *nn* is the TP number). Instead of using a JCL file, the $Run image is entered through the CLI.

If all channels of the CP are dedicated to asynchronous service, modification of the Cpnn.cnf file is also necessary to provide a synchronous line for the gateway. Likewise, if all channels of the terminal processor are dedicated to asynchronous service, you must modify Tpnn.cnf.

Modify the Cpnn.cnf file as follows:

1 Edit Cpnn.cnf (or Tpnn.cnf). Delete the line of text that inappropriately defines your chosen channel.
2 The command that invokes the X.25 Network Gateway from the Command Line Interpreter is:

   $Run [!Sys]<Sys>X25.Run,(supply installation parameters as described earlier under "Using JCL Files on XE 520 Systems.")

Parameters and Response Options

Software installation creates five new commands, including the Install X.25 command.

To invoke the Install X.25 command:

1 At the Command line, type **Install X.25** and press RETURN.
2 The following is diplayed:

   Install X.25
   [Net Address]
   [Number of VCs]
   [LC group no. for two-way VCs]
   [Starting LCn for two-way VCs]
   [Number of incoming-only VCs]
   [LC group no. for incoming-only VCs]
   [Starting LCn for incoming-only VCs]
   [Number of outgoing-only VCs]
   [LC group no. for outgoing-only VCs]
   [Starting LCn for outgoing-only VCs]
   [Number of permanent VCs]
   [Channel]
   [Modulus]
   [Max packet size]
   [Default packet size]
   [Default window size]
   [PDN Type]
   [Wait for DCE Ack]

Note: *VC* stands for virtual circuit; *LC* stands for logical channel.

Parameter Descriptions

[Net Address] The network address of the X.25 Network Gateway. (For further information, see Section 4.) Up to 15 digits may be entered.

[Number of VCs] The number of all virtual circuits to be allocated. This number ranges from 2 to 64. The default is either 2, for a standalone workstation, or the total number of workstations, including the master, in a cluster configuration.

Note: Although up to 64 virtual circuits (VCs) are supported, the actual number allocated usually equals the number of PDN VCs to which the user has subscribed plus 1 (for VC 0). This requirement may vary, depending on the PDN host country.

The number of two-way, incoming, outgoing, and permanent VCs cannot exceed the number of allocated VCs. X.25 Network Gateway calculates the number of two-way circuits by subtracting the number of incoming, outgoing, and permanent VCs from the number of allocated VCs.

The responses to the following set of parameters should correspond with those supplied by the PDN. See "Logical Channels and Virtual Circuits" in Section 2 for more information on how CCITT has defined logical channels for each DTE-DCE physical link.

[LC group no. for two-way VCs] The logical channel group number for two-way virtual circuits. The number can range from 0 (the default) to 15.

[Starting LCn for two-way VCs] The starting logical channel number for two-way virtual circuits. The number can range from 0 to 255. The default is the channel number of the last incoming-only virtual circuit plus 1.

[Number of incoming-only VCs] The number of virtual circuits to be allocated for incoming only calls. The number can range from 0 (the default) to the total number of allocated virtual circuits.

[LC group no. for incoming-only VCs] The logical channel group number for incoming virtual circuits. The number can range from 0 (the default) to 15.

[Starting LCn for incoming-only VCs] The starting logical channel number for incoming virtual circuits. The number can range from 0 to 255. The default is the number of permanent virtual circuits plus 1.

[Number of outgoing-only VCs] The number of virtual circuits to be allocated to outgoing calls only. The number can range from 0 (the default) to the total number of allocated virtual circuits.

[LC group number for outgoing-only VCs] The logical group number for outgoing virtual circuits. The number can range from 0 (the default) to 15.

[Starting LCn for outgoing-only VCs] The starting logical channel number for outgoing virtual circuits. The number can range from 0 to 255. The default is the channel number of the last two-way virtual circuit plus 1.

[Number of permanent VCs] The number of permanent virtual circuits can range from 0 (the default) to the total number of allocated virtual circuits.

[Channel] The communication channel, which can be A, B (the default), or one of the four-port data communications (DCX) ports: 1A, 1B, 1C, 1D, 2A, 2B, 2C, and 2D. 1A through 1D correspond to the channels closest to the CPU module; 2A through 2D correspond to the four farthest from the CPU module. Your workstation can have only two four-port data communications modules attached to it. For the XE 520, channel values are A or B for a cluster processor; for a terminal processor, channel values are A, B, C, or D.

[Modulus] The packet sequence numbering scheme. Modulo counts represents the number of packets that can be transmitted without an acknowledgment by the receiver. This release supports modulo 8.

[Max packet size] The maximum packet size, in bytes, that the gateway can send or receive. Possible values are 16, 32, 64, 128 (the default), 256, 512, and 1024.

[Default packet size] The default packet size, in bytes. The default packet size can be negotiated on a per-call basis. Possible values are 16, 32, 64, 128 (the default), 256, 512, and 1024. This parameter cannot exceed the [Max packet size].

[Default window size] The default packet window size. The number can range from 1 and 6. The default is 2.

[PDN Type] The identification number of your public data network. Possible values range from 0 to 6. Functional differences are:

0 Telenet, Tymnet (USA), or Datanet-1

1 Datex–P (Germany, Austria)
T1 timer is three seconds
N2 counter is 10
VC 0 may be used for data

2 PSS (United Kingdom)

3 Transpac (France)
VC 0 may be used for data
No subaddressing

4 Austpac (Australia)

5 Datapac (Canada)

6 SpecialTranspac (France)
VC 0 may be used for data
Subaddressing is allowed
The calling request is not placed in the Call Request and Call Accept packets

For more information on specific PDN differences, contact your subsidiary or PDN support representative.

[Wait for DCE Ack] Type **Y** to instruct the gateway to wait for acknowledgment from the network before allowing the application to send more packets to it. Type **N** to tell the gateway not to expect a packet-level ACK until a default-window-size number of packets have been sent.

Deinstalling the Gateway

Type **Deinstall X.25** at the Command line and press GO to remove the X.25 Network Gateway from system memory. The workstation on which you invoke the command must:

- Contain the gateway
- Have no active virtual circuits
- Be using a multipartition operating system (which is not running Context Manager)

Note: You cannot deinstall the gateway on an XE 520 master or an IDS module.

Packet Access Method

This section describes packet access method operations and network interaction.

The packet access method provides access to the capabilities of the X.25 Network Gateway system service through BTOS. It enables your program to send and receive individual X.25 data and control packets and to directly monitor the establishment of X.25 connections to a public data network (PDN). The packet access method supports all BTOS user languages, including Pascal, COBOL, FORTRAN, and BASIC.

PAM procedures for accessing the gateway now permit access over a B-NET transport. The gateway still supports the old procedures. For example, AcceptX25Call still works, but you must use AcceptX25CallNet in order to access the gateway via B-NET. In any event, incorporate the new procedures in your applications.

Use

The packet access method allows the communications system programmer to design sophisticated, CCITT Recommendation X.25-based communication products such as interfaces to other computer networks, server programs, and packet assembly/disassembly (PAD) facilities. The applications programmer must understand the format and contents of X.25 packets, as specified in Section 6 of CCITT Recommendation X.25 (see Appendix D for reference information).

Linking a Packet Access Method Program

After writing a program using the packet access method, the object modules you create must be linked by the BTOS Linker to produce a run file. This can be done with the Link command. On the [Libraries] line it is necessary to specify [sys]<sys>rqlablx25.lib. Without this, calls to the X.25 Gateway using the procedural interface will not link properly.

A standard X.25 user session consists of a call establishment phase, a data transfer phase, and a call clearing phase. Status can be monitored during sessions.

The call establishment phase places or receives a call. The data transfer phase reads and/or writes data, and the call clearing phase terminates a session on a nonpermanent virtual circuit.

Operations

Detailed descriptions of each packet access method operation follow Table 4-3.

Conventions for Naming Application Requests

One BTOS mechanism for accessing application features is a procedural access. Procedural access is how PAM accesses the gateway. This method automatically formats all messages between BTOS system services and an application system process (in this case the gateway) into a formalized structure: the request block.

A request block interface contains the specification and parameters of the desired system service. The user can format his own request block. The block contains a request code field, a response exchange field, and several application-specific fields. These fields show the size of data (preceded by an s, such as sCntInfo or sFacilities), contain a status code (preceded by erc, such as ercRet), or point to the memory location where certain data resides (preceded by a *p*, such as pFacilities). See Section 3 of the *BTOS Reference Manual, Volume 1* for a comprehensive description of BTOS interprocess communication.

Call Establishment and Clearing

Five operations enable a calling process to begin (establish) and end (clear) calls over virtual circuits. Table 4-1 describes them in detail.

Packet and Window Size Negotiation

The gateway allocates packet buffers according to the maximum packet size (default = 128) specified at installation time. Do not negotiate for a larger packet size per call.

The gateway checks the maximum packet size to be used during the call to be sure it is within the size selected at installation, and that the window size is within the range 1 through 7, specified by CCITT.

Table 4-1 **Call Establishment and Clearing Operations**

Operation	Description
AcceptX25CallNet	Requests transmission of a CALL ACCEPT packet to complete the establishment of an incoming call.
ClearX25CallNet	Requests transmission of a CLEAR REQUEST packet to either refuse an incoming call or terminate an existing call.
ConnectX25PermanentNet	Reserves a permanent virtual circuit.
InitiateX25CallNet	Requests transmission of a CALL REQUEST packet to initiate a call to the called number.
NotifyNextIncomingCallNet	Notifies the X.25 Network Gateway system service that the calling process wishes to receive an incoming call.

Data Transfer

Once a virtual circuit has been established, four operations can transfer data over it: ReadX25PacketNet, ResetX25CallNet, WriteX25InterruptNet, and WriteX25PacketNet. These operations can be used in any sequence. Table 4-2 describes the operations. Refer to the alphabetized listing of all operations following Table 4-3 for more information.

Table 4-2 **Data Transfer Operations**

Operation	Description
ReadX25PacketNet	Returns DATA and INTERRUPT packets received by the X.25 Network Gateway system service from the PDN to the application.
ResetX25CallNet	Requests that a RESET REQUEST packet be transmitted in order to clear all outstanding read and write requests. It also reinitializes the flow control procedures for a virtual circuit without clearing the call.
WriteX25InterruptNet	Transmits an INTERRUPT packet over the PDN.
WriteX25PacketNet	Allows users to transmit a DATA packet over the PDN.

Status Monitoring

The QueryX25StatusNet operation monitors the status of gateway activity over the public data network. Table 4-3 gives a general description of the operation. For more details see the individual descriptions that follow.

Table 4-3 **Status Monitoring Operation**

Operation	Description
QueryX25StatusNet	Returns a set of statistics about gateway activity over the PDN.

AcceptX25CallNet

The AcceptX25CallNet operation requests transmission of a CALL ACCEPT packet to complete the establishment of an incoming call. AcceptX25CallNet can be issued after the successful completion of a NotifyNextIncomingCallNet operation.

Only one AcceptX25CallNet can be outstanding for a virtual circuit at any given time. If more are sent, the excess requests are returned with status code 8502.

To take advantage of the fast select facility or to engage in facility negotiation, the optional facilities field and/or the user data field in the CALL ACCEPT packet can be sent to the caller. Details of the required facilities and user data fields formats are given in Section 7 of CCITT Recommendation X.25 (see Appendix D). For further information on facilities, see "X.25 Network Interaction," later in this section.

Procedural Interface

AcceptX25CallNet *(vch, pFacilities, sFacilities, pUserData, sUserData, bGfi, wFh): ErcType*

vch	The virtual circuit handle returned by a NotifyNextIncomingCall operation.
pFacilities sFacilities	A buffer containing facilities data to be included in the CALL ACCEPT packet.
pUserData sUserData	A buffer containing user data to be included in the CALL ACCEPT packet. This field is used for fast select calls only.
bGfi	The one-byte general format indicator. Bit 6 of this byte can be set if the D bit is to be used during the call (bit 7 is the most significant bit). (See "Specifying and Setting M, D, and Q bits" later in this section).
wFh	The file handle returned by the gateway for B-NET routing use.

Request Block

Offset	Field	Size (bytes)	Contents
0	sCntInfo	2	8
2	nReqPbCb	1	2
3	nRespPbCb	1	0
4	userNum	2	
6	exchResp	2	
8	ercRet	2	
10	rqCode	2	16503
12	Fh	2	
14	lVch	2	
16	reserved	2	
18	bGFI	1	
19	reserved	1	
20	pFacilities	4	
24	sFacilities	2	
26	pUserData	4	
30	sUserData	2	

ClearX25CallNet

The ClearX25CallNet operation requests transmission of a CLEAR REQUEST packet to either refuse an incoming call or terminate an existing call. If a call is cleared by the PDN, the status code 8513 (DCE clear) is returned to any outstanding data transfer requests.

Only one ClearX25CallNet can be outstanding for a virtual circuit at any given time. If more are sent, the excess requests are returned with a status code of 8502.

Procedural Interface

ClearX25CallNet *(vch, bReason, pUserData, sUserData, wFh): ErcType*

vch	The virtual circuit handle.
bReason	A one-byte diagnostic code to be included in the CLEAR REQUEST packet.
pUserData sUserData	pUserData points to and sUserData describes a buffer containing user data to be included in the CLEAR REQUEST packet. This field should be used only if ClearX25CallNet is used to reject a Fast Select call (this is not checked by the X.25 Network Gateway system service).
wFh	The file handle returned by the gateway for B-NET routing use.

Request Block

Offset	Field	Size (bytes)	Contents
0	sCntInfo	2	8
2	nReqPbCb	1	1
3	nRespPbCb	1	0
4	userNum	2	
6	exchResp	2	
8	ercRet	2	
10	rqCode	2	16505
12	Fh	2	
14	iVch	2	
16	bReason	1	
17	reserved	3	
20	pUserData	4	
24	sUserData	2	

ConnectX25PermanentNet

The ConnectX25PermanentNet operation reserves a permanent virtual circuit for a user. ConnectX25PermanentNet returns a virtual circuit handle that can then be used to reference the permanent virtual circuit in subsequent data transfer requests.

Only one ConnectX25PermanentNet can be outstanding for a virtual circuit at any given time. If more are sent, the excess requests are returned with a status code of 8502.

No packets are transmitted by ConnectX25PermanentNet.

Procedural Interface

ConnectX25PermanentNet *(lcn, pVchRet, fLongLive, pNodeName, sNodeName, pFHRet): Erctype*

lcn	A logical channel number ranging from 1 through the maximum number of allocated virtual circuits. The logical channel number must have been specified as a permanent virtual circuit at installation.
pVchRet	The memory address of a word where the virtual circuit handle is to be returned.
flonglive	Should be set to *false* unless you are a system service wishing to run under a single-partition operating system.
pNodeName	A pointer to the node name where the X.25 Network Gateway is installed. Cannot exceed 12 characters, including curly brackets.
sNodeName	A word describing the length of the node name.
pFHRet	The memory address of a word to which the B-NET file handle is returned.

Request Block

Offset	Field	Size (bytes)	Contents
0	sCntInfo	2	4
2	nReqPbCb	1	1
3	nRespPbCb	1	2
4	userNum	2	
6	exchResp	2	
8	ercRet	2	
10	rqCode	2	16512
12	inLCN	2	
14	flonglive	1	
15	Reserved	1	
16	pNodeName	4	
20	sNodeName	2	
22	pFHRet	4	
26	sFHMax	2	
28	pVchRet	4	
32	sVchRet	2	2

InitiateX25CallNet

The InitiateX25CallNet operation requests transmission of a CALL REQUEST packet to initiate a call to the called number, which can be any user in the PDN. The local network address number, specified at installation, is automatically filled in as the calling number by the X.25 Network Gateway system service, except in the case of the special Transpac PDN type.

Only one InitiateX25CallNet can be outstanding for a virtual circuit at any given time. If more are sent, the excess requests are returned with a status code of 8502.

Optional parameters for facilities and user data can be specified. The specified parameters must be in accordance with applicable standards, either those of CCITT Recommendations X.25 and X.29, those of other international or national standards bodies, or those unique to the specific PDN. For further information on facilities, see "X.25 Network Interaction" later in this section.

Procedural Interface

InitiateX25CallNet *(iNet, pVchRet, pPacketRet, sPacketMax, psPacketRet, pCalled, sCalled, pFacilities, sFacilities, pUserData, sUserData, bGfi, fLongLive, pNodeName, sNodeName): ErcType*

iNet	The communications network identifier, for which the only value currently allowed is 0.
pFhVchRet	A pointer to a double word; the high-order word contains the memory address of a word to which the virtual circuit handle is returned. The low-order word contains the memory address of a word to which the B-NET file handle is returned.
pPacketRet sPacketMax	Describe a buffer into which either the CALL ACCEPT packet is copied if the call is established or the CLEAR REQUEST packet is copied if the call is rejected. pPacketRet is the pointer to the memory location of the buffer and sPacketMax indicates the maximum size of the buffer.
psPacketRet	The memory address of a word where the size of the received CALL ACCEPT or CLEAR REQUEST packet is to be returned.
pCalled sCalled	Describe a buffer containing the called number in ASCII. (The gateway converts this number to binary coded decimal (BCD) and places it in the CALL REQUEST packet.) pCalled points to the buffer memory location and sCalled contains the buffer size in bytes. The buffer can contain up to 15 ASCII characters and cannot contain hexadecimal ASCII values.
pFacilities sFacilities	Describe a buffer containing facilities data to be included in the CALL REQUEST packet. pFacilities is a pointer to the b uffer memory location while sFacilities contains the buffer size in bytes.
pUserData sUserData	Describe a buffer containing user data to be included in the CALL REQUEST packet. The buffer can contain up to 16 bytes for normal calls or 128 bytes for fast select calls. The first four bytes identify high-level protocols. (For more information, see CCITT Recommendation X.25.) pUserData is a pointer to the buffer memory location while sUserData contains the size of the buffer in bytes.

bGfi	The one-byte general format indicator. Bit 6 of this byte should be set if the D bit is to be used during the call (bit 7 is the most significant bit). (See "Specifying and Setting M, D, and Q Bits" later in this section.)
fLonglive	Should be set to *false* unless you are a system service wishing to run under a single-partition operating system.
pNodeName	The pointer to the node name where the X.25 Network Gateway is installed. Cannot exceed 12 characters, including curly brackets.
sNodeName	Describes the length of the node name.

Request Block

Offset	Field	Size (bytes)	Contents
0	sCntInfo	2	4
2	nReqPbCb	1	4
3	nRespPbCb	1	3
4	userNum	2	
6	exchResp	2	
8	ercRet	2	
10	rqCode	2	16504
12	iNet	2	
14	buserGfi	1	
15	fLonglive	1	
16	pNodeName	4	
20	sNodeName	2	
22	pFacilities	4	
26	isFacilities	2	
28	pUserData	4	
32	isUserData	2	
34	pCalled	4	
38	isCalled	2	
40	pFHVchRet	4	
44	sFHVchRet	2	
46	pPacketRet	4	
50	sPacketMax	2	
52	psPacketRet	4	
56	ssPacketRet	2	2

NotifyNextIncomingCallNet

The NotifyNextIncomingCallNet operation notifies the X.25 Network Gateway system service that the calling process wishes to receive an incoming call of a certain protocol and within a certain port range. For NotifyNextIncomingCallNet to successfully complete, both the protocol and the port number of a received CALL REQUEST packet must match those specified in NotifyNextIncomingCallNet.

The gateway queues NotifyNextIncomingCallNet requests on the order of 1 queue per virtual circuit. This queue can contain as many requests as there are virtual circuits per line. If more are sent, the excess requests are returned with status code 8502.

The first byte of the user data field of the CALL REQUEST packet may contain a protocol identifier. That is matched against the set of protocol identifiers specified in NotifyNextIncomingCallNet. This set is specified by mask/value pairs, where the mask is a byte that is logically ANDed with the received protocol identifier, and the value is compared with the result. If mask/value pairs are not specified (sRgClass of 0), any incoming protocol is accepted.

If no protocol is specified in the CALL REQUEST packet (that is, no user data is present), 0 is used as the value of the received protocol identifier.

Subaddressing

The port number, or subaddress, is the last two digits of the called address in CALL REQUEST packets. You can specify subaddress when the gateway is installed by appending it to the network address. This assigns the subaddress to the entire gateway. In this case, applications cannot specify a different subaddress. Otherwise, you may install the gateway without a subaddress. The specific application would then assign a subaddress by using the high and low port parameters in the NotifyNextIncomingCallNet request block.

A port number is valid only if the number of digits of the incoming address is two digits longer than the network address specified at installation. This incoming port number must be within the specified range of high and low port numbers for NotifyNextIncomingCallNet to successfully complete and return the call to the user. To receive a call on a specified port, the last two digits of the called address (the subaddress) in the InitiateX25Call and InitiateX25CallNet must match the specified port (if low port and high port are equal), or be within the specified port range (if the range option is used) of the [NotifyNextIncomingCallNet] or [NotifyNextIncomingCall] requests. The gateway will not accept calls where these addresses do not match. If a port number is not present in the called address of the INCOMING CALL packet, port 0 is assumed.

To use subaddressing when initiating calls, the subaddress of the party you wish to call may be appended to the called address in the InitiateX25Call and InitiateX25CallNet requests.

No packets are transmitted by NotifyNextIncomingCallNet.

Procedural Interface

NotifyNextIncomingCallNet *(iNet, pVchRet, pPacketRet, sPacketMax, psPacketRet, pRgClass, sRgClass, lowPort, highPort, timeOut, fLongLive, pNodeName, sNodeName): ErcType*

iNet	The communications network identifier for which the only value currently allowed is 0.
pFhVchRet	A pointer to a double word; the high-order word contains the memory address of a word to which the virtual circuit handle is returned. The low-order word contains the memory address of a word to which the B-NET file handle is returned.
pPacketRet sPacketMax	Describe a buffer to which the received INCOMING CALL packet is copied. pPacketRet is a pointer to the buffer; sPacketMax indicates the buffer size in bytes.
psPacketRet	The memory address of a word that contains the size of the received INCOMING CALL packet.

pRgClass sRgClass	Describe an array of protocol class identifiers; each identifier consists of a one-byte mask that is followed by a one-byte value. If sRgClass is 0, checking for protocols is disabled for this NotifyNextIncomingCallNet.
lowPort	As word containing the lower bound on the range of port numbers (the last two digits of the called address) for which calls should be notified. The required format for the port number is a pair of ASCII digits, ranging from 0 to 9, with the high-order digit in the high-order byte of the word. No hexadecimal ASCII values are accepted. (See "Subaddressing" under "NotifyNextIncomingCallNet" earlier in this section.)
highPort	A word containing the upper bound on the range of port numbers for which calls should be notified. The required format for the port number is a pair of ASCII digits, ranging from 0 to 9, with the high-order digit in the high-order byte of the word. No hexadecimal ASCII values are accepted. (See "Subaddressing" under "NotifyNextIncomingCallNet" earlier in this section.)
timeOut	A word containing the maximum amount of time to wait for an incoming call, in units of 100 ms. However, the X.25 Network Gateway system service rounds the value to the next second. A value of 0 indicates no waiting; NotifyNextIncomingCall returns with status code 8507 (user-specified time-out) if no packets have been received from the PDN. A value of 0FFFFh indicates an indefinite wait.
fLonglive	Set to false unless you are a system service wishing to run under a single-partition operating system.
pNodeName	A pointer to the node name where the gateway is installed. Cannot exceed 12 characters including curly brackets.
sNodeName	A word defining the length of the node name.

Request Block

Offset	Field	Size (bytes)	Contents
0	sCntInfo	2	10
2	nReqPbCb	1	2
3	nRespPbCb	1	3
4	userNum	2	
6	exchResp	2	
8	ercRet	2	
10	rqCode	2	16502
12	iNet	2	
14	ilowPort	2	
16	ihighPort	2	
18	itimeOut	2	
20	fLonglive	1	
21	Reserved	1	
22	pNodeName	4	
26	sNodeName	2	
28	pRgClass	4	
32	sRgClass	2	
34	pFhVchRet	4	
38	sFhVchMax	2	
40	pPacketRet	4	
44	sPacketMax	2	
46	psPacketRet	4	
50	ssPacketRet	2	2

QueryX25StatusNet

The QueryX25StatusNet operation returns a set of statistics concerning gateway activity over the PDN. Information can be provided for either a single logical channel or several logical channels.

Procedural Interface

QueryX25StatusNet *(iNet, pStatusBuffer, sStatusBuffer, blcnFirst, blcGnFirst, nLc, vch, pnLcRet, pNodeName, sNodeName): ErcType*

iNet	The communications network identifier for which the only value currently allowed is 0.
pStatusBuffer sStatusBuffer	Describe a buffer to which the status information is to be returned. For more information, see "Network Statistics Buffer" and Table 4-6.
bLcnFirst	The logical channel number of the first of several virtual circuits for which circuit-specific statistics are to be returned. bLcnFirst is ignored if vch is not 0.
bLcGnFirst	The logical channel group number of the first of several virtual circuits for which circuit-specific statistics are to be returned. BLcGnFirst is ignored if vch is not 0.
nLc	The number of logical channels for which circuit-specific statistics are to be returned. If vch is not 0, nLc is ignored. If sStatusBuffer is too small, the information is truncated. If vch and nLc are both 0, logical channel statistics are not returned.
vch	The handle of a single virtual circuit for which statistics are to be returned. If vch is not 0, blcnFirst, blcGnFirst, and nLc are ignored. If vch is 0, statistics for several logical channels are returned, based on the values in blcnFirst, blcGnFirst, and nLc.
pnLcRet	The memory address of a byte. The number of logical channels for which statistics are returned is placed in that byte.
pNodeName	A pointer to the node name where the gateway is installed. Cannot exceed 12 characters including curly brackets.
sNodeName	Describes the length of the node name.

Request Block

Offset	Field	Size (bytes)	Contents
0	sCntInfo	2	8
2	nReqPbCb	1	2
3	nRespPbCb	1	1
4	userNum	2	
6	exchResp	2	
8	ercRet	2	
10	rqCode	2	16511
12	iNet	2	
14	bnlcFirst	1	
15	bnlcGnFirst	1	
16	nLc	2	
18	vch	2	
20	pNodeName	4	
24	sNodeName	2	
26	pStatusBuffer	4	
30	sStatusBuffer	2	
32	pnLcRet	4	
36	snLcRet	2	1

Network Statistics Buffer

The network statistics buffer is set up by the user application and is pointed to by pStatusBuffer and sStatusBuffer. See Table 4-4. The first 64 bytes contain line information and are always returned. If you request logical channel status, the X.25 Network Gateway system service appends logical channel status blocks. One 34-byte logical channel status block is appended for each logical channel.

The layout of the network statistics buffer pointed to by pStatusBuffer and sStatusBuffer allow an extra byte for the localNetAddress. An extra byte has also been added to each logical channel status block appended to the buffer for the RemoteNetAddress and to iLine.

Note: The Queryx25Status request continues to return the previous buffer layout. Refer to 5.0 release documentation.

Table 4-4 **Network Statistics Buffer**

Offset	Field	Size (bytes)
0	iLine	2
2	nVcConf	2
4	nVcActive	2
6	nVcOutofOrder	2
8	fLinkLevelUp	1
9	fPacketLevelUp	1
10	nLinkResets	2
12	nDteRestarts	2
14	nDceRestarts	2
16	dataPacketsReceived	4
20	dataPacketsTransmitted	4
24	nIncomingCalls	2
26	nOutgoingCalls	2
28	timeLastChange	4
32	localNetAddress	8
40	cbLocalNetAddress	1
41	cTicks	4
45	cRxActiveTicks	4
49	cTxActiveTicks	4
53	reserved	11
64	rgLcStatus	nLc*34

The following is a description of the network statistics buffer fields.

iLine	The communications channel represented by two ASCII characters.
nVcConf	The maximum number of configured virtual circuits.
nVcActive	The number of virtual circuits currently active.
nVcOutofOrder	The number of virtual circuits currently out of order.
fLinkLevelUp	*True* if the X.25 Network Gateway link-level protocol is operational.
fPacketLevelUp	*True* if the X.25 Network Gateway packet-level protocol is operational.
nLinkResets	The number of link-level protocol resets that have occurred since the connection of the physical line.
nDteRestarts	The number of packet-level protocol restarts initiated by the X.25 Network Gateway system service since the last link-level protocol restart.

nDceRestarts	The number of packet-level protocol restarts initiated by the PDN since the last link-level protocol restart.
dataPacketsReceived	The number of DATA packets received since the last packet-level protocol restart.
dataPacketsTransmitted	The number of DATA packets transmitted since the last packet-level protocol restart.
nIncomingCalls	The the number of calls answered since the last packet-level protocol restart.
nOutgoingCalls	The number of calls established since the last packet-level protocol restart.
timeLastChange	The system date and time, in BTOS format (32 bits), that was recorded the last time either the link level changed state or the packet level became operational, whichever was most recent.
localNetAddress	The local network address inserted in all outgoing CALL REQUEST packets, in binary coded decimal (BCD).
cbLocalNetAdddress	The number of digits in the local network address.
cTicks	The number of elapsed 100 ms timer ticks since the last link-level protocol restart.
cRxActiveTicks	The number of elapsed timer ticks since the last link-level protocol restart in which the receive side of the line was active
cTxActiveTicks	The number of elapsed time ticks since the last link-level protocol restart in which the transmit side of the line was active.
rgLcStatus	An array of nLc logical channel status blocks. One block is written for each virtual circuit that is monitored. For more information, see Table 4-5.

For each logical channel status requested, the logical channel status block shown in Table 4-5 is appended to the buffer.

Table 4-5 **Logical Channel Status Blocks**

Offset	Field	Size (bytes)
0	lcn	1
1	lcGn	1
2	fActive	1
3	fOutofOrder	1
4	dataPacketsReceived	2
6	dataPacketsTransmitted	2
8	nDteReset	2
10	nDceReset	2
12	nUser	2
14	remoteNetAddress	8
22	cbRemoteNetAddress	1
23	receiveWindowSize	1
24	sendWindowSize	1
25	receivePacketSize	2
27	sendPacketSize	2
29	lastCause	1
30	lastdiagnostic	1
31	callState	1
32	circuitType	1
33	dummy	1

The following describes the logical channel status blocks.

lcn	The logical channel number.
lcGn	The logical channel group number.
fActive	*True* if the virtual circuit is active.
fOutofOrder	*True* if the virtual circuit is out of order.
dataPacketsReceived	The number of DATA packets received since this virtual circuit was established.
dataPacketsTransmitted	The number of DATA packets transmitted since this virtual circuit was established.
nDteReset	The number of flow control resets initiated by the X.25 Network Gateway system service over this virtual circuit since it was established.
nDceReset	The number of flow control resets initiated by the PDN over this virtual circuit since it was established.
nUser	The number of the user who last established a connection over this virtual circuit.
remoteNetAddress	The called (remote) network address.

cbRemoteNetAddress	The number of digits in the called (remote) network address.
receiveWindowSize	The maximum number of DATA packets the gateway can receive over this virtual circuit without acknowledgment by the X.25 Network Gateway system service. This is a sliding window; more than one unacknowledged packet can be outstanding. If the maximum number is reached, no more packets can be received by the gateway until some (or all) of the outstanding packets are acknowledged.
sendWindowSize	The maximum number of DATA packets the X.25 Network Gateway system service will send to the PDN over this virtual circuit before waiting for acknowledgment. This is a sliding window; more than one unacknowledged packet can be outstanding. If the maximum number is reached, no more packets can be sent until some (or all) of the outstanding packets are acknowledged.
receivePacketSize	The size of the largest DATA packet the gateway can accept over this virtual circuit.
sendPacketSize	The size of the largest DATA packet the PDN will accept from this virtual circuit.
lastCause	The reason why this virtual circuit was previously cleared or reset. (For further information on clear and reset operations, see Appendix C.)
lastDiagnostic	The diagnostic from the previous time this virtual circuit was cleared or reset. (For further information, see Appendix C.)
callState	The internal state of the virtual circuit, as defined in Annex 2 of CCITT Recommendation X.25. It can take one of the values shown in Table 4-6. (Refer to CCITT in Appendix E.)
circuitType	The type of virtual circuit; it can take one of the following values: 0 two-way virtual circuit. 1 permanent virtual circuit. 2 incoming-only virtual circuit. 3 outgoing-only virtual circuit.

Table 4-6 **CallState Values**

Value	Acronym	Meaning
0	R1	Ready
1	P2	DTE waiting
2	P3	DCE waiting
3	P6	DTE clear
4	P7	DCE clear
5	D1	Flow control ready
6	D2	DTE reset
7	D3	DCE reset
8	R2	DTE restart
9	R3	DCE restart

ReadX25PacketNet

The ReadX25PacketNet operation returns a DATA or INTERRUPT packet received by the X.25 Network Gateway system service from the PDN. The returned status code is 0 (OK) for a DATA packet, or 8519 (interrupt data) for an INTERRUPT packet. In either case, only the data portion of the full X.25 packet is returned.

Only two ReadX25PacketNet requests can be outstanding for a virtual circuit at any given time. If more are sent, the excess requests are returned with status code 8502.

No packets are transmitted by ReadX25PacketNet.

Procedural Interface

ReadX25PacketNet *(vch, pPacketRet, sPacketMax, psPacketRet, pGfiRet, timeOut, wFh): ErcType*

vch	The virtual circuit handle.
pPacketRet sPacketMax	Describe a buffer to which the user data portion of the next DATA or INTERRUPT packet received over the specified virtual circuit is to be returned.
psPacketRet	The memory address of a word to which the size of the received DATA or INTERRUPT packet is to be returned.

pGfiRet	The memory address of a byte into which the general format identifier (see Table 4-8) is to be returned.
timeOut	A word containing the maximum amount of time, in units of 100 ms, to wait; however, the system rounds the value to the next second. A value of 0 indicates no waiting; ReadX25PacketNet returns with the status code 8507 (user-specified time out) if no packets have been received from the PDN. A value of 0FFFFh indicates an indefinite wait.
wFh	The file handle returned by the gateway for B-NET routing use.

Request Block

Offset	Field	Size (bytes)	Contents
0	sCntInfo	2	8
2	nReqPbCb	1	0
3	nRespPbCb	1	3
4	userNum	2	
6	exchResp	2	
8	ercRet	2	
10	rqCode	2	16507
12	Fh	2	
14	inVch	2	
16	intimeOut	2	
18	reserved	2	
20	pPacketRet	4	
24	isPacketMax	2	
26	psPacketRet	4	
30	ssPacketRet	2	
32	pGfiRet	4	
36	sGfiRet	2	1

ResetX25CallNet

The ResetX25CallNet operation requests that a RESET REQUEST packet be transmitted in order to clear all outstanding read and write requests and to reinitialize the flow control procedures for the specified virtual circuit without clearing the call.

Only one ResetX25CallNet can be outstanding for a virtual circuit at any given time. If more are sent, the excess requests are returned with status code 8502.

Procedural Interface

ResetX25CallNet *(vch, bReason, wFh): ErcType*

vch	The virtual circuit handle.
bReason	A one-byte diagnostic code to be included in the RESET REQUEST packet.
wFh	The file handle returned by the gateway for B-NET routing use.

Request Block

Offset	Field	Size (bytes)	Contents
0	sCntInfo	2	5
2	nReqPbCb	1	0
3	nRespPbCb	1	0
4	userNum	2	
6	exchResp	2	
8	ercRet	2	
10	rqCode	2	16510
12	Fh	2	
14	invch	2	
16	bReason	1	

WriteX25InterruptNet

The WriteX25InterruptNet operation transmits an INTERRUPT packet over the PDN, bypassing the normal flow control procedures associated with DATA packet transmission. Unlike the WriteX25PacketNet operation (described later in this section), WriteX25InterruptNet transfers only a single byte of interrupt data; DATA packet formatting is performed by the X.25 Network Gateway system service.

Only one WriteX25InterruptNet can be outstanding for a virtual circuit at any time. If more are sent, the excess requests are returned with status code 8502.

Procedural Interface

WriteX25InterruptNet *(vch, bInterruptData, wFh): ErcType*

vch	The virtual circuit handle.
bInterruptData	The byte of interrupt data to be transmitted over the specified virtual circuit.
wFh	The file handle returned by the gateway for B-NET routing use.

Request Block

Offset	Field	Size (bytes)	Contents
0	sCntInfo	2	5
2	nReqPbCb	1	0
3	nRespPbCb	1	0
4	userNum	2	
6	exchResp	2	
8	ercRet	2	
10	rqCode	2	16509
12	Fh	2	
14	invch	2	
16	bIntData	1	

WriteX25PacketNet

The WriteX25PacketNet operation causes transmission of the DATA packet over the PDN. As with ReadX25PacketNet, the packet buffer should contain only user data. The three- or four-octet packet header is built by the X.25 Network Gateway system service and contains the qualifier, delivery confirmation, and the data bits specified in the general format identifier (see Table 4-8).

Only five WriteX25PacketNet requests can be outstanding for a virtual circuit at any given time. If more are sent, the excess requests are returned with status code 8502.

Procedural Interface

WriteX25PacketNet *(vch, bGfi, pPacket, sPacket, wFh): ErcType*

vch	The virtual circuit handle.
bGfi	The one-byte general format identifier in which M, D, and Q bits are specified. For further information on these bits, see "Specifying and Setting M, D, and Q Bits" later in this section.
pPacket sPacket	Describe a buffer containing the data to be transmitted over the specified virtual circuit. sPacket should be less than or equal to the Maximum Packet Size parameter used when installing X.25 Network Gateway.
wFh	The file handle returned by the gateway for B-NET routing use.

Request Block

Offset	**Field**	**Size (bytes)**	**Contents**
0	sCntInfo	2	8
2	nReqPbCb	1	1
3	nRespPbCb	1	0
4	userNum	2	
6	exchResp	2	
8	ercRet	2	
10	rqCode	2	16508
12	Fh	2	
14	invch	2	
16	bUserGfi	1	
17	reserved	3	
20	pPacket	4	
24	sPacket	2	

Specifying and Setting M, D, and Q Bits

M (More data indicator), *D* (Delivery confirmation), and *Q* (Qualified data) bit values can be specified when packets are transmitted or examined when packets are received on a per-packet basis with the following packet access method operations:

- AcceptX25CallNet
- InitiateX25CallNet
- NotifyNextIncomingCallNet
- ReadX25PacketNet
- WriteX25PacketNet

Not all bits are meaningful for all packets. Table 4-7 lists the valid bits.

Table 4-7 **Valid M, D, and Q Bits**

Packet	**Valid Bits**		
	M	D	Q
CALL ACCEPT		x	
INCOMING CALL		x	
DATA	x	x	x

Invalid bits are ignored, if specified, or set to 0, if returned. For DATA packets, all bit settings are considered valid by the X.25 Network Gateway.

For received CALL REQUEST packets, the entire packet is returned to the buffer specified in the NotifyNextIncomingCallNet operation. You can then examine the bit values.

For received CALL ACCEPT packets, the entire packet is returned to the buffer specified in the InitiateX25CallNet operation. You can then examine the bit values.

For all other packets, bit values are specified or examined with the general format identifier parameter. The format of the general format identifier is shown in Table 4-8. Refer to CCITT Recommendation X.25 for additional information concerning these bits.

For AcceptX25CallNet, InitiateX25CallNet, and WriteX25PacketNet operations, the general format identifier is a request parameter. Bit values are specified by the user in the bGfi field of the request.

For ReadX25PacketNet operations, the general format identifier is a response parameter. Bit values are returned to the memory location pointed to by the user-specified parameter pGfiRet.

Table 4-8 **General Format Identifier**

Bit	Meaning
0	More data indicator (M bit). Indicates start and end of multipacket messages (DATA packets only).
1-5	Reserved.
6	Delivery confirmation bit (D bit). Indicates local (immediate) or end-to-end (delayed) delivery confirmation (DATA, CALL REQUEST, and CALL ACCEPT packets).
7	Qualified data bit (Q bit). Indicates special X.29 packet content (DATA packets only).

X.25 Network Interaction

The X.25 Network Gateway supports, with a few exceptions, both the CCITT facilities (options) described in the 1980 CCITT Recommendation X.25 and the non-CCITT facilities of a particular PDN as long as the facilities of the PDN conform to the facilities formats defined in CCITT Recommendation X.25.

In general, it is your responsibility to:

- Coordinate the facilities used with the appropriate PDN administration.
- Ensure that the PDN to which the X.25 Network Gateway is connected supports the desired facilities.
- Ensure that the specification of a facility agrees with the parameters expected by the PDN.

The X.25 Network Gateway checks outgoing per-call facilities to ensure that their length does not exceed 63 bytes (CCITT specification). If this length is exceeded, the data contained in the facilities fields are truncated.

The three general categories of CCITT facilities are:

- Those transparent to the X.25 Network Gateway
- Those nontransparent to the X.25 Network Gateway (that is, requiring special processing)
- Those not supported by the X.25 Network Gateway

Transparent Facilities

Most CCITT facilities are transparent to the X.25 Network Gateway. Except for those facilities listed in the following two subsections, the X.25 Network Gateway operates without regard for the facilities environment in which it is used, and without knowledge of the facilities data contained in the packet. If the X.25 Network Gateway system service is communicating with a PDN that supports non-CCITT facilities, these facilities are handled transparently if they conform to the CCITT facilities standards.

Nontransparent Facilities

Nontransparent facilities consist of installation facilities and per-call facilities.

Installation Facilities

The following facilities must be specified at the X.25 Network Gateway installation:

- Nonstandard default window size (other than 2)
- Nonstandard default packet size (other than 128)

Per-Call Facilities

Some per-call facilities affect the X.25 Network Gateway operation; therefore, it must recognize when these facilities are used. To do this, the X.25 Network Gateway examines the facility fields of received call-establishment packets (CALL ACCEPT and CALL REQUEST) for these per-call facilities. The X.25 Network Gateway modifies its operation appropriately if these per-call facilities are found.

The facility fields are examined using an algorithm based on CCITT Recommendation X.25. This algorithm is independent of all facilities other than the ones being searched for; the coding of the facility fields, however, must match CCITT Recommendation X.25. If these fields are coded in other formats, errors can occur.

Facilities data are transferred between the calling process and the X.25 Network Gateway system service using three operations: AcceptX25CallNet, InitiateX25CallNet, and NotifyNextIncomingCallNet.

The AcceptX25CallNet and InitiateX25CallNet operations enable data to be included in the facilities field of transmitted CALL ACCEPT and CALL REQUEST packets, respectively. The InitiateX25CallNet and NotifyNextIncomingCallNet operations return the CALL ACCEPT and CALL REQUEST packets, respectively, which contain facilities fields transmitted by the remote DTE.

Facilities data are scanned for three per-call facilities:

- Fast select
- Window size negotiation
- Packet size negotiation

Fast Select Facility The X.25 Network Gateway examines the facility fields of both incoming and outgoing CALL REQUEST packets for the fast select facility. A fast select facility is a Type A facility with a facility code of 1 and a parameter field with its high-order bits set. No special actions are taken for a restricted fast select facility beyond those taken for a nonrestricted fast select facility. If the fast select facility is used, the X.25 Network Gateway generates and accepts call establishment and call clearing packets according to the fast select packet formats during call establishment. Once call establishment is complete, the X.25 Network Gateway reverts to normal packet format. The gateway can send 128 bytes of data in a fast select during call establishment.

Window and Packet Size Negotiation Facilities The X.25 Network Gateway examines the facilities fields of both received and transmitted CALL ACCEPT packets for the window size and packet size negotiation facilities. If either of these facilities is used, the X.25 Network Gateway modifies its parameters for window and/or packet size as long as the negotiated maximum packet size is less than the maximum default packet size. If the negotiated maximum packet size is larger, then the gateway returns an error code to the application and clear the call.

The window size negotiation facility is a Type B facility with a facility code of 43h. The elements of the parameter field are assumed to contain appropriate values for window sizes; therefore, these values are not validated. The packet size negotiation facility is a Type B facility with a facility code of 42h. The elements of the parameter field are assumed to contain appropriate values for packet sizes (log base 2); therefore, these values are not validated.

For further information on these facilities, refer to Section 7 of CCITT Recommendation X.25 (see Appendix D).

Unsupported Facilities

- Datagram service
- Packet retransmission facilities
- Throughput negotiation

Multiplexing

When a packet is transmitted to the link level, the number of the virtual circuit on which it is transmitted is saved. The next time the link level is ready to accept a packet for transmission, virtual circuit 0 is examined for a restart-related packet requiring transmission. If no restart-related packet is pending, all other active virtual circuits are examined in circuit number order, starting at the virtual circuit following the virtual circuit that transmitted last. The first virtual circuit found with a pending requirement to transmit a packet is allowed to transmit. Thus, no particular virtual circuit can dominate transmission.

Flow Control

When a DATA packet is received on a virtual circuit, a flow-control packet (RR or RNR; see CCITT Recommendation X.25 and D), which acknowledges that data packet, is made pending for transmission. If additional DATA packets are received before the flow-control packet can be issued, a single flow-control packet acknowledging multiple DATA packets is made pending transmission. If the type of flow-control packet to be sent changes (from RR to RNR, or vice versa) once it has been made pending, the type is altered accordingly. Thus, a flow-control packet is sent to acknowledge a DATA packet as soon as transmission is possible. With this scheme, a minimum number of flow-control packets is transmitted.

Error Handling

Virtual circuit cause and diagnostic codes, as indicated in received RESTART REQUEST, CLEAR REQUEST, RESET REQUEST packets, are stored for each virtual circuit as they are received. The last virtual circuit cause and diagnostic codes received are accessible to each virtual circuit with the QueryX25StatusNet operation.

A process can specify the value of the diagnostic code field when it requests that CLEAR and RESET REQUEST packets be generated. Refer to Appendix B for the values inserted in the diagnostic code field for CLEAR and RESET REQUEST packets generated by the X.25 Network Gateway in response to errors.

Error conditions for user requests to the X.25 Network Gateway packet occur in three circumstances:

- The request contains invalid parameters or is invalid in the context of prior requests.
- The request is superseded by a later request from the same application system.
- The state of a virtual circuit or of the X.25 Network Gateway service as a whole makes it impossible to fulfill the request.

The gateway checks requests to see if they are valid and appropriate. If an error condition is detected, the request is immediately returned with an appropriate status code. If a valid and proper request is received, the X.25 Network Gateway system service attempts to fulfill the request.

In most cases, a request cannot be fulfilled immediately and must be held by the X.25 Network Gateway system service momentarily. If a request awaiting fulfillment is canceled by a later action from the application system or the X.25 Network Gateway system service, the request is canceled and an appropriate status code is returned.

Generally, you cannot assume that requests will be returned in the same order that they were sent either after an error occurs or under normal conditions. However, requests for reads and writes are returned in the same order that they were issued if errors occur while they are being held by the gateway.

Sequential Access Method

The X.25 Sequential Access Method (X.25 SAM) provides device-independent input and output through the BTOS Sequential Access Method (SAM). Because this level of PDN access requires no detailed knowledge of CCITT Recommendation X.25, it is appropriate for application programmers. X.25 SAM supports all BTOS user languages, including Pascal, COBOL, FORTRAN, and BASIC.

This section describes how to use the X.25 SAM, and includes the following topics:

- Device file specification
- Configuration files
- Device-independent operations
- Network interaction

How SAM Accesses a PDN

X.25 SAM accesses the X.25 Network Gateway through device-dependent and device-independent operations. Device-dependent operations are specified by parameters defined in the X.25 configuration file. Device-independent operations occur through a programmatic interface, which uses byte streams.

A byte stream is a readable (input) or writeable (output) sequence of 8-bit bytes. Each X.25 byte stream corresponds to a virtual circuit that is established when the byte stream is opened and cleared when the byte stream is closed. An X.25 byte stream is established by a set of byte stream procedures. (Procedures and their definitions are described later in this section.) For a comprehensive discussion of BTOS byte streams, refer to the *BTOS Reference Manual.*

X.25 SAM procedures access the appropriate packet access method procedures. PAM procedures access the PDN through the gateway. Thus, X.25 SAM provides you with the tools to send data to other systems over an X.25 PDN without requiring you to learn the sophisticated formatting techniques necessary to use the packet access method.

Note: X.25 SAM operates on switched virtual circuits only; operation over permanent virtual circuits is not supported.

The configuration file defines the call establishment parameters for the virtual circuit (byte stream). You can create or modify configuration files with the Configure X.25 command.

X.25 SAM performs packet assembly and disassembly (PAD) in a manner compatible with CCITT Recommendations X.3 and X.29 (see Section 2).

Linking a Sequential Access Method Program

After you write a program using the sequential access method, the object modules you create must be linked by the BTOS Linker to produce a run file. This is done using the Link command. On the *Object Modules* line you must specify all the object modules that make up your program as well as the module [Sys]<Sys>X25Sam.lib(samgenx25). On the *[libraries]* line you must specify [Sys]<Sys>X25Sam.lib. Without this, any attempt to open an X.25 bytestream returns error code 7. The library X25SAM.LIB contains the X.25 SAM object module procedures.

Note: X25Sam.Lib was modified for release 6.0. In order to use byte streams over B-NET, you must relink your X.25 SAM applications with the new version of X25Sam.Lib.Programs linked with this new version require a new X.25 SAM configuration file that conforms to the updated format. These programs cannot operate with an old format configuration file (relative to release 5.1 or less).

Device File Specification

The device file specification contains the information that defines an X.25 virtual circuit. The device file specification supplied to the OpenByteStream operation that opens an X.25 byte stream is:

[X25]n&{node}[volname]<dirname>filename

where *n* is a network specifier. Either leave this field blank or specify 0.

{node}[volname]<dirname>filename describes an optional configuration file containing the operational characteristics. A default configuration file is used if none is specified. The default file name is:

[Sys]<Sys>X25Config.Sys

Configuration Files

A configuration file is generated with the Configure X.25 command. Configure X.25 takes the file specification for a configuration file as a parameter, then displays a form that prompts for configuration file parameters.

Command **Configure X.25** RETURN
Configure X.25
Configuration file name

X.25 Configuration File Parameters

The following form is displayed after you enter the configuration file name in the Configure X.25 command. To view the default values, press GO (you exit the utility), reenter the configuration file, and this screen is redisplayed along with the default values.

X.25 Parameters
[Node Name]
[Disable transmit buffering?]
[Initiate or Accept (default = accept)]
[Local or End-to-end acknowledgment (default = local)]
[Read timeout (default = forever)]
[Called address]
[Reverse charging?]
[Call data]
[Low port (default = 0)]
[High port (default = 99)]
[Notify timeout (default = forever)]
[Max Packet Size (default = 128)]

[Node Name]
The B-NET node name where the gateway resides. It can be specified with up to 12 characters (including curly brackets if used).

[Disable transmit buffering?]
Either *Yes* or *No* (*No* is the default). Outgoing data is usually stored in a write buffer until the transmit packet size for the X.25 Network Gateway is reached, at which time the actual transfer occurs. Entering *Yes* disables internal data buffering from WriteByte and WriteBsRecord operations. Data is then transferred for each X.25 SAM write operation.

[Initiate or Accept (default = accept)]
Specifies if the call is to be initiated by the X.25 Network Gateway or by the remote user.

For initiate, an OpenByteStream operation causes an InitiateX25CallNet operation to be issued with the following attributes specified in the configuration file:

- Called address
- Call data
- Reverse charging
- Node name

Call data and reverse charging attributes are optional.

For accept, an OpenByteStream operation causes a NotifyNextIncomingCallNet operation with the following values specified in the configuration file:

- Low port
- High port
- Time out
- Node name

If an incoming call received within the timeout interval matches the specified port values, an AcceptX25CallNet operation is issued to complete the establishment of the call.

[Local or end-to-end acknowledgment (default = local)]
Entering *end-to-end* specifies the delivery confirmation (D-bit) feature. This feature allows you to specify the value of the D bit in X.25 DATA packets transmitted by an X.25 byte stream. The D bit is set if end-to-end acknowledgment is specified and reset if local acknowledgment is specified.

[Read timeout (default = forever)]
The timeout interval, in seconds, for read operations. A value of 0 indicates data is to be returned if buffered data is available at the byte stream or at the X.25 Network Gateway. Otherwise, the read operation should immediately time-out. Any value equal to or greater than 60,000 indicates an indefinite wait.

[Called address]
The network address to be called. Used only if initiate is specified in the [Initiate or Accept] field.

[Reverse charging?]
No is the default. Entering *Yes* specifies that a collect call is made when a call is initiated (initiate is specified in the [Initiate or Accept] field).

[Call data]
Can contain up to 16 characters of ASCII data to be transmitted to the remote user during call establishment and is used only if initiate is specified in the [Initiate or Accept] field.

[Low port (default = 0)]
Indicates the lower bound of the port range for calls to be accepted. Used only if accept is specified in the [Initiate or Accept] field. Low port value must be equal to or less than the high port value.

[High port (default = 99)]
Indicates the upper bound of the port range for calls to be accepted. Used only if accept is specified in the [Initiate or Accept] field. High port value must be equal to or greater than the low port value.

[Notify timeout (default = forever)]
Contains the timeout interval, in seconds, for the OpenByteStream operation. A value of 0 indicates data is to be returned if buffered data is available at the byte stream or at the X.25 Network Gateway. Otherwise, the operation should immediately time–out. Any value equal to or greater than 60,000 indicates an indefinite wait. The timeout is used only if accept is specified in the [Initiate or Accept] field.

[Max Packet Size (default = 128)]
The maximum packet size, in bytes, that the gateway can send or receive. Possible values are 16, 32, 64, 128 (the default), 256, 512, and 1024.

Device-Independent Operations

The following device-independent byte stream sequential access method operations are supported:

CheckPointBs
CloseByteStream
OpenByteStream
PutBackByte
ReadBsRecord
ReadByte
ReadBytes
ReleaseByteStream
WriteBsRecord
WriteByte

X.25 SAM can be opened for read, write, or modify operations. For compatibility, the text and append modes are allowed in the OpenByteStream operation. The text mode is treated as read mode, and the append mode is treated as write mode.

Multiple X.25 byte streams can be opened concurrently by an individual user. Each open byte stream transfers data over a single X.25 virtual circuit.

The X.25 protocol allows data to be transferred during the establishment of a virtual circuit. If initiate is specified in the configuration file, data can be transferred during call establishment. If accept is specified in the configuration file, any data received during call establishment is placed in the read buffer. This data is accessible with standard read operations once the byte stream is opened.

Sequential Access Method Operations

Sequential access method operations are categorized by function in Table 5-1. These operations are handled automatically by the X.25 SAM. (Note that X.25 SAM never issues ConnectX25PermanentNet, ResetX25CallNet, or WriteX25InterruptNet operations.)

Table 5-1 **Sequential Access Method Operations by Function**

Call Establishment and Clearing Operations *

AcceptX25CallNet
ClearX25CallNet
InitiateX25CallNet
NotifyNextIncomingCallNet

Data Transfer Operations **

ReadX25PacketNet
WriteX25PacketNet

* *For a summary description of these operations, see Table 4-1.*

** *For a summary description of these operations, see Table 4-2.*

For further information on these operations under the packet access method, see Section 4.

Note: Because X25Sam.Lib has been modified, X.25 SAM applications must be relinked with the new version of X25Sam.Lib to be fully compatible with the new features of the 6.0 release of the X.25 gateway (such as selectable packet size and B-NET).Programs linked with the new version of X25Sam.Lib require an X25 SAM configuration file that conforms to the updated format. These programs cannot operate with an old format configuration file.

CheckPointBs

The CheckPointBs procedure checkpoints the open byte stream identified by the memory address of the Byte Stream Work Area (BSWA) by waiting for all write operations to finish before returning. The byte stream remains open for subsequent output.

Procedural Interface

CheckPointBs (pBSWA): ErcType

pBSWA	The memory address of the same byte stream work area that was supplied to OpenByteStream.

CloseByteStream

The CloseByteStream procedure closes the open byte stream identified by the memory address of the Byte Stream Work Area (BSWA). If the byte stream was open for output, then CloseByteStream writes any partially full buffers and waits for all write operations to finish before returning. After calling CloseByteStream, the process can reuse the BSWA and the buffer area. If an error occurs during a CloseByteStream operation, then the byte stream is closed and a status code is returned.

Procedural Interface

CloseByteStream (pBSWA): ErcType

pBSWA	The memory address of the same byte stream work area that was supplied to OpenByteStream.

OpenByteStream

The OpenByteStream procedure attempts to open up a virtual circuit as a byte stream. The address of the Byte Stream Work Area (BSWA) returned from a OpenByteStream operation must be supplied to subsequent operations such as ReadBytes, WriteBsRecord, and ReleaseByteStream to identify a particular byte stream.

Procedural Interface

OpenByteStream (pBSWA, pbFileSpec, cbFileSpec, pbPassword, cbPassword, mode, pBufferArea, sBufferArea): ErcType

pBSWA	The memory address of a 130-byte memory work area for use by sequential access method procedures.
pbFileSpec cbFileSpec	Describe the X.25 byte stream device file specification. The device file specification is defined as: *[X25]n&*{node}[volume]*<dir>filename.* *n* is a communications channel/network specifier. This field must either be left blank or 0. *{node}[volume]<dir>filename* describes an X.25 SAM configuration file containing operational characteristics. The default configuration file—[Sys]<Sys>X25Config.Sys—is used if none is specified. This file may be created using the Configure X.25 utility.
pbPassword cbPassword	No passwords are required or supported by X.25 byte streams, so 0 should be specified for each of these fields. X.25 ignores any password specified in these fields.
mode	Read, text, write, append, or modify. Mode is indicated by 16-bit values representing the ASCII constants mr (mode read), mt (mode text), mw (mode write), ma (mode append), or mm (mode modify). In these ASCII constants, the first character (m) is the high-order byte and the second character (r, t, w, a, or m, respectively) is the low-order byte.

To maintain device independence in SAM, mode text and mode append are supported by X.25 SAM even though those modes are not really applicable to communications byte streams. X.25 SAM processes mode text in the same way that it processes mode read, and it processes mode append in the same way that it processes mode write.

To support CCITT Recommendation X.29, X.25 SAM transmits and receives X.29 data packets, regardless of the mode (read only or write only) for which the byte stream is opened. The open mode of an X.25 byte stream affects only user access to directions of data transfer. Data transfer occurs in two directions, regardless of the mode.

pBufferArea sBufferArea	Describe a memory area provided for the exclusive use of SAM procedures. To ensure device independence, this area must be at least 1024 bytes and word-aligned.

PutBackByte

The PutBackByte procedure returns one byte to the open input byte stream identified by the memory address of the Byte Stream Work Area (BSWA). Only one byte can be put back before reading again. An attempt to put back more than one byte returns status code 2305.

Procedural Interface

PutBackByte (pBSWA, b): ErcType

pBSWA	The memory address of the same byte stream work area that was supplied to OpenByteStream.
b	The 8-bit byte to be put back.

ReadBsRecord

The ReadBsRecord procedure reads the specified count of bytes from the open input byte stream identified by the memory address of the Byte Stream Work Area (BSWA) to the specified memory area. ReadBsRecord always reads the count of bytes specified except when a read timeout occurs. 8 If fewer than the specified count of bytes (or no bytes) remain in the buffer, status code 1 (End of file) is returned in conjunction with the actual count of bytes read.

Procedural Interface

ReadBsRecord (pBSWA, pBufferRet, sBufferMax, psDataRet): ErcType

pBSWA	The memory address of the same byte stream work area that was supplied to OpenByteStream.
pBufferRet	The memory address of the first byte of the buffer to which the data is to be read.
sBufferMax	The count of bytes to be read to memory.
psDataRet	The memory address of the word to which the count of bytes successfully read is returned.

ReadByte

The ReadByte procedure reads one byte from the open input byte stream identified by the memory address of the Byte Stream Work Area (BSWA) to the specified memory area. ReadByte returns status code 1 (End of file) when a read timeout occurs.

Procedural Interface

ReadByte (pBSWA, pbRet): ErcType

pBSWA	The memory address of the same byte stream work area that was supplied to OpenByteStream.
pbRet	The memory address of the byte to which the data is returned.

ReadBytes

The ReadBytes procedure reads up to the specified count of bytes from the open input byte stream identified by the memory address of the Byte Stream Work Area (BSWA). If fewer than the specified count of bytes or no bytes are available, ReadBytes returns a status code 1 and the actual count of bytes read.

ReadBytes returns the memory address of the data bytes in its buffer rather than moving the data to a specifed location. This optimizes performance but imposes the restriction that the calling process completely process the data before calling ReadBytes again. If this restriction is inconvenient, the ReadBsRecord opeation should be used instead. If fewer than the specified count of bytes (or no bytes) remain in the buffer, status code 1 is returned in conjunction with the actual count of bytes read.

Procedural Interface

ReadBytes (pBSWA, cbMax, ppbRet, pcbRet): ErcType

pBSWA	The memory address of the same byte stream work area that was supplied to OpenByteStream.
cbMax	The maximum count of bytes of data that the calling process will accept.
ppbRet	The memory address of four bytes to which the memory address of the data is returned.
pcbRet	The memory address of a word to which the actual count of data bytes made available is returned.

ReleaseByteStream

The ReleaseByteStream procedure abnormally closes the open output byte stream identified by the memory address of the Byte Stream Work Area (BSWA). ReleaseByteStream, unlike CloseByteStream, does not write any partially full buffers. ReleaseByteStream should be used only when a WriteBsRecord, WriteByte, or CheckpointBs operation fails due to an unrecoverable error.

When a ReleaseByteStream call is made, X.25 SAM asks the X.25 gateway to either clear the call or send an X.29 invitation to clear, depending on the call direction.

Procedural Interface

ReleaseByteStream (pBSWA): ErcType

pBSWA	The memory address of the same byte stream work area that was supplied to OpenByteStream.

WriteBsRecord

The WriteBsRecord procedure writes the specified count of bytes to the open output byte stream identified by the memory address of the Byte Stream Work Area (BSWA) from the specified memory area. Because output is buffered, there is no guarantee of the time at which output is actually written. Only the CheckpointBs and CloseByteStream operations ensure that data was actually written.

Procedural Interface

WriteBsRecord (pBSWA, pbRecord, cbRecord, pcbRet): ErcType

pBSWA	The memory address of the same byte stream work area that was supplied to OpenByteStream.
pbRecord	The memory address of the data to be written.
cbRecord	The count of bytes to write.
pcbRet	The memory address of the word to which the count of bytes successfully written is returned.

WriteByte

The WriteByte procedure writes one byte to the open output byte stream identified by the memory address of the Byte Stream Work Area (BSWA). Because output is buffered, there is no guarantee of the time at which output is actually written. Only the CheckpointBs and CloseByteStream operations ensure that data was actually written.

Procedural Interface

WriteByte (pBSWA, b): ErcType

pBSWA	The memory address of the same byte stream work area that was supplied to OpenByteStream.
b	The 8-bit byte to write.

Network Interaction Under Sequential Access Method

The following describes how X.25 SAM supports the protocols in CCITT Recommendations X.3, X.25, and X.29.

Packet Assembly/Disassembly Facility (PAD)

The X.25 byte stream transfers and receives data from the X.25 Network Gateway in packets of variable size. Packets are buffered within the byte stream and are assembled and disassembled as user input and output is performed by the X.25 SAM read and write operations. Separate read and write buffers are provided.

Outgoing data is usually stored in the write buffer until the transmit packet size for the X.25 Network Gateway is reached; at which time, the actual transfer is performed. The configuration file can be used to disable this write buffering feature; data is then transferred for each X.25 SAM write operation.

Each packet of incoming data is stored in the read buffer until all data in the packet is read; at that time, a new incoming packet is solicited from the X.25 Network Gateway. The write buffer is flushed before every read request to the X.25 Network Gateway. This avoids a deadlock condition in which you have to wait for a response to outgoing data being held in the buffer.

X.29 packets are treated in a manner similar to user data packets, except that X.29 packets are never handed over to the user. Data packets and X.29 packets are accumulated in the read buffer when received from the network. When a read request is encountered, all X.29 packets accumulated are immediately responded to, and all user data packets are handed over to the user.

X.29 Procedures

CCITT Recommendation X.29 defines procedures for exchanging information between a PAD and a packet-mode Data Terminal Equipment (DTE) over a public data network. X.25 SAM incorporates elements of these procedures into its operation to support interaction among users of BTOS workstations, and between users of BTOS workstations and users of other equipment.

Byte streams opened by X.25 SAM are compatible with CCITT Recommendation X.29 procedures. The provided X.29 support depends on the direction of call establishment. Asymmetric operation is required by the nature of CCITT Recommendation X.29, which only allows calls to be established in one direction; a PAD can establish calls to a packet-mode DTE, but a packet-mode DTE cannot establish calls to a PAD. However, once the call is established, data can be transferred in both directions.

For calls accepted by X.25 SAM, the workstation functions as a packet-mode DTE communicating with a PAD according to CCITT Recommendation X.29. This allows data to be exchanged with an asynchronous DTE over a PDN.

For calls initiated by X.25 SAM, the workstation functions as a PAD interfacing an asynchronous DTE to a PDN. This allows the use of commercial time-sharing facilities that communicate with asynchronous DTEs over public data networks using PADs. In this case, your workstation is still a packet-mode DTE using the X.25 Network Gateway to exchange data according to the X.25 packet switching protocol. Actual public data network PADs are not required. Rather, the packet-mode capability is enhanced by allowing your workstation to function in place of both an asynchronous DTE and a PAD while maintaining the benefits of a packet-mode DTE.

Supported Procedures

X.25 SAM is compatible with most CCITT Recommendation X.29 procedures for communicating control and user information between a PAD and an X.25 packet-mode device; this operation is user-transparent. X.29 support specifications are published in CCITT Recommendation X.29 (see Appendix E).

X.25 SAM support of CCITT Recommendation X.29 procedures allows workstation-to-workstation communication over a PDN through X.25 SAM. Support of X.25 SAM initiating a call is compatible with that of X.25 SAM accepting a call.

X.25 SAM can be used to communicate with pure packet-mode DTEs if the DTE initiates and/or accepts calls with the CCITT Recommendation X.29 protocol identifier in the Call User Data field of the CALL REQUEST packet. Only CCITT Recommendation X.29 calls are either initiated or accepted by X.25 SAM. Because X.29 support of X.25 SAM is essentially passive, the only action required during the call is that the remote DTE must use the X.29 protocol identifier during call establishment. Generally, the protocol identifier will be a 01 hex placed in the first byte of the user data field.

Data Transfer

X.29 information is transmitted within X.25 DATA packets. This method of data transfer provides special error recovery as well as call termination and parameter setting procedures for PAD operation. X.29 messages are automatically intercepted and generated by X.25 SAM independently of the operations that package user data.

In supporting X.29, X.25 SAM transmits and receives DATA packets regardless of the mode (read only or write only) for which the byte stream is opened. Thus, the open mode of an X.25 byte stream affects only user access to directions of data transfer. Data transfer occurs in two directions, regardless of the mode.

Packet-Mode DTE

When X.25 SAM accepts a call (the workstation functions as a packet-mode DTE), X.25 SAM supports standard CCITT Recommendation X.29 control procedures for PAD functions. This support can be termed passive because X.25 SAM does not set PAD parameters to force any particular mode of operation upon the PAD or upon the asynchronous terminal. X.25 SAM recognizes and issues PAD messages when necessary to ensure correct data transfer with the PAD, but in no other case.

Actions taken on receipt of PAD messages (Q bit set) are shown in Table 5-2.

Table 5-2 **Actions Taken by X.25 SAM (as a Packet-Mode DTE) on Receipt of X.29 Messages**

Message	Action
Illegal message	Send X.29 error message
Error message	Clear the call
Indication of break	Send a set-parameters message to reset parameter 8 for normal delivery of data to asynchronous DTE
Any other message	Ignored

An invitation-to-clear message is transmitted if the ReleaseByteStream or CloseByteStream operation is called. This message asks the remote user to clear the call once the remote user has received all transmitted data. The byte stream itself does not clear the call. However, termination of your program results in the gateway clearing the call. You may lose some data if your program terminates before the PAD has responded to the invitation-to-clear message by clearing the call.

Asynchronous DTE/PAD

When X.25 SAM initiates a call (the workstation functions as an asynchronous DTE interfaced to a PAD), X.25 SAM supports standard CCITT Recommendation X.29 control procedures for PAD functions, but it does not act on the PAD parameters set by the packet-mode DTE. That is, while the packet–mode DTE can set and read back values of Recommendation X.3 PAD parameters, the X.25 SAM operation is not modified by the values of these parameters.

Actions taken on receipt of PAD messages (Q bit set) are shown in Table 5-3.

Parameter values are maintained for the 12 X.3 parameters (see CCITT Recommendation X.3). Any other parameters included in a parameter field are considered an error, and an X.29 error message is issued. The call is cleared if the ReleaseByteStream or CloseByteStream operation is called.

Table 5-3 **Actions Taken by X.25 SAM (as Asynchronous DTE/PAD) on Receipt of X.29 Messages**

Message	Action
Illegal message	Send X.29 error message
Error message	Clear the call
Indication of break	Ignored
Invitation to clear	Clear the call
Read parameters	If a parameter field is present, send parameters-indication with values of specific parameters. If no parameter field is present, send the values of all parameters.
Set and read	If a parameter field is parameters-present, store new values of specific parameters and send parameters-indication with their values. If there is no parameter field present, reset all parameters to 0 and send parameters-indication with values of all the parameters.

Limitations of SAM X.25 Communications

The following are not supported by X.25 SAM:

- Transmission of indication-of-break message
- Local specification or interpretation of X.29 parameters
- Asynchronous requests to X.25
- M bit
- INTERRUPT packets
- Facilities other than reverse charging
- Permanent virtual circuits

Error Handling

X.29 error handling is transparent to the user.

The X.29 Network Gateway performs the following error handling procedures:

- If X.25 SAM receives a message violating CCITT Recommendation X.29 procedures, it sends an X.29 error message, but does not clear the call. IF X.25 SAM sends a message violating CCITT Recommendation X.29 procedures, X.25 SAM will receive an X.29 error message; X.25 SAM will then clear the call.
- If an X.25 error occurs during a SAM operation, the call is cleared, but the byte stream is not closed. A ReleaseByteStream or CloseByteStream operation must be done to close the byte stream.
- If a user-specified time-out occurs, the operation in question is terminated, but the call is not cleared.

Multimode Terminal Program X.25 Communications

The Multimode Terminal Program (MTP) enables a workstation user to communicate with a host computer over a public data network using X.25. For more information on MTP functions and capabilities, refer to the *BTOS Multimode Terminal Program (MTP) Programming Reference Manual* and the *BTOS Multimode Terminal Program (MTP) Operations Guide.*

This section describes the interaction between MTP X.25 communications and the PDN; it is written for the programmer who is knowledgeable about communications.

Note: MTP procedure calls have not migrated to the new procedures. Thus, MTP does not support use over B-NET or permanent virtual circuits (PVC).

Operations

Multimode Terminal Program operations are categorized by function in Table 6-1. These operations are handled automatically by the MTP X.25. Detailed descriptions of each operation (in alphabetical order) follow Table 6-1. (Note that MTP X.25 never issues the ConnectX25Permanent or the ResetX25Call operation.)

Table 6-1 **Multimode Terminal Program Operations by Function**

Call Establishment and Clearing Operations*

AcceptX25Call
ClearX25Call
InitiateX25Call
NotifyNextIncomingCall

Data Transfer Operations**

ReadX25Packet
WriteX25Interrupt
WriteX25Packet

**For a summary description of these operations, see Table 4-1. For MTP, ignore the parameters that refer to BNET operation.*

***For a summary description of these operations, see Table 4-2.*

See Section 4 under the Packet Access Method for further information on these operations.

AcceptX25Call

To issue AcceptX25Call, press CODE-F2, then GO. The NotifyNextIncomingCall operation completes without an error.

Procedural Interface

AcceptX25Call *(vch, pFacilities, sFacilities, pUserData, sUserData, bGfi):* ErcType

vch	An internal variable
pFacilities sFacilities	0
pUserData sUserData	0
bGfi	0 (all bits are off)

ClearX25Call

ClearX25Call is issued when you press FINISH to clear the call or when an X.29 error occurs. The call must have already been established before you try to initiate or accept another call.

Procedural Interface

ClearX25Call *(vch, bReason, pUserData, sUserData):* ErcType

vch	An internal variable
bReason	0
pUserData sUserData	0

InitiateX25Call

InitiateX25Call is issued when the Initiate Call command (CODE–F1, then GO) is executed.

Procedural Interface

InitiateX25Call *(iNet, pVchRet, pPacketRet, sPacketMax, psPacketRet, pCalled, sCalled, pFacilities, sFacilities, pUserData, sUserData, bGfi):* ErcType

iNet	0
pVchRet	An internal variable
pPacketRet sPacketMax	An internal buffer
psPacketRet	An internal variable
pCalled sCalled	Forward the user-specified value from the MTP initiate call form
pFacilities sFacilities	0
pUserData sUserData	The X.29 protocol identifier plus the user-specified value from the MTP initiate call form
bGfi	0 (all bits are off)

NotifyNextIncomingCall

NotifyNextIncomingCall is issued when the Accept Call command (CODE-F2, then GO) is executed.

Procedural Interface

NotifyNextIncomingCall *(iNet, pVchRet, pPacketRet, sPacketMax, psPacketRet, pRgClass, sRgClass, lowPort, highPort, timeOut):* ErcType

iNet	0
pVchRet	An internal variable
pPacketRet sPacketMax	An internal buffer
psPacketRet	An internal variable
pRgClass sRgClass	An X.29 protocol mask-value pair
lowPort highPort	User-specified values from the MTP accept call form
timeOut	0FFFFh (indefinite wait)

ReadX25Packet

Reads occur synchronously, and one read always outstanding unless the MTP input buffer is full.

Procedural Interface

ReadX25Packet *(vch, pPacketRet, sPacketMax, psPacketRet, pGfiRet, timeOut):* ErcType

vch	An internal variable
pPacketRet sPacketMax	An internal buffer
psPacketRet	An internal variable
pGfiRet	An internal variable
timeOut	0FFFFh (indefinite wait)

WriteX25Interrupt

WriteX25Interrupt is issued when you press BREAK (F3) and the break options specify an INTERRUPT packet is to be sent.

Procedural Interface

WriteX25Interrupt *(vch, bInterruptData):* ErcType

vch	An internal variable
bInterruptData	0

WriteX25Packet

WriteX25Packet is issued when characters are transmitted to the host computer and when an X.29 message is sent.

Procedural Interface

WriteX25Packet *(vch, bGfi, pPacket, sPacket):* ErcType

vch	An internal variable
bGfi	D bit is 0; M bit is 0; Q bit is 1 for X.29 messages, 0 for user data
pPacket sPacket	An internal buffer

Network Interaction

The rest of the section describes how MTP X.25 supports the protocols in CCITT Recommendations X.3, X.28, and X.29.

Packet Assembly/Disassembly Facility (PAD)

MTP transfers and receives data from the X.25 Network Gateway in variable-size packets. Packets are buffered within MTP and are assembled and disassembled as user output and input occurs. Separate write and read buffers are provided.

Data transfer is handled by an individual process that transmits DATA packets to the X.25 Network Gateway. As data is received for transmission, the process constructs a packet to send to the X.25 Network Gateway. When the request is completed, another packet is constructed for transmission to the X.25 Network Gateway. If the X.25 Network Gateway falls behind, the transmission process sends a second DATA packet. Two DATA packets are maintained until either transmission is complete or the X.25 Network Gateway catches up.

Each incoming DATA packet is stored in the read buffer until all data in the packet are read by MTP, at which time a new incoming DATA packet is solicited from the X.25 Network Gateway.

X.29 Procedures

CCITT Recommendation X.29 defines procedures for exchanging information between a PAD and a packet-mode DTE over a public data network. MTP incorporates elements of these procedures into its operation to support interaction between and among users of Unisys equipment and users of other equipment.

MTP supports CCITT Recommendation X.29 procedures. The provided X.29 support depends on the direction of call establishment. This asymmetric operation is required by the nature of CCITT Recommendation X.29, which allows calls to be established only in one direction. That is, a PAD can establish calls to a packet mode DTE, but a packet mode DTE cannot establish calls to a PAD.

For calls accepted by MTP, the workstation functions as a packet-mode DTE communicating with a PAD, according to CCITT Recommendation X.29. This allows data to be exchanged with an asynchronous DTE over a public data network.

For calls initiated by MTP, the workstation functions as a PAD interfacing an asynchronous DTE to a public data network. This allows the use of commercial time-sharing facilities that communicate with asynchronous DTEs over public data networks with PADs. In this case, the workstation still functions as a packet mode DTE using the X.25 Network Gateway to exchange data according to CCITT Recommendation X.25 packet-switching protocol. Actual public data network PADs are not required. Rather, the packet-mode capability is enhanced by allowing the user's workstation to function in place of both an asynchronous DTE and a PAD while maintaining the benefits of a packet mode DTE.

Supported Procedures

The Multimode Terminal Program supports most CCITT Recommendation X.29 procedures for communicating control and user information between a PAD and a packet mode DTE; this is transparent to the user. X.29 support specifications are published in CCITT Recommendation X.29 (see Appendix D).

MTP support of CCITT Recommendation X.29 procedures allows workstation-to-workstation communication over a PDN through MTP. Support of the MTP initiating a call is compatible with that of the MTP accepting a call.

MTP can communicate with pure packet-mode DTEs if the DTE initiates and/or accepts calls with the X.29 protocol identifier in the Call User Data field of the CALL REQUEST packet. Only CCITT Recommendation X.29 calls are initiated or accepted by MTP. Because support of X.29 communications is essentially passive, the only X.29 action required during the call is the use of the X.29 protocol identifier during call establishment.

MTP is affected by two X.3 parameters: echo and break. These parameters affect both the echo (full-duplex versus half-duplex) in conversational transmission and the action taken when BREAK (F3) is pressed.

Data Transfer

X.29 information is transmitted within X.25 DATA packets. Special error recovery as well as call termination and parameter-setting procedures for PAD operation are supported. X.29 messages are automatically intercepted and generated by MTP independently of the operations that package user data.

Packet-Mode DTE

When MTP accepts a call (the workstation functions as a packet-mode DTE), it supports standard CCITT Recommendation X.29 control procedures for PAD functions. This support is passive because MTP does not set PAD parameters to force any particular mode of operation upon the PAD or upon the asynchronous DTE. MTP recognizes and issues most PAD messages only when necessary to ensure correct data transfer with the PAD.

Actions taken on receipt of PAD messages (Q bit set) are shown in Table 6-2.

Table 6-2 **Actions Taken by the MTP (as a Packet-Mode DTE) on Receipt of X.29 Messages**

Message	Action
Illegal message	Send X.29 error message, then clear the call.
Error message	Clear the call.
Indication of break	Set-parameters message; reset parameter 8 for normal delivery of data to asynchronous DTE.
Any other message	Ignored.

An invitation-to-clear message is transmitted when the connection is terminated. This message requests that the remote user clear the call once all transmitted data has been received. MTP itself does not clear the call. However, once MTP is terminated, the gateway clears the call. Thus, data can be lost if MTP terminates before the PAD has responded to the invitation-to-clear message by clearing the call.

Asynchronous DTE/PAD

When MTP initiates a call (the workstation functions as an asynchronous DTE interfaced to a PAD), MTP supports standard CCITT Recommendation X.29 control procedures for PAD functions, but it acts only on the echo and break PAD parameters set by the packet-mode DTE. That is, while the packet-mode DTE can both set and read back values of CCITT Recommendation X.3 PAD parameters, MTP X.25 communications operation is modified only by the values of the echo and break parameters.

Actions taken on receipt of PAD messages (Q bit set) are shown in Table 6-3.

Parameter values are maintained for X.3 parameter numbers 1 through 12 (refer to CCITT Recommendation X.3). Any other parameters included in a parameter field are flagged as an error and the call is cleared.

Table 6-3 **Actions Taken by the MTP (as an Asynchronous DTE/PAD) on Receipt of X.29 Messages**

Message	Action
Illegal message	Send X.29 error message, then clear the call.
Error message	Clear the call.
Indication of break	Ignored.
Invitation to clear	Clear the call.
Read parameters	If a parameter field is present, send parameters indication with values of specific parameters. If no parameter field is present, send the values of all parameters.
Set and read parameters	If a parameter field is present, store new values of specific parameters and send parameters indication with their values. If no parameter field is present, reset all parameters to 0 and send parameters indication with values of all parameters.

Limitations

The following are not supported by MTP X.25 communications:

- User specification or interpretation of X.29 parameters, except echo and break options
- D bit
- M bit
- Facilities
- Permanent virtual circuits
- Nonstandard default packet sizes
- Nonstandard default window sizes
- Communication over a B-NET transport

Error Handling

X.29 errors clear the call.

X.25 Status Program

This section provides an item-by-item description of the X.25 status program display. You can monitor X.25 gateway connections from a local, master, or remote (via B-NET) workstation.

The X.25 status program displays information about the state of the X.25 Network Gateway system service. The BTOS command is:

Command **X.25 Status** RETURN
[Node Name]

[Node Name] refers to the B-NET node the remote user wishes to access. Non-B-NET users may leave this parameter blank to monitor the status of a gateway on the local or master workstation.

The X.25 Status command displays a node name parameter for use over B-NET. This parameter accepts a B-NET node name. If the parameter entered is {*local*} but the X.25 Gateway is on the master system, the status monitor displays {*master*}.

When the X.25 status program is operational, it displays status information on the video display in one of three formats (see Figures 7-1, 7-2, and 7-3). The information is updated at one-second intervals.

Each status item on the video display is described in this section. The items are numbered, and these numbers are keyed to Figure 7-1. The numbered items are grouped by function in the following categories:

- General information, items 1-2
- Line status data, items 3-13
- Line use, items 14-21
- Version, item 22
- Circuit status data, items 23-30

Figure 7-2 **Connected Status of the X.25 Network Gateway**

R6.00 Feb 9,1986 5:10 PM

X.25 LINE MONITOR

CHANNEL B ADDRESS 311040800044 #CIRCUITS 17 CALLS 0
NODE NAME [LOCAL]
CONNECTED SINCE Feb 9, 1986 4:45 PM

DATA RX	120	CALLS IN	12	RESTARTS IN	1
DATA TX	216	CALLS OUT	32	RESTARTS OUT	0

RX UTIL:	LAST SEC. 12%	LAST 5 SECS. 23%	LAST 60 SECS. 24%
TX UTIL:	LAST SEC. 68%	LAST 5 SECS. 70%	LAST 60 SECS. 76%

CIRCUIT STATUS

			DATA		RESETS		
VC#	STATE	NET ADDRESS	RX	TX	IN	OUT	DIAG
0	P1		0	0	0	0	007250
1	P1	31106090042	3	8	0	0	000000
2	P1	31106090043	9	12	0	0	000000
3	P1	31105080011	3	21	0	0	000001
4	P1	31106090042	6	6	1	0	000001
5	P1	31106120098	8	3	0	1	000001
6	P1	31103050012	5	5	0	0	000001
7	P1	31107140016	81	82	0	0	000001
8	P1	31102130001	4	77	0	0	000003
9	P1	31102130005	1	2	0	0	000000

☐ Highlight

E7786

Status Items

Status item numbers are keyed to Figure 7-1.

Items 1 and 2 display general status information.

Item 1. Current date and time.

Item 2. *X.25 LINE MONITOR* means the X.25 status program is active.

Line Status Data

Items 3 through 21 indicate the status of the line as a whole. The significance of the data depends on the X.25 Network Gateway system service state (refer to item 7). If the state is operational, the other line status data reflects the current state of the line. If the state is connected or disconnected, the other line status data reflects the state of the line when it was last operational. The data, with the exception of the restarts (items 10 and 13), is reset each time the X.25 Network Gateway system service state goes from connected to operational.

Item 3. *CHANNEL* is the communications channel for the X.25 Network Gateway system service being monitored. This value is entered as a parameter when the X.25 Network Gateway is installed.

Item 3A. *NODE NAME* refers to the B-NET node the user wishes to access. Non-B-NET users should specify local or master to monitor the status of a gateway on the local or master workstation.

Item 4. *ADDRESS* is the network identification code for the X.25 Network Gateway system service being monitored. This code is assigned by the PDN and entered as a parameter when the X.25 Network Gateway is installed.

Item 5. *#CIRCUITS* is the number of virtual circuits allocated on the PDN that are being monitored. This value is entered as a parameter when the X.25 Network Gateway is installed.

Item 6. *CALLS* is the number of virtual circuits currently connected.

Item 7. *x...x SINCE y...y* is the state of the X.25 Network Gateway system service where *y...y* is the date and time when the current state was entered and *x...x* is one of the following (the field has a different video attribute, depending on the state):

- *OPERATIONAL* The X.25 Network Gateway is communicating with the PDN. This is shown in bright highlight.
- *CONNECTED* The connection to the PDN is established. This is shown in highlight.
- *DISCONNECTED* No connection to the PDN exists. This is shown in normal video.

Item 8. *DATA RX (received)* is the number of data packets received since the X.25 Network Gateway system service became operational.

Item 9. *CALLS IN* is the number of incoming calls established on nonpermanent virtual circuits since the X.25 Network Gateway system service became operational.

Item 10. *RESTARTS IN* is the number of times the X.25 Network Gateway system service was restarted by the PDN since the X.25 Network Gateway system service was installed.

Item 11. *DATA TX (transmitted)* is the number of data packets transmitted since the X.25 Network Gateway system service became operational.

Item 12. *CALLS OUT* is the number of calls established on nonpermanent virtual circuits by the user since the X.25 Network Gateway system service became operational.

Item 13. *RESTARTS OUT* is the number of times the X.25 Network Gateway system service has been restarted by the workstation since the X.25 Network Gateway system service was installed.

Line Use

Items 14 through 21 are the approximate percentage usage of the line for the transmit and receive directions for the previous 1, 5, and 60 seconds.

Items 14 through 17. *RX UTIL* is the usage on the receive direction (RX UTIL Item 14) during the previous second (LAST SEC. Item 15), the previous 5 seconds (LAST 5 SECS. Item 16), and the previous 60 seconds (LAST 60 SECS. Item 17).

Items 18 through 21. *TX UTIL* is the usage on the transmit direction (TX UTIL Item 18) during the previous second (LAST SEC. Item 19), the previous 5 seconds (LAST 5 SECS. Item 20), and the previous 60 seconds (LAST 60 SECS. Item 21).

RX Util and TX Util indicate the interarrival-time probability density of packets on the receive and transmit connections, respectively. The status monitor inquires X.25 status from the gateway, calculates Rx and Tx use by the elapsed time counters (cTicks, cRxActive, cTxactive) in the statistics buffer, and stores them in a temporary buffer on a second-to-second basis.

The formulas it uses are as follows:

$$RxUtil(i) = \frac{(cRxActive(i) - cRxActive(i-1))*100}{cTicks(i) - cTicks(i-1)}$$

$$TxUtil(i) = \frac{(cTxActive(i) - cTxActive(i-1))*100}{cTicks(i) - cTicks(i-1)}$$

where

i	= 1,2,...60 in increments of 1
cTicks	= number of elapsed 100 ms timer ticks since the last link level protocol restart (returned in current request)
cRxActive	= number of elapsed timer ticks since the last link level protocol restart in which the receive side of the line was active (returned in current request)
cTxActive	= number of elapsed timer ticks since the last link level protocol restart in which the transmit side of the line was active (returned in current request)

RxUtil(i) - Last 5 secs = (RxUtil(i)+...RxUtil(i-5))/5
RxUtil(i) - Last 60 secs = (RxUtil(i)+...RxUtil(i-60))/60
TxUtil(i) - Last 5 secs = (TxUtil(i)+...TxUtil(i-5))/5
TxUtil(i) - Last 60 secs = (TxUtil(i)+...TxUtil(i-60))/60

Version

Item 22. *Rn.n.n* is the version of the X.25 status program, where *n.n.n* is the version number.

Circuit Status Data

The status of ten circuits at a time can be displayed in the circuit status area. For more than ten circuits, the scroll feature is used. (See "Keyboard Input" following the item descriptions).

The status line for a circuit consists of eight entries (items 23 through 30). The significance of these entries depends on the state of the circuit. Circuits can be in one of three states, which are determined by the X.25 Network Gateway system service state and the current use of the virtual circuit. Each state has a particular video attribute.

Normal video indicates that a virtual circuit is out of order. If the X.25 Network Gateway system service is operational, a permanent virtual circuit is out of order in two cases:

1 Permanent virtual circuit was not initiated by the PDN.
2 A fatal protocol error occurred.

A nonpermanent circuit is out of order if a fatal protocol error occurred. If the X.25 Network Gateway system service state is not operational (disconnected or connected), then all virtual circuits are out of order.

Highlight indicates that a virtual circuit is idle. A virtual circuit is idle if it is operational but is not in use.

Bright highlight indicates a virtual circuit is in use. A virtual circuit is in use when a user is connected to it.

Circuit status for permanent virtual circuits reflects activity over the permanent virtual circuit since it was initialized by the PDN. Circuit status for permanent virtual circuits is reinitialized when the permanent virtual circuit's state changes from out of order to idle.

Circuit status for virtual circuits in use reflects activity on that virtual circuit since the current call was established. Circuit status for virtual circuits that are idle or out of order reflects activity on the virtual circuit during the last call established on that virtual circuit. Circuit status for nonpermanent virtual circuits is reinitialized when a call is established on the virtual circuit.

Item 23. *VC#* is the virtual circuit number; it is preceded by a *p* if the circuit is permanent, *i* if the circuit is incoming only, *o* for outgoing only, and blank for two-way. Virtual circuit 0 is reserved for protocol functions and is always in use when the X.25 Network Gateway system service is operational.

Item 24. *STATE* is the circuit state. The BTOS implementation of CCITT Recommendation X.25 defines ten internal circuit states (independent of the idle/in-use/out-of-order state) that correspond approximately to the circuit states defined in CCITT Recommendation X.25. A two-character abbreviation is displayed for the current state of the circuit (see Table 7-1).

Table 7-1 **Circuit State Abbreviations**

Acronym	Meaning
R1	Ready
R2	DTE restart
R3	DCE restart
P2	DTE waiting
P3	DCE waiting
P6	DTE clear
P7	DCE clear
D1	Flow control ready
D2	DTE reset
D3	DCE reset

Item 25. *NET ADDRESS* is the remote user address, that is, the PDN identification code of the user at the remote end of the call. (This field is blank for permanent virtual circuits.)

Item 26. *DATA RX* is the number of packets (circuit data) received since call establishment (or since circuit initialization for permanent virtual circuits).

Item 27. *DATA TX* is the number of packets (circuit data) transmitted since call establishment (or since circuit initialization for permanent virtual circuits).

Item 28. *RESETS IN* is the number of times the PDN has reset the protocol since call establishment (or since circuit initialization for permanent virtual circuits).

Item 29. *RESETS OUT* is the number of times the X.25 Network Gateway system service has reset the protocol since call establishment (or since circuit initialization for permanent virtual circuits).

Item 30. *DIAG (diagnostic)* includes the cause and diagnostic codes transmitted in the last clear/reset/restart packet over this virtual circuit. The cause code is displayed in the first three digits, and the diagnostic code is displayed in the last three digits (see Appendix B).

For a permanent virtual circuit, the cause and diagnostic codes are always from the last reset. For virtual circuit 0, they are from the last restart. For all other virtual circuits:

- If the virtual circuit is in use, the cause and diagnostic codes are from the last reset.
- If the virtual circuit is idle, the cause and diagnostic codes are from the last clear.
- If the virtual circuit is out of order, the cause and diagnostic codes can be from either the last reset or the last clear.

Keyboard Input

The following BTOS workstation keys can be used during status monitoring; all other keys (except ACTION-FINISH) are ignored:

- FINISH. Returns to the Executive.
- SCROLL UP/SCROLL DOWN. If more than ten circuits are allocated for the X.25 Network Gateway system service being monitored, the scroll keys are active and allow the user to select the ten circuits for which data is displayed in the circuit status area. Circuit is are displayed in circuit number order. However, circuit numbers may not be contiguous, depending on how circuit ranges were specified at installation of the X.25 Network Gateway.

Status Codes

This section lists the status codes and their meanings for the X.25 packet access method and the X.25 sequential access method.

X.25 Packet Access Method

Error conditions for requests to the X.25 Network Gateway packet access method occur in three circumstances:

- The request either contains invalid parameters or is invalid in the context of prior requests.
- The request is superseded by a later request.
- The state of a virtual circuit or of the packet level as a whole makes it impossible to fulfill the request.

When an incoming request is received, the gateway checks to make sure that the parameters are valid and appropriate. If an error is detected, the gateway immediately returns the request with an appropriate status code.

If a valid and appropriate request is received, the packet level attempts to fulfill the request. A fulfilled request is returned with a status code of 0 (no error). In most cases, the request cannot be fulfilled immediately and must be held momentarily by the packet level. If a request awaiting fulfillment is canceled by a later action from the user or the packet level, the request is returned with an appropriate status code.

Generally, you cannot assume that requests are returned in the order they are issued for either error or normal completion. However, requests for reads and writes are returned in the order they were issued (and are held by the packet level) whenever errors occur.

X.25 Packet Access Method Status Codes

Decimal	Hex Value	Meaning
60	3C	Invalid Device Spec. Channel specified in the Install X.25 command is invalid.
8500	2134	Link level down. The link level of the X.25 Network Gateway system service is not operational. This situation occurs either at power up, before communication with the PDN is established, or during operation, if an irrecoverable link-level error occurs. The X.25 Network Gateway system service link-level software should reestablish communication as soon as possible. If the link level remains down for an extended period, an irrecoverable error at the physical level or the link level exists. Contact a PDN representative.
8501	2135	Packet level down. The packet level of the X.25 Network Gateway system service is not operational. The situation occurs at power up, during operation following a link-level failure and subsequent reestablishment of link-level communications, or following an irrecoverable packet level error condition. The X.25 Network Gateway system service should reestablish the packet level as soon as possible. If the packet level remains down for an extended period. Contact a PDN representative.
8502	2136	Maximum number of this request has been queued. Previously submitted requests of this type must be completed before more can be issued. The maximum number of each request type is as follows: NotifyNextIncomingCallNet requests: the number of virtual circuits per line. ReadX25PacketNet requests: two per virtual circuit. WriteX25PacketNet requests: five per virtual circuit.

Decimal	Hex Value	Meaning
		All other packet access method operation requests: one per virtual circuit. Generally, since the packet level should complete requests in a short time, the request should be resubmitted. If this condition persists, QueryX25StatusNet should be used to examine the state of the X.25 Network Gateway system service to determine the cause of the delay.
8503	2137	X.25 Network Gateway system service is busy. Insufficient memory is available for the X.25 Network Gateway system service to process any more requests at this time. In normal operation, the X.25 Network Gateway system service should complete enough requests to free the memory required to accept new requests. If this error persists, you might want to consider reinstalling the X.25 Network Gateway with additional memory.
8504	2138	Process termination. All requests were (or shortly will be) returned and all virtual calls were (or shortly will be) cleared because your process has terminated.
8505	2139	Bad port parameter. A NotifyNextIncomingCallNet operation contains a port range with one of two error conditions: The high port number is less than the low port number. The low and/or high port number is not in ASCII digits.
8506	213A	No virtual circuit available. An InitiateX25CallNet operation was received, but all virtual circuits were either in use or out of order.
8507	213B	User-specified time-out. A ReadX25PacketNet or a NotifyNextIncomingCallNet operation could not be fulfilled by the packet level during the specified maximum time.
8508	213C	Virtual circuit in use. A request was received for a virtual circuit (or permanent virtual circuit) already in use.

Decimal	Hex Value	Meaning
8509	213D	Call collision. An incoming call was received on a virtual circuit before an InitiateX25CallNet operation that had been allocated to that virtual circuit could be completed. The process should resubmit the InitiateX25CallNet operation.
8510	213E	Call cleared. An AcceptX25CallNet operation was made on a circuit for which no call was pending.
8511	213F	Virtual circuit not in use. A request was received for a virtual circuit (or permanent virtual circuit) that was not allocated to any user.
8512	2140	DTE clear. Either an erroneous packet was received from the PDN, or the process requested that the call be terminated. The X.25 Network Gateway system service cleared the call that was on this virtual circuit. Data in the process of being transferred may have been lost.
8513	2141	DCE clear. The PDN cleared the call that was on this virtual circuit. Data in the process of being transferred may have been lost.
8514	2142	DTE reset. Either an erroneous packet was received from the PDN, or the process requested the call be reset. The X.25 Network Gateway system service reset the call on this virtual circuit. Data in the process of being transferred may have been lost.

Decimal	Hex Value	Meaning
8515	2143	DCE reset. The PDN reset the call on this virtual circuit. Data in the process of being transferred may have been lost.
8516	2144	DTE restart. An erroneous packet was received from the PDN, and the X.25 Network Gateway system service was restarted. All active calls were cleared. Data in the process of being transferred may have been lost.
8517	2145	DCE restart. The PDN restarted the packet level. All active calls were cleared. Data in the process of being transferred may have been lost.
8518	2146	Virtual circuit not in data transfer mode. A read, write, reset, or interrupt request was received for a virtual circuit that was not in the correct state. Either no call was present or the circuit was in the process of being cleared or reset.
8519	2147	Interrupt data. Normal completion of a read request, but with an interrupt data packet rather than a normal data packet. Interrupt data is returned to the process before any normal packets being held for the process by the packet level are returned.
8520	2148	Virtual circuit out of order. An irrecoverable error occurred on this virtual circuit, and the X.25 Network Gateway system service declared it out of order. The call on this circuit was cleared. The circuit can be restored only by the PDN.
8521	2149	Internal time out. The PDN did not respond to the packet generated by the request in the required time period. The process should resubmit the request.
8522	214A	Invalid virtual circuit number. Either one of the following conditions exists: A request was received for a virtual circuit with a vch parameter that is either out of range or is 0 (circuit 0 is reserved). A ConnectX25PermanentNet operation was received for a nonpermanent virtual circuit.

Decimal	Hex Value	Meaning
8523	214B	Data truncated. Data to be returned to the process exceeded the size of sPacketRet as specified by the process. The data were truncated to the size of the buffer.
8524	214C	No buffer. A read or write operation was attempted, with sBuffer equal to 0.
8525	214D	Permanent circuit. ClearX25CallNet or AcceptX25CallNet was issued with the vch parameter of a permanent virtual circuit. Note: This status code should never occur with this release.
8526	214E	The number of VCs requested at X.25 installation time is not enough to support the current system configuration. There must be at least one VC per workstation in a cluster environment.
8527	214F	The number of VCs requested at X.25 installation time exceeds the maximum number of 64 VCs supported by the X.25 Network Gateway.
8528	2150	Invalid maximum packet size value. The allowed values are 16, 32, 64, 128 (default), 256, 512, and 1024.
8529	2151	Maximum number of Permanent VCs exceeds the total number of VCs assigned.
8530	2152	Number of Incoming Only VCs exceeds the total number of VCs assigned.
8531	2153	Number of Outgoing Only VCs exceeds the total number of VCs assigned.
8532	2154	Total of all Outgoing Only VCs, Incoming Only VCs, and Permanent VCs exceeds the assigned number of VCs.
8533	2155	Invalid Window Size. Window size specified in the Install X.25 command is invalid.
8534	2156	Invalid Packet Size. Packet size specified in the Install X.25 command is invalid.
8535	2157	Invalid Local Address. Address specified in the Install X.25 command is too long.

Decimal	Hex Value	Meaning
8536	2158	Invalid Modulus. Only modulo 8 is supported.
8537	2159	Invalid PDN Type. PDN type specified in the Install X.25 command is invalid. PDN must be an integer between 0 and 6, or it must be left blank.
8555	216B	Invalid channel value.
31178	79CA	Conflicting Packet Size. Max packetsize and default packetsize specified in the Install X.25 command are incompatable.
31179	79CB	Invalid PVC Param. PVC install parameter not numeric.
31180	79CC	Invalid Ack Param. Wait for DCE Ack Install parameter not *yes* or *no*.
31181	79CD	ercInvalidVCParam. Install number of VCs not numeric.
31182	79CE	Invalid Default Packet Size. Install value for default packet size is invalid. Valid values are 16, 32, 64, 128,256, 512, 1024.
31183	79CF	Invalid Incoming-Only LCGN. Value specified as LCGnumber for first incoming-only VC is not numeric.
31184	79D0	Invalid Incoming-Only LC. Value specified as LC number for first incoming-only VC is not numeric.
31185	79D1	Invalid Incoming-Only VC. Value specified for the number of incoming-only VCs is not numeric.
31186	79D2	Invalid Two-Way LCGN. Value specified as starting LCG number for first Two-way VC is not numeric.
31187	79D3	Invalid Two-Way LC. Value specified as starting LC number for first Two-way VC is not numeric.
31188	79D4	Invalid Outgoing LCGN. Value specified as starting LCG number for first outgoing VC is not numeric.
31189	79D5	Invalid Outgoing LC. Value specified as starting LC number for first outgoing VC is not numeric.

Decimal	Hex Value	Meaning
31190	79D6	Invalid Outgoing VC. Value specified for the number of outgoing VC's is not numeric.
31191	79D7	User Data Too Large. Four requests return this error if the user data specified is too large: InitiateX25CallNet, AcceptX25CallNet, ClearX25CallNet, WriteX25PacketNet. If the error was returned from an Accept, the gateway clears the call.
31192	79D8	Bad Incoming Accept. The Accept Packet received by the gateway contained an invalid facility request. The call was cleared.
31195	79DB	Bad Request Facility. Improper window or packet size value specified in optional facilities. Either the accept packet has an improper value or the original incoming request had improper data and was not overridden by accept (specific to InitiateX25CallNet).
31196	79DC	Bad Accept Facility. Improper value in the incoming Accept packet per-call facility data. The gateway clears the call (specific to AcceptX25CallNet).

Sequential Access Method Error Codes

Decimal	Hex Value	Meaning
2350	092E	X.25 SAM error occurred during operation. If an X.25 SAM error occurs during a byte stream operation, the call is cleared, but the byte stream is not closed. A ReleaseByteStream or CloseByteStream operation must be implemented to close the byte stream.
2351	092F	Time out. The specified time out elapsed before the X.25 Network Gateway system service operation could finish. The operation in question is terminated, but the call is not cleared.

X.25 Cause and Diagnostic Codes

CCITT Recommendation X.25 specifies two diagnostic fields within CLEAR REQUEST, RESTART REQUEST, and RESET REQUEST packets. The Cause field indicates the reason for the operation, and the Diagnostic field provides additional information.

CCITT Recommendation X.25 defines cause and diagnostic codes for clear, restart, or reset operations initiated by a PDN. These codes are listed in the subsections "CCITT Standard Cause Codes" and "CCITT Standard Diagnostic Codes."

For clear, restart, or reset operations initiated by a DTE, CCITT Recommendation X.25 specifies that the value of the Cause field is 000 and that the value of the Diagnostic field is defined by the DTE.

CLEAR REQUEST, RESTART REQUEST, and RESET REQUEST packets can be generated locally either by the X.25 Network Gateway system service itself (in response to protocol errors) or by the user of the X.25 Network Gateway system service (for application-specific reasons). The diagnostic codes included in CLEAR REQUEST, RESTART REQUEST, and RESET REQUEST packets generated by the X.25 Network Gateway system service are listed under "BTOS Diagnostic Codes."

The same set of values is used in CLEAR REQUEST, RESTART REQUEST, and RESET REQUEST packets; however, some codes are appropriate only to one packet type. For packets that you generate you must supply the value of the diagnostic code.

The most recent cause code for a virtual circuit is displayed as the first three digits of the virtual circuit's Diagnostic (DIAG) field in the circuit status area of the X.25 status program. (See Section 7 for more information on the status program.) The cause code is also returned by the QueryX25Status operation.

The most recent diagnostic code for a virtual circuit is displayed as the last three digits of the circuit status area of the X.25 Status Program. (See Section 7.) The code is also returned by the QueryX25Status operation.

CCITT Standard Cause Codes

CLEAR Packets

Code	Definition
000	DTE originated.
001	Number is busy.
003	Invalid facility request.
005	Network congestion.
009	Out of order.
011	Access barred.
013	Not obtainable.
017	Remote procedure error.
019	Local procedure error.
021	RPOA out of order.
025	Reverse charging acceptance not subscribed (can be received only if the reverse charging user facility is used).
033	Incompatible destination.
041	Fast-Select facility acceptance not subscribed (can be received only if the fast select user facility is used).

RESET Packets

Code	Definition
000	DTE originated.
001	Out of order (applicable only to permanent virtual circuits).
003	Remote procedure error.
005	Local procedure error.
007	Network congestion.
009	Remote DTE operational (applicable only to permanent virtual circuits).
015	Network operational.
017	Incompatible destination (applicable only to permanent virtual circuits).

RESTART Packets

Code	Definition
000	DTE originated.
001	Local procedure error.
003	Network congestion.
007	Network operational.

BTOS Diagnostic Codes

Code	Definition
000	No additional information.
001	Process termination. The virtual circuit is being cleared (permanent virtual circuit is being reset) because the process using it terminated.
002	Reset request time out. The maximum number of transmissions of a RESET REQUEST packet were performed without receipt of a RESET CONFIRM packet from the remote DTE.
003	CALL REQUEST time out. No CALL CONFIRM or CLEAR INDICATION packet was received in response to a CALL REQUEST packet.
004	Window time out. Unacknowledged packets were outstanding for an excessive period of time.
005	No answer. A CLEAR REQUEST packet was transmitted in response to an INCOMING CALL packet because no NotifyNextIncomingCall operations that match the incoming call were present.
006	No session established. For virtual circuits: A packet other than an INCOMING CALL packet was received, but no virtual circuit was established. For permanent virtual circuits: A packet other than a RESET REQUEST packet with a cause code of 007 (Network operational) was received while the permanent virtual circuit was waiting to be initialized with such a packet.

Code	Definition
007	Invalid interrupt confirm. An INTERRUPT CONFIRM packet was received, but no INTERRUPT packet was outstanding.
008	Bad p(r). The p(r) sequencing number of a received RR, RNR, or DATA packet was in error.
009	Bad p(s). The p(s) sequencing number of a received DATA packet was in error.
010	No packet header. A packet has been received with a length of less than three bytes.
011	Bad packet ID. A packet has been received with a packet ID that was not recognized by the X.25 Network Gateway system service.
012	Packet too small. A packet has been received with a length less than the minimum number of bytes specified for its type.
013	Packet too big. A packet has been received with a length greater than the maximum number of bytes specified for its type.
014	Remote error in P2. A packet has been received that is inappropriate in the current state.
015	Remote error in P3. A packet has been received that is inappropriate in the current state.
016	Remote error in P7. A packet has been received that is inappropriate in the current state.
017	Remote error in D1. A packet has been received that is inappropriate in the current state.
018	Remote error in D2. A packet has been received that is inappropriate in the current state.
019	Remote error in D3. A packet has been received that is inappropriate in the current state.
020	Remote error in R1. A packet has been received that is inappropriate in the current state.
021	Remote error in R3. A packet has been received that is inappropriate in the current state.

CCITT Standard Diagnostic Codes

Code	Definition
000	No additional information.
001	Invalid p(s).
002	Invalid p(r).
016	Packet type invalid.
017	Packet type invalid for state r1.
018	Packet type invalid for state r2.
019	Packet type invalid for state r3.
020	Packet type invalid for state p1.
021	Packet type invalid for state p2.
022	Packet type invalid for state p3.
023	Packet type invalid for state p4.
024	Packet type invalid for state p5.
025	Packet type invalid for state p6.
026	Packet type invalid for state p7.
027	Packet type invalid for state d1.
028	Packet type invalid for state d2.
029	Packet type invalid for state d3.
032	Packet not allowed.
033	Packet not allowed; unidentifiable packet.
034	Packet not allowed; call on one-way logical channel.
035	Packet not allowed; invalid packet type on a permanent virtual circuit.
036	Packet not allowed; packet on unassigned logical channel.
037	Packet not allowed; REJECT packet not subscribed to.
038	Packet not allowed; packet too short.
039	Packet not allowed; packet too long.
040	Packet not allowed; invalid general format identifier.
041	Packet not allowed; RESTART packet with other than 0 in bits 1 through 4 and 9 through 16.
042	Packet not allowed; packet type is not compatible with the facility.
043	Packet not allowed; unauthorized interrupt confirmation.
044	Packet not allowed; unauthorized interrupt.
048	Timer expired.
049	Timer expired for INCOMING CALL packet.
050	Timer expired for CLEAR REQUEST packet.
051	Timer expired for RESET REQUEST packet.
052	Timer expired for RESTART REQUEST packet.
064	Call establishment problem.
065	Call establishment problem; facility code not allowed.
066	Call establishment problem; facility parameter not allowed.
067	Call establishment problem; invalid called address.
068	Call establishment problem; invalid calling address.
069	Call establishment problem; invalid facility request.

Hardware Considerations

The X.25 Network Gateway is intended for use with synchronous modems such as the Bell 201, 208, or 209 Data sets (Data-Phone 2400, 4800, or 9600 Service). Modems must be either leased from the telephone company, purchased from independent vendors offering compatible products, or supplied by the PDN vendor.

Usually, the BTOS system is permanently connected to the PDN over a leased line.

When other considerations permit, Unisys recommends the following optional modem features:

- Internally timed transmitter
- Switched carrier
- Without new sync
- Four-wire operation

A double-male RS232C extension cable must be used to connect the workstation to the modem. It should be a straight-through terminal-to-modem cable rather than the crossover (null modem) type.

When running at speeds greater than 19.2 K baud (available on IDS port B), an external RS232 to V.35 converter is necessary.

RS232C signals used in synchronous operation are shown in Table C-1.

Table C-1 **RS232C Signals in Synchronous Operation**

Pin Number	Signal Name
1,7	Ground
2	Transmit Data
3	Receive Data
4	Request to Send
5	Clear to Send
6	Data Set Ready
8	Data Carrier Detect
15	Transmit Clock
17	Receive Clock
20	Data Terminal Ready
22	Ring Indicator

References

This appendix lists Unisys and CCITT documents recommended for further information on X.25 Network Gateway.

Unisys Corporation References

The following documents contain information on the Multimode Terminal Program and its capabilities:

- *BTOS Multimode Terminal Program (MTP) Programing Reference Manual*
- *BTOS Multimode Terminal Program (MTP) Operations Guide*

Additional References

Public Data Networks, CCITT Comite Consultatif Internationale de Telegraphique et Telephonique (International Telegraph and Telephone Consultative Committee) Seventh Plenary Assembly, Document No. 7, Study Group 7, Contribution No. 489, (June 1980).

- Recommendation X.3, "Packet Assembly/Disassembly Facility (PAD) in a Public Data Network."
- Recommendation X.21bis, "Use on Public Data Networks of Data Terminal Equipment (DTEs) Which Are Designed for Interfacing to Synchronous V-Series Modems."
- Recommendation X.25, "Interface Between Data Terminal Equipment (DTE) and Data Circuit-Terminating Equipment (DCE) for Terminals Operating in the Packet Mode on Public Data Networks."
- Recommendation X.28, "DTE/DCE Interface for a Start-Stop Mode Data Terminal Equipment Accessing the Packet Assembly/Disassembly Facility (PAD) in a Public Data Network Situated in the Same Country."
- Recommendation X.29, "Procedures for the Exchange of Control Information and User Data Between a Packet Mode DTE and a Packet Assembly/Disassembly Facility (PAD)."

The CCITT documents are also contained in *McGraw-Hill's Compilation of Data Communications Standards*, edited by Harold C. Folts and Harry R. Karp (1978).

Glossary

Balanced Link-Access Procedure (LAPB) A particular implementation of a high-level data link control (HDLC) used to manage data flow between a DTE and a public packet-switched network.

CCITT Comite Consultatif Internationale de Telegraphique et Telephonique (International Telegraph and Telephone Consultive Committee). Part of the International Telecommunications Union (ITU), which is an agency of the United Nations. CCITT develops telecommunication standards.

Data Circuit-Terminating Equipment (DCE) Equipment that provides communications services between data terminal equipment (such as a BTOS workstation) and a physical link to a public data network.

Data Terminal Equipment (DTE) Equipment, such as a BTOS workstation, that uses communications services.

extended packet-sequence numbering An option that allows the maximum window size of outstanding packets to be increased.

facility An option provided by the PDN according to CCITT standards, such as reverse charging, fast-select, or closed user group.

fast select facility An option within CCITT Recommendation X.25, which allows both the called and calling data terminal equipment to transfer 128 bytes of data during call establishment and/or clearing.

High-Level Data Link Control (HDLC) A bit-synchronous, link-access protocol. Variants of the HDLC protocol are used as the link-level protocol in the X.25 Network Gateway in the BTOS cluster. See also *balanced link-access procedure.*

link-level protocol A protocol, also known as link-access protocol, that defines the link-access procedure for transferring data between data terminal equipment and data communications equipment.

Logical Channel (LC) A single stream of multiplexed data sent over a physical line.

Multimode Terminal Program (MTP) Software that allows a cluster workstation to operate on public packet-switching networks that support CCITT Recommendation X.25.

multiplexing A technique that enables several data streams to be sent over a single physical link between the data terminal equipment and the network. The link is shared by multiple users, although it appears as a private link to each user.

one-way virtual circuit A circuit that accepts either incoming or outgoing calls, but not both. Also see *Permanent Virtual Circuit, Virtual Circuit,* and *Two-Way Virtual Circuit.*

packet The basic unit of data transfer over an X.25 communications network. A packet contains control information as well as data.

Packet Assembly/Disassembly Facility (PAD) The special software in a public data network that processes the data from nonpacket mode terminals into a format compatible with CCITT Recommendation X.25.

packet-level protocol The X.25 layer that defines procedures for communication between two pieces of data terminal equipment through a public data network.

packet switching A type of data transfer that uses a network communications link only during the time of actual data transmission. The data are transmitted in small segments, called packets.

permanent virtual circuit A logical path that is permanently assigned for communication between two DTEs. A permanent virtual circuit does not require call establishment and cannot be cleared. See also *one-way virtual circuit, two-way virtual circuit,* and *virtual circuit.*

physical-level protocol The protocol that defines the modem interface between data terminal equipment and the data communications equipment.

protocol A set of procedures that governs (for the X.25 implementation) the exchange of data over a communications link.

protocol identifier Information, transferred during call establishment, that identifies the higher level protocol to be used for a virtual call.

Public Data Network (PDN) A regulated provider of communications services.

two-way virtual circuit A circuit that accepts both incoming and outgoing calls. See also *one-way virtual circuits, permanent virtual circuits,* and *virtual circuits.*

Virtual Circuit (VC) A communications link between data terminal equipment through a packet-switching network. A virtual circuit uses a logical channel which is multiplexed over a physical communications line. See also *one-way virtual circuit, permanent virtual circuit,* and *two-way virtual circuit.*

virtual circuit handle A 16-bit number used to reference and identify a virtual circuit within the gateway.

window The visible portion of display memory contained in the data frame of the video display.In a public data network, a window is the number of X.25 packets that can be transmitted before receiving an acknowledgment. The default window size is two; the maximum window size is six.

X.3 A set of parameters (for example, speed, line editing characters) used by a packet assembler/disassembler to control a start/stop terminal.

X.21 A physical interface that may be used to connect a DTE and communication equipment in a public circuit switched data network.

X.21bis RS-232 compatible interface between a DTE and communication equipment.

X.25 Internationally accepted interface to connect terminals or computers to public packet switched networks.

X.28 A set of commands that can be used by a start/stop terminal to communicate with a packet assembler/disassembler.

X.29 A set of commands or information that can be exchanged by a packet assembler/disassembler and a remote DTE.

Index

F

F

G

H

I

L

P

Q

R

S

Help Us To Help You

Publication Title

Form Number Date

Unisys Corporation is interested in your comments and suggestions regarding this manual. We will use them to improve the quality of your Product Information. Please check type of suggestion:

☐ Addition ☐ Deletion ☐ Revision ☐ Error

Comments

Name

Title Company

Address (Street, City, State, Zip)

Telephone Number

Help Us To Help You

Publication Title

Form Number Date

Unisys Corporation is interested in your comments and suggestions regarding this manual. We will use them to improve the quality of your Product Information. Please check type of suggestion:

☐ Addition ☐ Deletion ☐ Revision ☐ Error

Comments

Name

Title Company

Address (Street, City, State, Zip)

Telephone Number

NO POSTAGE NECESSARY IF MAILED IN THE UNITED STATES

BUSINESS REPLY CARD

FIRST CLASS PERMIT NO. 817 DETROIT, MI 48232

POSTAGE WILL BE PAID BY ADDRESSEE

Unisys Corporation
Product Services - East
P.O. Box 500
Blue Bell, PA 19424

ATTN: Corporate Product Information

NO POSTAGE NECESSARY IF MAILED IN THE UNITED STATES

BUSINESS REPLY CARD

FIRST CLASS PERMIT NO. 817 DETROIT, MI 48232

POSTAGE WILL BE PAID BY ADDRESSEE

Unisys Corporation
Product Services - East
P.O. Box 500
Blue Bell, PA 19424

ATTN: Corporate Product Information