IBM

X.25 Network Control Program Packet Switching Interface

SC30-3502-2

Host Programming

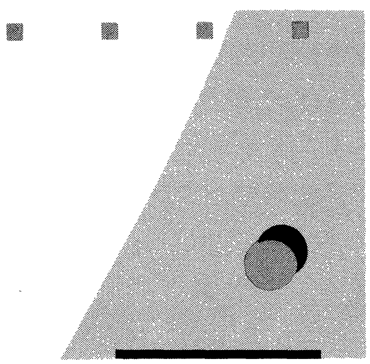
Version 3

X.25 Network Control Program Packet Switching Interface

SC30-3502-2

Host Programming

Version 3



File Number
S370/4300/30xx-50

Program Number
5688-035

Third Edition (May 1990)

This edition applies to the IBM licensed program X.25 NCP Packet Switching Interface Version 3 (program number 5688-035) Releases 1, 2, and 3.

Publications are not stocked at the address given below. If you want more IBM publications, ask your IBM representative or write to the IBM branch office serving your locality.

A form for your comments is provided at the back of this document. If the form has been removed, you may address comments to IBM Corporation, Department E15, P.O. Box 12195, Research Triangle Park, North Carolina, 27709, U.S.A. IBM may use or distribute any of the information you supply in any way it believes appropriate without incurring any obligation to you.

Special Notices

The licensed program described in this manual, and all licensed material available for it, are provided by IBM under terms of the Agreement for IBM Licensed Programs. Changes are made periodically to the information herein; before you use this document in connection with the operation of IBM systems, consult the latest *IBM System/370, 30xx, 4300, and 9370 Processors Bibliography*, GC20-0001, for the editions that are applicable and current.

Any reference to an IBM licensed program in this document is not intended to state or imply that only IBM's program may be used.

IBM may have patents or pending patent applications covering subject matter described in this document. The furnishing of this document does not of itself constitute or imply a grant of any license or immunity under any patents, patent applications, trademarks, copyrights, or other similar rights of IBM or of any third party, or any right to refer to IBM in any advertising or other promotional or marketing activities. IBM assumes no responsibility for any infringement of patents or other rights of third parties that may result from use of the subject matter disclosed in this document or for manufacture, use, lease, or sale of machines or programs described herein outside of any responsibilities assumed via the *Agreement for Purchase of IBM Machines* and the *Agreement for IBM Licensed Programs*.

Licenses under IBM's utility patents are available on reasonable and nondiscriminatory terms and conditions. IBM does not grant licenses under its appearance design patents. Inquiries relative to licensing should be directed in writing to the IBM Director of Commercial Relations, International Business Machines Corporation, Armonk, New York, 10504.

It is possible that this material may contain reference to, or information about, IBM products (machines and programs), programming, or services that are not announced in your country. Such references or information must not be construed to mean that IBM intends to announce such products, programs, or services in your country.

The following terms, used in this publication, are trademarks of the IBM Corporation in the United States and/or other countries:

NetView VTAM

The following term, used in this publication, is a registered trademark of the IBM Corporation in the United States and/or other countries:

IBM

Contents

Figures

Tables

About This Book

This *X.25 NPSI Host Programming* manual is intended to provide the format, description, and protocol of the Command PIUs and PAD parameters used with the GATE, DATE, and Transparent PAD functions of X.25 NPSI. It contains general-use programming interfaces, which allow the customer to write host applications that use the services of X.25 NPSI.

X.25 NPSI Host Programming assists application programmers in writing programs to interface with IBM's X.25 Network Control Program Packet Switching Interface (X.25 NPSI) licensed program. X.25 NPSI offers Systems Network Architecture (SNA) users the ability to use communication facilities that support the X.25 Interface as defined by the International Telegraph and Telephone Consultative Committee (CCITT) at Geneva in 1980 and Malaga-Torremolinos in 1984.

Who Should Use This Book

X.25 NPSI Host Programming is intended for the following audiences:

- Application programmers who are responsible for designing, coding, compiling, executing, debugging, and testing application programs that work with X.25 NPSI connections
- System programmers who are customizing subsystems (such as TSO, CICS, and IMS) that support application programs
- Programmers who write communication and transmission control programs (CTCPs) to interface with the GATE and DATE functions of X.25 NPSI.

How to Use This Book

Use this book to help you design application programs, customize subsystems, and write CTCPs to interface with X.25 NPSI.

How This Book Is Organized

Chapter 1, "Methods of Communication Using X.25 NPSI," describes items you must consider in designing and programming a host application that can communicate with remote DTEs. It includes information about communication with SNA and non-SNA DTEs, as well as a description of the X.25 NPSI LU simulator. It also provides general information about CTCP programming.

Chapter 2, "Programming Using PCNE and PAD Support," describes PCNE and the integrated and transparent PAD support functions of X.25 NPSI. It also explains how to program virtual circuits for use with PCNE and PAD support.

Chapter 3, " Programming a GATE CTCP," explains virtual circuit programming using the GATE and fast connect GATE functions of X.25 NPSI. This chapter also describes how to develop CTCPs and how to program virtual circuits for use by the GATE functions, fast connect GATE functions, and X.21 switched connections that use GATE.

Chapter 4, "Programming of a DATE CTCP," describes virtual circuit programming using the DATE function of X.25 NPSI. This chapter also describes how to develop CTCPs and how to program virtual circuits for use by the DATE function.

Chapter 5, "Host Application Programming," describes the definitions and other requirements that must be satisfied when you use X.25 NPSI to support communication between host IBM subsystems and remote DTEs.

Appendix A, "Example of Programming a DATE CTCP," provides you with an example of a DATE CTCP.

A glossary, bibliography, and index are also included in this book.

Abbreviations and Terms Used in This Book

Throughout the book, the following abbreviations and terms apply.

CCITT International Telegraph and Telephone Consultative Committee

CTCP Communication and Transmission Control Program

DATE Dedicated Access to X.25 Transport Extension

GATE General Access to X.25 Transport Extension

ISDN Integrated services digital network

ISO International Organization for Standardization

MVS Multiple Virtual Storage/System Product (MVS/SP) operating system, Multiple Virtual Storage/Extended Architecture (MVS/XA) operating system, or Multiple Virtual Storage/Enterprise System Architecture (MVS/ESA)

ACF/NCP Advanced Communications Function for Network Control Program

PAD Packet Assembler/Disassembler

PSDN Packet Switched Data Network

SNA Systems Network Architecture

SSP System Support Programs

SSCP System Services Control Point

SVCSC Switched Virtual Circuit Subarea Communication

VM Virtual Machine/System Product (VM/SP) operating system or Virtual Machine/Extended Architecture (VM/XA) operating system

VSE Disk Operating System/Virtual Storage Extended (DOS/VSE) operating system

VTAM Virtual Telecommunications Access Method

X.25 NPSI X.25 NCP Packet Switching Interface.

Other abbreviations used in this book are listed in the "Glossary."

How the Term Network Is Used

The term network has at least two meanings. A public network is established and operated by communication common carriers or telecommunication administrations for the specific purpose of providing circuit-switched, packet-switched, and leased-circuit services to the public.

A user application network is a configuration of data processing products, such as processors, controllers, and terminals, established and operated by users for data processing or information exchange, which can use transport services offered by communication common carriers or telecommunication administrations.

Network, as used in this book, refers to a user application network.

How Version and Release Are Abbreviated

The terms version and release are abbreviated as "V" and "R." For example, X.25 NPSI Version 3 Release 3 is abbreviated as X.25 NPSI V3R3.

How Numbers Are Written

In this book, numbers over four digits are represented in metric style. A space is used rather than a comma to separate groups of three digits. For example, the number ten thousand five hundred fifty-two is written 10 552.

Symbols Used in This Book

Figure 1 illustrates the networking symbols used throughout this book.

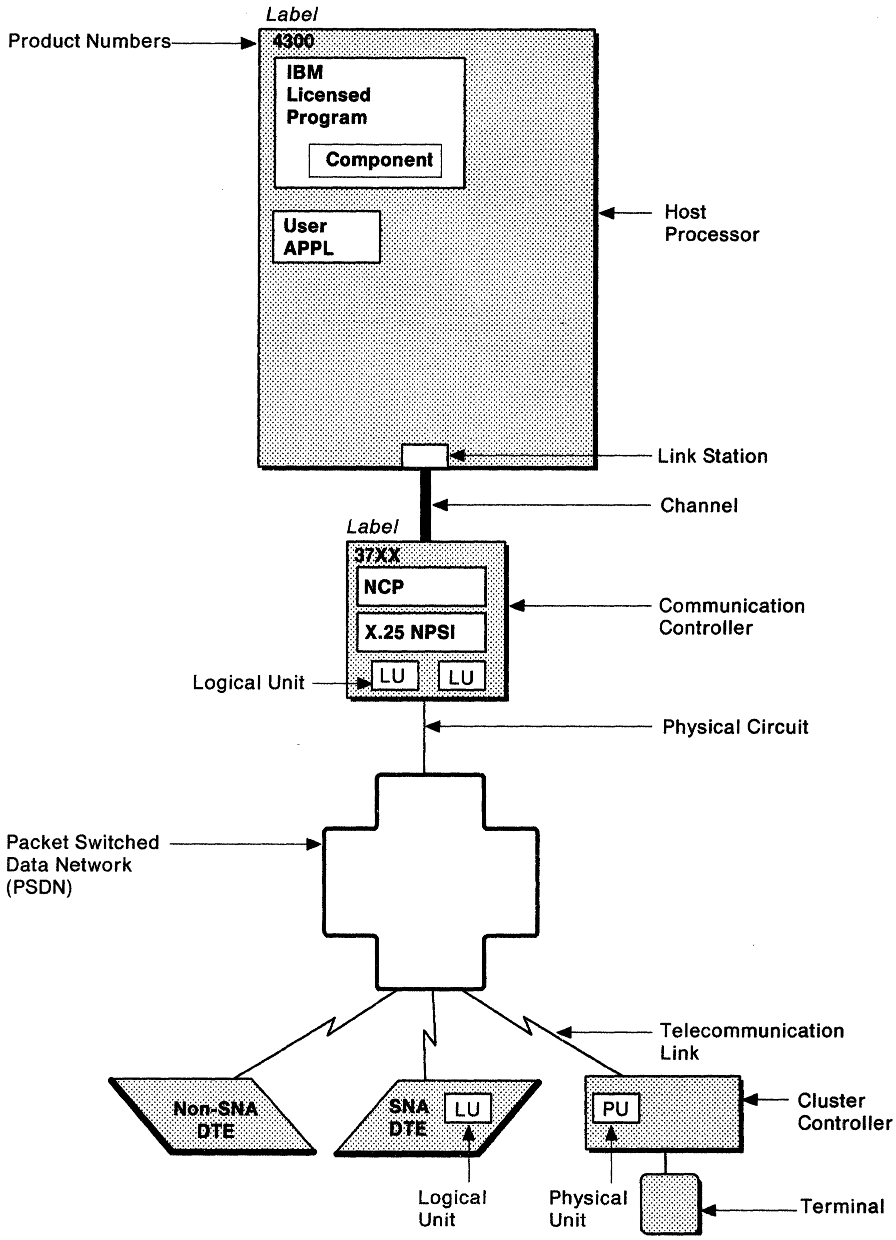


Figure 1. Networking Symbols Used in This Book

What Is New in This Book

This book has been changed to reflect the enhancements available for X.25 NPSI Version 3 Release 3. These changes include:

- Enhanced multichannel link compatibility
- NCP V5R3 support
- Ability to establish link session priority
- Enhanced SNA Type 2.1 boundary function support (casual connection)
- RU chaining support for long non-SNA messages
- Improved conformance to the International Organization for Standardization (ISO) 7776 and 8208
- Enhanced PAD support
- X.21 switched connections support
- Enhanced capability to activate, load, and dump remote NCPs
- Miscellaneous enhancements, which include:
  - Ability to clear an SVC based on an inactivity time-out
  - Ability to use billing units as statistics
  - Improved inbound flow control
  - Improved flow control negotiation in GATE and DATE
  - Improved integrated PAD support
  - Improved reset processing.

These changes are described in the following chapters:

- Chapter 1, "Methods of Communication Using X.25 NPSI," has been expanded to describe the enhanced SNA type 2.1 support and the interpretation of the flow control parameters in the Call Connected packet for GATE and DATE. This chapter also contains information regarding the ability to perform RU chaining for long non-SNA messages.
- Chapter 2, "Programming Using PCNE and PAD Support," has been expanded to reflect that PAD parameters can be set by the PADPARM keyword. A description of the enhanced SIGNAL/BREAK processing for X.25 NPSI V3R3 has also been added.
- Chapter 3, " Programming a GATE CTCP," has been expanded to include the establishment of an X.21 switched connection using GATE.
- Chapter 4, "Programming of a DATE CTCP," has been updated to describe DATE support of the integrated PAD function.
- In Chapter 5, "Host Application Programming," references to CLISTs have been changed to command lists.
- Appendix A, "Example of Programming a DATE CTCP," has been added to provide an example of a DATE CTCP.
- The Glossary has been expanded, and the Bibliography and Index have been updated.

Where to Find More Information

Before using this book, you should be familiar with SNA concepts and products as described in *Systems Network Architecture Concepts and Products*, and with the various logical link control (LLC) support functions of X.25 NPSI. If you are using this book to write CTCPs, you should be familiar with VTAM™ application programming[1] techniques and the information contained in the following books:

- *VTAM Programming*
- *VTAM Programming for LU 6.2*
- *X.25 NPSI Planning and Installation*
- *X.25 NPSI Diagnosis, Customization, and Tuning.*

For information on these and other related publications, see "Bibliography" at the back of this book.

1 VTAM is a trademark of International Business Machines Corporation.

Chapter 1. Methods of Communication Using X.25 NPSI

Chapter 1. Methods of Communication Using X.25 NPSI

X.25 NPSI is an IBM licensed program that provides VTAM users with an X.25 interface to connect remote data terminal equipment (DTEs) with Systems Network Architecture (SNA) networks. This chapter provides you with information to design or modify a host application program to allow communication with remote DTEs through X.25 NPSI.

X.25 NPSI makes a virtual circuit appear to the access method and NCP as a Synchronous Data Link Control (SDLC) link with an associated physical unit (PU) and logical unit (LU). Three types of SNA sessions result when the virtual circuit is active: an SSCP-PU session, an SSCP-LU session, and an LU-LU session.

- An SSCP-PU (system services control point to physical unit) session is established with each multichannel link physical unit (MCH PU) and each virtual circuit PU. The SSCP-PU session allows the SSCP to monitor and change the status of the circuits by sending SNA requests to the PU. These commands are processed by the NCP and X.25 NPSI.
- An SSCP-LU session is associated with each MCH LU and virtual circuit LU. The SSCP-LU session allows the SSCP to transfer commands to and from the LUs.
- An LU-LU session occurs between an LU associated with the host application program and the virtual circuit LU. This LU-LU session allows the application program to transfer messages to and from the virtual circuit LUs. The MCH LU to APPL LU session exists only if the APPL LU is in a communication and transmission control program (CTCP).

Communication with an SNA DTE

X.25 NPSI supports connection to SNA devices, including the following session types:

- Type 0
- Type 1
- Type 2
- Type 6.1
- Type 6.2.

There are no special considerations for the host application program when it communicates with an SNA device. However, the host application program must ensure that the remote device has the correct device specifications. If the application uses Customer Information Control System (CICS) or Information Management System (IMS), entries for the correct specifications must be added to the table of devices.

X.25 NPSI must have the virtual circuit defined as logical link control (LLC) type 2 or type 3. The type you should use depends on the network path configuration. If you are using a Network Interface Adapter (NIA) or a physical services header (PSH) interface at the remote DTE, use LLC type 2; otherwise, use LLC type 3.

Switched Virtual Circuit Subarea Communication (V3R2 and Later Releases)

Switched virtual circuit subarea communication (SVCSC) allows two subarea nodes to be connected by a switched virtual circuit (SVC). The connection is used primarily for the following reasons:

- Connecting two subarea nodes in different PSDNs
- Adding transfer capability through new virtual routes that map to VCs of different MCHs.

Note: A PSDN is required to execute the SVCSC function.

X.25 NPSI Considerations

X.25 NPSI V3R2 and later releases support SVCSC. With SVCSC, communication with another communication controller running X.25 NPSI or communication with a host computer that has the X.25 telecommunication subsystem controller (TSC) is possible.

SVCSC supports SNA traffic between subareas. The NCPs supporting these subareas can provide SNA network interconnect (SNI) functions. SVCSC provides connectivity between any combination of 3720 and 3745 communication controllers and X.25 TSC. The subarea node qualified logical link control (QLLC) is used by SVCSC.

Both the access method and the NCP are aware that a switched virtual circuit is used. All connection parameters, such as dial number, VCCPT index, and OUFT index, are defined to the access method. A VTAM operator or an automated operator facility, such as NetView™[2] program command list, must initiate the dial connection.

Switched virtual circuits used for SVCSC communication do not have to be dedicated to this function.

Short Hold Mode (SHM)

The short hold mode feature that is managed by X.25 NPSI can provide a significant savings in network fees. When there is an absence of traffic on an X.25 SVC for an amount of time specified by the user, and if SHM is supported by both sides, X.25 NPSI breaks the virtual connection without informing NCP or VTAM. From an SNA point of view, the connection is unchanged, and a logical connection continues to be in place. When traffic resumes from either side, X.25 NPSI automatically reestablishes the connection.

SNA Type 2.1 Node Support (V3R2 and Later Releases)

X.25 NPSI V3R2 and later releases, in conjunction with VTAM V3R2, allows logical units (LUs) that are connected to an SNA type 2.1 peripheral node to communicate through a PSDN.

Two peer systems can communicate using the SNA backbone network without requiring the participation of a host application program for relay purposes. X.25 NPSI also allows SNA LU type 6.2 multiple session communication for the LUs that reside in the SNA type 2.1 node.

---

2 NetView is a trademark of the International Business Machines Corporation.

Enhanced SNA Type 2.1 Boundary Function Support (Casual Connection) (V3R3 Only)

The enhanced SNA type 2.1 boundary function support provided in X.25 NPSI V3R3 extends the NCP's X.25 boundary node support by allowing a primary SNA type 2.1 peripheral node to be attached to an NCP through an X.25 NPSI that is acting as the secondary partner.

The enhanced SNA type 2.1 boundary function support is also called casual connection. Casual connection allows a VTAM, NCP, and X.25 NPSI configuration, working as a node T2.1, to communicate with another node T2.1. This adjacent node T2.1 can be another VTAM, NCP, and X.25 NPSI configuration, or a native node T2.1. With casual connection, you do not have to predefine the identity of the adjacent link station.

Communication with a Non-SNA DTE

Communication between a non-SNA DTE and a host application program is *always* performed through the X.25 NPSI LU simulator. The LU simulator is used for LLC types 0, 4, and 5 and for CTCP communication with the MCH LU. The LU simulator allows communication with applications that support SNA LU type 1 devices, including CICS, IMS, and TSO. These applications are described in Chapter 5, "Host Application Programming."

LU Simulator

The LU simulator converts the SNA transmission header (TH) and request/response header (RH) into appropriate X.25 headers when SNA data is sent to X.25 NPSI for transmission to a non-SNA device. X.25 NPSI then performs any other processing necessary for transmission of the message to the PSDN.

Communication from a non-SNA DTE to an SNA host application program flows in the reverse direction. After X.25 NPSI satisfies all X.25 processing considerations, the LU simulator then converts the X.25 headers into appropriate SNA headers and forwards the message to the destination application.

The X.25 NPSI LU simulator operates as an SNA secondary LU type 1 in a half-duplex mode using either contention or flip-flop protocol. Only one LU-LU session is allowed on a virtual circuit to a non-SNA DTE at any given time.

For a basic understanding of the SNA functions and concepts involved in a type 1 LU-LU session, see the following publications:

- *SNA Concepts and Products*
- *SNA Sessions between Logical Units*
- *IBM 3767 Models 1 and 2 Communication Terminal Component Description*
- *SNA Technical Overview*.

Bracket Protocol

Most exchanges between LUs are organized into logical units of data for transmission or reception. The LUs ensure that the data units are not broken up during the exchange.

Bracket protocol is initiated by an issuer placing the begin bracket (BB) indicator in the SNA header. An end bracket (EB) is used to terminate the logical unit of work.

When neither LU is engaged in an exchange of data, a state known as *between bracket* exists. A second-speaker LU, as defined in the BIND, can *bid* for initiation of a bracket. The partner LU sends a positive or negative response. If a positive response is received, the bidding LU can begin the bracket. The partner can choose to reject the *bid* if, for instance, the partner has already initiated a bracket. The LU simulator sends an X'0813' negative response to reject the BID request if the BID request cannot be accepted.

Bracket protocol works the same with the LU simulator as it does with all LU type 1 devices.

Response Types

SNA defines two different types of responses: *definite* and *exception* responses.

Definite response requests are used to ensure that a message has arrived at its destination. When this type of message is received by the LU simulator, a response is immediately returned. Unless the delivery confirmation bit (D bit) is used, the LU simulator does not wait for the data to be sent out or for the data to arrive at the destination before responding to the host.

Exception response requests are used when the arrival of the message is not critical or when there is a high degree of traffic on the communication path. In such cases, the sender is notified, by means of a negative response, only when there is an error.

Half-Duplex Flip-Flop Protocol

In a half-duplex flip-flop session, the LU that is the first speaker to send is the owner of the change of direction (CD) indicator. The CD owner continues sending requests until it reaches the end of the data it wishes to send or until the receiver of the data issues a SIGNAL command. In either case, after the sender (CD owner) sends the last request of the last chain, the sender turns on the CD indicator to give control of the session to the other session partner.

An LU that is awaiting a CD is prohibited from sending data. It is free to send response and expedited-flow requests, such as SIGNAL.

If a receiving LU wishes to initiate data transmission, it must obtain the CD indicator. It can notify the sender by issuing the SIGNAL command. The sender (CD owner) can then either relinquish the CD indicator or refuse the request.

The LU simulator always sends a CD indicator in host-bound PIUs, which allows the host to respond immediately.

Half-Duplex Contention Protocol

In a half-duplex contention session, either LU can initiate data transmission. If both LUs send at the same time, the *contention winner* is the winner defined in the BIND image at the session's initiation. The loser, as defined in the BIND, receives a negative response with a sense code of X'081B'.

With X.25 NPSI, this negative response occurs with definite response only. In the case of exception response, data from either side is processed immediately or queued in X.25 NPSI.

Session-Level Pacing

For session-level pacing, the LU simulator supports only send pacing. The simulator sends a pacing response each time it receives a request with the pacing bit set to 1 in the RH. For an LU associated with a virtual circuit or a physical circuit, the pacing parameter must be 1.

Although the pacing information placed by the LU simulator in a response does not cause the release of the buffers containing the corresponding request unit, session-level pacing allows the data flow between the host and the NCP to be regulated. An RU from the host entering the NCP is processed by the LU simulator only when the packet window is open — that is, when the previous RUs in the window have been sent to the network.

Message Transmission (V3R2 and Previous Releases)

X.25 NPSI does not perform RU chaining with the LU simulator. X.25 NPSI maps outbound chained RUs into independent packets, regardless of the RU chaining bits. X.25 NPSI assembles inbound packets into single-segment, only-in-chain (OIC) RUs.

X.25 NPSI does not support SNA segmenting with the LU simulator. If the value specified for MAXDATA is not large enough, X.25 NPSI returns a negative response with a sense code of X'08F5' to the host.

Support of RU Chaining for Long Non-SNA Messages (V3R3 Only)

X.25 NPSI V3R3 supports RU chaining for long non-SNA messages on both inbound and outbound flows. On inbound flows, X.25 NPSI converts each long non-SNA message into an SNA RU chain. The conversion produces a first-in-chain (FIC) PIU, middle-in-chain (MIC) PIU, or last-in-chain (LIC) PIU. On outbound flows, each SNA chain is converted into a packet sequence. X.25 NPSI can create two types of packet sequences:

- Complete packet sequence (CPS)

  A CPS contains contiguous full data packets with the M bit set to 1 and the Delivery bit (D bit) set to 0, followed by any other data packet with the M bit set to 0.

- M bit sequence.

  An M bit sequence contains a CPS series. Each packet within the series has both the M bit and D bit set to 1, except for the last packet of the last CPS, which has the M bit set to 0.

On inbound flows, X.25 NPSI provides optional support for RU chaining through the MBITCHN keyword on the X25.MCH statement. When a CPS is received, if MBITCHN=YES and DBIT=NO is specified, X.25 NPSI accumulates the data packets and builds an RU chain (FIC, MIC, or LIC). The size of the PIU in the chain is determined by either the value of the X25.MAXPIU keyword on the BUILD statement, or the MAXRU size on the BIND command. When MBITCHN=YES and DBIT=YES are specified, X.25 NPSI accumulates CPS data packets and builds an RU. The length of the RU is determined by the length of the CPS series. When the M-bit sequence is received (CPS series), RUs are chained as FIC, MIC, or LIC, depending on the length of the CPS series.

Note: If the last packet of an M-bit sequence has the D bit set to 1, X.25 NPSI sends an RR packet for end-to-end acknowledgment.

If the application sends a negative response to X.25 NPSI, X.25 NPSI builds a CANCEL CHAIN command and sends the DTE a Reset Request packet containing a Diagnostic code of X'B3'.

On outbound flows, if MBITCHN=YES and DBIT=YES is specified, each RU chain is converted into a CPS series that is linked together using the D bit. The last packet in the last CPS has the M-bit set to 0. This series is sent to the non-SNA DTE as an M-bit sequence. If MBITCHN=YES is specified and DBIT=NO, each RU chain is converted into a single CPS. The CPS is then sent to the non-SNA DTE.

On outbound flows, X.25 NPSI processes the CANCEL CHAIN command by sending to the DTE a Reset Request packet containing a diagnostic code of X'B3'.

For both inbound and outbound flows, when MBITCHN=YES is specified, and a Reset Indication packet is received during the process of an M-bit sequence, X.25 NPSI builds a CANCEL CHAIN command RU and sends it to the host.

Note: If the LU-simulator LU-application supports a definite response, X.25 NPSI sets the D bit to 1 in the last packet of an M-bit sequence.

Actions on RUs

X.25 NPSI does not perform any processing on the SNA data RU *except* when you use the optional character code translation between ASCII and EBCDIC, and when you use integrated PAD support with password protection.

The two types of bidirectional control RUs that flow between the LU and the LU simulator are session control RUs and data flow RUs.

- Session control RUs manage:
  - LU activation and deactivation
  - Session initiation and disconnection
  - Data traffic initiation
  - LU status.
- Data flow RUs manage:
  - Directions of the data on the LU-LU session
  - Traffic assessment between LUs.

  Some of these RUs are partial implementations of SNA. The extent of implementation for each supported RU type is specified in Table 1 on page 9 and Table 2 on page 10.

Session Control RUs: The LU simulator recognizes and processes the session-control RUs described in Table 1.

Table 1. Session Control RUs

Command	Description
ACTLU	An ACTIVATE LOGICAL UNIT command is received from the SSCP. The LU simulator returns a positive response to the SSCP checking for the activation type. If the session is not bound, the ACTLU is for an initial activation and is followed by a BIND. If the session is already bound, the ACTLU is for a session continuation case and is not followed by a BIND. The ACTLU establishes a connection between VTAM (SSCP) and the target LU. Logon requests can then be transmitted to the host.
BIND	A BIND SESSION command is received from the host LU. The LU simulator validates the bind image and completes the session connection. BIND establishes a connection between the host application program and the virtual circuit LU. This is a non-negotiable BIND. The validity of the BIND parameters are checked against the values listed in Table 3 on page 13. If BIND is accepted, the LU simulator sends a positive response to the primary LU (PLU).
SDT	A START DATA TRAFFIC command is received from the host LU. The LU simulator becomes eligible to send and receive data and returns a positive response to the application LU.
CLEAR	A CLEAR command is received from the host LU. The LU simulator clears all the local pending conditions and returns a positive response to the LU. The status of the LU, as managed by the LU simulator, is reinitialized so that further processing can continue.
UNBIND	An UNBIND SESSION command is received from the host LU. The LU simulator returns a positive response to the host and releases the remote DTE from the session. LOGON requests for a new session can then be transmitted to the host.
DACTLU	A DEACTIVATE LOGICAL UNIT command is received from the SSCP. The LU simulator returns a positive response to the SSCP and breaks the connection between the virtual circuit LU and VTAM.

Data Flow Control RUs: The LU simulator accepts and processes the data-flow control RUs described in Table 2.

Table 2. Data Flow Control RUs

Command	Description
CANCEL	A CANCEL command is received from the host LU. The LU simulator returns a positive response and takes no other action. Because the chaining concept is local to the LU simulator, the host application program needs a higher-level end-to-end protocol to cancel a chain of RUs being sent to a remote non-SNA DTE. **For V3R2 and previous releases:** The LU simulator never sends a CANCEL command to the host. **For V3R3 only:** When MBITCHN=YES, if a CANCEL command is received from the host LU, the LU simulator returns a positive response and resets the VC with a Diagnostic code of X'B3'. On receipt of a RESET INDICATION, the LU simulator sends a CANCEL command to the host.
SIGNAL	A SIGNAL command is received from the host LU. The LU simulator returns a positive response and takes no other action unless X.25 NPSI integrated PAD support is used. The SLU never generates a SIGNAL command unless the integrated PAD function is used to support the remote DTE. When the integrated PAD function receives a SIGNAL from the host, the integrated PAD sends an INDICATION OF BREAK message to the remote PAD. On receipt of an Interrupt, Reset, or qualified packet from the remote PAD, the integrated PAD interface sends a SIGNAL request to the host.
BID	A BID command is received from the host LU. The LU simulator enters begin bracket pending status if its status is such that it can accept the request and return a positive response. An exception response is returned if the LU simulator cannot accept the BID. All data packets received from the network are buffered until an RU is received specifying begin bracket.
CHASE	A CHASE command is received from the host LU. The LU simulator returns a positive response and takes no other action.
SHUTD	A SHUTDOWN command is received from the host LU. The LU simulator returns SHUTDOWN COMPLETE (SHUTC) as a response as soon as the LU enters the between bracket state. Any additional incoming data packets are discarded. If the LU simulator can close the bracket, the SHUTC command is returned after receiving the first packet with the more data bit (M bit) set to 0 (end of packet sequence). With integrated PAD, a SHUTDOWN can be optionally converted to an INVITATION TO CLEAR message for an SVC, or a Reset packet for a PVC, and sent to the remote PAD.

Session Initiation

A host application program establishes communication with an X.25 NPSI remote DTE in the same way it does with any LU type 1 device. The session initiation requests used to establish this communication can originate from any one of the following sources:

- Remote users

  Remote users can initiate a session from their devices by entering a LOGON command.

  The LOGON command is translated into a formatted initiate (INIT) request by the unformatted system services (USS) table in use at the host. The application program need not have previous knowledge of the LUs existence. However, if your network uses non-EBCDIC devices, the USS table must be coded to recognize the characters used by these devices.

- Network operators

  Network operators can establish a session by issuing:

  - The V NET,LOGON=*application* command

  - The V NET,ACT command when LOGAPPL=*application* is coded on the LU statement.

  When either of these commands is issued, a session is initiated between the controlling application (specified by *application*) and the LU simulator by the application issuing the OPNDST macro in its logon exit.

- Application programs

  Application programs can initiate a session by issuing:

  - The OPNDST OPTCD=ACQUIRE macroinstruction
  - The SIMLOGON and OPNDST OPTCD=ACCEPT macroinstructions.

  In the second case, a logon exit must be coded as well. The application program must have previous knowledge of the LUs existence to use these instructions.

- Network definition

  The remote user can be connected directly to a specific application if you code both of the following:

  - The LOGAPPL=*application* keyword on the LU macro

  - The OPNDST OPTCD=ACCEPT macro in the logon exit of the CTCP or application program.

  In the LOGAPPL keyword, *application* is the name of the application program to which the remote user is connected.

- NetView command list

  If the processing initiated by the entry of a V NET, LOGON command or the activation of a resource with the LOGAPPL keyword specified does not result in a logon, the request is not queued and the request is not reattempted. In this instance, you could write a command list to periodically check if the LU is in session, and, if not, the command list could issue a V NET,LOGON=appl name.

The request parameter list (RPL) and the node initialization block (NIB) are additional control blocks used for session establishment. The RPL contains information that describes the session and how to establish it. The NIB contains additional information about the session. The NIB must specify the symbolic name of the LU (NIBSYM).

You also have the option of specifying the BIND image in the NIBNDAR field of the NIB. The BIND image contains the session parameters that establish the communication rules to be followed for session establishment. The session parameters enable each end of the session to know what the other end of the session will or will not do in different communication situations. If you do not specify a BIND image, the VTAM default is used.

If a mode name is required, a corresponding mode table must be defined to VTAM. The following is a typical mode table entry that can be used with the X.25 NPSI LU simulator:

```
name   MODEENT FMPROF=X'03',                        X
               TSPROF=X'03',                        X
               PRIPROT=X'B1',                       X
               SECPROT=X'B0',                       X
               COMPROT=X'3040',                     X
               PSNDPAC=1
```

name is the mode name used in the NIB definition.

Table 3 on page 13 represents a BIND image required to establish a session.

See *VTAM Customization* for information on coding and installing a mode table and *VTAM Programming* for more information on controlling session initiation through a CTCP.

Table 3. Bind Parameter List

Byte	Bits	Value	Meaning
0	0–7	X'31'	BIND code
1	0–3	B'0000'	BIND format = 0
	4–7	B'0001'	BIND type = 1
2	0–7	X'03'	FM profile = 3
3	0–7	X'03'	TS profile = 3
			Definition of Primary Protocol:
4	0	B'1'	Multiple request chains
	1	B'0'	Required value
	2–3	B'01'	Exception response
		B'10'	Definite response
		B'11'	Exception or definite response
	4–5		Not used
	6	B'0'	Compression not used
	7	B'1'	PLU can send EB
		B'0'	PLU will not send EB
			Definition of Secondary Protocol:
5	0	B'1'	Multiple request chains (the SLU always sends OIC to the host)
	1	B'0'	Required value
	2–3	B'01'	Exception response
		B'10'	Definite response
		B'11'	Exception or definite response
	4–5		Not used
	6	B'0'	Compression not used
	7	B'1'	Secondary can send EB (Not allowed with flip-flop)
		B'0'	Secondary will not send EB
			Common Protocol:
6	0		Not used
	1	B'0'	FM header not used
	2	B'1'	Bracket used
		B'0'	Bracket not used
	3	B'1'	Bracket termination rule 1
	4	B'0'	EBCDIC
	5–7		Not used
			Common Protocol:
7	0–1	B'01'	Half-duplex contention
		B'10'	Half-duplex flip-flop
	2	B'0'	Primary LU recovery responsibility
	3	B'0'	First speaker is secondary
	4–7		Not used
8	0–7		Not Used
9	0–1		Not Used
	2–7	B'000001'	Pacing

Session Termination

Termination of a session between a host application program and an X.25 NPSI resource occurs in much the same way as it does for any other LU type 1 device. Session termination requests can come from any of the following sources:

- Remote users

  Remote users can terminate sessions from their devices by:

  - Entering the LOGOFF command
  - Pressing a key that causes the X.25 connection to be terminated.

- Network operators

  Network operators can terminate sessions by issuing one of the following commands:

  - V NET,INACT,ID=*luname*

    When this command is issued, the logical unit is deactivated.

  - V NET,TERM,ID=*luname*

    When this command is issued, a termination request is sent to the SSCP.

- Application programs

  Application programs can terminate sessions normally by issuing the CLSDST macroinstruction.

- Network failures

  Network failures can originate from many sources, such as the failure of a hardware or software component. In most cases, the session is abnormally terminated.

See Chapter 2, "Programming Using PCNE and PAD Support," Chapter 3, " Programming a GATE CTCP," and Chapter 4, "Programming of a DATE CTCP," for more information on how the CTCP controls session termination.

For more specific information on controlling session termination through a CTCP, see *VTAM Programming*.

Session Continuation (V3R2 and Later Releases)

X.25 NPSI V3R2 and later releases, in conjunction with VTAM V3R2 and NCP V5R2 and later releases, provides optional session continuation capabilities, including takeover and return of ownership.

Ownership of resources by a system services control point (SSCP) is determined by either initial activation or acquisition.

Session continuation provides communication with no disruptions for active LU-LU sessions, if the owning SSCP fails or connectivity between the LU simulator and the owning SSCP is lost. The LINE, PU, and LU ownership can be taken over by another SSCP, and given back to the original SSCP when it becomes active again. With X.25 NPSI V3R2 and later releases, the NCP allows a switched line, PU, and LU to remain active. Thus, when the session partners are still accessible, the active LU-LU sessions belonging to that PU are not disturbed. Session continuation operates in both SNA and non-SNA environments, and applies to PVC and SVC.

CTCP Programming

When using GATE, with or without fast connect, or DATE, you must use a communication and transmission control program (CTCP) as an interface between host application programs and X.25 NPSI. The CTCP must communicate with the application program it controls. In particular, the CTCP and the application program must be able to communicate to accomplish the following:

- To perform a Call Request
- To handle SIGNAL outbound
- To handle SIGNAL inbound (caused by BREAK at a remote DTE).

The CTCP must be able to process all valid control and qualified packets for the virtual circuit. A list of the control packets for GATE can be found in Table 5 on page 54. The control packets for DATE are listed in Table 7 on page 97 and Table 8 on page 98. The CTCP is able to specify any valid options within these packets.

An example of these options is the CALL REQUEST command from the CTCP to X.25 NPSI. In a message to DATE or GATE, the CTCP can specify *any* combination of facilities that are acceptable to the PSDN within the FACILITIES field of the Call Request packet or the Call Accepted packet.

It is *not* the responsibility of the CTCP to manage the packet level processor (PLP) counters, the packet-level modulo, or the M bit. They are managed by X.25 NPSI. When communicating through a CTCP, the enhancements for ISO are not implemented by X.25 NPSI. It is the responsibility of the host programmer of the CTCP to ensure that the ISO functions are implemented.

Selecting GATE or DATE

While GATE and DATE both operate in conjunction with the LU simulator, these X.25 NPSI functions are designed to satisfy different operating requirements. Table 4 on page 16 shows the major differences between GATE and DATE processing. If you specify MBITCHN=YES on the X.25 MCH statement, X.25 NPSI does not support the M-bit sequence. On inbound flow, X.25 NPSI builds an OIC PIU for each CPS received. After receiving a CPS, the RU size is determined by combining either the value of the X.25 MAXPIU keyword on the BUILD statement, or the MAXRU size on the BIND command with the remaining accumulated data packets. On outbound flow, X.25 NPSI converts each FIC, MIC, or LIC PIU into an OIC, then builds and sends a corresponding CPS.

Table 4. GATE and DATE Processing Differences

GATE	DATE
The logon is always built by X.25 NPSI.	Building of the logon by X.25 NPSI is optional when using DATE. The logon can be built by X.25 NPSI if this service is requested by the CTCP in the CALL REQUEST or CALL CONFIRMATION command.
GATE always uses LLC type 4.	For DATE, the LLC type is determined by the CTCP, including SNA sessions to SNA destinations. See the list of virtual circuit types and corresponding hexadecimal codes in "Determination of Virtual Circuit Type" on page 89.
RU byte 0 contains control information for qualified, control, and data packets.	RU byte 0 contains no control information when DATE is communicating with the VC LU, except when transparent PAD is used. RU byte 3 contains the packet type for PIUs passed to or from the MCH LU.
The CTCP is in session with the MCH LU and the VC LU.	The CTCP is in session only with the MCH LU. The application program is in session with the VC LU.
After CLEAR exchange, the CTCP must issue the CLSDST macro.	After CLEAR exchange, X.25 NPSI requests that the NCP send an INOP message (INOP RU) to VTAM. If the application issues the CLSDST macro, X.25 NPSI sends an information message to the CTCP, and the CTCP must initiate a CLEAR exchange.
X.25 NPSI manages the packet level processor timers.	The CTCP manages the packet level processor timers.
The CTCP acts as a relay program for session data. This GATE function provides better transmission synchronization than is possible with DATE.	The CTCP does not act as a relay program for session data, because the data is transmitted directly to the application program. Because the CTCP does not process session data, DATE can be used to achieve better host performance than is possible with GATE.
The CTCP cannot send the CLEAR CONFIRMATION command or the RESET CONFIRMATION command.	The CTCP must send the CLEAR CONFIRMATION and RESET CONFIRMATION commands.

Application Programming Requirements

When using either GATE or DATE, you must satisfy the following requirements for an interface between the CTCP and a host application program:

- Virtual circuit establishment
- Virtual circuit termination.

Additional requirements specific to GATE and DATE are described in "GATE Function of X.25 NPSI" on page 47 and in Chapter 4, "Programming of a DATE CTCP."

Virtual Circuit Establishment

Virtual circuit management is performed on the CTCP LU to MCH LU session when GATE and DATE are used. When using fast connect, the virtual circuit is established on the CTCP LU to VC LU session. For these reasons, establishment of the control session must be the first step in virtual circuit establishment. The control session can be established using the procedures described in "Session Initiation" on page 11.

Call-In: When a call comes in to the application program from a remote DTE, the Incoming Call packet is translated by X.25 NPSI into a CALL REQUEST command. This command is then sent to the CTCP.

The CTCP determines packet length and window size, but when using GATE or fast connect, X.25 NPSI determines the LLC type by using CUD0 or subaddressing.

The CTCP also determines the LLC type when using the DATE function.

The packet length, window size, and LLC type (for DATE) are communicated back to X.25 NPSI through a CALL ACCEPTED command. X.25 NPSI then converts this command to a Call Accepted packet and forwards it to the PSDN.

The CTCP is responsible for all required facility negotiations. In particular, the CTCP can negotiate the packet and window sizes with the PSDN. X.25 NPSI does not participate in these negotiations.

If the CTCP cannot accommodate the packet and window sizes specified by the PSDN, the CTCP must clear the call. When the negotiations are successful, the CTCP passes the negotiated packet and window sizes to X.25 NPSI in the CALL ACCEPTED command.

Call-Out: When the user application or the CTCP operator requests that the CTCP connect with a call-out destination, the operation performed is similar to the one executed for call-in. The CTCP builds a corresponding CALL REQUEST command, including the Call Request packet that is sent to the DCE. The CTCP also builds parameters that include:

- Window size
- Packet size.

The CALL REQUEST command is received by the GATE or DATE function in X.25 NPSI. X.25 NPSI then selects an available virtual circuit, updates the associated control blocks with the provided parameters, and sends the Call Request packet to the PSDN.

If an answer is not received for the Call Request packet within the time limit set for the T21 timer, the CTCP can retry the call by resending the same CALL REQUEST command to X.25 NPSI. In this case, X.25 NPSI keeps the same virtual circuit and resends the Call Request packet. It is up to the CTCP to decide how many retries must be performed and what must be done if the retries are unsuccessful.

When the remote DTE accepts the Call Request, a Call Connected packet is sent to X.25 NPSI. This Call Connected packet is forwarded to the CTCP as a CALL CONNECTED command.

Logon Processing: If X.25 NPSI builds the logon, it does so when the call setup is completed. This logon request is issued to the access method after the ACTLU command has been processed for the LU associated with the virtual circuit for the duration of the call. The input queue for that virtual circuit is locked. All incoming data is queued until SDT (start data traffic) is received from the user application. At this point, all data traffic is passed to the application.

For V3R3 only: A keyword has been added to the X25.MCH statement that gives X.25 NPSI the option to participate in packet and window size negotiation. If you specify INTFAC=NO on the X25.MCH statement, X.25 NPSI does not participate in packet or window size negotiation. If you specify INTFAC=YES on the X25.MCH statement, X.25 NPSI interprets and uses the packet and window sizes that are contained in the Call Connected packet under GATE and DATE.

Virtual Circuit Termination

A virtual circuit can be deactivated in any of the following ways:

- The remote DTE requests the session termination.

  In this instance, a Clear Request packet is received from the PSDN, and X.25 NPSI transfers this information to the CTCP as a CLEAR REQUEST command.

- The user application program can initiate session termination by issuing the CLSDST macroinstruction.

  In this case, X.25 NPSI receives an ABCONN request from VTAM.

- The user application program can notify the CTCP to disconnect the virtual circuit.

  When the CTCP receives this request, the CTCP builds a CLEAR command, including the cause and diagnostic bytes as determined by the CTCP. This command is sent to X.25 NPSI. X.25 NPSI then sends a Clear Request packet to the PSDN. When the Clear Confirmation packet is received, X.25 NPSI transfers it to the CTCP in the CLEAR CONFIRMATION command.

- The network operator can terminate the session by issuing a V NET,INACT command for the VC PU or the VC LU.

  When this command is processed, the status of the virtual circuit is set to INACT (inactive). This status change is communicated to the host and the CTCP. The SNA and X.25 resources are disassociated at this point.

Chapter 2. Programming Using PCNE and PAD Support

Chapter 2. Programming Using PCNE and PAD Support

This chapter explains how to program virtual circuits when you use either of the following:

- The protocol converter for non-SNA equipment (PCNE)
- Integrated and transparent packet assembler/disassembler (PAD) support functions of X.25 NPSI.

PCNE Function of X.25 NPSI

The host application programs exchange data with the LU simulator using only LU type 1 protocol in half-duplex contention mode or half-duplex flip-flop mode. The LU simulator provides buffering of inbound data for asynchronous data flow. In the case of exception response mode and contention mode without brackets, buffering of outbound data is also provided by X.25 NPSI.

When operating in half-duplex contention mode, the LU simulator immediately sends incoming data to the host, unless it is awaiting a response from the host. If the LU simulator is waiting for a definite response, and the host sends data to the LU simulator instead, X.25 NPSI sends an SNA sense code of X'081B' to the host application program.

When operating in half-duplex flip-flop mode, the LU simulator must wait to be in the correct state before transmitting or receiving data with the host. This process is accomplished by the change direction (CD) bit in the SNA request/response header (RH). If the host sends data to X.25 NPSI and it is not allowed because X.25 NPSI has the CD bit, the data is rejected with a sense code of X'081B'.

For V3R3 only: X.25 NPSI supports RU chaining for the PCNE function.

Figure 2 on page 22 shows the data flow for a network configuration that uses PCNE.

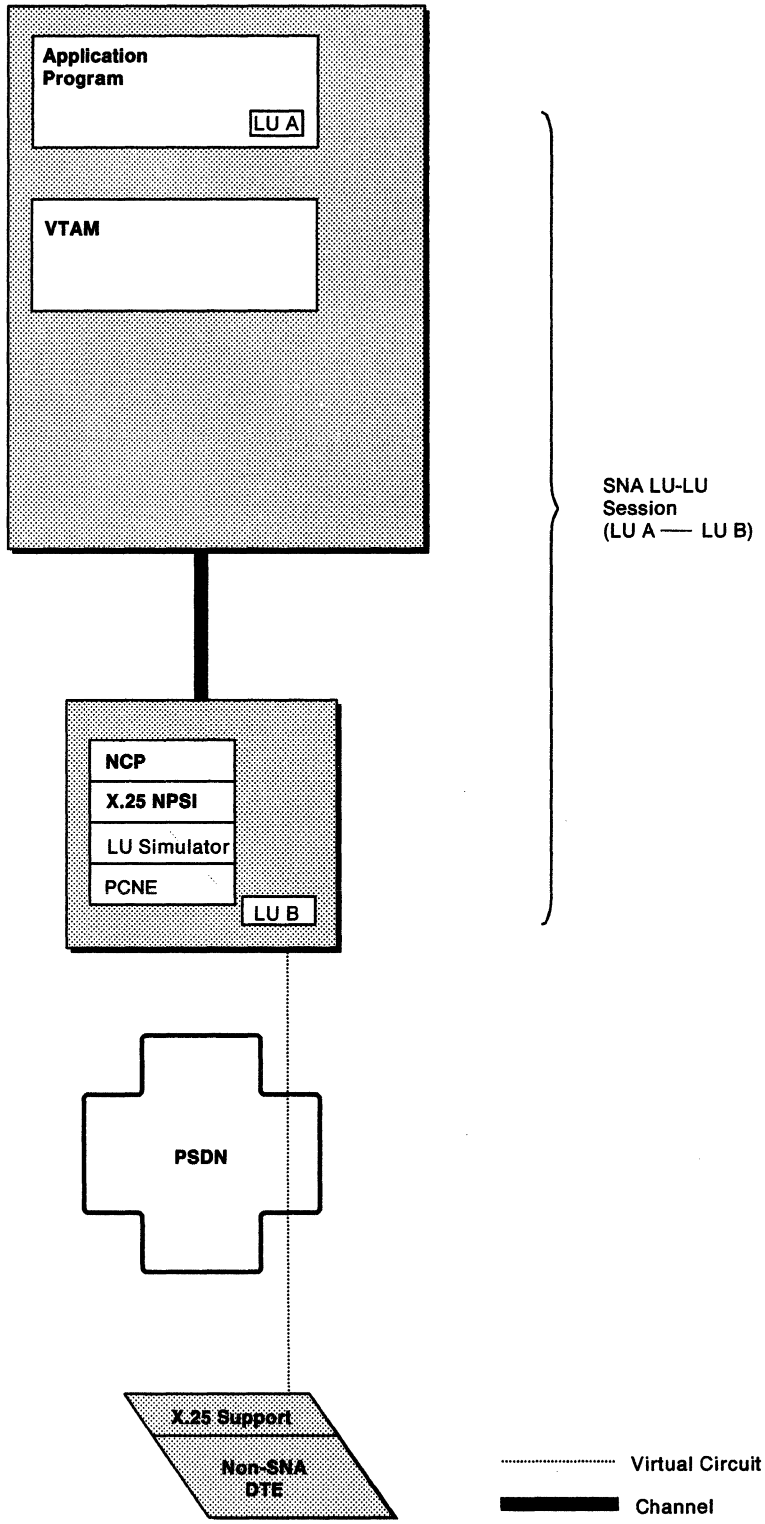


Figure 2. SNA Host Node to Non-SNA DTE (PCNE)

Delivery Confirmation Bit (V3R2 and Previous Releases)

When a type 0 virtual circuit is used, X.25 NPSI can support an end-to-end delivery confirmation service to verify that an outbound RU, which has been converted to X.25 data, has reached its destination. The LU simulator does this through a bit in the packet header known as the *delivery confirmation bit* (D bit). X.25 NPSI provides three levels of D-bit support.

1. The secondary protocol for the LU simulator supports *only* definite response when:

   On an outbound request, if an OIC RU is presented with a definite response request, the LU simulator converts this request to a packet with the D bit set to 1. Upon receipt of a corresponding Receive Ready (RR) packet, the LU simulator constructs a definite response and presents it to the request sender (point of origin).

   Conversely, on an inbound request, if X.25 NPSI receives a packet with the D bit set to 1, the LU simulator sets the definite response indicator on in the request/response header (RH) of the message sent to the host application program. When the response is returned, X.25 NPSI responds to the remote DTE's request for delivery confirmation by sending an RR packet to the network.

2. The secondary protocol for the LU simulator allows *either* definite response or exception response when:

   X.25 NPSI sets the D bit to 1 only when the outbound PIU requests a definite response.

   A data packet received with the D bit set to 1 is mapped into a PIU to the host requesting definite response. A data packet received with the D bit set to 0 is mapped into a PIU requesting exception response.

3. The secondary protocol for the LU simulator supports *only* exception response.

   In this case, an RR packet is returned to the network as soon as X.25 NPSI receives a data packet with the D-bit set to 1. The D bit is used to inform the remote DTE that the packet was received by X.25 NPSI.

You can combine the second and third levels of support (one at each end of the connection) to form an abbreviated level of D-bit support. This abbreviated level can be used in PCNE-to-PCNE sessions in which X.25 NPSI is running on both ends of the PSDN. In this situation, D bit use can help you improve response time and avoid deadlocks.

Notes:

1. The end-to-end delivery confirmation bit is supported only in the case of half-duplex end-to-end data flow; that is, the incoming and outgoing data flows need to be correlated. The D-bit is not supported with uncorrelated (duplex-like) data flow.

2. The use of the D bit should not be regarded as providing equivalent function to the SNA response protocol. SNA allows an end-to-end indication of successful, positive response or unsuccessful, negative response delivery of a request unit. The X.25 D bit indicates delivery to the destination, and whether the delivery was successful. If the application responds negatively to an RU that was received with the D bit set to 1, X.25 NPSI INOPs the virtual circuit and sends a RESET to the sender. This RESET is followed by a CLEAR for an SVC, where CRAFTRC=YES was coded on the corresponding X25.MCH statement. Because protocols that are at a higher level than those defined by Recommendation X.25

may be needed to obtain a function equivalent to that provided by SNA response protocols, the X.25 D bit may serve no useful purpose.

Delivery Confirmation Bit (V3R3 Only)

When a type 0 virtual circuit is used, X.25 NPSI can support an end-to-end delivery confirmation service to verify that an outbound RU, which has been converted to X.25 data, has reached its destination. The LU simulator does this through a bit in the packet header known as the *delivery confirmation bit* (D bit). D-bit support differs in V3R3 from the support provided in previous releases, because X.25 NPSI V3R3 supports RU chaining for non-SNA connections. You can specify whether you wish to use RU chaining by coding the MBITCHN keyword on the MCH statement at SYSGEN. If MBITCHN=YES is specified, RU chaining is supported. The following describes the levels of D-bit support provided as it differs from V3R2 when RU chaining is supported:

1. The secondary protocol for the LU simulator supports *only* definite response when:

   An RU is presented with a definite response requested, the LU simulator converts this request to a complete packet sequence with the D bit of the last packet set to 1. Upon receipt of a Receive Ready (RR) packet, which acknowledges the last packet of the M bit sequence, the LU simulator constructs a definite response and presents it to the request sender (point of origin).

   Conversely, if X.25 NPSI receives a packet with the D bit set to 1 and the M bit set to 0, the LU simulator sets the definite response indicator on in the request/response header (RH) of the message sent to the host application program. When the response is returned, X.25 NPSI responds to the remote DTE's request for delivery confirmation by sending an RR packet to the network.

2. The secondary protocol for the LU simulator allows *either* definite response or exception response and functions identically to V3R2.

Notes:

1. The X.25 D bit indicates delivery to the destination, and whether the delivery was successful. If the application responds negatively to an RU that was received with the D bit set to 1, X.25 NPSI resets the VC with a diagnostic code of X'B3'. The sending of the RESET is dependent upon the setting of the RESETINO keyword on the X.25 NET statement.

2. When X.25 NPSI issues a Call Request packet with the D bit set to 1, but receives a Call Connected packet with the D bit set to 0, the data packets should have the D bit set to 0.

   However, X.25 NPSI still accepts data packets with the D bit set to 1, without resetting the logical channel. This avoids migration problems with DTEs that are not aware of how the D-bit procedure is used during call establishment.

X.25 NPSI Support for RU Chaining (V3R3 only)

This function allows non-SNA equipment to use the SNA RU chaining function when communicating with an SNA application through X.25 NPSI. This function can be used for outbound and inbound flows using a generation parameter.

The chain of PIUs is composed of:

- First-in-chain (FIC)
- Middle-in-chain (MIC)
- Last-in-chain (LIC)
- Only-in-chain (OIC).

Previously, when receiving an RU, X.25 NPSI (under LLC 0, 4, and 5) acts as if it were an OIC RU. That is, after removing the TH and RH fields, it sends a complete packet sequence (CPS). These packets are linked together with an M bit by setting the M bit to 1 in all but the last packet. On inbound flow, when receiving a CPS, X.25 NPSI builds an OIC RU.

With X.25 NPSI V3R3, if MBITCHN=YES, X.25 NPSI sends the entire chain of PIUs as either one CPS (if DBIT=NO) or a CPS series, such as an MBS (if DBIT=YES).

On inbound flow, X.25 NPSI maps each CPS on one or more SNA elements of the chain.

Figure 3 on page 26 and Figure 4 on page 27 show examples of outbound flow.

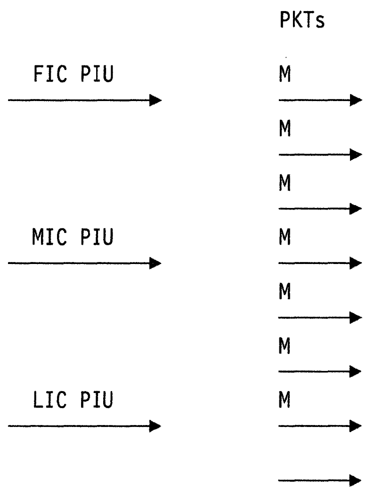
Figure 5 on page 28 and Figure 6 on page 29 show examples of inbound flow.

Outbound Flow

On Outbound Flow:

The following figure illustrates the conversion of each chain of RUs into one CPS when the CPS packets are linked together with the M bit.

Case 1



Case 2

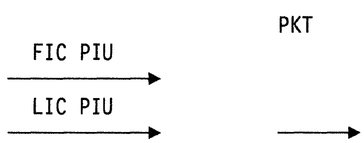


Figure 3. X.25 NPSI RU Chaining Outbound Flow (MBITCHN=YES, DBIT=NO)

Note: Each packet sent into the network with the M bit on must be a full data packet. If the last packet from an FIC or MIC RU is not full, X.25 NPSI must wait for the next RU to fill the non-full packet with the data from this RU.

The following figure illustrates the conversion of each RU chain into a CPS chain linked together with the M and D bits, if D bit support is allowed.

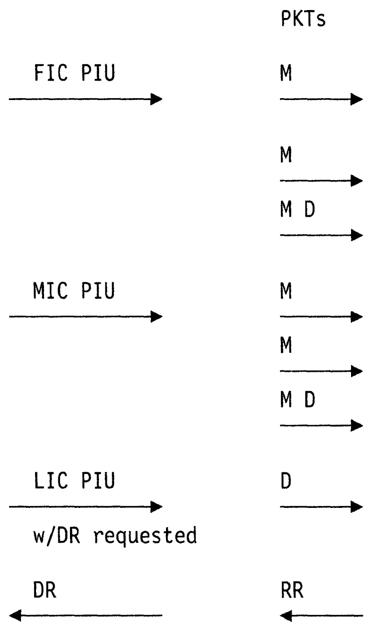


Figure 4. X.25 NPSI RU Chaining Outbound Flow (MBITCHN=YES, DBIT=YES)

Note: If Definite Response (DR) is allowed in bind, RR is immediately sent by X.25 NPSI.

Inbound Flow

On Inbound Flow:

The following figure illustrates the conversion of a CPS into an SNA RU chain.

Packets containing the M bit are accumulated until MAXRU of the BIND command is reached. The PIU is then forwarded to the host. Otherwise, the size of the accumulated data needed to build a PIU is determined by the length of the CPS received.

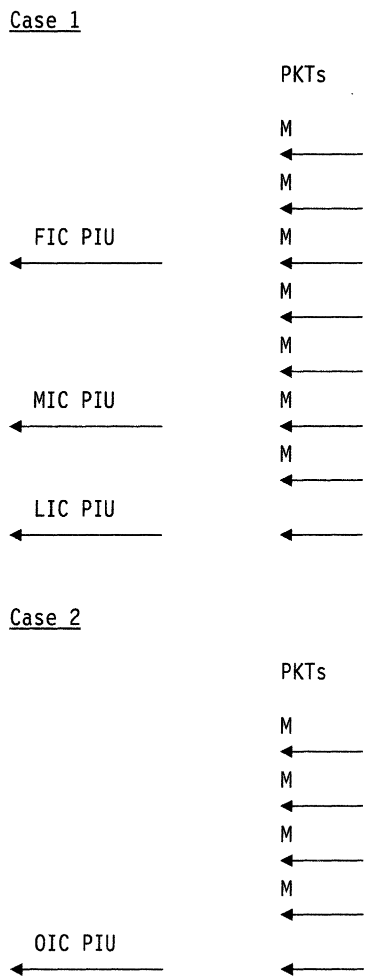


Figure 5. X.25 NPSI RU Chaining Inbound Flow (MBITCHN=YES, DBIT=NO)

The following figure illustrates the conversion of each CPS chain linked together with the D bit into an RU chain, if D-bit support is allowed. The size of the PIU forwarded to the host is determined by either the length of the CPS received, or the size of the X25.MAXRU of the BUILD statement, whichever is smaller.

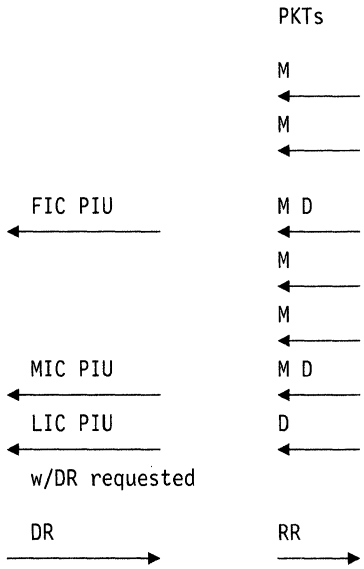


Figure 6. X.25 NPSI RU Chaining Inbound Flow (MBITCHN= YES, DBIT= YES)

Note: If DR is allowed in Bind, RR is sent immediately by X.25 NPSI.

Establishing a Session over a PCNE-to-PCNE Connection

In addition to an SNA-to-SNA session link, two application programs can establish a PCNE-to-PCNE session link across a type 0 virtual circuit. This connection can be between any two programs.

The process of session initiation in a PCNE-to-PCNE environment can be the following:

1. APPL 1 initiates a session with the LU of the virtual circuit that communicates with the other host.
2. APPL 1 constructs a LOGON message acceptable to APPL 2, and sends it to APPL 2.
3. APPL 1 and APPL 2 exchange data using either the half-duplex flip-flop protocol or the half-duplex contention protocol.
4. APPL 1 issues a CLSDST to terminate the session.

In a PCNE-to-PCNE session, the initiator of the session can construct a LOGON message to cause the LOGON exit routine to activate in the partner LU. The LOGON message must be formatted as follows:

```
LOGON APPLID(aaaaaaaa) DATA(dddddddd)
```

where:

aaaaaaaa	Is the LU partner name (APPLID).
dddddddd	Is any appropriate user data that is to be sent to the partner LU.

Figure 7 on page 31 is an example of a session that can be established. It shows the data flow for a session that uses PCNE.

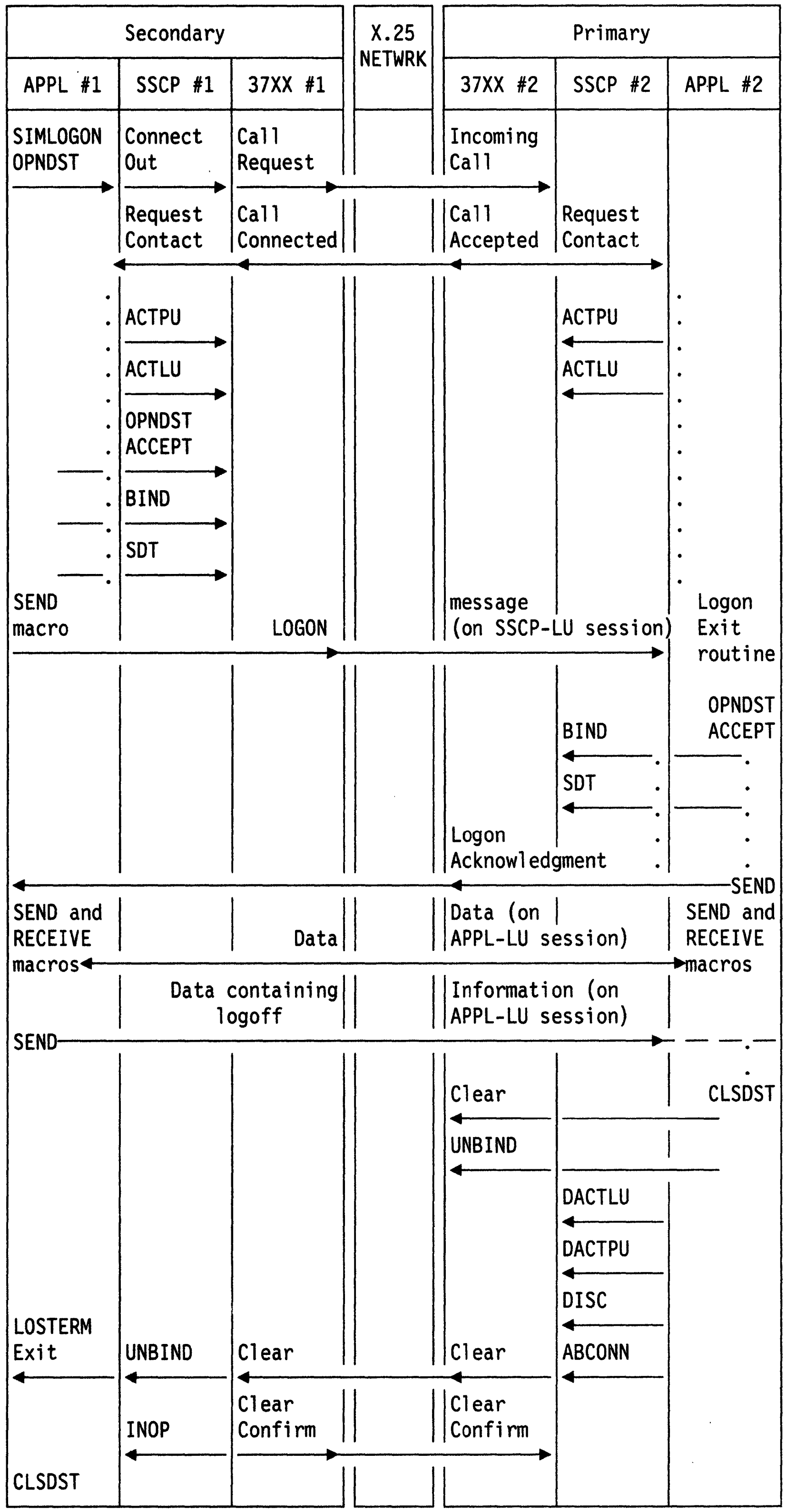


Figure 7. APPL-to-APPL Session through PCNE-to-PCNE

PCNE-to-PCNE Considerations

PCNE-to-PCNE sessions require special planning. You must solve some PCNE-specific problems when choosing the method of synchronization between endpoints and the level of delivery confirmation to be used for a PCNE-to-PCNE session.

Application-Level Synchronization

The session between the two PCNE endpoints must be controlled carefully. The connection is not controlled by any X.25 or SNA flow control. In addition, the line between the endpoints is a duplex line. Because the connection is able to support communication in both directions from the host *simultaneously*, the session can lose synchronization easily.

If synchronization is required, the PCNE applications must create their own synchronization method. Methods for creating synchronization include (but are not limited to):

- The inclusion of special data to signify the passage of the flow direction
- The use of half-duplex flip-flop protocol.

PAD Support Functions of X.25 NPSI

X.25 NPSI provides two levels of support for a packet assembler/disassembler (PAD):

- Integrated PAD support

  If the PAD complies with Recommendations X.3, X.28, and X.29, you can use the integrated PAD support function of X.25 NPSI to interface with the PAD. Integrated PAD support features are based on those provided by PCNE.

- Transparent PAD support

  If you need to use the facilities of the PAD, as defined by Recommendations X.3, X.28, and X.29, other than those provided by integrated PAD support, or the PAD service does not follow Recommendation X.29, you can use X.25 NPSI's transparent PAD function. Transparent PAD support provides an interface between host application programs and any type of PAD.

  When the application wants to use interrupt packets to control the flow direction or the Reset packet to monitor the operation of the network, transparent PAD can be used to control an X.25 remote DTE that does not use a PAD.

Note: The X.25 NPSI PAD support should not be used for connections through SDLC PADs. Such connections are supported by the QLLC (or LLC 3) or through the SPNQLLC specification in the X.25 MCH statement.

Integrated PAD Support

A host application program communicating through the integrated PAD function of X.25 NPSI is largely isolated from the intricacies of communication with the remote PAD. The application program operates in its normal fashion without regard for the interface at the remote end. Integrated PAD provides support for the extra processing that is required for operation of the PAD. The application program is generally not involved in sending or receiving PAD commands or responses.

The considerations for running with the LU simulator apply in the case of integrated PAD support. Integrated PAD code within X.25 NPSI interacts with the remote PAD according to CCITT Recommendation X.29. X.25 NPSI supports RU chaining for the integrated PAD function.

PAD Parameters

PAD parameters 1, 7, and 8 are of particular interest in an X.25 NPSI environment. When under the control of the integrated PAD support function, X.25 NPSI sends a PAD message to the PAD at ACTLU for SVC and at BIND for PVC. X.25 NPSI uses this PAD message to set the following PAD parameter values:

PAD Parameter	Default Setting (Packed Decimal)
1	0
7	21
8	0

For more information about X.25 NPSI PAD parameters, see *X.25 NPSI Planning and Installation*. For more information about X.25 NPSI PAD parameter settings, see the customization section of *X.25 NPSI Diagnostics, Customization, and Tuning*.

PAD Parameters (V3R3 Only)

X.25 NPSI V3R3 allows you to specify, at system generation, one or more strings of PAD parameters using the PADPARM keyword on the X25.PAD statement. The PADINDX keyword on the X25.MCH statement provides an index, which points to a selected parameter string.

The PAD parameter string is passed to the PAD at ACTLU for SVC and at BIND for PVC.

For more information about PAD parameter generation, see *X.25 NPSI Planning and Installation*.

SIGNAL/BREAK Processing (V3R2 and Previous Releases)

When X.25 NPSI receives an Interrupt packet from the PAD, X.25 NPSI sends a SIGNAL command to the host application program. The application program should stop sending data to the remote terminal.

When the BREAK key is pressed at the terminal, X.25 NPSI sends the host application program a null RU with the change direction (CD) flag set on. This CD is sent *only* if:

- The LU simulated by X.25 NPSI is in transmit state.
- The session is operating in half-duplex flip-flop mode.

A qualified packet is sent to restore data delivery, because data delivery is inhibited by the PAD when it sends an indication of BREAK message to X.25 NPSI. This occurs if PAD parameter 7 is set to 21 decimal.

If PAD parameter 7 is set to 21 decimal, the PAD discards the outbound data that is in its buffer when the PAD receives a BREAK. At this point, X.25 NPSI does not discard any inbound or outbound data.

The host application program can cause X.25 NPSI to send an indication of BREAK message to the remote PAD by sending an SNA SIGNAL command to the LU associated with the virtual circuit.

After sending the indication of BREAK message to the PAD, X.25 NPSI sends a null RU with a change of direction to the host, if the following conditions are met:

- The simulated LU uses half-duplex flip-flop protocol.
- The simulated LU is in the transmit state.

This allows the host to send data to the terminal, after having sent out the SIGNAL (flow direction reversal).

Note: Integrated PAD support and GATE can be on the same MCH as long as USS message 10 is not being sent to the remote PAD and its attached devices. X.25 NPSI V3R2 does not allow the integrated PAD option on an MCH controlled by a DATE CTCP.

SIGNAL/BREAK Processing (V3R3 Only)

When X.25 NPSI receives a SIGNAL from the host application program, the type of action taken is dependent on the setting of the PAD parameter 7.

When the PAD parameter setting is 0705 or 0721, X.25 NPSI converts a SIGNAL from the host application program to an indication of BREAK message that is sent to the PAD.

Additionally, if the PAD parameter setting is 0702, X.25 NPSI sends a Reset Indication message to the virtual circuit. If the PAD parameter setting is 0701, X.25 NPSI sends an Interrupt Indication message to the virtual circuit.

When the BREAK key is pressed at the terminal, X.25 NPSI's actions are dependent on the setting of PAD parameter 7.

If the PAD parameter setting is 0701, X.25 NPSI receives an Interrupt packet from the PAD. X.25 NPSI then sends a SIGNAL to the host application and an Interrupt Confirmation packet to the PAD. This sequence causes the application to discontinue output to the terminal until a change of direction (CD) flag is received.

If the PAD parameter setting is 0702, X.25 NPSI receives a Reset packet from the PAD, and then sends a Reset Confirmation packet to the PAD. If the cause and diagnostic code in the Reset packet matches one of the values in PADBRKCD, X.25 NPSI sends a SIGNAL to the host application. The PADBRKCD keyword is coded on the MCH statement. The PADBRKCD keyword is coded on the X25.MCH statement. Additionally, X.25 NPSI purges its input and output queues. This sequence causes the application to discontinue output to the terminal.

If the PAD parameter setting is 0705, X.25 NPSI receives an Interrupt packet and indication of BREAK message from the PAD. X.25 NPSI then sends a SIGNAL, which is built on the Break packet, to the host and an Interrupt Confirmation packet to the PAD. When a DATE CTCP is used, a SIGNAL is sent to the application on the data session. If the BREAK was entered during output data, this sequence causes the application to discontinue output to the terminal until a CD flag is received. If the BREAK was entered during input data, this sequence causes the application to delete the characters that were entered just before the SIGNAL arrived.

If the PAD parameter setting is 0708 (which leads to a DATA ESCAPE PAD state when the BREAK key is used), X.25 NPSI sends a SIGNAL, which is built on a Reset or an Interrupt packet, to the host application. If the cause and diagnostic code in the RESET packet matches one of the values in PADBRKCD, X.25 NPSI sends a SIGNAL to the host application. If the BREAK was entered during data input, the host application purges the last block of data that was entered. If the BREAK was entered during output data, the host application discontinues output until a CD flag is received.

The sequence of events that occurs when the PAD parameter setting is 0721 remains as in previous X.25 NPSI releases, except that the SIGNAL that is sent to the host application contains PAD setting information.

Note: Integrated PAD, PCNE, and GATE devices can be on the same MCH if you code different IDBLK numbers for the IDBLKP, IDBLKC, and IDBLKG keywords. On a DATE MCH, the SIGNAL/BREAK processing is the same for an Integrated PAD.

SHUTD Considerations

When a terminal or printer is connected to a host through a PAD, it is common for the device to need attachment to more than one VTAM application program. The device might need to move between application programs or, as is true of a printer, the device itself might be passed between application programs. X.25 NPSI provides two methods for devices to be shared by more than one application program. To specify the option you want to use, you must code the SHUTD keyword of the X25.MCH statement in one of the following ways:

- `SHUTD=INVCLR`

  Code the INVCLR parameter for the SHUTD keyword when you want the device to be disconnected from one application program before being connected to another.

  When an SNA SHUTD command is sent by the application and the SHUTD keyword is coded as INVCLR, X.25 NPSI interprets the SHUTD command and converts it into one of the following:

  - An X.25 Invitation to Clear packet, if the virtual circuit is switched
  - A Reset packet, if the circuit is permanent.

  X.25 NPSI immediately returns to the host a positive response to the SHUTD command.

  The PSDN responds to the packet by returning either a Clear Indication packet (for SVCs) or a Reset Confirmation packet (for PVCs). These packets return the virtual circuit to its initial status, breaking the connection to the host.

  Reception of these packets by X.25 NPSI results in an SNA SHUTC command being sent to the host. In response to this command, the application program sends an UNBIND command to the LU simulator function of X.25 NPSI, which unbinds the SNA session.

  Once the session is broken, the device must either redial the host, if the connection was through an SVC, or logon to the new application program, if the connection was by way of a PVC.

  Figure 8 on page 36 shows the data flow of the SNA commands and the X.25 packets used to switch attachment of a device from one application program to another when the INVCLR parameter is specified.

- `SHUTD=NOINVCLR`

  Code the NOINVCLR parameter for the SHUTD keyword when you want to keep the X.25 connection active on SHUTD. In that case, no Invitation to Clear packet or Reset packet is issued to the PSDN. With this option, the virtual circuit is not cleared, and the device can be passed between application programs.

  Figure 9 on page 37 shows the data flow when the NOINVCLR parameter is specified. In the case of a PVC, if the SNA UNBIND command includes an indication that an SNA BIND command is forthcoming, a Set PAD message is not sent with the following SNA BIND command.

For V3R3 only: SHUTD = NOINVCLR is mandatory when coding PAD = INTEG and GATE = DEDICAT on the same MCH.

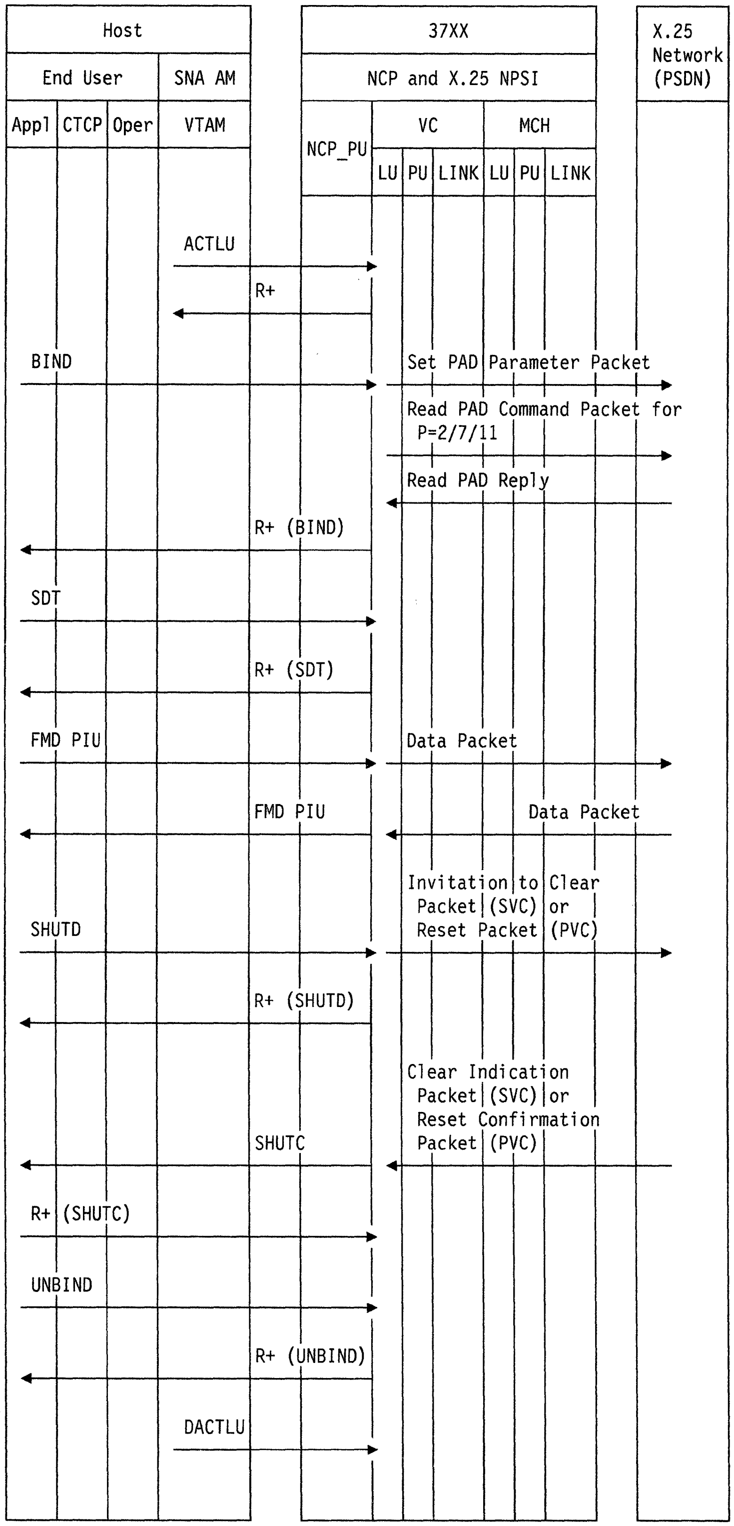


Figure 8. Data Flow with SHUTD = INVCLR Option

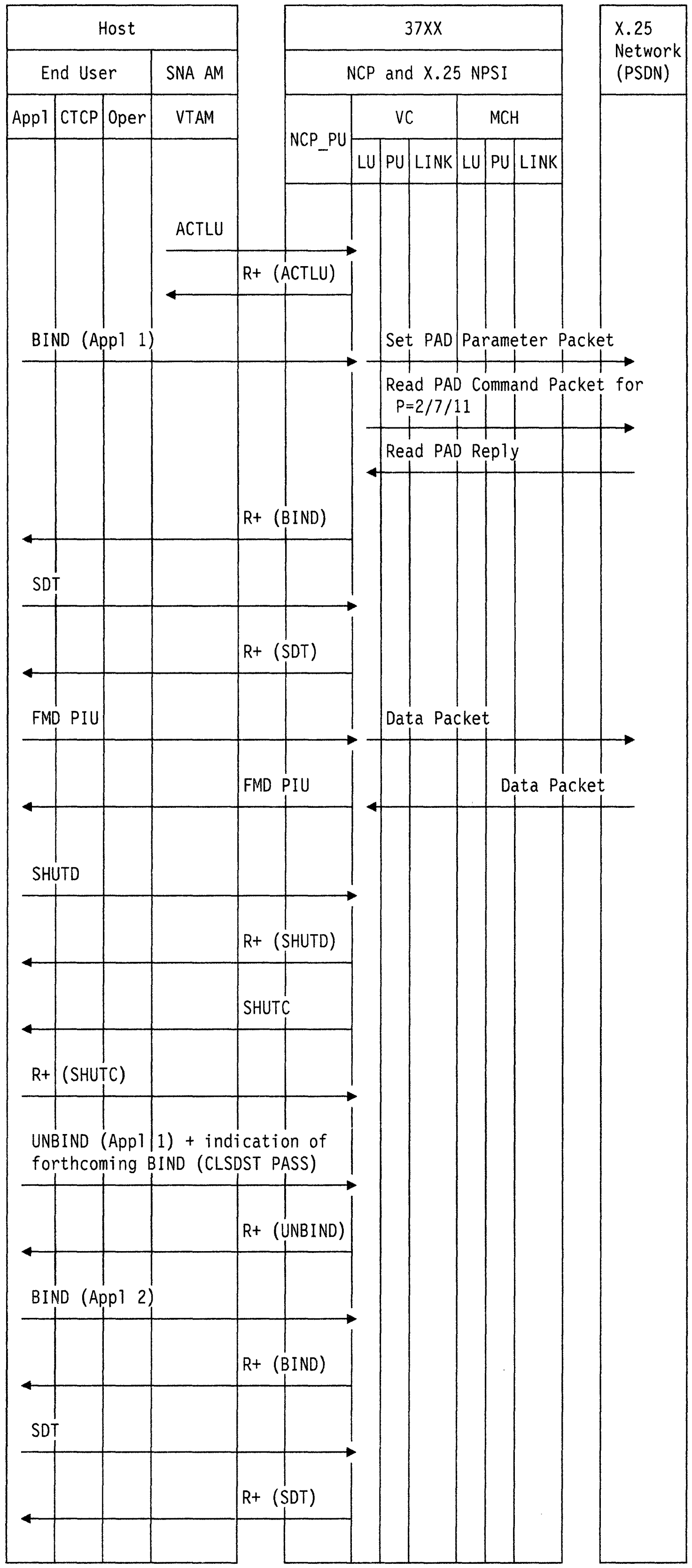


Figure 9. Data Flow with SHUTD=NOINVCLR Option

Application Program Support for Password Protection

If the application program wants to use the password protection feature of X.25 NPSI, it can use the enable presentation (ENP) and inhibit presentation (INP) characters to define when a field is not to be echoed to the terminal. The hexadecimal values for these control characters are:

X'24' Inhibit presentation (INP)
X'14' Enable presentation (ENP).

Note: If X.25 NPSI does not translate EBCDIC to ASCII, the control character used for inhibit presentation is X'12' rather than X'24', and the overstrike message is translated to ASCII EVEN code. If this default value must be changed, see the customization section of *X.25 NPSI Diagnosis, Customization, and Tuning*.

To use the password protection function, the application program places an INP character at the end of the data stream prompting the protected information. The protected information is interpreted and converted into the appropriate PAD commands to disable display at the device. Disabling display at the device can be accomplished through one of the following methods:

- The inhibition of echoing data back to the device, if the device is operating with echoplexing
- The transmission of an 8-character blackout string, if the device is not operating with echoplexing.

Note: Both of the options for this function work with typewriter-like devices. However, the inhibition of echoing data back to a device works only with video display terminals.

X.25 NPSI determines whether to inhibit echo or to transmit a blackout message to support password protection. This determination is made by interrogating the setting of PAD parameter 2 (echo). If the echo mode is ON, the echo facility is used to inhibit the display. Otherwise, the blackout message is used to hide the printing.

Upon receiving and processing the response, the host application program starts the next output buffer with the ENP character to initiate the redisplaying of characters on the device.

X.25 NPSI is responsible for all processing required for the password protection function. To implement this function, you must specify PWPROT=YES on the X25.MCH statement. If X.25 NPSI processing of the password protection function does not meet your requirements, you can use the GATE, DATE, or transparent PAD function rather than the integrated PAD support function.

Transparent PAD Support

Transparent PAD support is provided to allow an application program to control the functions of a packet assembler/disassembler (PAD). When a program uses transparent PAD, the first byte of the data passed between the application program and X.25 NPSI is used as a packet type indicator. This byte notifies X.25 NPSI and the host application program of the type of packet and allows each end to process the packet accordingly.

Uses of Transparent PAD Support

Transparent PAD is used for two reasons. The first reason is to permit you to use facilities of the PAD, as defined by Recommendations X.3, X.28, and X.29, other than those provided by integrated PAD support. Using transparent PAD support, a host application program can control its interaction with a remote DTE connected through a PAD. This higher level of control is required by some application programs and users of X.25 NPSI.

The second reason to use transparent PAD is to provide control of PAD services, which *do not* follow Recommendation X.29. Transparent PAD support allows any type of PAD to be supported and controlled by an application program through X.25 NPSI. An example of such a X.3 PAD service is a bisynchronous PAD. This type does not support the X.3, X.28, and X.29 interface recommendations and often uses a vendor-specific interface that must be provided for by the host application program.

Transparent PAD support can be used on a physical circuit defined for any mode of operation; that is, the X25.MCH statement can specify GATE=NO, GATE=GENERAL, or GATE=DEDICAT. Even when MBITCHN=YES is coded on the X25.MCH statement, the SNA RU chaining function is not supported. On inbound flow, packets containing the M bit are accumulated until MAXRU of the BIND command is reached. The PIU is then forwarded to the host. Otherwise, the size of the accumulated data needed to build a PIU is determined by the length of the CPS received. On outbound flow, X.25 NPSI converts each FIC, MIC, or LIC PIU into an OIC, and then builds and sends a corresponding CPS.

Figure 10 on page 40 shows the data flow in a network configuration using transparent PAD support.

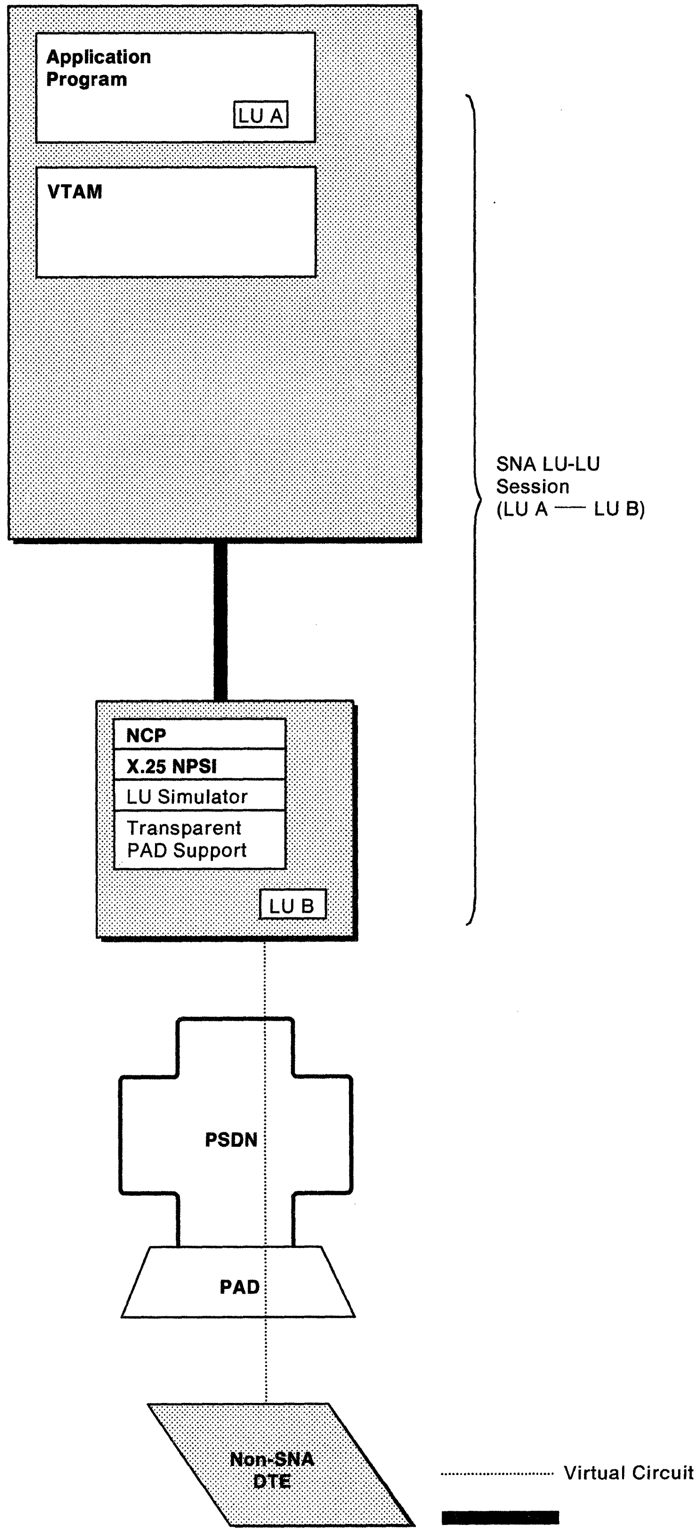


Figure 10. SNA Host Node to Non-SNA DTE (Transparent PAD)

Programming Requirements for Transparent PAD

The transparent PAD support function is designed for application programs that need to control the remote PAD by means of commands exchanged between the application program and X.25 NPSI.

For application programs using transparent PAD support, the virtual circuit setup and takedown is identical to that for the type 0 virtual circuit. However, the contents of the following types of packets are sent from or routed to the application program over the LU-LU session between the application program LU and the virtual circuit LU.

- Data packets
- Qualified data packets
- Interrupt packets
- Reset packets.

Commands and information for PAD control are contained in qualified data packets.

An application program must use the first byte (byte 0) of the RU to specify the packet type indicator, which denotes the type of data contained in the packet. This byte is used in communication between the application program and the transparent PAD function within X.25 NPSI. The first byte of each RU must contain one of the following values:

RU Byte 0	Packet Type
X'00'	Data without Q bit
X'02'	Data with Q bit
X'1B'	Reset
X'1F'	Reset Confirmation
X'23'	Interrupt
X'27'	Interrupt Confirmation.

Qualified data packets (that is, data packets with the Q bit set to 1) are used to exchange information between the application program and the remote PAD. Data packets with the Q bit set to 0 are used to exchange information between the application program and the remote DTE.

For Reset packets, the second and third bytes of the RU contain cause and diagnostic values. When Reset packets are exchanged, the only action of X.25 NPSI is to set the P(R) and P(S) counters to 0. The application is responsible for issuing a CLSDST if required, or to perform a checkpoint/restart with the remote DTE, if this was planned in the convention between the application and the remote DTE.

Upon receipt of a Reset Indication packet from the network, the application program must send the Reset Confirmation to X.25 NPSI. X.25 NPSI then forwards the Reset Confirmation to the network.

For Interrupt packets, the second byte of the RU contains the interrupt cause (usually set to 0). For Interrupt packets from the PAD, the application program must initiate the transmission of the Interrupt Confirmation packet.

X.25 NPSI performs normal packetization and recombining through the M bit in packet headers.

Depending on the value of the TRAN keyword coded on the X25.MCH statement, X.25 NPSI does or does not translate between EBCDIC and International Alphabet

Number 5. When translation is requested, data beyond byte 0 (the first byte) in RUs flowing on transparent PAD sessions is translated by X.25 NPSI in the case of unqualified data packets.

The following describes the format for each transparent PAD RU.

Data without Q Bit: The data packet without a Q bit is used to signify that the associated data is only user data. There is no special significance to the attached data. The format of this packet is:

Byte 0	X'00'
Bytes 1 through n	Variable-size field containing the data packet.

Data with Q Bit: The data packet with a Q bit is used for data that is sent to or from the remote PAD. All PAD commands are contained in data that start with this indicator.

To instruct the PAD to perform a function, place the value X'02' in byte 0 of the data and place the PAD command directly behind it. X.25 NPSI recognizes your desire to send a command to the PAD. It responds by placing the PAD command into the packet and by setting the Q bit to 1. The packet is then sent to the remote PAD. The PAD performs the desired function, and the host application program continues with its required processing.

The format of the data packet with a Q bit is:

Byte 0	X'02'
Bytes 1 through n	Variable-size field containing the data packet.

Qualified data can be used to perform any type of PAD command. The valid X.29 PAD commands are:

Code	**Command**
X'00'	Parameter Indication
X'01'	Invitation to Clear
X'02'	Set PAD
X'03'	Indication of Break
X'04'	Read PAD
X'05'	Error
X'06'	Set and Read PAD
X'07'	Reselection.

For detailed information on the coding requirements for these PAD commands, see *X.25 NPSI Diagnosis, Customization, and Tuning*. This book contains the formats of all commands and the status and error codes.

Reset: A Reset packet is used to reset the PAD. When the PAD is reset, the communication between X.25 NPSI and the remote DTE is reset to its initial state. This reset can be totally disruptive to the session state; all outstanding packets are discarded. When an application program receives a Reset Indication packet, the application program must send a Reset Confirmation command back to X.25 NPSI. X.25 NPSI then forwards a Reset Confirmation packet to the PAD. If an application sends a Reset command, a Reset Confirmation packet is returned to the application. The format of this packet is:

Byte 0	X'1B'
Bytes 1 and 2	Cause and diagnostic fields
Bytes 3 through n	Variable-size field containing optional user data.

Reset Confirmation: Reset requires the use of a confirmation packet called the Reset Confirmation packet. The format of this packet is:

Byte 0	X'1F'
Bytes 1 through n	Variable-size field containing optional user data.

Interrupt: It is possible for an application program to start the Interrupt procedure by sending an Interrupt Indication command to the PAD through X.25 NPSI. When an Interrupt packet is received by X.25 NPSI, it is passed on to the PAD through the network. The PAD returns an interrupt confirmation to X.25 NPSI, which relays it to the application.

When an application program receives an Interrupt Indication packet, the application program must send an Interrupt Confirmation command back to X.25 NPSI. X.25 NPSI then forwards an Interrupt Confirmation packet to the PAD. The format of this packet is:

Byte 0	X'23'
Byte 1	Interrupt cause
Bytes 2 through n	Variable-size field containing optional data.

Interrupt Confirmation: An application program must be able to handle either sending or receiving an Interrupt Confirmation command. The format of the Interrupt Confirmation packet is:

Byte 0	X'27'
Byte 1	Optional user data.

Recommendation for Transparent PAD

Half-duplex contention is recommended, because unexpected control packets can come from the network at any time.

Chapter 3. Programming a GATE CTCP

Chapter 3. Programming a GATE CTCP

This chapter explains how to program virtual circuits when you use one of the following:

- The GATE function
- The fast connect GATE function of X.25 NPSI
- The X.21 switched connection using GATE.

GATE Function of X.25 NPSI

When you use the GATE function of X.25 NPSI, communication is through the CTCP. All traffic passes through the CTCP. GATE uses the X.25 NPSI LU simulator to operate and communicate with non-SNA LUs. For GATE, you do not need to specify the LLC type in the CALL REQUEST command, because GATE always uses a type 4 LLC.

The CTCP is responsible for handling all commands and data that pass between the application program and X.25 NPSI. GATE resides in X.25 NPSI and communicates with the CTCP, which contains a VTAM application interface. Two types of LU-LU sessions perform the command and data communications.

- The CTCP conducts a *control session* to control the initiation and termination of sessions with remote DTEs. The control session is between the CTCP and the MCH LU.
- A *data session* is created for each virtual circuit for data transmission. The data session is between the CTCP and the LU associated with the virtual circuit for the duration of the call. This LU is called VC LU in this chapter for wording simplicity.

Note: The packet level processor (PLP) timers are started and controlled by X.25 NPSI. The CTCP does not need to start these timers. When the network does not respond to a message, X.25 NPSI sends an INFORMATION REPORT message to the CTCP on the MCH LU session.

RU Chaining Support Using GATE (For V3R3 only)

Even when MBITCHN=YES is coded on the X25.MCH statement, the RU chaining function is not supported. On inbound flow, packets with the M bit are accumulated until MAXRU of the BIND command is reached. Then, the OIC PIUs of MAXRU size are forwarded to the host. Otherwise, the size of the accumulated data needed to build a PIU is determined by the length of the CPS received.

Figure 11 on page 48 shows the data flow for a network configuration that uses GATE.

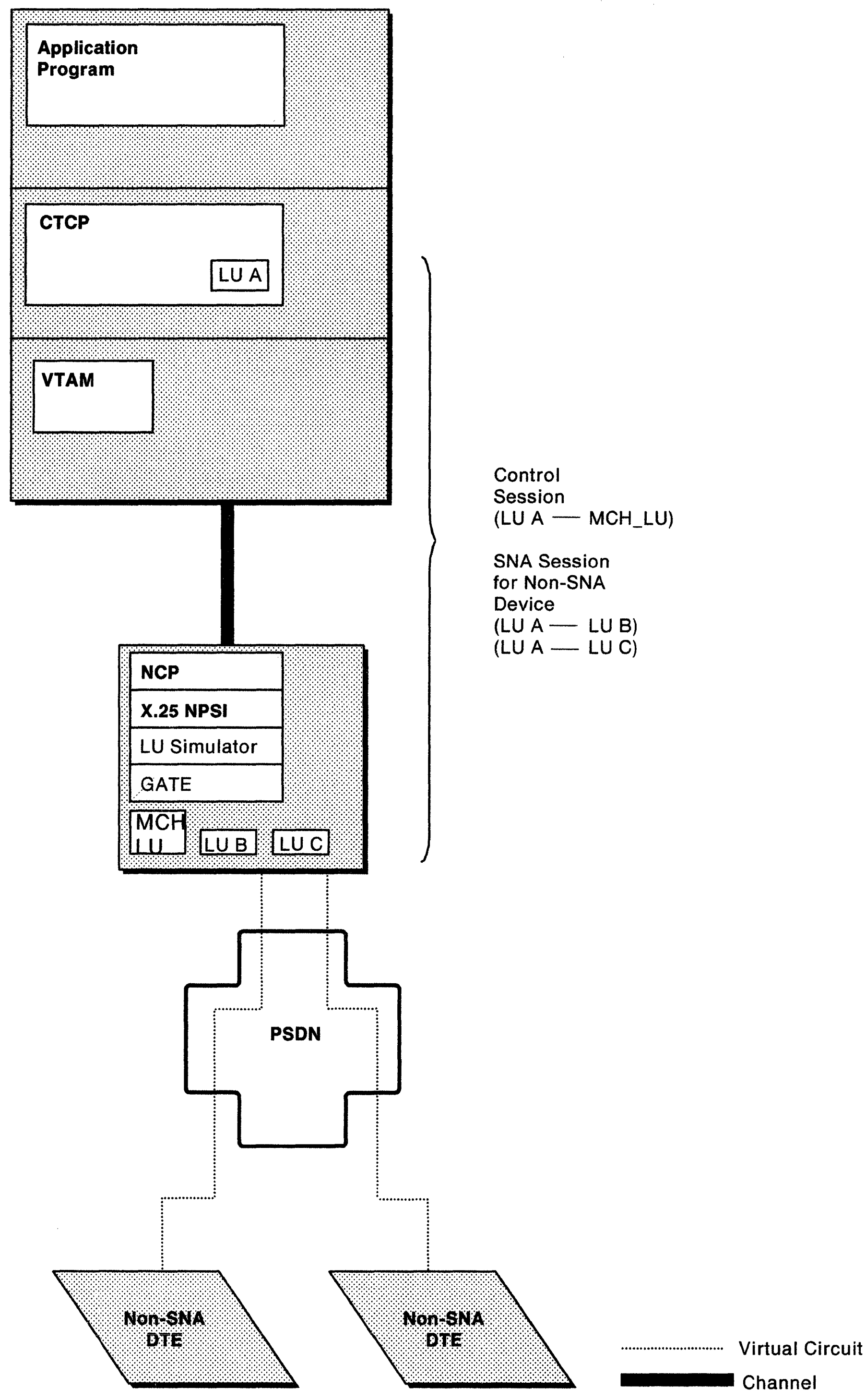


Figure 11. SNA Host Node to Non-SNA DTE (GATE)

Control Session Establishment

The control session for GATE can be established using any of the following session initiation requests:

- The network operator can issue one of the following commands:
  - V NET,LOGON=*application*
  - V NET,ACT with LOGAPPL=*application* coded on the LU statement.

  When either of these commands is issued, a session is automatically initiated between the controlling application (specified by *application*) and the LU simulator.

- The application program can issue one of the following:
  - The OPNDST OPTCD=ACQUIRE statement instruction
  - The SIMLOGON and OPNDST OPTCD=ACCEPT statement instructions.

  In either case, the application program must have previous knowledge of the LU's existence to use these instructions. In the second case, a LOGON exit must be coded as well.

The following processes are unique to GATE:

- All data for GATE processing passes between the CTCP and the virtual circuit LU. The CTCP must either process the request or pass it to the appropriate application.
- The use of GATE is determined on a virtual circuit-by-virtual circuit basis. Therefore, a single physical circuit can support the GATE LLC, as well as other types of LLCs.
- Call-out is simulated as a call-in to NCP and the access method. For this reason, you do not need to code a PATH statement in the switched major node definition.

Virtual Circuit Establishment

You can use different methods to establish virtual circuits, depending on whether the circuits are permanent or switched. The methods available for the establishment of permanent and switched virtual circuits are described in the following sections.

Session Establishment through a Permanent Virtual Circuit

The control session (CTCP LU to MCH LU) performs virtual circuit management. For this reason, establishment of the control session must be the first step in virtual circuit establishment. The permanent virtual circuit can be established by issuing any of the following activation requests and statement instructions:

- V NET,ACT with LOGAPPL=*appl1* coded on the VC LU definition statement
- V NET,LOGON=*appl1*
- OPNDST OPTCD=ACQUIRE
- SIMLOGON and OPNDST OPTCD=ACCEPT
- LOGON from a remote DTE.

Session Establishment through a Switched Virtual Circuit

To set up switched virtual circuits, the CTCP must send commands to X.25 NPSI over the APPL LU to MCH LU session.

Your CTCP must open the session with the physical circuit LU before GATE virtual circuit sessions can be established. The GATE code in X.25 NPSI finds the name of the CTCP application program in the primary-LU name field of the BIND SESSION command.

One CTCP can manage several physical circuits, so the user data field of the BIND SESSION command must contain the symbolic name of the physical circuit LU. This symbolic name, which is associated with the SNA resource identifier, enables the CTCP to identify the SNA resource and MCH where the logon message is received.

Once the session between the LU associated with the CTCP and the LU associated with the physical circuit is established, a connection from a switched virtual circuit between the CTCP and a remote DTE can be established in one of two ways:

- A CALL REQUEST command from the CTCP to the physical circuit LU
- An Incoming Call packet containing a type 4 virtual circuit identifier.

BIND SESSION Command Format for a Physical Circuit LU

The format of the BIND SESSION command between the CTCP LU and the physical circuit LU (MCH LU) is the same as that described in Table 3 on page 13 with the following additions:

Position in BIND Image	**Description**
Byte 7	The physical circuit LU must work in contention mode
Byte 26	Crypto options field length
Bytes 27 through k	Crypto options
Byte k+1	Length of primary LU name (L1)
Bytes k+2 through m	Primary LU name (CTCP name)
Byte m+1	Length of secondary LU name (L2)
Bytes m+2 through n	Secondary LU name (physical circuit LU name).

The following is the layout of this section of the BIND command without crypto fields:

Position in BIND Image	**Description**
26	Length of crypto field (Value = 0)
27	Length of PLU name (Value = L1)
28	PLU name
28+L1+1	Length of SLU name
28+L1+2	SLU name.

BIND SESSION Command Format for a Virtual Circuit LU

The BIND SESSION command format for a virtual circuit LU (VC LU) is the same as shown in Table 3 on page 13.

Logon Message Format for GATE

Once a switched virtual circuit is set up between the CTCP and a remote DTE, the X.25 NPSI GATE function generates a logon message. This message sets up the LU-LU session between the LU associated with the CTCP and the LU associated with the virtual circuit for the duration of the call. The logon message has the following format:

```
LOGON APPLID(XXXXXXXX) DATA(YYYYZZZZZZZZ)
```

where:

XXXXXXX — Is the primary LU (CTCP) name of up to 8 characters.

YYYY — Is a 2-byte connection identifier (CNID) in the case of a call request made by the CTCP, or a 2-byte SNA resource identifier (RESID) in the case of an incoming call from the network. RESID has an X'F' as the first digit. See the CNID and RESID definitions on page 56.

ZZZZZZZZ — Is the secondary LU (physical circuit) name of 8 characters. It is padded to the right with blanks.

The following example assumes that the CTCP application name is CTCP1 and that the physical circuit LU name is XU038:

```
LOGON APPLID(CTCP1) DATA(0001XU038   )
```

GATE constructs and sends to the SSCP a logon message that allows the LU associated with the virtual circuit to enter into a session with the CTCP.

Processing of Data Received before Session Establishment

Data received before the session is established between the VC LU and the CTCP is treated in the following manner:

PVCs — After the ACTLU command is processed, the first message is treated as a logon and is passed to the SSCP on the SSCP to LU session. The other data is passed to the CTCP on the CTCP to VC LU session after start data traffic (SDT) processing.

SVCs — The data is queued in X.25 NPSI. The data is passed to the CTCP on the CTCP to VC LU session after SDT processing. The logon message is built by the X.25 NPSI GATE function and passed to the SSCP on the SSCP to VC LU session after the ACTLU command is processed.

Virtual Circuit Termination

A virtual circuit can be deactivated using any of the following requests and commands:

- Clear Request from remote DTE
- CLSDST from the application program
- Request from the application program for the CTCP to CLEAR
- V NET,INACT command
- V NET,TERM command.

In the simplest case, the remote DTE requests the session termination. In this instance, a Clear Request packet is received from the PSDN, and X.25 NPSI transfers this packet to the CTCP as a CLEAR REQUEST command.

The user application program can initiate session termination. In this case, X.25 NPSI receives an ABCONN request from VTAM and clears the SVC, or sends a reset out for a PVC.

The user application program can notify the CTCP to disconnect the virtual circuit. When the CTCP receives this request:

- The CTCP builds a CLEAR command, including the cause and diagnostic bytes as determined by the CTCP.
- The CLEAR command is sent to X.25 NPSI.
- X.25 NPSI then sends a Clear Request packet to the PSDN.
- A Clear Confirmation packet is sent to X.25 NPSI. X.25 NPSI transfers this packet to the CTCP in the CLEAR CONFIRMATION command.
- A CLSDST macro is issued to end the session with the VC LU. X.25 NPSI does not generate an INOP message.

The network operator can terminate the session by issuing a V NET,INACT or V NET,TERM command for the VC PU or the VC LU. The SNA and X.25 resources are disassociated when this command is issued. X.25 NPSI then clears the SVC or sends a reset out for a PVC.

Control Session Termination

The control session for GATE can be terminated in any of the following ways:

- The application program can request that the CTCP issue a CLSDST macro for the MCH LU.
- The network operator can issue one of the following commands:
  - V NET,INACT,ID=*mch lu*
  - V NET,TERM,LU=*mch lu*
  - CANCEL *the CTCP name*

  In the first and second commands, *mch lu* is the identifier of the physical circuit to be terminated. In the third command, the CTCP is canceled.
- If the CTCP abends, the physical circuit can no longer process GATE calls.

Note: If the CTCP is canceled or abends, the resources used with this CTCP must be deactivated and reactivated. This reactivation is necessary, because any LUs undergoing logon processing at the time the CTCP fails will be waiting for a response from the CTCP that will not be forthcoming. Therefore, the switched major nodes where the LUs are defined must be deactivated and reactivated.

MCH Failure Handling

Error reporting for MCH failures is handled in the following ways:

Failure	Action
MCH INOP	This occurs if MKBMDREC equals 1x or 2x. The LU also becomes inoperable, and the session with the CTCP fails.
MCH REINIT	This occurs if MKBMDREC equals 3x. The PU and LU MCH stay up and the session with the CTCP does not fail. However, the active virtual circuit PUs are all made inoperable. Note that the MKBMDREC value is reported as a qualifier in the NetView Hardware Monitor event detail panel, which describes the MCH disruption event.

Command Interface between GATE and the CTCP

In the following command descriptions, the RESID field in the RU format corresponding to the resource identifier designates either the set of SNA resources (LINE, PU, and LU) that are associated with a virtual circuit (VC), or the X.25 VC number (systems generation option VCID). SNA resource identifiers are associated with VCs for each MCH. They correspond to the number of the SNA resource within the set of SNA resources associated with the MCH. The *range* of resource identifier numbers is the same as the *range* of virtual circuit numbers. An SNA resource number consists of 3 digits in hexadecimal format. The first digit is the logical group number and the next two digits are the logical channel numbers.

The following example illustrates how resource identifier numbers are determined.

```
X25.LCG  LCGN=0
X25.VC   LCN=(1,2),TYPE=P ....  2 PVCs
X25.VC   LCN=(3,99),TYPE=S ... 97 SVCs
X25.LCG  LCGN=2
X25.VC   LCN=(1,3),TYPE=P ....  3 PVCs
X25.VC   LCN=(4,98),TYPE=S ... 95 SVCs
```

In this example, SNA identifiers X'003' through X'063' and SNA identifiers X'204' through X'262' are used to map all SVCs of this MCH. This setup allows CTCPs to continue to run without change.

The commands recognized or generated by GATE are listed in Table 5 on page 54.

Table 5. CTCP-GATE Control Commands

Command Code	Command	Session with CTCP
00	DATA EXCHANGE (Q bit set to 0)	Data Session
02	DATA EXCHANGE (Q bit set to 1)	Data Session
0B	INCOMING CALL/CALL REQUEST	Control Session
0F	CALL CONNECTED/CALL ACCEPTED	Control Session
13	CLEAR indication/CLEAR request	Control Session
17	CLEAR CONFIRMATION	Control Session or Data Session*
1B	RESET	Control Session or Data Session*
1F	RESET CONFIRMATION	Data Session
23	INTERRUPT	Data Session
27	INTERRUPT CONFIRMA-TION	Data Session
F1	DIAGNOSTIC	Control Session
FF	INFORMATION REPORT Message	Control Session

Note: An asterisk (*) indicates that the command flows on the data session if the VC LU is in session; otherwise, the command flows on the control session.

The following describes the data exchange commands and the control commands used by GATE and the CTCP. The data exchange commands are followed by user data. The control commands are followed by additional control information.

DATA EXCHANGE without Q Bit (CTCP to GATE, GATE to CTCP through the Data Session)

A DATA EXCHANGE command with the Q bit set off is issued from the CTCP to GATE and from GATE to the CTCP through the data session. The command is sent in the following format:

```
       ┌────┬─//─┐
RU:    │ cc │user│
       └────┴─//─┘
```

Bytes	Field	Description
0	cc	1-byte command code. For this command, the value is X'00'.
1–n	user	Variable-size field containing user data following the packet header.

Note: The command byte is set only in FIC and OIC RUs.

DATA EXCHANGE with Q Bit (CTCP to GATE, GATE to CTCP through the Data Session)

A DATA EXCHANGE command with the Q bit set on is issued from the CTCP to GATE and from GATE to the CTCP through the data session. The command is sent in the following format:

```
       ┌────┬─//─┐
RU:    │ cc │user│
       └────┴─//─┘
```

Bytes	Field	Description
0	cc	1-byte command code. For this command, the value is X'02'.
1–n	user	Variable-size field containing user data following the packet header.

Note: The command byte is set only in FIC and OIC RUs.

CALL REQUEST (CTCP to GATE through the Control Session)

The CALL REQUEST command is issued from the CTCP to GATE through the control session. The command is sent in the following format:

RU:

cc	cnid	pw	psiz	crpk

Bytes	Field	Description
0	cc	1-byte command code. For this command, the value is X'0B'.
1–2	cnid	2-byte unique connection identifier determined by the CTCP. This 3-digit number is right-aligned. The leftmost halfbyte *must* be X'0'. To avoid conflicts, you should use numbers greater than the highest SNA resource identifier numbers (virtual circuit numbers).
3	pw	1-byte packet window size.
4–5	psiz	2-byte packet size, in hexadecimal, used for this virtual circuit.
6–n	crpk	Variable-size field containing the Call Request packet without the packet header.

Note: Packet and window sizes specified in the CALL REQUEST command are taken into account by X.25 NPSI. The flow control parameters included in the facilities of the Call Connected packet are ignored by X.25 NPSI. The CTCP should clear the call and do a recall if these flow control parameters do not match the flow control parameters that were passed to X.25 NPSI in the CALL REQUEST command.

For V3R3 only: Depending on the generation option specified, the flow control parameters included in the facilities of the CALL CONNECTED packet are taken into account by X.25 NPSI.

CALL CONNECTED (GATE to CTCP through the Control Session)

This CALL CONNECTED command is issued from GATE to the CTCP through the control session. The command is sent in the following format:

RU:

cc	cnid	resid	ccpk

Bytes	Field	Description
0	cc	1-byte command code. For this command, the value is X'0F'.
1–2	cnid	2-byte connection identifier as specified in the CALL REQUEST command.
3–4	resid	2-byte resource identifier. The leftmost halfbyte is set to X'F'.
5–n	ccpk	Variable-size field containing optional bytes that followed the packet header of the Call Connected packet.

INCOMING CALL (GATE to CTCP through the Control Session)

The INCOMING CALL command is issued from GATE to the CTCP through the control session. The command is sent in the following format:

```
RU: | cc | resid | icpk //|
```

Bytes	Field	Description
0	cc	1-byte command code. For this command, the value is X'0B'.
1−2	resid	2-byte resource identifier. The leftmost halfbyte is set to X'F'.
3−n	icpk	Variable-size field containing the Incoming Call packet received from the network (without the 3-byte packet header).

CALL ACCEPTED (CTCP to GATE through the Control Session)

The CALL ACCEPTED command is issued from the CTCP to GATE through the control session. The command is sent in the following format:

```
RU: | cc | resid | pw | psiz | capk //|
```

Bytes	Field	Description
0	cc	1-byte command code. For this command, the value is X'0F'.
1−2	resid	2-byte resource identifier as found in the INCOMING CALL command.
3	pw	1-byte packet window size.
4−5	psiz	2-byte packet size in hexadecimal used for this virtual circuit.
6−n	capk	Variable-size field containing the Call Accepted packet to be sent to the network without the 3-byte packet header.

CLEAR on Incoming Call (CTCP to GATE through the Control Session)

The CLEAR command can be issued from the CTCP to GATE through the control session to refuse an incoming call. The command is sent in the following format:

```
              ┌────┬───────┬──────┬─//─┐
RU:           │ cc │ resid │ crdf │user│
              └────┴───────┴──────┴─//─┘
```

Bytes	Field	Description
0	cc	1-byte command code. For this command, the value is X'13'.
1–2	resid	2-byte resource identifier as found in the INCOMING CALL command.
3–4	crdf	2-byte CAUSE and DIAGNOSTIC fields that are to be inserted in the Clear Request packet.
5–n	user	Variable-size field containing optional user data to be put in the Clear Request packet after the CAUSE and DIAGNOSTIC fields.

CLEAR after Incoming Call (GATE to CTCP through the Control Session)

The CLEAR command can be issued after the Incoming Call is sent from GATE to the CTCP through the control session. A Clear Indication clears an unacknowledged Incoming Call from the network. The command is sent in the following format:

```
              ┌────┬───────┬──────┬─//─┐
RU:           │ cc │ resid │ crdf │user│
              └────┴───────┴──────┴─//─┘
```

Bytes	Field	Description
0	cc	1-byte command code. For this command, the value is X'13'.
1–2	resid	2-byte resource identifier. The leftmost byte is set to X'F'.
3–4	crdf	2-byte CAUSE and DIAGNOSTIC fields from the Clear Indication packet.
5–n	user	Variable-size field containing optional user data found in the Clear Indication packet after the CAUSE and DIAGNOSTIC fields.

CLEAR CONFIRMATION after CLEAR Command for Incoming Call Refused (GATE to CTCP through the Control Session)

The CLEAR CONFIRMATION command is issued after the CLEAR command for the incoming call that is refused. The command is issued from GATE to the CTCP through the control session. The command is sent in the following format:

```
         ┌────┬───────┬──//──┐
    RU:  │ cc │ resid │ user │
         └────┴───────┴──//──┘
```

Bytes	Field	Description
0	cc	1-byte command code. For this command, the value is X'17'.
1–2	resid	2-byte resource identifier as found in the CLEAR on an incoming call. The leftmost halfbyte is set to X'F'.
3–n	user	Variable-size field containing optional user data if present in the DCE Clear Confirmation packet.

CLEAR on Outgoing Call (GATE to CTCP through the Control Session)

The CLEAR command on the outgoing call is issued from GATE to the CTCP through the control session, if an outgoing call cannot be completed. The command is sent in the following format:

```
         ┌────┬──────┬──────┬──//──┐
    RU:  │ cc │ cnid │ crdf │ user │
         └────┴──────┴──────┴──//──┘
```

Bytes	Field	Description
0	cc	1-byte command code. For this command, the value is X'13'.
1–2	cnid	2-byte connection identifier as specified in the CALL REQUEST command. The leftmost halfbyte is 0.
3–4	crdf	2-byte CAUSE and DIAGNOSTIC fields from the Clear Indication packet.
5–n	user	Variable-size field containing optional user data found in the Clear Indication packet after the CAUSE and DIAGNOSTIC fields.

Note: If the X.25 NPSI timer elapses on a call request, X.25 NPSI sends a Clear packet to the PSDN and sends this CLEAR command to the CTCP.

CLEAR (GATE to CTCP, CTCP to GATE through the Data Session)

This CLEAR command is issued from GATE to the CTCP and from the CTCP to GATE through the data session. The command is sent in the following format:

```
RU: | cc | ccdf | user //
```

Bytes	Field	Description
0	cc	1-byte command code. For this command, the value is X'13'.
1−2	ccdf	2-byte clear CAUSE and DIAGNOSTIC fields.
3−n	user	Variable-size field containing optional user data following the CAUSE and DIAGNOSTIC fields.

CLEAR CONFIRMATION (GATE to CTCP through the Data Session)

The CLEAR CONFIRMATION command is issued from GATE to the CTCP through the data session. The command is sent in the following format:

```
RU: | cc | user //
```

Bytes	Field	Description
0	cc	1-byte command code. For this command, the value is X'17'.
1−n	user	Variable-size field containing any optional user data found in the DCE Clear Confirmation packet.

RESET (GATE to CTCP, CTCP to GATE through the Data Session)

The RESET command is issued from GATE to the CTCP and from the CTCP to GATE through the data session. The command is sent in the following format:

```
RU: | cc | rcdf | user //
```

Bytes	Field	Description
0	cc	1-byte command code. For this command, the value is X'1B'.
1−2	rcdf	2-byte reset CAUSE and DIAGNOSTIC fields.
3−n	user	Variable-size field containing optional user data following the CAUSE and DIAGNOSTIC fields.

If X.25 NPSI detects an unrecoverable situation, GATE sends a RESET command to the CTCP and resets the VC:

Type of Error	Reset CMD to CTCP	Reset Req to Network
Pkt rcvd w/ invalid P(R)	Diag ab	Diag ab
Pkt rcvd w/ invalid P(S)	Diag ac	Diag ac
Pkt rcvd w/ invalid D-bit	Diag ad	Diag ad
Inconsistent Q-bit setting	Diag 53	Diag 53
Chain cancelled by APPL	Diag b3	Diag b3

RESET CONFIRMATION (GATE to CTCP through the Data Session)

The RESET CONFIRMATION command is issued from GATE to the CTCP through the data session. The command is sent in the following format:

```
       ┌────┬─//─┐
RU:    │ cc │user│
       └────┴─//─┘
```

Bytes	Field	Description
0	cc	1-byte command code. For this command, the value is X'1F'.
1 – n	user	Variable-size field containing any optional user data found in the DCE Reset Confirmation packet.

INTERRUPT (CTCP to GATE, GATE to CTCP through the Data Session)

The INTERRUPT command is issued from the CTCP to GATE and from GATE to the CTCP through the data session. The command is sent in the following format:

```
       ┌────┬────┬─//─┐
RU:    │ cc │ iu │user│
       └────┴────┴─//─┘
```

Bytes	Field	Description
0	cc	1-byte command code. For this command, the value is X'23'.
1	iu	1-byte interrupt user data byte.
2 – n	user	Variable-size field containing optional user data following the interrupt user data byte.

INTERRUPT CONFIRMATION (CTCP to GATE, GATE to CTCP through the Data Session)

The INTERRUPT CONFIRMATION command is issued from the CTCP to GATE and from GATE to the CTCP through the data session. The command is sent in the following format:

```
       ┌────┬─//─┐
RU:    │ cc │user│
       └────┴─//─┘
```

Bytes	Field	Description
0	cc	1-byte command code. For this command, the value is X'27'.
1 – n	user	Variable-size field containing optional user data following the packet header.

DIAGNOSTIC (GATE to CTCP through the Control Session)

The DIAGNOSTIC command is issued from GATE to the CTCP through the control session. The command is sent in the following format:

```
    ┌────┬────┬─//─┐
RU: │ cc │ dc │ od │
    └────┴────┴─//─┘
```

Bytes	Field	Description
0	cc	1-byte command code. For this command, the value is X'F1'.
1	dc	1-byte diagnostic code.
2−n	od	Variable-size field containing diagnostic explanation.

ERROR/INFORMATION REPORT (GATE to CTCP through the Control Session or Data Session)

The ERROR/INFORMATION REPORT command is issued from GATE to the CTCP through the control session for a CLEAR command sent on the incoming call. Otherwise, the command is sent through the data session. The command is sent in the following format:

```
    ┌────┬───────┬────┬────┐
RU: │ cc │ resid │ ce │ ec │
    └────┴───────┴────┴────┘
```

Bytes	Field	Description
0	cc	1-byte command code. For this command, the value is X'FF'.
1−2	resid	2-byte resource identifier. The leftmost halfbyte is set to X'0'.
3	ce	1-byte command code for the command in error.
4	ec	1-byte error cause with a value of X'30'. Timer elapsed before a response from the network was received for the last control packet sent out.

Note: No ERROR/INFORMATION REPORT is sent to the CTCP if the X.25 NPSI timer elapses on a call request. Instead, a CLEAR command is sent to the CTCP, and a Clear packet is sent to the PSDN.

RESET because Permanent Virtual Circuit Not in SNA Session (GATE to CTCP through the Control Session)

A RESET command is issued from GATE to the CTCP when a Reset packet is received and the permanent virtual circuit is not in session with the host. This RESET command is issued through the control session and is sent in the following format:

```
              ┌────┬───────┬──────┬──//─┐
RU:           │ cc │ resid │ rcdf │ user │
              └────┴───────┴──────┴──//─┘
```

Bytes	Field	Description
0	cc	1-byte command code. For this command, the value is X'1B'.
1−2	resid	2-byte resource identifier. The leftmost halfbyte is set to X'0'.
3−4	rcdf	2-byte reset CAUSE and DIAGNOSTIC fields.
5−n	user	Variable-size field containing optional user data.

The network has sent a Reset packet on the indicated permanent virtual circuit that is not yet in session. The CTCP might want to establish a session using this permanent virtual circuit. X.25 NPSI sends the Reset Confirmation automatically, and the CTCP must not send the Reset Confirmation to X.25 NPSI.

Notes:

1. When control commands from the CTCP to GATE contain optional user data, such data is put into the resulting control packet without being checked by GATE. The M bit is not used with control packets. For this reason, all user data must fit into a single packet.

2. When X.25 NPSI receives a RESET from the network, X.25 NPSI responds immediately with a RESET CONFIRMATION. This is the reason why the CTCP does not have to send a RESET CONFIRMATION to GATE.

   When X.25 NPSI receives a CLEAR from the network, X.25 NPSI responds with a CLEAR CONFIRMATION at ABCONN time from VTAM. This is to avoid having the next call arrive while the previous session is closing down. The CTCP does not send a CLEAR CONFIRMATION to GATE.

   Although the CTCP cannot send the CLEAR CONFIRMATION command or the RESET CONFIRMATION command, the CTCP must be able to receive these commands.

Interfacing with Multiple CTCPs

If more than one CTCP is specified in an X25.MCH statement, the MCH has more than one LU. The CTCPs can reside in the same host or in different hosts. Each MCH LU is in session with a CTCP. Once a CTCP to MCH LU session is established, X.25 NPSI knows the CTCP application name and places it in the logon command that is generated by X.25 NPSI when a call-in or call-out occurs on a virtual circuit.

The selection of the CTCP is based on subaddressing or on the value of the first byte of the CALL USER DATA (CUD) field. If subaddressing is used and the subaddressing digit is found, only CTCP 0 can be chosen. If the subaddressing digit is *not* found, CUD0 is used. See *X.25 NPSI Planning and Installation* for more information about CTCP selection.

Once the appropriate MCH LU and CTCP are selected, the interface between these two LUs is the same as the interface for GATE with a single CTCP. If several CTCPs can be accessed from a given MCH, all VC LUs that can be associated with this MCH must be defined in all the CTCPs, if the CTCPs require the definition of the LUs that go in session with them. In the case of GTMOSI™[3] for instance, all VC LUs associated with an MCH must be defined in each generation of GTMOSI related to this MCH.

Fast Connect GATE Option

Fast connect is an option of the X.25 NPSI GATE function. Fast connect is used only for switched virtual circuits connected to non-SNA DTEs.

Fast connect operates similarly to GATE. With fast connect, X.25 resources are dynamically connected to SNA sessions with the host. These sessions can be pre-established before the MCH is activated.

The reduction in session setup time results in a much quicker connection to the host. This faster connection is useful when connections are characterized by:

- Short connection duration
- Small data size
- Quick response time for connection establishment
- Heavy calling and clearing rate.

VTAM is not aware that the SNA sessions are carrying data from different virtual circuits, because only X.25 NPSI is involved in the mapping of the virtual circuit to the SNA session.

Specifying fast connect permits the creation of a number of SNA resources and a *different* number of X.25 resources. You can create a pool of available SNA resources that can be dynamically connected to different X.25 resources on demand. Virtual calls can be directed to different CTCPs, depending on the amount of traffic at the time.

Fast connect supports only the following logmode table entry:

```
FAST1   MODEENT   LOGMODE=FAST,
                  TSPROF=X'03',
                  FMPROF=X'03',
                  PRIPROT=X'B0',  (- multi RU chain from PLU   )
                                  (- definite or exception resp)
                                  (- primary will not send EB  )
                  SECPROT=X'B0',  (- multi RU chain from PLU   )
                                  (- definite or exception resp)
                                  (- secondary will not send EB)
                  COMPROT=X'0040' (- bracket not used          )
                                  (- half-duplex contention    )
```

Sessions between the CTCPs and the virtual circuit LUs can be established before MCH activation so that the sessions are ready to receive incoming calls when the MCH becomes active.

3 General Teleprocessing Monitor for Open Systems Interconnection (GTMOSI). In certain countries IBM can provide these CTCPs for use with GATE.

Figure 12 on page 65 shows the data flow for a network configuration that uses fast connect GATE.

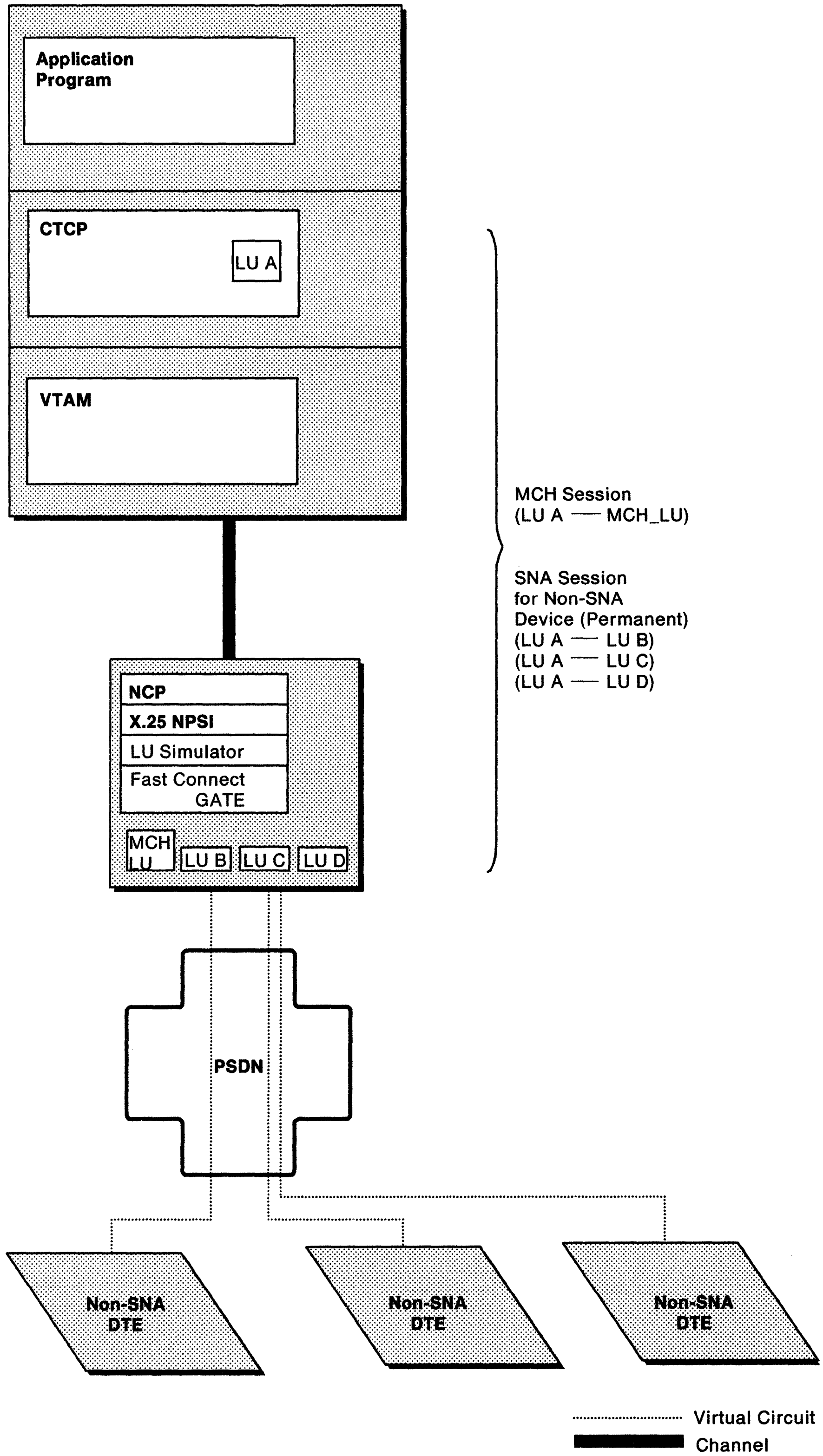


Figure 12. SNA Host Node to Non-SNA DTE (Fast Connect GATE)

Call-In (GATE to CTCP)

When an incoming call is received from the PSDN, fast connect GATE sends a CALL IN command to the host processor through a CTCP LU to VC LU session. This processing differs from non-fast connect GATE processing, in which CALL commands are sent on the CTCP LU to MCH LU session. The reason for this difference is the high calling rate that can occur with the fast connect option. If this traffic were placed on the CTCP LU to MCH LU session, a bottleneck for session setup could occur. Placing the control requests on the CTCP LU to VC LU session spreads the control requests across more sessions, thereby reducing the delay because of serialization on a single session.

X.25 NPSI chooses an available SNA resource to place the incoming call and allocates that SNA resource to the calling VC. X.25 NPSI then associates the SNA resource and the X.25 resource.

Once the virtual circuit is established, fast connect operates similarly to the GATE interface. Call operation is performed with the same commands as would occur without the fast connect option. (See "Command Interface between Fast Connect and the CTCP" on page 70 for the exact format of the commands passed between the CTCP and fast connect GATE operating in X.25 NPSI.)

If the CTCP does not accept the call, the CTCP returns a CLEAR command that carries the resource identifier of the call to fast connect GATE. Fast connect transforms the CLEAR command into a Clear packet and forwards the packet to the PSDN along with the cause and diagnostic fields specified by the CTCP. See *X.25 NPSI Diagnosis, Customization, and Tuning* for a list of supported cause and diagnostic codes.

Call-Out (CTCP to GATE)

When a fast connect GATE CTCP performs a call-out procedure, the CTCP sends a CALL REQUEST command on a CTCP to VC LU session.

If a CTCP tries to send a CALL REQUEST command to the LU on the physical circuit, fast connect GATE returns an X'0801' sense code (resource not available). This negative response tells the CTCP that it is communicating with X.25 NPSI V3R1, rather than with the fast connect request for price quote (RPQ).

When receiving a CALL REQUEST command from a fast connect CTCP, fast connect GATE selects a free X.25 resource, dynamically links it to an SNA resource, and uses the information provided in the command to create a Call Request packet that is sent to the PSDN.

The PSDN acknowledges the Call Request packet with a Call Connected packet. Fast connect GATE returns a CALL CONNECTED command to the CTCP. The resource identifier in the CALL CONNECTED command is used by the CTCP and fast connect GATE for subsequent commands and data exchange.

If the PSDN responds to a Call Request with a Clear packet, a CLEAR command is sent to the CTCP. The CLEAR command contains a clearing cause and diagnostic byte.

If a response to the Call Request is not received within the duration of the X.25 T21 timer, X.25 NPSI sends a CLEAR command (built by fast connect GATE) to the network and then to the CTCP through the data session.

Call Collision

Figure 13 shows the order of virtual circuit and SNA resource assignment for call-in and call-out.

If contention occurs, and depending on the call that arrives last at X.25 NPSI, one of the following actions is performed by X.25 NPSI:

- Discards an incoming call from the PSDN using the same virtual circuit as the call-out session.
- Responds negatively to a CALL command from the CTCP using the same SNA resource ID as the call-in session.

```
               Order of                     Order of
              Assignment                   Assignment
              of X.25 VCs               of SNA Sessions
              by a Network                  by CTCP
 ┌──────────┐      │                          │      ┌────────────────┐
 │X25 VC 00 │      │                          │      │ SNA session 00 │
 │X25 VC 01 │      │                          │      │ SNA session 01 │
 │X25 VC 02 │      │                          │      │ SNA session 02 │
 │X25 VC 03 │      ▼                          ▼      │ SNA session 03 │
 │X25 VC 04 │                                        │ SNA session 04 │
 │    .     │                                        │       .        │
 │    .     │                                        │       .        │
 │    .     │                                        │       .        │
 │    .     │      ▲                          ▲      │       .        │
 │    .     │      │                          │      │       .        │
 │          │      │                          │      │                │
 │X25 VC nn │      │                          │      │ SNA session nn │
 └──────────┘      │                          │      └────────────────┘
               Order of                     Order of
              Assignment                   Assignment
              of X.25 VCs                of SNA Sessions
                by NPSI                      by NPSI
             (Call Command                (Incoming Call
           Received from CTCP)        Received from Network)
```

Figure 13. Virtual Circuit and SNA Resource Assignment Order

MCH Reinitialization Consideration

When an MCH reinitialization occurs, the fast connect SNA resources are not changed to INOP status. For SNA resources engaged in an active call, a CLEAR command with a diagnostic code of X'30' is sent to the CTCP. For SNA resources not engaged in an active call, no action is taken.

Fast Connect and CTCP Interface during Virtual Circuit Connection

Once an X.25 connection is established, data and commands (for example, RESET and INTERRUPT) can be exchanged between fast connect GATE and a CTCP.

The data exchanged can be either qualified or unqualified. Qualified data is used to communicate data to a PAD; unqualified data is used to communicate user data. An example of qualified (control) data is a packet containing a PAD command.

If the data is specified as qualified, the Q bit is on in the X.25 packet. The data type is communicated to fast connect GATE through the use of the first byte of the RU. The distinction is:

RU byte 0	Usage
X'00'	Unqualified data
X'02'	Qualified data.

X.25 NPSI Connection Termination

X.25 NPSI communications are terminated by the CLEAR command. The CLEAR command is processed differently by X.25 NPSI depending on whether a normal session termination is possible or a collision must be resolved.

Normal Clearing

An X.25 communication can end either on CTCP request (a CLEAR command) or on request from either the PSDN or the terminal (a Clear Indication packet).

CLEAR commands (X'13') flow between the CTCP and fast connect GATE on the CTCP LU to VC LU session.

If the CLEAR indication signals a routine end of session, the cause code is X'00' or X'C6'.

After a CLEAR command exchange, the X.25 virtual circuit is freed by the PSDN. However, because the associated SNA resource remains bound and available for a subsequent call, X.25 NPSI discards any unexpected packet (except an Incoming Call packet) received from the PSDN. Requests from the CTCP, with the exception of CALL REQUEST commands, are rejected with an SNA sense code of X'081C'.

When a Clear packet is received from the PSDN, fast connect GATE sends a Clear Confirmation packet to the network, and generates a RECFMS type 0 request unit (which appears as an event to the NetView Hardware Monitor) unless the clear cause code is X'00' or X'C6'. The SNA resource set and the X.25 virtual circuit are immediately available for the next call.

If you request the billing record option by specifying the TAXUNIT keyword of the X25.MCH statement, the CLEAR or CLEAR CONFIRMATION command returned to the CTCP will contain billing information. The TAXUNIT keyword specifies the size of the X.25 accounting unit.

The billing function of fast connect GATE provides the CTCP with the option of obtaining the following information:

- Number of X.25 accounting units
- User session start time and date
- User session end time and date.

The billing information can be used by the CTCP or an auxiliary program to assist in utilization tracking, accounting, and other user requirements.

When you request the billing option, billing information is appended to the last command (CLEAR, CLEAR CONFIRMATION, or ERROR/INFORMATION REPORT) sent by X.25 NPSI to a CTCP. The RU of the last command is sent in the format shown in Figure 14.

CLEAR, CLEAR CONFIRM, or ERROR/ INFO. RPT. as Received from Network	(	Session Start Time and Date	Session End Time and Date	Re- served	Number of X.25 Accounting Units Received	Number of X.25 Account. Units Sent	(

Figure 14. Format of CLEAR, CLEAR CONFIRMATION, and ERROR/INFORMATION REPORT Commands with Billing Option

Notes:

1. The two left parentheses are delimiters and can be used by the CTCP to locate the billing information.
2. An X.25 unit is the network billing unit as defined in the TAXUNIT keyword of the X25.MCH statement at X.25 NPSI generation.

The CLEAR command is sent in the following format:

Bytes	Value
0	X'13'
1 through 2	Resource ID
3 through 4	Cause and diagnostic fields
5 through n	User data
n+1	(
n+2 through n+21	Session start time and date
n+22 through n+41	Session end time and date
n+42 through n+44	Reserved
n+45 through n+48	Number of X.25 accounting units received
n+49 through n+52	Number of X.25 accounting units sent
n+53	(

The CLEAR CONFIRMATION command is sent in the following format:

Bytes	Value
0	X'17'
1	(
2 through 21	Session start time and date
22 through 41	Session end time and date
42 through 44	Reserved
45 through 48	Number of X.25 accounting units received
49 through 52	Number of X.25 accounting units sent
53	(

The ERROR/INFORMATION REPORT command is sent in the following format:

Bytes	Value
0	X'FF'
1 through 2	Resource identifier
3	Command that failed
4	Error cause
5	(
6 through 25	Session start time and date
26 through 45	Session end time and date
46 through 48	Reserved
49 through 52	Number of X.25 accounting units received
53 through 56	Number of X.25 accounting units sent
57	(

The billing information is sent to the CTCP in the ERROR/INFORMATION REPORT command when X.25 NPSI receives no response to a CLEAR within the value set for the T23 timer (CLEAR time-out).

No billing information is provided, even if it is requested, when an:

- Incoming or outgoing call is cleared before confirmation is received.
- LU-LU session is disrupted.
- MCH experiences a failure.

CLEAR Collision

CLEAR collision is detected in X.25 NPSI when either:

- A CLEAR command is received while a Clear packet is processed.
- A Clear packet is received while a CLEAR command is processed.

If a CLEAR command collides with a previously received Clear Indication packet from the PSDN, fast connect GATE rejects the CLEAR command with an SNA sense code of X'081C' (function not executable).

When X.25 NPSI receives a Clear Request packet from the PSDN, prior to receiving the Clear Confirmation packet for a previously sent CLEAR command, X.25 NPSI sends a CLEAR command to the CTCP to confirm the previous CLEAR command.

Command Interface between Fast Connect and the CTCP

In the command descriptions described in Table 6 on page 71, the RESID field in the RU format that corresponds to the resource identifier designates either the set of SNA resources (LINE, PU, and LU) that are associated with a VC, or the X.25 VC number (systems generation option VCID). SNA resource identifiers are associated with VCs for each MCH. The range for the resource identifier numbers is the same as the range for the VC numbers, if there is only one CTCP for each MCH. Otherwise, the range is defined by the X25.FCG statements.

An SNA resource number consists of 3 digits in hexadecimal format. The leftmost halfbyte maps the logical channel group number and the next 2 digits map the logical channel number.

The following example shows how the X25.MCH statement can be coded to define SVCs. In this example, SNA IDs X'003' through X'063' and SNA IDs X'204' through X'262' are used to map all SVCs of this MCH. This setup allows CTCPs to continue to run without change.

```
X25.LCG  LCGN=0
X25.VC   LCN=(1,2),TYPE=P ....  2 PVCs
X25.VC   LCN=(3,99),TYPE=S ... 97 SVCs
X25.LCG  LCGN=2
X25.VC   LCN=(1,3),TYPE=P ....  3 PVCs
X25.VC   LCN=(4,98),TYPE=S ... 95 SVCs
```

Table 6 lists the commands recognized or generated by fast connect GATE.

Table 6. CTCP-Fast Connect GATE Control Commands

Command Code	Command	Session with CTCP
00	DATA EXCHANGE (Q bit set to 0)	Data Session
02	DATA EXCHANGE (Q bit set to 1)	Data Session
0B	INCOMING CALL/CALL REQUEST	Data Session
0F	CALL CONNECTED/CALL ACCEPTED	Data Session
13	CLEAR indication/CLEAR request	Data Session
17	CLEAR CONFIRMATION	Data Session
1B	RESET	Data Session
1F	RESET CONFIRMATION	Data Session
23	INTERRUPT	Data Session
27	INTERRUPT CONFIRMATION	Data Session
F1	DIAGNOSTIC	Control Session
FF	INFORMATION REPORT Message	Data Session

The following descriptions explain data exchange commands and control commands. The data exchange commands are followed by user data. The control commands are followed by additional control information.

DATA EXCHANGE without Q Bit (CTCP to GATE, GATE to CTCP through the Data Session)

A DATA EXCHANGE command with the Q bit set off is issued from the CTCP to GATE and from GATE to the CTCP through the data session. The command is sent in the following format:

```
RU: | cc | user //|
```

Bytes	Field	Description
0	cc	1-byte command code. For this command, the value is X'00'.
1 − n	user	Variable-size field containing user data following the packet header.

Note: The command byte is set only in FIC and OIC RUs.

DATA EXCHANGE with Q Bit (CTCP to GATE, GATE to CTCP through the Data Session)

A DATA EXCHANGE command with the Q bit set on is issued from the CTCP to GATE and from GATE to the CTCP through the data session. The command is sent in the following format:

```
RU: | cc | user //|
```

Bytes	Field	Description
0	cc	1-byte command code. For this command, the value is X'02'.
1 − n	user	Variable-size field containing user data following the packet header.

Note: The command byte is set only in FIC and OIC RUs.

INCOMING CALL (GATE to CTCP through the Data Session)

The INCOMING CALL command is issued from GATE to the CTCP through the data session. The command is sent in the following format:

```
                     //
RU: | cc | resid | icpk |
                     //
```

Bytes	Field	Description
0	cc	1-byte command code. For this command, the value is X'0B'.
1−2	resid	2-byte resource identifier. The leftmost halfbyte is set to X'F'.
3−n	icpk	Variable-size field containing the Incoming Call packet received from the network (without the 3-byte packet header).

CALL ACCEPTED (CTCP to GATE through the Data Session)

The CALL ACCEPTED command is issued from the CTCP to GATE through the data session. The command is sent in the following format:

```
                                 //
RU: | cc | resid | pw | psiz | capk |
                                 //
```

Bytes	Field	Description
0	cc	1-byte command code. For this command, the value is X'0F'.
1−2	resid	2-byte resource identifier as found in the INCOMING CALL command.
3	pw	1-byte packet window size.
4−5	psiz	2-byte packet size in hexadecimal used for this virtual circuit.
6−n	capk	Variable-size field containing the Call Accepted packet to be sent to the network without the 3-byte packet header.

CALL REQUEST (CTCP to GATE through the Data Session)

The CALL REQUEST command is issued from the CTCP to GATE through the data session. The command is sent in the following format:

```
                                   //
RU: | cc | cnid | pw | psiz | crpk |
                                   //
```

Bytes	Field	Description
0	cc	1-byte command code. For this command, the value is X'0B'.
1–2	cnid	2-byte unique connection identifier determined by the CTCP. This 3-digit number is right-aligned. The leftmost halfbyte *must* be X'0'.
3	pw	1-byte packet window size.
4–5	psiz	2-byte packet size in hexadecimal used for this virtual circuit.
6–n	crpk	Variable-size field containing the Call Request packet without the packet header.

Note: Packet and window sizes specified in the CALL REQUEST command are taken into account by X.25 NPSI. The flow control parameters included in the facilities of the Call Connected packet are ignored by X.25 NPSI. The CTCP should clear the call and try again if these flow control parameters do not match the flow control parameters that were passed to X.25 NPSI in the CALL REQUEST command.

For V3R3 only: Depending on the generation option, the flow control parameters included in the facilities of the CALL CONNECTED packet are taken into account by X.25 NPSI.

CALL CONNECTED (GATE to CTCP through the Data Session)

The CALL CONNECTED command is issued from GATE to the CTCP through the data session. The command is sent in the following format:

```
                              //
RU: | cc | cnid | resid | ccpk |
                              //
```

Bytes	Field	Description
0	cc	1-byte command code. For this command, the value is X'0F'.
1–2	cnid	2-byte connection identifier as specified in the CALL REQUEST command.
3–4	resid	2-byte resource identifier. The leftmost halfbyte is set to X'F'.
5–n	ccpk	Variable-size field containing optional bytes that followed the packet header of the Call Connected packet.

CLEAR on Incoming Call (CTCP to GATE through the Data Session)

The CLEAR command to clear an Incoming call is issued from the CTCP to GATE through the data session. The command is sent in the following format:

```
RU:  | cc | resid | ccdf | user //
```

Bytes	Field	Description
0	cc	1-byte command code. For this command, the value is X'13'.
1−2	resid	2-byte resource identifier as found in the Incoming call command.
3−4	crdf	2-byte CAUSE and DIAGNOSTIC fields that are to be inserted in the Clear Request packet.
5−n	user	Variable-size field containing optional user data to be put in the Clear Request packet after the CAUSE and DIAGNOSTIC fields.

CLEAR CONFIRMATION after CLEAR Command for Incoming Call Refused (GATE to CTCP through the Data Session)

The CLEAR CONFIRMATION command is issued after the CLEAR command for the incoming call is refused. It is issued from GATE to the CTCP through the data session. The command is sent in the following format:

```
RU:  | cc | resid | user //
```

Bytes	Field	Description
0	cc	1-byte command code. For this command, the value is X'17'.
1−2	resid	2-byte resource identifier as found in the CLEAR command on the incoming call. The leftmost halfbyte is set to X'F'.
3−n	user	Variable-size field containing optional user data if present in the DCE Clear Confirmation packet.

CLEAR on Call Request (GATE to CTCP through the Data Session)

The CLEAR command on the Call Request is issued from GATE to the CTCP through the data session. The command is sent in the following format:

```
                               //
RU: | cc | cnid | crdf | user |
                               //
```

Bytes	Field	Description
0	cc	1-byte command code. For this command, the value is X'13'.
1–2	cnid	2-byte connection identifier as specified in the CALL REQUEST command. The leftmost halfbyte is X'0'.
3–4	crdf	2-byte CAUSE and DIAGNOSTIC fields from the Clear packet.
5–n	user	Variable-size field containing optional user data found in the Clear packet after the CAUSE and DIAGNOSTIC fields.

Note: This CLEAR command can be received by the CTCP either when the network clears the outgoing call or if no call confirmation is received from the network within the value set for the T23 timer (X.25 clear time-out).

CLEAR (GATE to CTCP through the Data Session)

The CLEAR command is issued from GATE to the CTCP through the data session. The command is sent in the following format:

```
                              //      //
RU: | cc | resid |ccdf| user | billing |
                              //      //
```

Bytes	Field	Description
0	cc	1-byte command code. For this command, the value is X'13'.
1–2	resid	2-byte resource identifier as specified in the INCOMING CALL or CALL CONNECTED command. The leftmost halfbyte is X'F' for the direction GATE to CTCP, and X'0' for the direction CTCP to GATE.
3–4	ccdf	2-byte clear CAUSE and DIAGNOSTIC fields.
5–n	user	Variable-size field containing optional user data following the CAUSE and DIAGNOSTIC fields.
n–m	billing	Variable-size field containing billing information, if the billing option is requested at X.25 NPSI generation. See page 70 for billing format details.

Note: When the CLEAR command is sent by GATE to the CTCP, a definite response is requested.

CLEAR (CTCP to GATE through the Data Session)

The CLEAR command is issued from the CTCP to GATE through the data session. The command is sent in the following format:

```
    ┌────┬──────┬──//──┐
RU: │ cc │ ccdf │ user │
    └────┴──────┴──//──┘
```

Bytes	Field	Description
0	cc	1-byte command code. For this command, the value is X'13'.
1−2	ccdf	2-byte CAUSE and DIAGNOSTIC fields that are to be inserted in the Clear Request packet.
3−n	user	Variable-size field containing optional user data to be put in the Clear Request packet after the CAUSE and DIAGNOSTIC fields.

CLEAR CONFIRMATION (GATE to CTCP through the Data Session)

The CLEAR CONFIRMATION command is issued from GATE to the CTCP through the data session. The command is sent in the following format:

```
    ┌────┬──//─────┐
RU: │ cc │ billing │
    └────┴──//─────┘
```

Bytes	Field	Description
0	cc	1-byte command code. For this command, the value is X'17'.
1−n	billing	Variable-size field containing billing information, if the billing option is requested at X.25 NPSI generation. See page 70 for billing format details.

RESET (GATE to CTCP, CTCP to GATE through the Data Session)

The RESET command is issued from GATE to the CTCP and from the CTCP to GATE through the data session. The command is sent in the following format:

```
    ┌────┬──────┬──//──┐
RU: │ cc │ rcdf │ user │
    └────┴──────┴──//──┘
```

Bytes	Field	Description
0	cc	1-byte command code. For this command, the value is X'1B'.
1−2	rcdf	2-byte reset CAUSE and DIAGNOSTIC fields.
3−n	user	Variable-size field containing optional user data following the CAUSE and DIAGNOSTIC fields.

If X.25 NPSI detects an unrecoverable situation, GATE sends a RESET command to the CTCP and resets the VC:

Type of Error	Reset CMD to CTCP	Reset Req to Network
Pkt rcvd w/ invalid P(R)	Cause 00 diag ab	Diag ab
Pkt rcvd w/ invalid P(S)	Cause 00 diag ac	Diag ac
Pkt rcvd w/ invalid D-bit	Cause 00 diag ad	Diag ad
Inconsistent Q-bit setting	Cause 00 diag 53	Diag 53
Chain cancelled by APPL	Cause 00 diag b3	Diag b3

RESET CONFIRMATION (GATE to CTCP through the Data Session)

The RESET CONFIRMATION command is issued from GATE to the CTCP through the data session. The command is sent in the following format:

```
    ┌────┬──//─┐
RU: │ cc │ user │
    └────┴──//─┘
```

Bytes	Field	Description
0	cc	1-byte command code. For this command, the value is X'1F'.
1−n	user	Variable-size field containing any optional user data found in the DCE Reset Confirmation packet.

INTERRUPT (CTCP to GATE, GATE to CTCP through the Data Session)

The INTERRUPT command is issued from the CTCP to GATE and from GATE to the CTCP through the data session. The command is sent in the following format:

```
    ┌────┬────┬──//─┐
RU: │ cc │ iu │ user │
    └────┴────┴──//─┘
```

Bytes	Field	Description
0	cc	1-byte command code. For this command, the value is X'23'.
1	iu	1-byte interrupt cause byte.
2−n	user	Variable-size field containing optional user data following the interrupt user data byte.

INTERRUPT CONFIRMATION (CTCP to GATE, GATE to CTCP through the Data Session)

The INTERRUPT CONFIRMATION command is issued from the CTCP to GATE and from GATE to the CTCP through the data session. The command is sent in the following format:

```
    ┌────┬──//─┐
RU: │ cc │ user │
    └────┴──//─┘
```

Bytes	Field	Description
0	cc	1-byte command code. For this command, the value is X'27'.
1−n	user	Variable-size field containing optional user data following the packet header.

DIAGNOSTIC (GATE to CTCP through the Control Session)

The DIAGNOSTIC command is issued from GATE to the CTCP through the control session. The command is sent in the following format:

```
RU: | cc | dc | od //|
```

Bytes	Field	Description
0	cc	1-byte command code. For this command, the value is X'F1'.
1	dc	1-byte diagnostic code.
2–n	od	Variable-size field containing diagnostic explanation.

ERROR/INFORMATION REPORT (GATE to CTCP through the Data Session)

The ERROR/INFORMATION REPORT command is issued from GATE to the CTCP through the data session. The command is sent in the following format:

```
RU: | cc | resid | ce | ec | billing //|
```

Bytes	Field	Description
0	cc	1-byte command code. For this command, the value is X'FF'.
1–2	resid	2-byte resource identifier.
3	ce	1-byte command code for the command in error.
4	ec	1-byte error cause.
5–n	billing	Variable-size field containing billing information, if the billing option is requested at X.25 NPSI generation. See page 70 for billing format details.

The only error cause that currently exists is X'30'. The timer elapsed before a response from the network was received for the control packet.

Although the CTCP cannot send CLEAR CONFIRMATION or RESET CONFIRMATION commands, the CTCP must be able to receive these commands. X.25 NPSI automatically sends out CLEAR CONFIRMATION and RESET CONFIRMATION commands.

X.21 Switched Connections Using GATE (V3R3 Only)

X.25 NPSI provides a connection to an integrated services digital network (ISDN) over a switched connection. To provide this connection capability, X.25 NPSI uses two logical interfaces.

- X.25 NPSI uses the standard GATE to CTCP upstream interface.
- X.25 NPSI uses a new downstream interface between X.25 NPSI Link Access Control and the X.21 switched function of the TSS to control the X.21 physical line.

Control Session Establishment

From the CTCP point of view, session establishment for an X.21 switched connection using GATE is similar to session establishment for a non-X.21 switched connection using GATE. See "Session Establishment through a Switched Virtual Circuit" on page 50 for more information.

First, the CTCP activates the control session with the physical circuit (CTCP to MCH_LU). When the CTCP to MCH_LU session is established, a session between the CTCP and a remote DTE can be established over a switched virtual circuit in one of the following ways:

1. A CALL REQUEST command from the CTCP.
2. An X.25 Incoming Call packet.

   The X.25 Incoming Call packet is received from the remote DTE, after the establishment of an X.21 Incoming connection on a physical interface.

Note: Only switched virtual circuits are supported on the MCH that is dedicated to X.21 connections.

Control sessions can be established by the CTCP using the set of requests described for GATE non-X.21 switched connections. See "Command Interface between GATE and the CTCP" on page 53 for more information.

At the end of the control session establishment, X.25 NPSI sets the physical interface in MONITOR INCOMING CALL state to detect X.21 Incoming calls.

Virtual Circuit Establishment

The following describes the establishment procedures for X.21 Incoming and Outgoing calls.

X.21 Incoming Call

When an X.21 Incoming Call signal is detected, the X.21 physical connection is performed by the TSS. Only after a successful X.21 connection, can X.25 NPSI perform the X.25 link level and packet level setup. X.25 NPSI then waits for an X.25 Incoming Call packet to be received. When a valid X.25 Incoming Call packet is received, the virtual circuit session is established.

If a valid X.25 Incoming Call packet is not received within the time specified on the X21INACT keyword of the X25.MCH statement, the X.21 connection is cleared.

X.21 Outgoing Call

When the CTCP wants to establish a session through a switched virtual circuit, it sends a CALL REQUEST command to GATE, which initiates the X.21 Outgoing Call. The called DTE address, which is contained within the X.25 Call Request packet, contains the X.21 dial digits that are used to perform X.21 Outgoing calls. These dial digits represent the destination of the call.

When the X.21 outgoing connection is successful, X.25 NPSI performs the X.25 link level and packet level setup.

Only when the link level and packet level are set up does GATE resume the processing of the CALL REQUEST command. The CALL REQUEST command is then processed by GATE as it would be processed on a non-X.21 connection.

CALL REQUEST Commands (CTCP to GATE)

Commands received during or after an X.21 Outgoing connection: While X.25 NPSI is establishing an X.21 Outgoing connection, any subsequent CALL REQUEST commands are refused until packet level setup is complete (X.25 Restart packets exchange completed).

Once the packet level is set up, X.25 NPSI compares the X.21 dial digits that are received in each subsequent CALL REQUEST command with the digits used for the X.21 Outgoing connection. If the X.21 dial digits are different, X.25 NPSI refuses the CALL REQUEST command by sending a negative response to the CTCP. If the X.21 dial digits match, the CALL REQUEST command is processed by GATE the same way a CALL REQUEST command would be processed on a non-X.21 connection.

- A free virtual circuit number is assigned to the call.
- The X.25 Call Request packet is sent on the SVC to the remote DTE.

Commands received during or after an X.21 Incoming connection: Once X.25 NPSI detects the X.21 Incoming Call signal, it refuses any CALL REQUEST commands that the CTCP sends until the X.25 link level and packet level are set up. To refuse these commands, X.25 NPSI returns negative responses to the CTCP.

Because calling party identification is not supported on X.21 switched connection, the origin of the call is not known by X.25 NPSI. In this case, checking of X.21 dial digits is not performed on CALL REQUEST commands following an X.21 incoming connection.

Once the packet level is set up, X.25 NPSI processes all CALL REQUEST commands received from the CTCP, as would occur in a typical CTCP to GATE operation.

Virtual Circuit Session Termination

A virtual circuit can be terminated by using the set of requests described for GATE non-X.21 switched connections. See "Virtual Circuit Termination" on page 51 for more information.

At the end of deactivation of a virtual circuit session on an X.21 switched connection, X.25 NPSI checks if it was the last VC active on the line.

- If the answer is yes, X.25 NPSI:
  - Disconnects the X.25 link level.
  - Releases the X.21 connection.
  - Sets the physical interface in MONITOR INCOMING CALL state.
- If the answer is no, X.25 NPSI resumes the process of virtual circuit deactivation as it would occur for a non-X.21 connection.

MCH Failure Handling for X.21 Connections

An MCH INOP occurs if MKBMDREC equals 1x, 2x, Cx, Dx, Ex, or Fx (except FA or FB). The LU becomes inoperable, and the session with the CTCP fails.

If MKBMDREC equals 3x, an X.25 link reinitialization occurs on the MCH. If MKBMDREC equals FA or FB, an X.21 disconnection occurs on the MCH. In both cases, the MCH PU and MCH LU remain active, and the control session with the CTCP does not fail. The X.21 connection is cleared, and the physical interface is set again to MONITOR INCOMING CALL state. However, the active virtual circuit PUs are all made inoperable.

Command Interface between GATE and the CTCP

The commands recognized or generated by GATE for X.21 switched connections are the same as the commands used by GATE for non-X.21 switched connections. See Table 5 on page 54 for a list of the commands.

Figure 15 on page 83 illustrates an X.21 switched connection using an IBM 7820 terminal adapter.

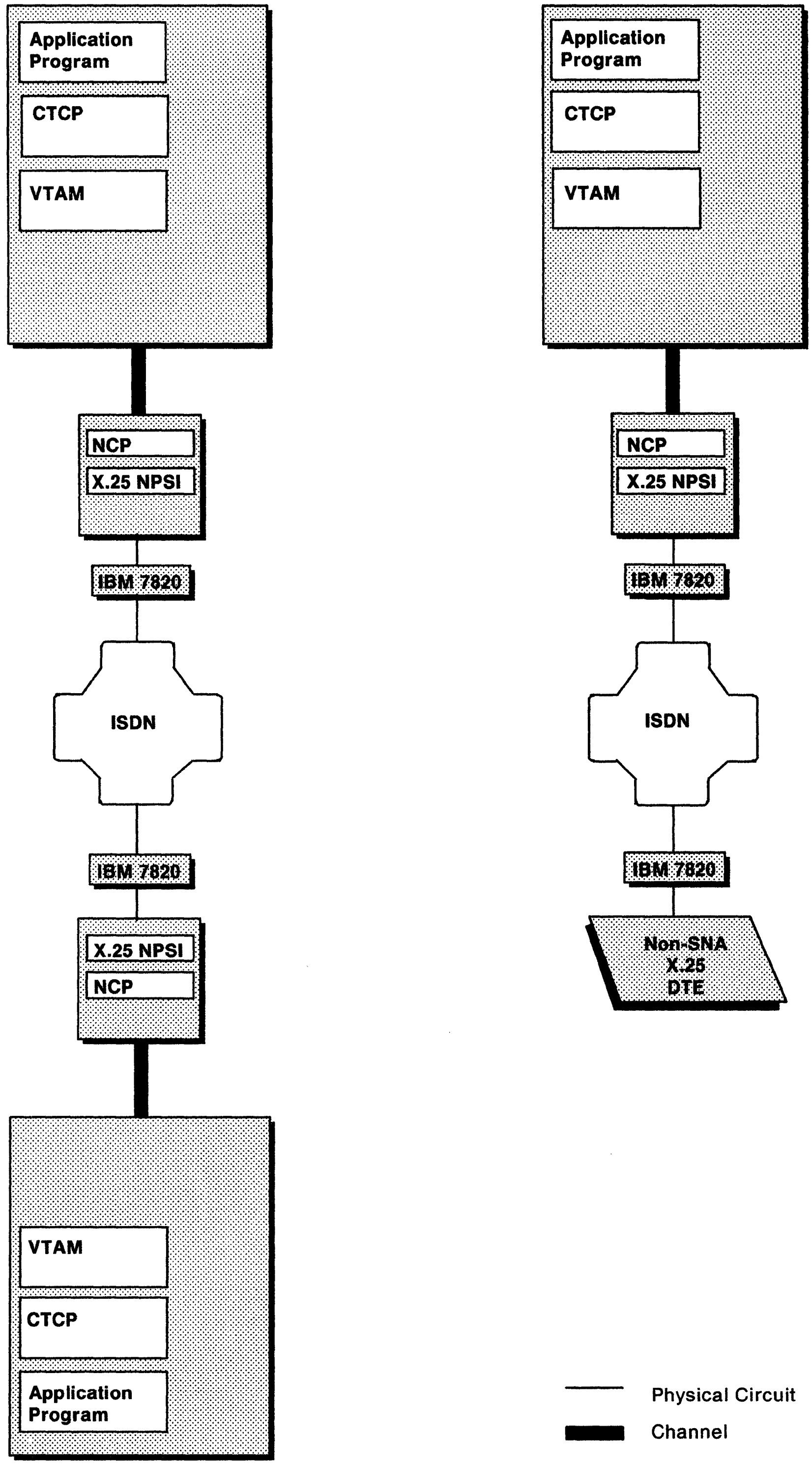


Figure 15. SNA Host to ISDN (X.21 Switched Connection)

Multiple CTCPs

If more than one CTCP is specified in an X25.MCH statement, the MCH has more than one LU. The CTCPs can reside in the same host or in different hosts. If different hosts are used, the LOGAPPL command might not be effective, and a CLIST will be needed to automatically put the LUs in session with the CTCPs. Each MCH LU is in session with a CTCP. The selection of the CTCP is based on subaddressing or CUD byte 0, depending on the option coded in the X25.MCH.

The number of LUs defined in each CTCP does not have to equal the number of virtual circuits defined on this MCH. Once the appropriate CTCP is selected, the interface is the same as when a single CTCP is present.

Chapter 4. Programming of a DATE CTCP

Chapter 4. Programming of a DATE CTCP

This chapter explains how to program virtual circuits using the DATE function of X.25 NPSI.

DATE Function of X.25 NPSI

With the DATE function of X.25 NPSI, you can use all the LLC types, except LLC type 4 (GATE) and the integrated PAD support option of LLC type 5. You can use these LLC types to gain greater control of X.25 NPSI control information than is possible with just X.25 NPSI LLC functions. The DATE CTCP processes all control packets. This feature provides maximum flexibility in the processing of these packet types. The control packets include virtual circuit setup and termination, as well as Interrupt, Reset, and qualified data packets.

Figure 16 on page 88 shows the data flow for a network configuration that uses DATE.

For V3R3 only: You can use the integrated PAD support option of LLC type 5.

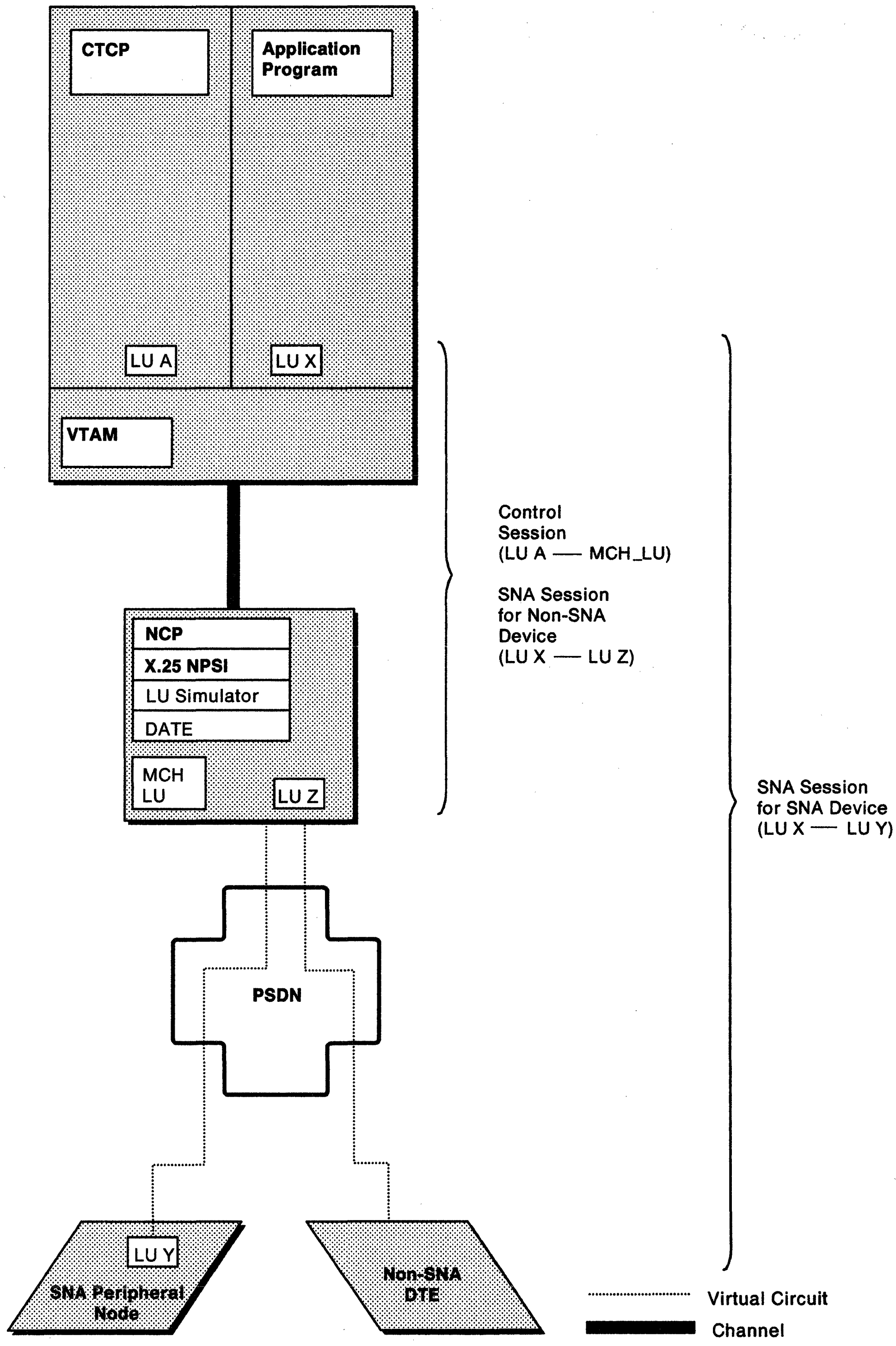


Figure 16. SNA Host Node to SNA Peripheral Node or Non-SNA DTE (DATE)

CTCP Interface with DATE

The type of data handled by the application program is determined by the virtual circuit type:

- Type 0

  The application program sends and receives data packets. The CTCP sends or receives Restart, Call, Clear, Reset, Interrupt, and qualified data packets.

- Type 2

  The application program sends and receives the data packets. The CTCP sends or receives Restart, Call, Clear, and Reset packets. Interrupt and qualified data packets are discarded.

- Type 3

  The application program sends and receives data packets. The CTCP sends or receives Restart, Call, Clear, and Reset packets. Interrupt packets are discarded, and qualified data packets are handled by X.25 NPSI.

- Type 5

  The application program sends or receives all packets except Restart, Call, and Clear packets. These packets are handled by the CTCP.

Determination of Virtual Circuit Type

The CTCP communicates the virtual circuit type (LLC type) to X.25 NPSI through the use of byte 6 of the CALL REQUEST or CALL ACCEPTED command. The value used in this byte is equal to the value used in the first byte of the call user data (CUD) field of the Call Request packet to select LLCs. The hexadecimal code for each LLC type is:

Hex Code	Virtual Circuit Type
X'C0'	LLC0
X'C2'	LLC2
X'C3'	LLC3
X'E3'	LLC3
X'C4'	LLC4
X'01'	LLC5
X'41'	LLC5
X'51'	LLC5
X'81'	LLC5.

Notes:

1. A type 5 virtual circuit using the DATE function *must* use the transparent PAD support function.
2. SVCSC is not supported on an MCH under DATE control.

For V3R3 only: A type 5 virtual circuit using the DATE function can use either the transparent or integrated PAD support function.

PLP Timers

The CTCP is responsible for managing the packet level processor (PLP) states and the PLP timers. When you are not using DATE, the default values used by X.25 NPSI for these timers are the values defined in the Recommendation X.25. You can change these values when you program a CTCP; or by using the T20, T21, T22, and T23 keywords on the X.25 NET statement. The defaults are:

- Restart Request time-out (T20 timer)

  This timer is started by the CTCP when a RESTART REQUEST command is sent to X.25 NPSI. The response must arrive within 180 seconds.

- Call Request time-out (T21 timer)

  This timer is started by the CTCP when a CALL REQUEST command is sent to X.25 NPSI. The response must arrive within 200 seconds.

- Reset Request time-out (T22 timer)

  This timer is started by the CTCP when a RESET REQUEST command is sent to X.25 NPSI. The response to this request must arrive within 180 seconds.

- Clear Request time-out (T23 timer)

  This timer is started by the CTCP when a CLEAR REQUEST command is sent to X.25 NPSI. The Clear Confirmation response must be received within 180 seconds.

Note: Only the applicable timers are used for a specific virtual circuit, because not all of the control packets are passed to the CTCP for all virtual circuit types.

Control Session Establishment

To communicate with a PSDN through DATE, the CTCP must first establish a session with the MCH LU. This session is a type 1 LU-LU session. Only after the START DATA TRAFFIC command is exchanged is the CTCP able to monitor the DATE function. The format of the BIND SESSION command is the same as the GATE format.

Note: Half-duplex contention mode must be used on the CTCP to MCH LU session.

When the CTCP initiates session setup, the CTCP starts by sending a RESTART command to X.25 NPSI. The T20 timer associated with this command must be started by the CTCP. Upon receipt of the RESTART command, X.25 NPSI clears all the virtual circuits associated with the physical circuit being restarted and initiates the link setup procedure.

The link setup procedure is executed in two steps:

1. Link level initiation is accomplished by the following:

   - X.25 NPSI sends a DISC (disconnect) frame to the DCE and waits for a DM (disconnect mode) frame or a UA (unnumbered acknowledgment) frame.

   - X.25 NPSI transmits an SABM (set asynchronous balanced mode) frame to the DCE and waits for a UA frame to be returned.

   - The DCE positively responds with a UA frame. At this point, the link level setup is complete.

2. Packet level initiation is accomplished by the following:
   - DATE sends the Restart packet to the network.
   - The DCE can respond with a Restart Confirmation packet, or it might have already sent a Restart packet.
   - The packet received is passed on to the CTCP.
   - If the T20 timer has not expired, it is now stopped.

The CTCP must initiate setup of the physical circuit dedicated to the DATE function by sending a RESTART command. When link setup is performed correctly, DATE sends the Restart Request packet corresponding to the command passed by the CTCP.

The response from the PSDN is passed on to the CTCP as a RESTART CONFIRMATION command or a RESTART command (in restart collision). In the case of restart collision, the CTCP does not need to respond with a RESTART CONFIRMATION command when it receives a RESTART command. When the CTCP receives a RESTART command or a RESTART CONFIRMATION command, it must then stop the T20 timer.

Once the control session with the physical circuit is established, the virtual circuits can be established. At this point, the virtual circuits are ready to operate as specified by the user application.

Virtual Circuit Establishment

For switched virtual circuits (SVCs), the CTCP must determine whether a user application program is ready to be called or is waiting to call out. The CTCP must know the LLC type used with a given application program. The CTCP passes this information to DATE in the CALL REQUEST or CALL ACCEPTED command. DATE then updates the corresponding control blocks in X.25 NPSI to allow the usual procedure at the data level.

Call-In

When DATE receives an Incoming Call packet, it transfers the packet to the CTCP as an INCOMING CALL command, along with an indication of the appropriate SNA resource ID. Using the INCOMING CALL command, the CTCP determines the type of virtual circuit to be set up. The CTCP also selects the packet length and packet window size to be used, based on the PSDN subscription parameters. Then, the CTCP passes this information to DATE along with the application PLU name (if appropriate) through a CALL ACCEPTED command. DATE translates the CALL ACCEPTED command into a Call Accepted packet and forwards it to the PSDN.

Call-Out

When an application program or operator requests that the CTCP connect an application with a destination, the CTCP sends a CALL REQUEST command to DATE. In this command, the CTCP passes on all the necessary information: virtual circuit type, packet length, packet window size, and an application PLU name for logon (if appropriate). The CTCP also sends a connection identifier that is used to associate the resource ID that is passed to the CTCP in the CALL CONNECTED command, with the CALL REQUEST command.

Upon receipt of the CALL REQUEST command, the DATE code in X.25 NPSI selects a free virtual circuit to send a Call Request packet to the PSDN. X.25 NPSI then associates that virtual circuit with an SNA resource set. The CALL REQUEST command passed by the CTCP is forwarded by X.25 NPSI to the PSDN.

If the connection is successful, the PSDN sends a Call Connected packet back to X.25 NPSI. DATE then transfers the packet to the CTCP in a CALL CONNECTED command. The CALL CONNECTED command associates the CTCP connection identifier with the resource ID associated with the virtual circuit for the duration of the call. Thus, the CTCP is aware of the resource ID that is used for this connection.

At this time, DATE causes the NCP to send a REQUEST CONTACT request unit to VTAM for that virtual circuit. Thus, a connection is established directly between the user application program and the corresponding remote DTE over the chosen virtual circuit.

Once a virtual circuit is active, the application program communicates data RUs *directly* with the remote DTE, without going through the CTCP or DATE.

Notes:

1. Call request is simulated by an incoming call to NCP and the access method. Consequently, no PATH statement is required in the SMN for a DATE call request.
2. A connection to a virtual circuit attached to a DATE MCH must be performed under the control of the CTCP. If an application tries to connect out directly through the access method on such a virtual circuit, DATE generates an INOP message for the involved virtual circuit.
3. The LLC type is determined by the CTCP. With DATE, byte 0 of the CUD field is not used by X.25 NPSI to select the LLC type. The LLC type is passed to X.25 NPSI through byte 6 of the CALL REQUEST or CALL ACCEPTED command. Similarly, the packet size and window size are determined by the CTCP. Their values are passed to X.25 NPSI in bytes 3, 4, and 5 of the CALL REQUEST or CALL ACCEPTED command.

   In the case of a CALL REQUEST sent by the CTCP, packet and window sizes specified in the CALL REQUEST command are taken into account by X.25 NPSI. The flow control parameters contained in the facilities of the Call Connected packet are ignored by X.25 NPSI. The CTCP should clear the call and try again if these parameters do not match the flow control parameters that were passed to X.25 NPSI in the CALL REQUEST command.

   For V3R3 only: Depending on the generation option, the flow control parameters included in the facilities of the Call Connected packet are taken into account by X.25 NPSI.

If the application is to decide when to initiate a connection, it must send a message to the CTCP, either directly or through the operator, to ask the CTCP to issue a CALL REQUEST. The application cannot directly trigger a CALL REQUEST through an OPNDST OPTCD=ACQUIRE, or a SIMLOGON followed by an OPNDST OPTCD=ACCEPT, as would a normal application.

An SNA session using the virtual circuit can be established in one of the following ways:

- A LOGON command is generated by DATE for LLC0 and LLC5 virtual circuits.
- A remote user enters LOGON or an equivalent command.
- A logon is performed automatically because of LOGAPPL coded on the LU statement at call setup.
- An operator issues a V NET,LOGON=*application* command.
- An operator issues a V NET,ACT command, and LOGAPPL=*application* is coded on the LU statement.
- CLIST does a V NET,LOGON=*application* when the LU is active.

Logging on to the Application

For DATE, if either LLC0 or LLC5 is selected for the call, the CTCP can request that X.25 NPSI build a logon with a specified application. In this case, X.25 NPSI builds the logon at call confirmation time.

When the CALL ACCEPTED command or the CALL REQUEST command carries an application name (when byte 7 is not equal to 0), DATE builds a logon message with the following format:

```
LOGON APPLID(XXXXXXXX) DATA(YYYYZZZZZZZZ)
```

where:

XXXXXXXX	Is the application program name given in the CALL REQUEST or CALL ACCEPTED command (up to 8 characters).
YYYY	Is the 2-byte SNA resource ID.
ZZZZZZZZ	Is the physical circuit LU name (up to 8 characters; padded to the right with blanks if less than 8 characters).

This logon request is issued to the access method after the ACTLU command is processed for the involved LU associated with the virtual circuit for the duration of the call. When any of the following conditions are true, all data is queued until start data traffic (SDT) is received:

- MAXOUT=6 is coded on the X25.PU statement or the X25.VC statement for PVCs.
- MAXOUT=6 is coded on the PU statement of the switched major node for SVCs.
- The logon is generated by X.25 NPSI.

Once SDT is received, all data traffic is passed to the application.

Protocol for a Type 0 Virtual Circuit

When a switched virtual circuit using LLC type 0 is requested (X'C0' in byte 6 of the CALL REQUEST command or CALL ACCEPTED command), the logon message can be either the first data received by DATE from the remote DTE or a logon message built by DATE. In the second case, the CTCP gives the name of the application program to DATE in the CALL REQUEST or CALL ACCEPTED command.

The logon message is sent as data from the selected LU attached to the virtual circuit for the duration of the call, through the access method, to the LU of the application program. The logon exit routine is scheduled. Execution of the VTAM OPNDST macro that is coded with OPTCD=ACCEPT in the application program causes the SNA BIND SESSION and START DATA TRAFFIC commands to flow. These commands establish the LU-LU session associated with the selected switched virtual circuit.

The application program then works under standard PCNE protocol, except when control packets, Interrupt packets, or qualified packets are received. The DATE code transmits the control, Interrupt, and qualified packets to the CTCP for processing on the MCH LU to CTCP LU session. DATE acts only on the Restart and Clear packets.

Protocol for a Type 2 Virtual Circuit

A switched virtual circuit LLC type 2 is requested by X'C2' in byte 6 of the CALL REQUEST or CALL ACCEPTED command. After the call exchange is performed, the switched virtual circuit is available to the application program, as in the case of a permanent type 2 virtual circuit.

The CTCP must establish an interface with the user application program to synchronize the different activities. A programmed operator interface should be developed with the CTCP.

Protocol for a Type 3 Virtual Circuit to SNA Peripheral Node

A switched virtual circuit LLC type 3 is requested by X'C3' in byte 6 of the CALL REQUEST or CALL ACCEPTED command. After the call exchange is performed, the switched virtual circuit is available to the application program, as in the case of a permanent type 3 virtual circuit.

The CTCP must establish an interface with the user application program to synchronize the different activities. A programmed operator interface should be developed with the CTCP.

For all type 3 virtual circuits, Interrupt packets and qualified packets are not authorized on the CTCP interface. The Interrupt packets are discarded; the qualified packets are processed by X.25 NPSI, as if the MCH was not under DATE.

Protocol for a Type 3 Virtual Circuit to SNA Subarea Node

For all switched virtual circuits using LLC type 3, Interrupt packets and qualified packets are not authorized on the CTCP interface. The Interrupt packets are discarded; the qualified packets are processed by X.25 NPSI, as if the MCH was not under DATE.

SVCSC is not supported on a DATE MCH, and only permanent virtual circuits are supported.

Protocol for a Type 5 Virtual Circuit

When a switched virtual circuit using LLC type 5 is selected (X'01', X'41', X'51', or X'81' in byte 6 of the CALL REQUEST or CALL ACCEPTED command), it is treated as if it were a type 0 virtual circuit for the logon message processing. Following the logon, the standard processing for a transparent PAD virtual circuit occurs. During the session, Reset, Interrupt, and qualified packets are handled directly by the application.

Note: Only transparent PAD is supported.

For V3R3 only: V3R3 supports both transparent and integrated PAD support function. When a switched virtual circuit using the integrated PAD support function of LLC type 5 is selected, the switched virtual circuit is treated as if it were a type 0 virtual circuit for the logon message processing. Following the logon request and upon receipt of Interrupt, Reset, or qualified packets with BREAK specified, X.25 NPSI sends an SNA SIGNAL to the application and confirmation to the network. See "SIGNAL/BREAK Processing (V3R3 Only)" on page 34 for more information. During the session, Interrupt, Reset, and qualified packets (because of a BREAK entered at the terminal) are not authorized on the CTCP interface.

Negotiation

The DCE can negotiate with the CTCP for the packet length and window size that apply to this virtual circuit. DATE keeps the values initially defined by the CTCP. The CTCP must analyze the CALL CONNECTED command to check whether a negotiation was performed, and to clear the connection, if necessary, before sending a new CALL REQUEST command containing negotiated values to update DATE.

For V3R3 only: Depending on the generation option specified, the flow control parameters included in the facilities of the Call Connected packet are taken into account by X.25 NPSI.

Virtual Circuit Termination

A virtual circuit can be deactivated using any of the following requests and commands:

- Clear Request from remote DTE
- CLSDST from the application program
- Request from the application program for the CTCP to CLEAR
- V NET,INACT command
- V NET,TERM command.

In the simplest case, the remote DTE requests the session termination. In this instance:

- A Clear Request packet is received from the PSDN, and X.25 NPSI transfers this information to the CTCP as a CLEAR REQUEST command.
- The CTCP returns a CLEAR CONFIRMATION command to X.25 NPSI.
- X.25 NPSI then generates an INOP for the VC LU.

The user application program can initiate session termination by issuing the CLSDST macroinstruction. In this case:

- X.25 NPSI receives an ABCONN request from the NCP and forwards an INFORMATION REPORT to the CTCP with a cause field of X'07'.
- The CTCP then sends a CLEAR command to X.25 NPSI that is forwarded to the PSDN.
- When the Clear Confirmation packet returns from the PSDN, it is passed to the CTCP as a CLEAR CONFIRMATION command by X.25 NPSI. At this point, deactivation is complete, and the SNA and X.25 resources are disassociated.

The user application program can notify the CTCP to disconnect the virtual circuit. When the CTCP receives this request:

- The CTCP builds a CLEAR command, including the cause and diagnostic bytes as determined by the CTCP.
- The CLEAR command is sent to X.25 NPSI.
- X.25 NPSI then sends a Clear Request packet to the PSDN.
- When the Clear Confirmation packet is received, X.25 NPSI transfers it to the CTCP in the CLEAR CONFIRMATION command.
- X.25 NPSI then generates an INOP to be sent to VTAM.
- The controlling application program's LOSTERM exit is scheduled, and the application program issues a CLSDST macroinstruction. At that point, the virtual circuit is freed and available for reuse.

The network operator can terminate the session by issuing a V NET,INACT command for the VC PU or the VC LU or a V NET,TERM command for the VC LU. When this command is issued, an Information Report message is sent to the CTCP. The CTCP must issue a CLEAR command to resynchronize communication between the CTCP and X.25 NPSI.

Control Session Termination

The DATE control session can be terminated in any of the following ways:

- The application program asks the CTCP to disconnect from the MCH LU.
- The operator issues the CANCEL command to cancel the CTCP.
- The operator issues the V NET,INACT,ID=*mch lu* or the V NET,TERM,LU=*mch lu* command to terminate the session with the physical circuit LU. *mch lu* is the name of the physical circuit LU.
- The CTCP abends.

Notes:

1. If the CTCP is canceled, abends, or is inactivated (through the V NET,INACT or V NET,TERM command), the switched major nodes used for communication with this CTCP must be deactivated and reactivated. This deactivation and reactivation clears any condition that is pending on the involved virtual circuits.
2. If the CTCP is terminated, the MCH LINK status is changed to INOP. The MCH LINK must be reactivated before the MCH can be reused.

Command Interface between DATE and the CTCP

Control and qualified data commands are exchanged between the CTCP LU and the MCH LU. With the exception of commands flowing on type 5 virtual circuits, commands exchanged between the application program LU and the LU associated with the virtual circuit do not contain command codes.

Command codes use conventions defined in the X.25 protocol. The command code is always in byte 2 of an RU flowing on a DATE session. For data exchange, byte 2 can have the value X'00', indicating data exchange without the Q bit, or the value X'02', indicating data exchange with the Q bit.

When control commands from the CTCP to DATE contain optional user data, that data is put into the resulting control packet without verification by DATE. Segmenting using the M bit in the packet header is *not* performed for control packets. For this reason, all user data within control packets must fit into a single packet. For qualified or unqualified data packets, user data can be spread over several packets.

For control commands, the other bytes of the RU also have significance to the CTCP and DATE. When sending commands to DATE, the CTCP must adhere to the command formats described in this section. For commands going from DATE to the CTCP, the CTCP uses these formats to interpret the command. The CTCP establishes one session with the DATE function within X.25 NPSI to handle the exchange of the following commands. These commands are contained in RUs that flow through the SNA session between the CTCP and the MCH LU.

Table 7 shows the commands that are processed by the CTCP over the physical circuit LU (MCH LU) session for each of these virtual circuit types.

Table 7. Commands on Physical Circuit LU Session

Commands	**Type 0**	**Type 2**	**Type 3 (Peripheral Node)**	**Type 3 (Subarea Node)**	**Type 5**
CALL REQUEST and INCOMING CALL	Yes *	Yes *	Yes *	No	Yes *
CALL ACCEPTED and CALL CONNECTED	Yes *	Yes *	Yes *	No	Yes *
CLEAR REQUEST and CLEAR INDICATION	Yes *	Yes *	Yes *	No	Yes *
CLEAR CONFIRMATION	Yes *	Yes *	Yes *	No	Yes *
DIAGNOSTIC	Yes	Yes	Yes	Yes	Yes
RESTART REQUEST and RESTART INDICATION	Yes	Yes	Yes	Yes	Yes
RESTART CONFIRMATION	Yes	Yes	Yes	Yes	Yes

Note: The commands followed by an asterisk (*) are processed by the CTCP on the MCH LU session *only* for switched virtual circuits.

Table 8 on page 98 shows, by virtual circuit type, the commands processed on the application program to virtual circuit LU (VC LU) session and those processed on the CTCP to physical circuit LU (MCH LU) session.

Table 8. Commands on Virtual Circuit or Physical Circuit LU Session

Commands	Type 0	Type 2	Type 3 (Peripheral Node)	Type 3 (Subarea Node)	Type 5
RESET REQUEST and RESET INDICATION	MCH LU	MCH LU	MCH LU	MCH LU	VC LU *
RESET CONFIRMATION	MCH LU	MCH LU	MCH LU	MCH LU	VC LU *
INTERRUPT REQUEST	MCH LU	No	No	No	VC LU *
INTERRUPT CONFIRMATION	MCH LU	No	No	No	VC LU *
QUALIFIED DATA	MCH LU	No	No	No	VC LU *
UNQUALIFIED DATA	MCH LU or VC LU	MCH LU or VC LU	MCH LU or VC LU	MCH LU or VC LU	MCH LU or VC LU

Note: The commands followed by an asterisk (*) are carried on the data session between the host application and the remote DTE. They are processed by X.25 NPSI transparent PAD support. The format of these commands is the same as the format of the transparent PAD RUs on page 41.

For V3R3 only: When using integrated PAD, RESET INDICATION and RESET CONFIRMATION commands can flow either on the VC LU or on the MCH LU session.

Identifiers

The connection identifier (CNID) referred to in the following command descriptions is a 2-byte field. It is chosen by the CTCP, when the CTCP makes a CALL REQUEST. It is then used by the CTCP to correlate the CALL REQUEST with the subsequent commands (CALL CONNECTED or CLEAR) related to the CNID. This field is used temporarily during the CALL REQUEST setup. Once the call is complete, it is no longer used.

The resource ID described in the following paragraph is used when the call is complete. The leftmost halfbyte of the 2-byte field is X'0', for outbound and inbound commands, except for the CLEAR command from the CTCP after a CALL REQUEST.

The resource ID (RESID) field designates the set of SNA resources (LINE, PU, and LU) that are associated with either a virtual circuit (VC), or the X.25 VC number (systems generation option VCID). SNA resource IDs are associated with SVCs for each MCH. The SNA resource IDs correspond to the number of SNA resources within the set of SNA resources associated with the MCH. The range for the resource ID numbers is the same as the range for the virtual circuit numbers. An SNA resource number consists of 3 digits in hexadecimal format. The first digit maps the logical group number, and the next two digits map the logical channel numbers.

The connection identifier (CNID) and resource ID must have different ranges to avoid problems at Call Collision.

The example on page 98 shows how the X25.MCH statement can be coded to define SVCs. In this example, SNA IDs X'003' to X'063' and SNA IDs X'204' to X'262' are used to map all SVCs of this MCH. This setup allows CTCPs to continue to run without change.

```
X25.LCG  LCGN=0
X25.VC   LCN=(1,2),TYPE=P ....  2 PVCs
X25.VC   LCN=(3,99),TYPE=S ... 97 SVCs
X25.LCG  LCGN=2
X25.VC   LCN=(1,3),TYPE=P ....  3 PVCs
X25.VC   LCN=(4,98),TYPE=S ... 95 SVCs
```

The following descriptions show data exchange commands and control commands. The data exchange commands are followed by user data. The control commands are followed by additional control information.

DATA EXCHANGE without Q Bit (CTCP to DATE through the Control Session)

Data without a Q bit can be sent out from the CTCP to DATE through the control session. The command is sent in the following format:

```
         ┌──────────┬─────┬─//──┐
RU:      │ resid    │ cc  │ user │
         └──────────┴─────┴─//──┘
```

Bytes	Field	Description
0–1	resid	2-byte SNA resource identifier.
2	cc	1-byte command code. For this command, the value is X'00'.
3–n	user	Variable-size field containing user data to be sent to the remote terminal. For example, this field can be used at connection time to ask the remote user not to clear the connection.

Note: For more information on this data exchange, see DATE Message on page 111.

DATA EXCHANGE with Q Bit (CTCP to DATE, DATE to CTCP through the Control Session)

A DATA EXCHANGE command with a Q bit is issued from the CTCP to DATE and from DATE to the CTCP through the control session for LLC type 0. (Exchange occurs on the data session for LLC type 5.) The command is sent in the following format:

```
         ┌──────────┬─────┬─//──┐
RU:      │ resid    │ cc  │ user │
         └──────────┴─────┴─//──┘
```

Bytes	Field	Description
0–1	resid	2-byte SNA resource identifier.
2	cc	1-byte command code. For this command, the value is X'02'.
3–n	user	Variable-size field containing user data related to this packet. For example, this field can be used to control a remote PAD.

Note: If you are using transparent PAD, see the command formats for the transparent PAD RUs on page 41.

CALL REQUEST (CTCP to DATE)

A CALL REQUEST command is issued from the CTCP to DATE. The command is sent in the following format:

```
         ┌────┬──┬──┬────┬──┬──┬────//┐
RU:      │cnid│cc│pw│psiz│vc│al│ crpk │
         └────┴──┴──┴────┴──┴──┴────//┘
```

Bytes	Field	Description
0–1	cnid	2-byte connection identifier contained in bytes 0 and 1.
2	cc	1-byte command code contained in byte 2. For this command, the value is X'0B'.
3	pw	Packet window size in hexadecimal contained in byte 3.
4–5	psiz	Packet size in hexadecimal used for this virtual circuit contained in bytes 4 and 5.
6	vc	The type of virtual circuit to be set up. Code in byte 6 the corresponding hexadecimal code shown next to the virtual circuit type in the following list: **Hexadecimal Code** — **Virtual Circuit Type** X'C0' — 0 X'C2' — 2 X'C3' — 3 (to SNA peripheral node) X'01', X'41', X'51', or X'81' — 5. **Notes:** 1. With a type 5 virtual circuit, only transparent PAD support can be used. 2. A type 4 virtual circuit cannot be specified. 3. **For V3R3 only:** With a type 5 virtual circuit, transparent PAD or integrated PAD support can be used.
7	al	The application name length (X'00' through X'08') contained in byte 7. A length of X'0' means that DATE must not generate the LOGON command.
8–n	crpk	Variable-size field starting in byte 8 that contains the application name followed by the Call Request packet (without the packet header).

CALL CONNECTED (DATE to CTCP)

The CALL CONNECTED command is issued from DATE to the CTCP. The command is sent in the following format:

RU:

resid	cc	cnid	ccpk

Bytes	Field	Description
0−1	resid	2-byte resource identifier contained in bytes 0 and 1. The left-most halfbyte is X'F'.
2	cc	1-byte command code contained in byte 2. For this command, the value is X'0F'.
3−4	cnid	2-byte connection identifier contained in bytes 3 and 4. This connection identifier matches the one given in bytes 0 and 1 of the corresponding CALL REQUEST command.
5−n	ccpk	Variable-size field starting in byte 5 that contains the Call Connected packet (without the packet header).

For V3R3 only: Depending on the generation option specified, the flow control parameters included in the facilities of the CALL CONNECTED packet are taken into account by X.25 NPSI.

INCOMING CALL (DATE to CTCP)

The INCOMING CALL command is issued from DATE to the CTCP. The command is sent in the following format:

RU:

resid	cc	icpk

Bytes	Field	Description
0−1	resid	2-byte resource identifier contained in bytes 0 and 1. The left-most halfbyte is X'F'. See note below.
2	cc	1-byte command code contained in byte 2. For this command, the value is X'0B'.
3−n	icpk	Variable-size field starting in byte 3 that contains the Call packet received from the network (without the 3-byte packet header).

Note: In the case of a Call Collision detected by X.25 NPSI, X.25 NPSI sends the Call Request to the network, and when X.25 NPSI receives the Incoming Call, it passes the Incoming Call to the CTCP. This Incoming Call contains a 2-byte connection identifier in bytes 0 and 1. The first digit is X'0', so that the CTCP can distinguish it from a normal Incoming Call. The CTCP can then discard this INCOMING CALL command (under normal circumstances, the network has already discarded the Incoming Call) and wait for the CALL CONNECTED command, or it may send a CLEAR REQUEST command that clears the Call Request.

CALL ACCEPTED (CTCP to DATE)

The CALL ACCEPTED command is issued from the CTCP to DATE. The command is sent in the following format:

```
     ┌───────┬────┬────┬──────┬────┬────┬──//──┐
RU:  │ resid │ cc │ pw │ psiz │ vc │ al │ capk │
     └───────┴────┴────┴──────┴────┴────┴──//──┘
```

Bytes	Field	Description
0−1	resid	2-byte resource identifier contained in bytes 0 and 1. This resource identifier matches the one given in bytes 0 and 1 of the corresponding INCOMING CALL command from DATE.
2	cc	1-byte command code contained in byte 2. For this command, the value is X'0F'.
3	pw	Packet window size contained in byte 3.
4−5	psiz	Packet size, in hexadecimal, contained in bytes 4 and 5.
6	vc	The type of virtual circuit to be set up. Code in byte 6 the corresponding hexadecimal code shown next to the virtual circuit type in the following list: **Hexadecimal Code** — **Virtual Circuit Type** X'C0' — 0 X'C2' — 2 X'C3' — 3 (to SNA peripheral node) X'01', X'41', X'51', or X'81' — 5. **Notes:** 1. With a type 5 virtual circuit, only transparent PAD support can be used. 2. A type 4 virtual circuit cannot be specified. 3. **For V3R3 only:** With a type 5 virtual circuit, transparent PAD or integrated PAD support can be used.
7	al	The application name length (X'00' through X'08') contained in byte 7. A length of X'0' means that DATE must not generate a LOGON command.
8−n	capk	Variable-size field starting in byte 8 that contains the application name followed by the Call Accepted packet (without the 3-byte packet header).

CLEAR on Incoming Call (CTCP to DATE)

The CLEAR command on the incoming call is issued from the CTCP to DATE. The command is sent in the following format:

```
RU: | resid | cc | crdf | user //|
```

Bytes	Field	Description
0−1	resid	2-byte resource identifier contained in bytes 0 and 1. This resource identifier matches the one given in bytes 0 and 1 of the corresponding INCOMING CALL command from DATE. However, the first four bits should be X'0' (CTCP turns off the X'F' in the first 4 bits of the Incoming Call).
2	cc	1-byte command code contained in byte 2. For this command, the value is X'13'.
3−4	crdf	The cause and diagnostic fields (contained in bytes 3 and 4) that are to be inserted in the Clear packet.
5−n	user	Variable-size field starting in byte 5 that contains optional user data.

CLEAR CONFIRMATION after CLEAR Command for Incoming Call Refused (DATE to CTCP)

The CLEAR CONFIRMATION command is issued from DATE to the CTCP after the CLEAR command for an incoming call is refused. The command is sent in the following format:

```
RU: | resid | cc |
```

Bytes	Field	Description
0−1	resid	2-byte resource identifier contained in bytes 0 and 1. The first digit is X'F'.
2	cc	1-byte command code contained in byte 2. For this command, the value is X'17'.

CLEAR on Outgoing Call (DATE to CTCP)

The CLEAR command on the outgoing call is issued from DATE to the CTCP. The command is sent in the following format:

```
     ┌───────┬────┬──────┬──────┬─//─┐
RU:  │ resid │ cc │ cnid │ crdf │user│
     └───────┴────┴──────┴──────┴─//─┘
```

Bytes	Field	Description
0−1	resid	2-byte resource identifier contained in bytes 0 and 1. The first digit is X'F'.
2	cc	1-byte command code contained in byte 2. For this command, the value is X'13'.
3−4	cnid	2-byte connection identifier contained in bytes 3 and 4. This connection identifier matches the one given in bytes 0 and 1 of the corresponding CALL REQUEST command.
5−6	crdf	Cause and diagnostic fields contained in bytes 5 and 6. These fields are taken from the Clear packet.
7−n	user	Variable-size field starting in byte 7 that contains optional user data.

CLEAR after Outgoing Call (CTCP to DATE)

The CLEAR command is issued after the outgoing call from the CTCP to DATE. This CLEAR command can be used when the CTCP detects a time-out after a CALL REQUEST command has been sent. The command is sent in the following format:

```
     ┌──────┬────┬──────┬─//─┐
RU:  │ cnid │ cc │ ccdf │user│
     └──────┴────┴──────┴─//─┘
```

Bytes	Field	Description
0−1	cnid	2-byte connection identifier in bytes 0 and 1. The first digit must be set to X'F' to signify to DATE that this is a CNID rather than a RESID.
2	cc	1-byte command code in byte 2. For this command, the value is X'13'.
3−4	ccdf	Cause and diagnostic bytes contained in bytes 3 and 4.
5−6	user	Variable-size field starting in byte 5 that contains optional user data.

CLEAR (DATE to CTCP, CTCP to DATE)

The CLEAR command is issued from DATE to the CTCP and from the CTCP to DATE. The command is sent in the following format:

```
RU: | resid | cc | ccdf | user //|
```

Bytes	Field	Description
0−1	resid	2-byte resource identifier, contained in bytes 0 and 1, that is provided to the CTCP in the CALL CONNECTED or INCOMING CALL command. The leftmost halfbyte is X'0' when the direction is CTCP to DATE, and X'F' when the direction is DATE to CTCP.
2	cc	1-byte command code contained in byte 2. For this command, the value is X'13'.
3−4	ccdf	Clear cause and diagnostic fields contained in bytes 3 and 4.
5−n	user	Variable-size field starting in byte 5 that contains optional user data.

CLEAR CONFIRMATION (CTCP to DATE, DATE to CTCP)

The CLEAR CONFIRMATION command is issued from the CTCP to DATE and from DATE to the CTCP. The command is sent in the following format:

```
RU: | resid | cc |
```

Bytes	Field	Description
0−1	resid	2-byte resource identifier contained in bytes 0 and 1. The leftmost halfbyte is X'0' when the direction is CTCP to DATE, and X'F' when the direction is DATE to CTCP.
2	cc	1-byte command code contained in byte 2. For this command, the value is X'17'.

RESET (DATE to CTCP, CTCP to DATE)

The RESET command is issued from DATE to the CTCP and from the CTCP to DATE.

For V3R3 only: When using integrated PAD, the CTCP is not authorized to send Reset packets. Reset packets flow on the physical circuit LU session only if they are *not generated* by a BREAK entered at the terminal.

The command is sent in the following format:

```
                            -//-
RU: | resid | cc | rcdf | user |
                            -//-
```

Bytes	Field	Description
0–1	resid	2-byte resource identifier, contained in bytes 0 and 1, that is provided to the CTCP in the CALL CONNECTED or INCOMING CALL command. The leftmost halfbyte is X'0' when the direction is CTCP to DATE, and X'F' when the direction is DATE to CTCP.
2	cc	1-byte command code contained in byte 2. For this command, the value is X'1B'.
3–4	rcdf	Reset cause and diagnostic fields contained in bytes 3 and 4.
5–n	user	Variable-size field starting in byte 5 that contains optional user data.

When X.25 NPSI receives a packet with invalid P(S) or invalid P(R), X.25 NPSI generates a RESET command that is passed to the CTCP. The diagnostic code is set to X'AB' for invalid P(S) or X'AC' for invalid P(R). This Reset packet is not sent to the PSDN. The CTCP must send a RESET command or clear the connection, if necessary.

Note: If you are using transparent PAD, see the command formats for the transparent PAD RUs on page 41.

RESET CONFIRMATION (CTCP to DATE, DATE to CTCP)

The RESET CONFIRMATION command is issued from the CTCP to DATE and from DATE to the CTCP. The command is sent in the following format:

```
RU: | resid | cc |
```

Bytes	Field	Description
0–1	resid	2-byte resource identifier contained in bytes 0 and 1. The left-most halfbyte is X'0' when the direction is CTCP to DATE, and X'F' when the direction is DATE to CTCP.
2	cc	1-byte command code contained in byte 2. For this command, the value is X'1F'.

Notes:

1. At RESET CONFIRMATION, the PLP counters are reset to 0, but the virtual circuit remains active. The CTCP may clear the virtual circuit, if necessary.
2. If you are using transparent PAD, see the command formats for the transparent PAD RUs on page 41.

INTERRUPT (CTCP to DATE, DATE to CTCP)

The INTERRUPT command is issued from the CTCP to DATE, from DATE to the CTCP, and is used for LLC0.

For V3R3 only: When using integrated PAD, Interrupt packets flow on the virtual circuit LU session. They are not authorized on the physical circuit LU session.

The command is sent in the following format:

```
RU: | resid | cc | iu | user //
```

Bytes	Field	Description
0–1	resid	2-byte resource identifier contained in bytes 0 and 1. The left-most halfbyte is X'0' when the direction is CTCP to DATE, and X'F' when the direction is DATE to CTCP.
2	cc	1-byte command code contained in byte 2. For this command, the value is X'23'.
3	iu	Interrupt user data contained in byte 3.
4–n	user	Variable-size field starting in byte 4 that contains optional user data.

Note: If you are using transparent PAD, see the command formats for the transparent PAD RUs on page 41.

INTERRUPT CONFIRMATION (CTCP to DATE, DATE to CTCP)

The INTERRUPT CONFIRMATION command is issued from the CTCP to DATE and from DATE to the CTCP, and is used for LLC0.

For V3R3 only: When using integrated PAD, Interrupt Confirmation packets flow on the virtual circuit LU session. They are not authorized on the physical circuit LU session.

The command is sent in the following format:

RU:	resid	cc

Bytes	Field	Description
0−1	resid	2-byte resource identifier contained in bytes 0 and 1. The left-most halfbyte is X'0' when the direction is CTCP to DATE, and X'F' when the direction is DATE to CTCP.
2	cc	1-byte command code contained in byte 2. For this command, the value is X'27'.

Note: If you are using transparent PAD, see the command formats for the transparent PAD RUs on page 41.

DIAGNOSTIC (CTCP to DATE, DATE to CTCP)

The DIAGNOSTIC command is issued from the CTCP to DATE and from DATE to the CTCP. The command is sent in the following format:

RU:	F000	cc	dc	// od //

Bytes	Field	Description
0−1	F000	2-byte null identifier.
2	cc	1-byte command code contained in byte 2. For this command, the value is X'F1'.
3	dc	Diagnostic code contained in byte 3.
4−n	od	Variable-size field starting in byte 4 that contains diagnostic explanation.

RESTART (CTCP to DATE, DATE to CTCP)

The RESTART command is issued from the CTCP to DATE and from DATE to the CTCP. The command is sent in the following format:

RU:

x000	cc	ca	dg

Bytes	Field	Description
0−1	x000	2-byte null identifier containing X'x000', where x is X'0' when the direction is CTCP to DATE and X'F' when the direction is DATE to CTCP.
2	cc	1-byte command code contained in byte 2. For this command, the value is X'FB'.
3	ca	Cause code contained in byte 3.
4	dg	Diagnostic code contained in byte 4.

RESTART CONFIRMATION (CTCP to DATE, DATE to CTCP)

The RESTART CONFIRMATION command is issued from the CTCP to DATE and from DATE to the CTCP. The command is sent in the following format:

RU:

x000	cc

Bytes	Field	Description
0−1	x000	2-byte null identifier containing X'x000', where x is X'0' when the direction is CTCP to DATE and X'F' when the direction is DATE to CTCP.
2	cc	1-byte command code contained in byte 2. For this command, the value is X'FF'.

ERROR/INFORMATION REPORT (DATE to CTCP)

The ERROR/INFORMATION REPORT command is issued from DATE to the CTCP. The command is sent in the following format:

RU:	resid	cc	ce	ec

Bytes	Field	Description
0–1	resid	2-byte resource identifier contained in bytes 0 and 1. The left-most halfbyte is X'F'.
2	cc	1-byte command code contained in byte 2. For this command, the value is X'00'.
3	ce	Command code for the command in error (contained in byte 3).
4	ec	Error cause. (See the list of error causes in Table 9.)

INFORMATION REPORT Messages Sent from DATE to the CTCP

The INFORMATION REPORT command allows for the exchange of information between DATE and the CTCP without involving the X.25 network. This INFORMATION REPORT message is sent from DATE to the CTCP. When the message is sent by DATE, the CTCP must respond to the code in the message as shown in Table 9.

Table 9. CTCP Responses to DATE INFORMATION REPORT Messages

Error Cause	Reason Sent by DATE	Required Action
X'00'	Current PLP state does not permit the sending of Reset, Interrupt, qualified data, or unqualified data packets	Correct the CTCP
X'01'	Virtual circuit type not consistent with DATE	Correct the CTCP
X'02'	Unknown format	Correct the CTCP
X'03'	Resource identifier not generated in X.25 NPSI	Correct the CTCP or the X.25 NPSI generation
X'04'	No virtual circuit available	The CTCP must retry a CALL REQUEST command
X'05'	Unsuccessful XIO to X.25 network	Send a message to the operator to check the MCH
X'06'	Physical circuit is not running	Issue a RESTART REQUEST command
X'07'	Application abandons connection	Issue a CLEAR REQUEST command
X'08'	Invalid packet size of the CALL REQUEST or CALL ACCEPTED command, or specified in the CALL REQUEST or CALL ACCEPTED command.	Correct the CTCP

When X.25 NPSI sends an INFORMATION REPORT message, DATE does not take action to resolve the problem reported by the message to the CTCP. The receiving buffer that caused the message is released, and DATE waits for new instructions from the CTCP.

Note: When DATE cannot determine the virtual circuit where the problem occurred, the INFORMATION REPORT message is sent with a resource identifier of 0.

DATE Message

A message can be sent from the CTCP LU to the MCH LU. This message is used to notify you of an impending session setup. The message can be used when the session setup will take a considerable amount of time to tell the operator that the session establishment is taking place, and that the operator should not clear the connection, even if this establishment takes a long time.

The CTCP uses the MCH LU session for this message, because of its constant availability and because data can be sent to the remote terminal before the virtual circuit session is active. Unlike the session to the VC LU, the session to the MCH LU is always available for use. This allows the message to be sent to the terminal operator, as soon as the call setup is complete but long before the SNA session setup for the VC is complete.

Program Operator Interface

Before communication between the DATE CTCP and the application program can take place, a program operator interface (POI) must be created. This interface is used to pass requests and the current status information between the two programs.

An example of how the communication interface (created by POI) operates is illustrated by the application program sending the CTCP a request to acquire a particular device. Without the use of the POI, the request cannot be sent because the application program requires a session with the MCH LU. This MCH LU is held by the CTCP, and VC LU can be acquired only through this session. The POI allows the necessary communication to notify the CTCP of the desired application.

In addition to handling requests, the POI must also be able to asynchronously notify the session partner of a change in virtual circuit status. A status change can be caused by a number of events. The following are examples of events that can cause status changes:

- Failure of a virtual circuit
- Activation of a virtual circuit LU
- Operator deactivation of a virtual circuit LU.

Methods of Creating a POI

The CTCP and the application program must be able to communicate before a POI can be created. This communication can use any method, including the following:

- SNA session

  This session can use any session protocol that is acceptable by both sides of the communication channel.

- Cross-memory services

  When cross-memory services are supported by the operating system.

- Connection through X.25 NPSI using PCNE.

The method of communication must be able to support asynchronous requests between the the CTCP and the application program. Although any SNA protocol allows for this method of communication, the use of SNA indicators signaling a change of direction or bracketing causes additional communication difficulties. These difficulties are caused by the problems associated with asynchronous requests that run counter to the direction of the communication path established at the time of the request.

For example, communication problems occur if the session is in the in-bracket state (INB), and the contention-losing partner must notify the other partner of an event. To eliminate the difficulty, a private protocol must be established that notifies the other partner that a message needs to be sent. When the private protocol is established, the contention-losing partner transmits a SIGNAL request that notifies the contention-winning partner of the need to reverse the flow of data. The contention-winning partner then responds by relinquishing control of the session, which permits the contention-losing partner to transmit the message.

A requirement for a more complex data flow can demand that a private protocol be established, even though it follows normal SNA request flows. A private protocol allows the data flow between partners to be streamlined by permitting either of the following transmissions:

- Multiple requests transmitted at one time
- Transmission of a different method of asynchronous requests.

The communication between the partners must have the flexibility to allow for any request. The POI should not:

- Restrict the type or size of the message
- Restrict the capacity of the interface
- Have a high utilization (this can change during a major network event, such as the failure of a communication controller).

Chapter 5. Host Application Programming

Chapter 5. Host Application Programming

This chapter presents the definitions and requirements needed to support application programs that communicate with resources through X.25 NPSI. Unusual application program occurrences, connection methods, and dial-in and dial-out methods are described.

CICS

The Customer Information Control System (CICS) subsystem is able to communicate with many types of SNA and non-SNA resources. The communication definitions for CICS are located in the terminal control table (TCT). This table defines device characteristics that include, but are not limited to:

- Network name
- Device type (such as 3270 or 3767)
- Access method
- Screen size and availability of an alternative size
- CICS terminal identifier.

The CICS Terminal Control Program (TCP) obtains the profile of the device from the terminal control table to determine the correct terminal driver.

The TCP also performs protocol and device-specific processing. This processing isolates the application program from the terminal-specific requirements. For example, the terminal driver can chain the output data into elements of an acceptable size for the destination device and can request attachment to the destination resource. The application program, unaware of this processing, can therefore support different types of devices simultaneously.

The X.25 NPSI LU simulator creates the LU-LU communication between the host application program and a non-SNA DTE for non-SNA devices.

Figure 17 on page 116 shows the data flow in a network configuration using CICS and X.25 NPSI.

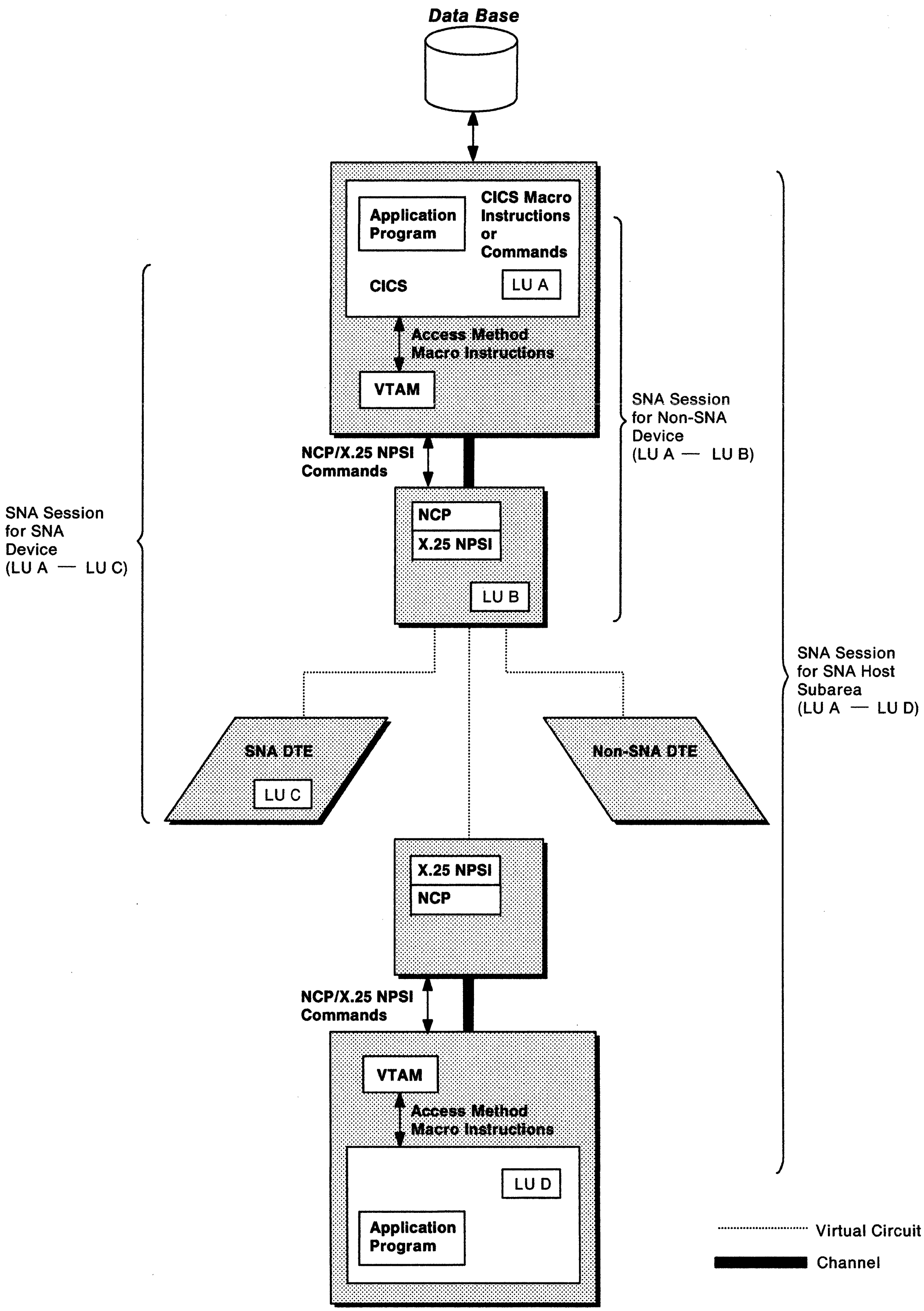


Figure 17. SNA Host Node to SNA Peripheral Node or Non-SNA DTE Using CICS

TCT Definition for SNA Resources

X.25 NPSI gives the impression of a direct connection (no PSDN in the middle) to a leased SNA device. The host application program is unaware of the session setup, access method, and network communication requirements. Instead, all this knowledge is off loaded onto the applicable software, hardware, and microcode to permit the isolation of these concerns.

The CICS/X.25 NPSI pair allows application support of all LUs supported by PU types 1 and 2.0, and PU type 2.1 nodes. Thus, this pair can support the following LU session types:

- LU1
- LU2
- LU4
- LU6.1
- LU6.2.

This support allows the LU to be connected to CICS as if the PSDN is not there. Because the connection is isolated from the PSDN, it is not necessary to change the SNA resources that communicate through X.25 NPSI. Thus, the definition of an SNA resource with X.25 NPSI is exactly the same as that without X.25 NPSI.

See *CICS/VS Resource Definition (Online)*, *CICS/VS Resource (Macro)*, and the *CICS/VS 3767/3770/6670 Guide* for more information about SNA terminal definitions.

TCT Definition for Non-SNA Resources

The TCT connection to non-SNA resources appears as an LU type 1 session.

The application programmer must set up a configuration using one of the two interface types supported by the CICS/X.25 NPSI interface. The types supported are half-duplex flip-flop mode and half-duplex contention mode. These standard connection facilities are described in the *CICS/VS 3767/3770/6670 Guide*.

Half-duplex flip-flop protocol is centered around the change of direction indicator (CD indicator). This request/response header (RH) indicator identifies the partner that can transmit data. The partner holding the CD indicator is the one able to send data. The partner receiving data can send a SIGNAL command if it needs to send data before the CD indicator would normally be relinquished.

Contention protocol is a half-duplex protocol that permits either session partner to initiate data transmissions. Both partners can receive data at any time.

Contention occurs if both session partners want to send data at the same time. In this case, a winner is determined and only the winner's transmissions continue. The contention winner is predetermined in the session BIND parameters. Thus, the winner of a contention is always known beforehand.

The contention of transmission is done at the *chaining* level. Contention occurs only at the start of a chain of data. The next chain initiates a new contention situation that must be administered individually.

Generation Requirements

The system programmer defines the LU1 device by coding the following values for the TRMTYPE keyword in the DFHTCT TYPE=TERMINAL macro or the CEDA DEFINE/ALTER TERMINAL transaction in the TYPETERM field.

The following example describes a typical TCT entry for an LU1 device. Options and defaults are noted to the right of some keywords.

```
DFHTCT   TYPE=TERMINAL,
         ACCMETH=VTAM,
         TRMTYPE=                (see list of options)
         NETNAME=xxxxxxxx,
         TRMSTAT=TRANSCEIVE,
         TIOAL=256
         BUFFER=256,
         RUSIZE=                 (default is 256)
         BRACKET=YES,
         PGESTAT=PAGE,
         PAGESIZE=               (default is (12,80))
```

Possible values for the TRMTYPE keyword are:

Protocol	Acceptable Values
Flip-flop	3767
	3767I
	INTLU
Contention	3767C

Note: Code TRMTYPE=TWX if the PAD device requires it at the remote end.

Application Requirements for Support

Application requirements for support of remote DTEs communicating through X.25 NPSI vary depending on whether the remote DTEs are SNA devices or non-SNA devices.

SNA Resources

The application need not be modified to support these devices. All changes applied to the data are isolated from the application.

Non-SNA Resources

Non-SNA resources must also be supported by a host application program. Because the host application knows only of the SNA resources, the support is made easier by the CICS/X.25 NPSI interface. However, communication needs can vary, so the host application program should be modified to support your network configuration. For example, if you use transparent PAD support, session data is prefixed with an extra byte. You must modify your CICS application programs to accommodate this prefix.

Integrated PAD Support: When a CICS application program communicates through integrated PAD support, the application program operates as if connected to an LU type 1 device.

If the application wants to use the password protection feature of X.25 NPSI, it can use the enable presentation (ENP) and inhibit presentation (INP) characters to define when a field should not be echoed to the terminal. The hexadecimal values for these control characters are:

X'24' Inhibit presentation (INP)
X'14' Enable presentation (ENP).

Note: If X.25 NPSI does not translate EBCDIC to ASCII, the control character used for inhibit presentation is X'12' rather than X'24', and the overstrike message is translated to ASCII EVEN code. If this default value must be changed, see *X.25 NPSI Diagnosis, Customization, and Tuning*.

To use the password protection function, the application program places an INP character anywhere in the data stream to prompt the protected information. The protected information is interpreted and converted into the appropriate PAD commands to disable display at the device. Disabling the display can be accomplished through either the inhibiting of the data from being echoed back to the device, if the device is a video display terminal, or the transmission of a blackout character string, if the device is a typewriter-like device.

A user LOGON transaction must be written to use the services of the X.25 NPSI password protection.

Upon receiving and processing the response, the next output RU should contain, at any position, the ENP character used to initiate the redisplaying of characters on the device.

To implement the password protection function, you must request that X.25 NPSI perform this processing by specifying PWPROT=YES on the X25.MCH statement. X.25 NPSI is responsible for all the required processing to implement the password protection function.

If a CICS transaction wants to pass the integrated PAD device to another application by using the EXEC CICS ISSUE PASS command, SHUTD=NOINVCLR should be coded in the corresponding X25.MCH statement. This prevents the SHUTD executed by CICS from clearing or resetting the virtual circuit.

X.25 NPSI is also responsible for *all* PAD processing for any activity. As soon as the virtual circuit connection is established with the PAD, X.25 NPSI sends a PAD message to the PAD to set the values of PAD parameters 1, 7, and 8. See *X.25 NPSI Diagnosis, Customization, and Tuning* for more information on the objective of integrated PAD support.

Transparent PAD Support: Communication through transparent PAD support requires that the CICS program be written so that the first byte of the data buffer is reserved for the command code that defines the type of data contained in the buffer.

The valid command codes are:

First Byte	**Type of Data**
X'00'	User data with Q bit set to 0 (unqualified data)
X'02'	User data with Q bit set to 1 (qualified data)
X'1B'	RESET command
X'1F'	RESET CONFIRMATION command
X'23'	INTERRUPT command
X'27'	INTERRUPT CONFIRMATION command.

Data that is passed between the remote DTE and the application program is sent as data with the Q bit set to 0. This is done by placing X'00' into the first byte of the data buffer sent to the remote DTE. Application data is returned to the application program with this same code in the first byte of the data buffer. The application program must be aware of the extra byte preceding the data processed by the application program.

Qualified data is used to exchange information between the application program and the remote PAD. Qualified data can be used to perform any type of PAD command.

Reset packets contain the cause and diagnostic codes in the second and third bytes, respectively. X.25 NPSI resets the send and receive counters to 0.

The second byte of the Interrupt packet contains the interrupt cause (usually set to X'00'). For Interrupt packets, the application program initiates the transmission of an Interrupt Confirmation packet when an Interrupt packet is received.

You can use the TRAN keyword on the X25.MCH statement to request that X.25 NPSI translate data when you use PAD support. When you request the TRAN option, X.25 NPSI performs translation of all data *after* the first byte of the RU for unqualified data.

Because the first 4 bytes of the input are not consistent, these characters cannot be used for transaction selection. As a result, the TCT entry for the session should specify the TRANSID keyword, or the TRANSID keyword should be used on the CICS RETURN command. This allows input to be processed correctly even though the keywords are preceded by a command code.

Bracket Protocol: A bracket is any complete unit of work that an application program and an LU have been programmed to accomplish. Bracket protocols are used to ensure that one or both ends of a session do not begin processing a new unit of work until the current one is complete.

A CICS transaction is a typical example of a bracket. In a CICS bracket, an LU requests information and CICS responds and ends the bracket. The LU can make further requests depending on the response, and the bracket will continue.

When a session is established, the bits in the session parameters determine who begins the bracket, who ends it, and who wins if contention occurs. One end of the session is assigned as the first speaker, and the other end is the bidder. The bidder normally asks permission to begin a bracket.

Flip-Flop Protocol: Half-duplex flip-flop data flow control protocol operates with the change of direction (CD) indicator. Only the partner that possesses the CD indicator is able to transmit data.

CICS automatically includes the CD indicator with the first data RU that is sent in a session. To change the CD indicator setting, choose one of the following actions:

- Explicitly use the INVITE parameter on the SEND command.

  Use of the INVITE parameter notifies the CICS Terminal Control Program that the program is requesting that a CD indicator be added to the data when it is sent to the access method.

- Implicitly use a SEND command with a RECEIVE command.

  CICS delays the transmission of output data as long as possible so it can piggyback as many indicators onto the data as is feasible. In this case, CICS *does not* actually send data out upon encountering a SEND command. Instead, CICS buffers the data and waits for a command that requires it to transmit the data.

  When CICS encounters the RECEIVE command, it transmits the data and adds a CD indicator to the data so that a response is returned. CICS performs the same processing for the CONVERSE command.

Using these methods, the CD indicator is passed between the session partners while data flow is maintained and controlled. Each session partner can control the data flow and manage its own internal resources.

To better manage resources, a session partner can send data to the partner that possesses the CD indicator. This is done by using the ISSUE SIGNAL command. This command indicates to the session partner holding the CD indicator that the partner sending the ISSUE SIGNAL command wants to send data. The partner with the CD indicator can satisfy this request by performing a command that sends the CD indicator to the other partner.

The session partner receiving the SIGNAL command is notified of this request by the execute interface block (EIB) field EIBSIG being set to X'FF' or through a HANDLE CONDITION SIGNAL command. For you to use the EIBSIG method, the application program must be coded to check the EIBSIG field after all terminal control commands (SEND, RECEIVE, CONVERSE, and so on). Use of the HANDLE CONDITION SIGNAL command is preferable because the specified routine branches whenever the SIGNAL command is received.

X.25 NPSI sends only-in-chain (OIC) and last-in-chain (LIC) RUs to the host with the CD indicator on. Integrated PAD can exchange SIGNAL commands with the host.

Contention Mode Protocol: In contention mode, contention for the right to send occurs only between *chains* within a bracket. If the session is not in bracket state, the normal rules for bracket initiation apply. X.25 NPSI does not have a contention state. It has send, receive, and standby states.

When CICS is in contention mode, it has three states:

- Send
- Receive
- Contention.

If a message arrives while a receiver is in contention state, the receiver's state switches to *receive*. Transmission of the end-of-chain session partners back to contention state.

Contention can occur when both session partners are in send state. The session is always established with CICS as the contention *loser*. SNA specifies the following rules for the resolution of contention:

- If the contention loser is sending, incoming messages must be queued.

- If the contention winner is sending, incoming messages can be either queued or rejected with an appropriate sense code.

For the CICS subsystem, (and its associated application programs), the SNA rules are implemented in the following manner:

- The contention resource sends data to CICS, but the CICS application program has not issued a RECEIVE command.

  In this case, CICS provides the queuing mechanism specified by SNA, if *read-ahead queuing* is specified for the transaction or the queuing is handled by VTAM. Read-ahead queuing can be specified by coding RAQ=YES in the DFHPCT macro or in the PROFILE command of the CEDA transaction.

  Note: If the CEDA transaction is used to create a new transaction profile, then this profile must be specified for all transactions that want to use read-ahead queuing.

  If read-ahead queuing is specified, incoming data is queued by CICS in its *temporary storage queue*. If the transaction subsequently issues a RECEIVE command, the input is recovered from the temporary storage queue rather than from the terminal. If the transaction terminates without issuing a RECEIVE command, CICS automatically purges the queue and the input is lost.

- The CICS application program issues a SEND command, but the contention resource is in *send* state.

  In this case, the contention resource returns a negative response with sense code X'081B'. This sense code specifies that the receiver is in *send* state.

  When the negative response is received, the Node Abnormal Condition Program (DFHZNAC) is scheduled for the task. The following default actions are taken by DFHZNAC:

  1. If the rejected chain is sent with definite response, the VTAM SEND command is purged, and the CICS transaction is abnormally terminated (DFHZNAC action flags X'60E000').
  2. If CICS is not in send state when the response is received, the VTAM SEND command is purged, the transaction is abnormally terminated, and a VTAM CLSDST macro is issued to break the connection (DFHZNAC action flags X'60E001').
  3. If exception response was requested, the VTAM SEND is purged, and the CICS transaction is abnormally terminated (DFHZNAC action flags X'60E000').

  For the first case, the CICS user can write a Node Error Program (NEP) to retry the failing SEND command. The NEP must do the following:

  - Reset the abend flag in TWAOPTL.
  - Reset the VTAM SEND purge flag in TWAOPTL.
  - Set flag TWANPFW in the NEP return code byte TWANEPR. This flag specifies *retry with FORCE* and results in CICS sending the SIGNAL command to X.25 NPSI and then retrying the SEND command.

  Note: For the second case, an NEP is not allowed to reset the *break connection* flag, so recovery is not possible. For the third case, care should be taken because the terminal input/output area (TIOA) might not be available.

Dial-In to CICS

An application program does not require any modification to support a dial-in resource. To the host application, the dial-in resource appears as a resource that has attached itself to the host.

Both physical and user security are cause for concern. Physical security *can* be implemented by the access method; however, additional security assurances, such as passwords or security packages, might be required.

Dial-Out from CICS

CICS dial-out can be performed in three ways:

- The first method uses the START command for a transaction on the terminal to which you want to dial out. This command performs an interval control process that initiates a new task to a specific terminal. When CICS recognizes that the terminal is not attached to the region, it attempts to acquire the terminal. The acquisition is passed to VTAM, and the SNA CONNOUT request is passed to the appropriate communication controller.

  In this case, the transaction program must establish a method of synchronization with the remote device. One method is to define a device that will be the first one to send data. A session partner would not send any data until requested to initiate data transfer. This simple protocol eliminates problems incurred in communication; this is especially true of non-SNA connections, because the flow control on these sessions is difficult to regulate. Once the data flow is synchronized between the session partners, communication can proceed.

  X.25 NPSI is invoked because the resource is identified as belonging to X.25 NPSI. An X.25 Call Request packet is transmitted to the remote DTE. Additional commands are passed between the network resources until the session is established (by a BIND being passed between the session partners) and start-data-traffic (SDT) is complete. At this point, the device is attached to CICS, and the transaction that is already started is attached to the device. Communication can now be performed with the dial-out device.

- The second method of initiating a dial-out operation is for an application program (or an operator command) to request the acquisition of a device. For an application program request, use the DFHTC TYPE=CONTROL macro or the EXEC CICS SET TERMINAL command. (The latter is available only in CICS Version 1.7 and later.) An operator can request acquisition of the device through the CEMT transaction.

- The third method of initiating a dial-out operation is for the network operator to issue one of the following commands:

  - V NET,ID=*termid*,LOGON=*appl*
  - V NET,ACT,ID=*termid* with LOGAPPL=*appl* coded on the LU statement.

  In both commands, *appl* specifies the CICS subsystem application ID name, and *termid* specifies the terminal to which you want to dial out. When either of these commands is issued, a session is established between the device and CICS.

Disconnection of Switched Virtual Circuits

An application program is able to disconnect a switched virtual circuit (SVC) when communication is complete. This disconnection reduces communication costs by reducing the connection time.

Termination of the virtual circuit is caused by an application program issuing an ISSUE DISCONNECT command to the SLU, which results in an UNBIND command flowing to the remote LU. When this command is processed, the SVC is cleared if DISCNT=YES is coded in the PU of the SMN, and this is the last or only LU to terminate its sessions.

When using CICS with the ISSUE PASS command, CICS causes a CLSDST OPTCD=PASS macroinstruction to be issued. In this case, to prevent a device from being disconnected when using integrated PAD support, specify SHUTD=NOINVCLR on the X25.MCH statement. The NOINVCLR parameter allows the device to be passed between applications. For example, a printer can be acquired by another application after it is passed from CICS without clearing the virtual circuit.

To use the ISSUE PASS command, you must code AUTH=PASS on the VTAM APPL statement for CICS.

Considerations when Creating a Fast Connect CTCP under CICS

You should understand the fast connect CTCP interface and have a very good understanding of CICS before attempting to use this application program interface (API). The following considerations should also be understood before using the API:

- Use the TRANSID on the TCT definition.

  The terminal control table (TCT) entries that are in session with the MCH LU and VC LUs should have a TRANSID specified. This results in CICS initiating the specified transaction *whenever* data arriving at CICS contains a diagnostic packet in the case of the MCH LU, or contains Call, Clear, Reset, or data packets in the case of the VC LUs.

- The fast connect CTCP created within a CICS application must be able to process *any* of the commands sent by X.25 NPSI. This includes commands that application programs are not usually able to recognize and process, such as the CLEAR, RESET, and CALL commands.

  All data from the X.25 NPSI LU is routed to the task defined in the TRANSID command located in the appropriate TCT entry. It is the duty of this task to analyze the data packets received from X.25 NPSI, and to act accordingly. This task can be programmed to handle X.25 commands, such as:

  - CLEAR
  - RESET
  - CALL

  For example, this task can be programmed to analyze the cause and diagnostic fields received in a Clear packet, and to reestablish the X.25 connection for certain causes.

- The VTAM node error program (DFHZNEP) must be customized.

  DFHZNEP is invoked by CICS whenever the operational status of a device controlled by VTAM changes. The CICS abnormal condition program (DFHZNAC) determines the necessary actions in response to the status change (such as session established, terminated, or device failure). DFHZNEP is able to modify some of the action settings determined by DFHZNAC. Upon returning from DFHZNEP, DFHZNAC performs the set actions. These actions can either be the default settings or those modified by DFHZNEP.

In addition, DFHZNEP can link up to other user-written programs. These user-written programs can determine the required actions for LUs attached to other subsystems, or even the same CICS system.

This process is invoked when the session to the MCH LU is disconnected. Although this is not a normal occurrence, the CICS CTCP must be able to handle this situation if it occurs.

When the session to the MCH LU is disconnected, DFHZNEP is invoked by CICS as a result of the session loss with the MCH LU. The CTCP should determine the cause of the failure by analyzing the cause code,[4] set by CICS within the communications area, that is passed from DFHZNAC to DFHZNEP.

IMS

Through X.25 NPSI, both SNA and non-SNA terminals can be connected to the Information Management System (IMS) subsystem. The system programmer defines each device communicating through X.25 NPSI with the COMM, TYPE, and TERMINAL macros.

Figure 18 on page 126 shows the data flow in a network configuration using IMS and X.25 NPSI.

4 These cause codes are not the same as the X.25 CAUSE codes that are contained in many of the packet formats. See *CICS Diagnosis Guide* for more information on the specific codes and their meanings.

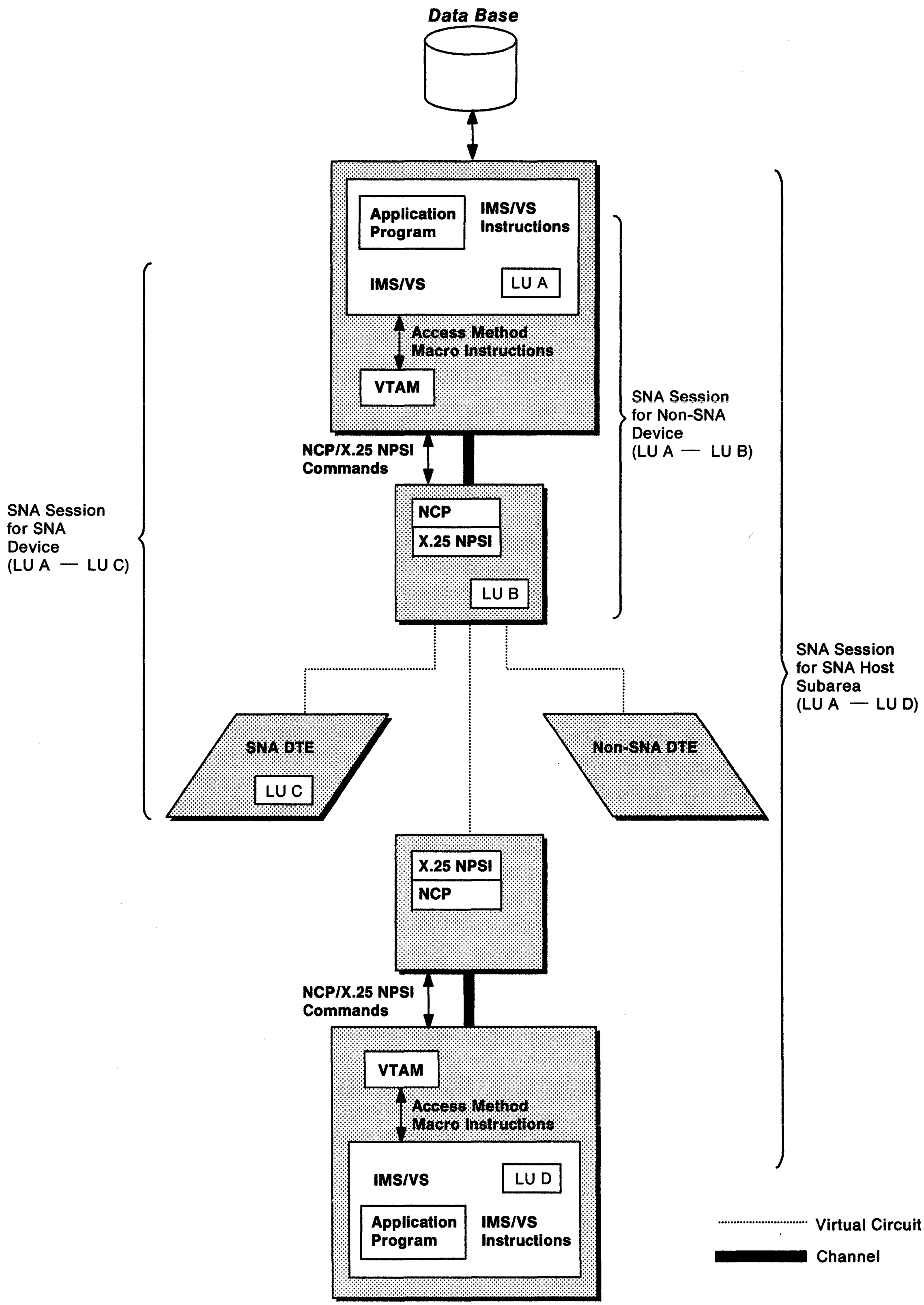


Figure 18. SNA Host Node to SNA Peripheral Node or Non-SNA DTE Using IMS

SNA Connection

SNA devices are defined to IMS and communicate with IMS without regard for the connection method. IMS is unaware that X.25 NPSI is in the communication path. No modification of IMS or IMS application programs is required to allow communication through X.25 NPSI to a remote DTE.

Non-SNA Connection

Non-SNA devices interface with IMS through the X.25 NPSI LU simulator. This function performs a conversion on data coming from the non-SNA device to give the appearance, to the IMS application program, that the data comes from an LU type 1 device.

For V3R2 and Previous Releases: The LU simulator does *not* map outbound chaining of PIUs to any X.25 protocol. Because there is no mapping, the size of the maximum RU must be set as large as the largest RU that the IMS system needs to transmit.

For V3R3 only: The LU simulator optionally maps chains of PIUs to the X.25 complete packet sequence M-bit sequence (CPS/MBS) when MBITCHIN=YES is specified. If this option is not chosen, the size of the maximum RU must be set as large as the largest RU that the IMS system needs to transmit. When using GATE or Transparent PAD with MBITCHIN=YES, only OIC PIUs will be generated and sent to the host.

This RU size must be set in two places. The first place is in the COMM macro. The maximum RU size on the RECANY keyword of the COMM macro must be set large enough so that no RUs are truncated. Secondly, the OUTBUF keyword of the TERMINAL macro must be set. This keyword is used by IMS to set the outbound RU size in the BIND.

These RU values should be optimized to reduce performance impacts. To conserve storage, set the maximum RU size as small as possible.

All devices are defined with the TYPE and TERMINAL macros as SNA devices. The TYPE macro defines some global values used in all subsequent TERMINAL macros.

LU type 1 devices are supported in communication through X.25 NPSI. In addition, IMS fast path devices are also supported. A more extensive list of supported devices can be found in *IMS General Information* and *IMS Installation Guide*.

An example of a terminal definition for a 3767 is:

```
COMM     RECANY=(number of buffers,maximum RU size)
TYPE     UNITYPE=3767,
         OPTIONS=NOASR
TERMINAL NAME=nnnnnnnn,
         COMPT=CONSOLE,
         FEAT=IGNORE,
         OPTIONS=(BSELM,...),
         OUTBUF=maximum RU size
```

IMS Application Support for Data Flow Control (DFC)

IMS is responsible for all data flow control requirements from the application program. The application program under IMS is not aware of the data flow control requirements. The application program is concerned only with the data; all higher-level protocol requirements are isolated from the application program by IMS.

Integrated PAD Support

When an IMS application program communicates through integrated PAD support, the application program operates as if connected to an LU type 1 device.

If the application program wants to use the password protection feature of X.25 NPSI, the application program can use the enable presentation (ENP) and inhibit presentation (INP) characters to define when a field should not be echoed to the terminal. The hexadecimal values for these control characters are:

X'24' Inhibit presentation (INP)
X'14' Enable presentation (ENP).

Note: If X.25 NPSI does not translate EBCDIC to ASCII, the control character used for inhibit presentation is X'12' rather than X'24', and the overstrike message is translated to ASCII EVEN code. If this default value must be changed, see the customization section of *X.25 NPSI Diagnosis, Customization, and Tuning*.

To use the password protection function, the application places an INP character at the end of the data stream to prompt the protected information. X.25 NPSI interprets this character and converts it into the appropriate PAD commands to disable the display at the device. Disabling the display can be accomplished through either the inhibition of echoing data back to the device, if the device is a video display terminal, or the transmission of a blackout character string, if the device is a typewriter-like device.

Upon receiving and processing the response, IMS starts the next output buffer with the ENP character to initiate the redisplaying of characters on the device.

X.25 NPSI is responsible for all processing required to implement the password protection function. You must request that X.25 NPSI perform this processing by specifying PWPROT=YES on the X25.MCH statement.

X.25 NPSI is also responsible for *all* PAD processing for any activity. As soon as the virtual circuit connection is established with the PAD, X.25 NPSI sends a PAD message to the PAD to set the values of PAD parameters 1, 7, and 8. See *X.25 NPSI Diagnosis, Customization and Tuning* for more information on the objective of integrated PAD support.

Transparent PAD Support

Communication through transparent PAD support requires that the IMS program reserve the first byte of the data buffer for the command code that defines the type of data to be contained in the buffer.

The valid command codes are:

First Byte	Type of Data
X'00'	User data with Q bit set to 0 (unqualified data)
X'02'	User data with Q bit set to 1 (qualified data)
X'1B'	RESET command
X'1F'	RESET CONFIRMATION command
X'23'	INTERRUPT command
X'27'	INTERRUPT CONFIRMATION command.

Qualified data is used to exchange information between the application program and the remote PAD. Qualified data can also be used to perform any type of PAD command.

Data that is passed between the remote DTE and the application program is sent as data with the Q bit set to 0. This is done by placing X'00' into the first byte of the data buffer being sent to the remote DTE. Application data is returned to the application program with the same code in the first byte of the data buffer. Thus, the application must be aware of the extra byte preceding the data processed by the application program.

Reset packets contain the cause and diagnostic codes in the second and third bytes, respectively. X.25 NPSI resets the send and receive counters to 0.

The second byte of the Interrupt packet contains the interrupt cause (usually set to X'00').

You can use the TRAN keyword on the X25.MCH statement to request that X.25 NPSI translate data when you are using PAD support. When you request the TRAN option, X.25 NPSI performs translation of all data *after* the first byte of the RU for unqualified data.

Note: If you are using transparent PAD, see the command formats for the transparent PAD RUs on page 41.

TSO

This section contains considerations specific to the Time Sharing Option (TSO) subsystem when running X.25 NPSI. Figure 19 on page 130 shows the data flow in a network configuration using TSO, and X.25 NPSI.

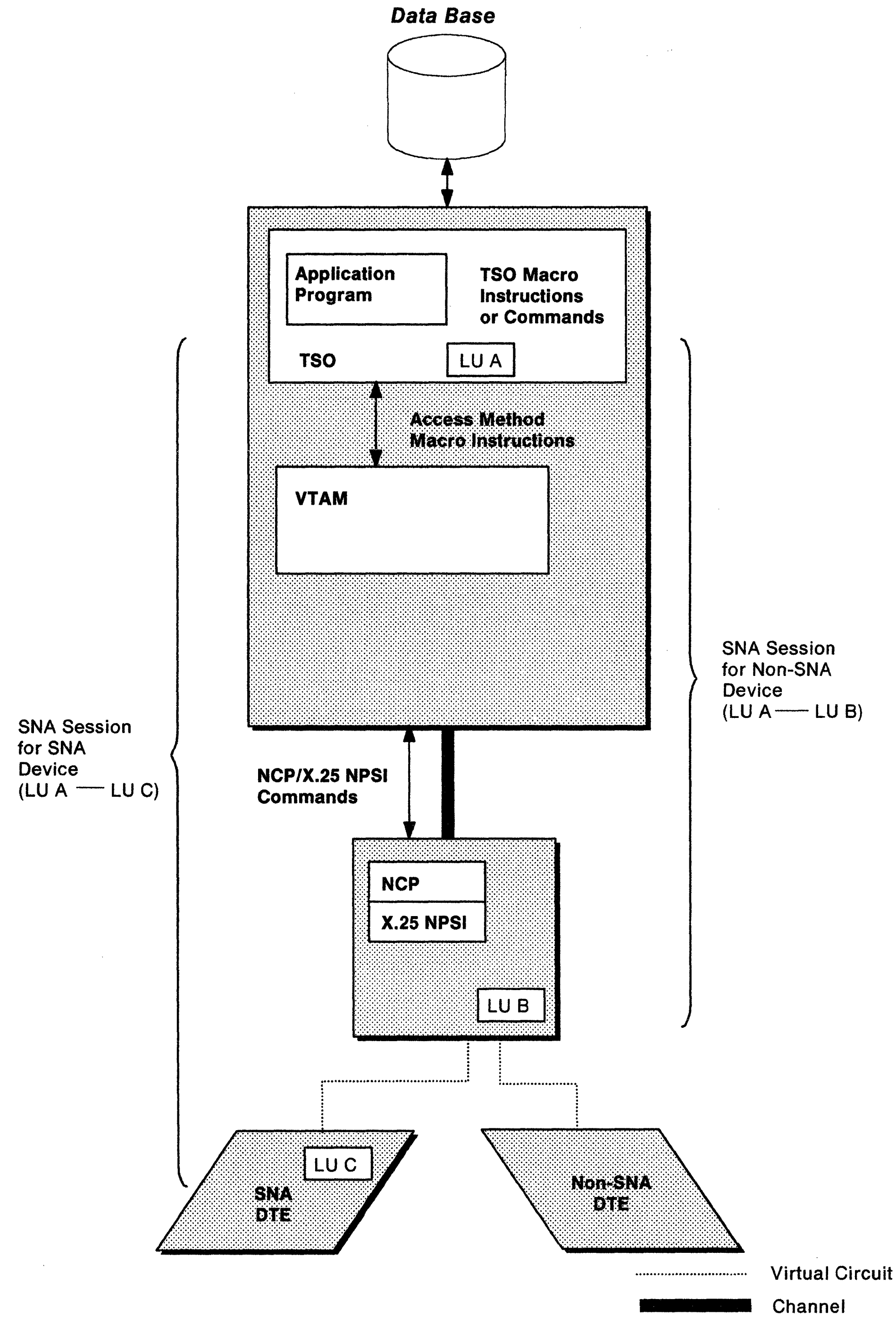


Figure 19. SNA Host Node to SNA Peripheral Node or Non-SNA DTE Using TSO

LU Definition

The LU simulator code in X.25 NPSI simulates an LU by a method similar to a Network Terminal Option (NTO) device for each start-stop mode DTE accessing the SNA host through X.25 NPSI. These devices appear to the host access method as LU type 1 SNA devices and support the same BIND parameters as the SDLC 3767 Communications Terminal. Therefore, you must define the appropriate profile for an LU type 1 device.

For start-stop mode DTEs running over a switched virtual circuit (SVC) using TSO, you should define the LU with the TERM=TWX keyword in the VTAM switched major node (SMN) definition as shown in Figure 20.

```
CA001    PU  ADDR=01,
             PACING=1,
             VPACING=2,
             PUTYPE=1,
             IDBLK=003,
             IDNUM=0F010,
             MAXDATA=261,
             DISCNT=YES
CX001    LU  LOCADDR=0,
             SSCPFM=USSNTO,
             MODTAB=MODRAL,
             DLOGMOD=SSCICSF,
             TERM=TWX,
             USSTAB=USSTAB
```

Figure 20. Non-SNA Entry for X.25 NPSI for Switched Major Node

The same parameters can be coded in the X25.PU and X25.LU statements, or in the X25.VC statement at X.25 NPSI generation for PVCs. See *X.25 NPSI Planning and Installation* for more information on coding these statements.

Line Control

The use of TSO with an LU type 1 device allows only line mode transactions. Problems controlling carriage return and line feed when simulating LU type 1 devices are quite common. For several start-stop mode DTEs, an End of Text (EOT) character is normally sent as the delimiting character rather than of a carriage return and line feed. To alleviate this problem, use TERM=TWX in the switched major node definition for the LU. The TERM=TWX specification tells TSO that the remote DTE is an NTO device.

Although NTO is not used within X.25 NPSI, the start-stop devices are actually simulated as NTO devices within the communication controller. This means VTAM expects to find a carriage return and line feed at the end of every input line. If either or both the carriage return and line feed are missing, VTAM inserts the missing character. The character VTAM inserts can cause a formatting problem. To avoid this problem, you should change the EOT character to a line feed using translation tables.

For more information about TSO LU definition and line control, see *VTAM Installation and Resource Definition*.

Password Protection

When using the TSO subsystem with LU type 1 devices, you must code the DCODE parameter on the MODEENT keyword of the mode table used for that session. It must be coded as:

MODEENT DCODE=devtype

See *VTAM Customization* for valid values for devtype. The value to code for password protection is X'80'. You must code this value whether the terminal at the remote end is typewriter or display type. Coding the DCODE keyword permits TSO to request the USERID first, and then the PASSWORD. The user's response to the PASSWORD prompt is suppressed.

During normal processing, the data entered by the user is returned to the terminal to be displayed. This process, often referred to as *echoing*, is a technique of visually assuring that the data was received correctly. In the special case of a PASSWORD, displaying the data that you entered is a security exposure.

X.25 NPSI uses the enable presentation (ENP) and inhibit presentation (INP) characters to define when a field is not to be echoed to the terminal.

TSO places an INP character at the end of the data stream when prompting for the password. X.25 NPSI scans the data for the INP character. If the character is present, X.25 NPSI ensures that, as the password is keyed, it cannot be read.

For terminals that are using echoing of input data, X.25 NPSI performs password protection by setting PAD parameter 2 to a value 0 before entry of the password and then returns PAD parameter 2 to a value of 1 after the password is entered. While PAD parameter 2 has the value 0, no input data is echoed.

For terminals that are not using echoing of input data, X.25 NPSI performs password protection by creating an overstrike sequence of 8 positions. Note that the echoing technique is also quite suitable for typewriter type terminals.

Upon receiving and processing the response, TSO starts the next output buffer with the ENP character. This causes the PAD parameter 2 with a value of 1 to be sent for terminals that are using echoing of input data.

Customization

When accessing TSO, you often need to customize the X.3 PAD parameters to suit your installation needs. For example, you may want to ensure that you can perform error correction in the PAD buffer by setting PAD parameter 16 to 8. This is an alternative to having erroneous data passed over the X.3 PSDN only to be corrected within TSO. To ensure that the PAD parameters are set to your desired values:

1. Select a profile from the supplier of the PAD service with the desired values.
2. Modify the profile immediately after the connection to the PAD service to set the desired values. (This can be automated if you are using a PC.)
3. Customize the X.25 NPSI integrated PAD support code so that the desired values are set when the SNA session is started using X.29 PAD messages.

For additional information on PAD parameter customization, including PAD parameter settings, see *X.25 NPSI Diagnosis, Customization, and Tuning*.

NetView

NetView™[5] is a licensed program that permits automated operation management of a communication network. This program provides a command facility that contains message automation, hardware monitor, session monitor, and status monitor. The following sections describe how to use the NetView program with X.25 NPSI attached terminals, and what functions of the NetView program work on X.25 NPSI attached resources.

5 NetView is a trademark of the International Business Machines Corporation.

Using NetView with X.25 NPSI Attached Terminals

The NetView program can be used from terminals connected through LLC type 2 or type 3. The NetView program cannot communicate through LLC types 0, 4, or 5 directly, but it can communicate through these LLC types through a relay program, such as the IBM-provided General Teleprocessing Monitor for Open Systems Interconnection (GTMOSI). The GTMOSI program maps non-SNA devices (such as a 3101 or a Minitel) into a 3278 to allow connection to the NetView program.

The NetView program also participates fully in the SVCSC environment, and is able to connect with another NetView system through an X.25 interface. This permits great flexibility in creating a network topology.

Figure 21 on page 134 shows the data flow for a network configuration that uses the NetView program and VTAM.

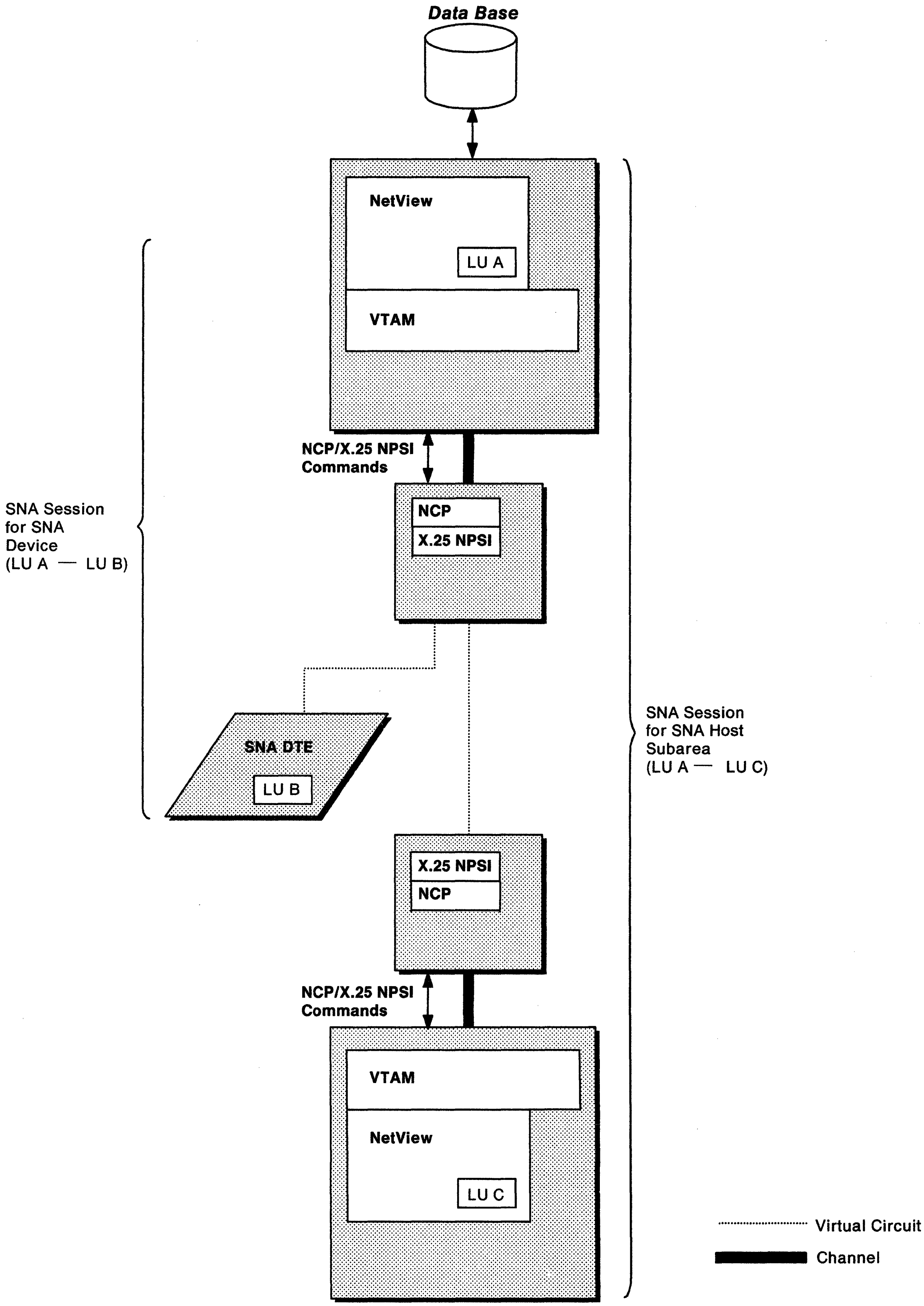


Figure 21. SNA Host Node to SNA DTE or Non-SNA DTE Using NetView and VTAM

NetView Functions Usable with X.25 NPSI Resources

The following sections describe the high-level functions and list the additional network-management capabilities of the NetView program. The relationship of the NetView program to SNA and non-SNA connections is also described. For more information, see *Planning and Reference for NetView, NCP, and VTAM*.

Command Facility

The command facility provides the environment to enter commands to all components of the network. These components include the host access method, application subsystems, the operating system, and the large and departmental processors comprising the network hosts.

Messages from all these network components can enter the command facility and commands can be automatically sent to the originating component, or any required destination.

To illustrate how these commands are automatically sent with X.25 NPSI, use command lists, or REXX EXECs, to retry the activation of an MCH PU when the PSDN is temporarily out of order. The VTAM message showing that the PU is inactive can be used to initiate an activate command after a period of time, without the need for any operator intervention. If the PSDN is still out of order after the wait, the MCH PU can be reactivated by the command facility.

Hardware Monitor

The hardware monitor enables you to access problem information generated at resources that are either link-attached or channel-attached to the host system. The information passed to the host consists of the following:

- Statistics: records of traffic and recoverable errors
- Events: unusual occurrences detected at a device or program.

The hardware monitor supports X.25 NPSI by managing and presenting the RECFMS request units, generated by X.25 NPSI, which contain event and statistical data.

Note: For V3R3 only, the use of billing units as statistical data affects panels NPDA-51F and NPDA-53F in NetView. In these panels, both the TRANSMISSION TOTAL and RECEIVE TOTAL can reflect billing units and not packet totals.

Session Monitor

The session monitor is used to collect and correlate data about sessions and routes, and to obtain online access to the collected data. It allows you to examine information related to the SNA network and to identify network problems.

Session monitor can be used to measure response time, when using LLC 2 or 3, if the remote DTE has the response time monitor (RTM) facility.

Status Monitor

The status monitor displays network status and accepts network operator commands. Status information that can be displayed includes the following:

- Status summaries
- Details for a domain
- Status of a single node
- Node descriptions.

SNA Connection

For SNA connections, those connections that flow through VCs using LLC types 2 and 3, X.25 NPSI provides path control functions only. Neither the SNA device nor the host destination is aware that the link uses X.25 protocols. Network management, functioning principally through the PU, is also unaffected by the presence of the PSDN. Management is not affected when using either connections to peripheral nodes or subarea nodes, whether through permanent or switched virtual circuits.

Support for Network Management

Network management using the NetView program is supported across the X.25 connection. The existence of an X.25 interface is transparent to the SNA requests and responses that are used for network management.

Non-SNA Connection

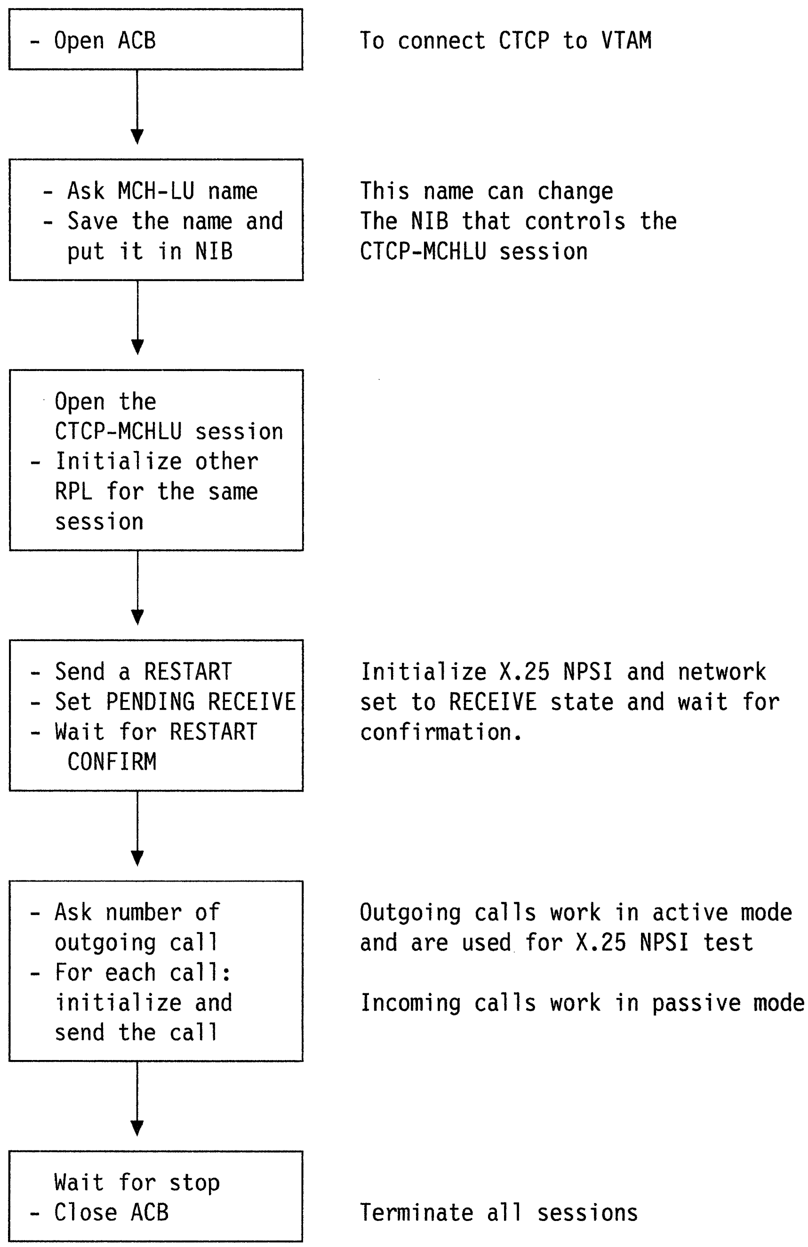
For non-SNA connections, that is LLC types 0, 4, and 5, X.25 NPSI simulates PUs and LUs within the communication controller. Thus, the SNA-oriented network management functions, which depend on the SNA resources, do not extend past the communication controller. The event and statistical data generated by X.25 NPSI are the only network management events provided by the NetView program for these types of virtual circuits. However, because the event data provided by X.25 NPSI includes diagnostic information created by the PSDN and the remote non-SNA DTE in control packets, as much network management data is presented through NetView as possible.

Appendix A. Example of Programming a DATE CTCP

Appendix A. Example of Programming a DATE CTCP

This appendix provides an example of the programming required to create a CTCP. The programming example includes use of the DATE function of X.25 NPSI.

Main Flow of the CTCP



Description of Sample CTCP

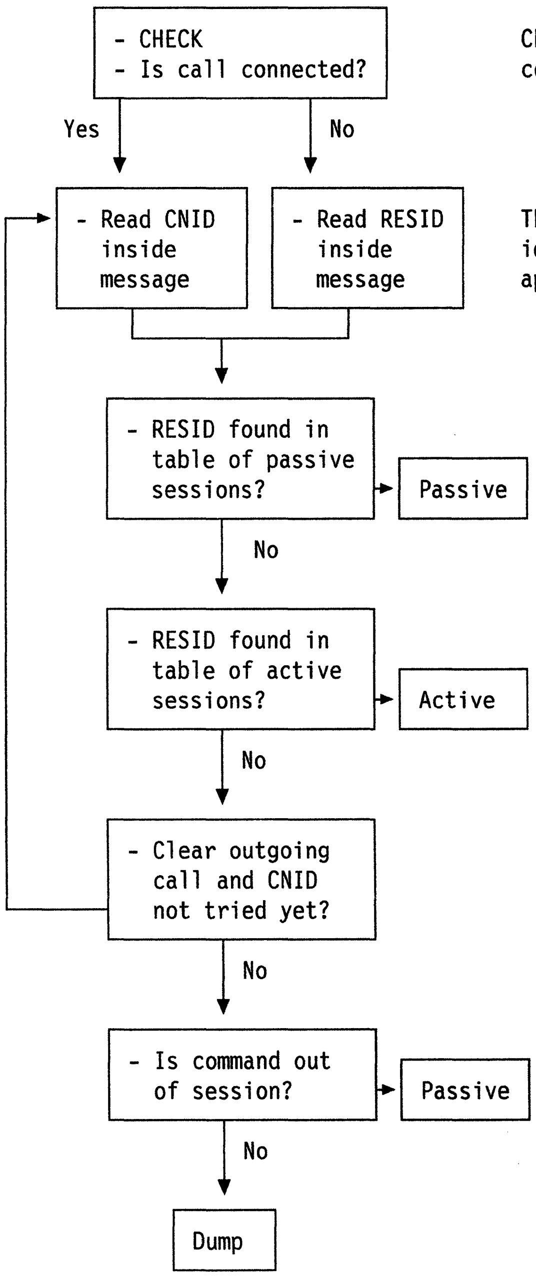
This CTCP has two parts:

- Active part: If callouts are requested, the callout part of this CTCP serves to test the X.25 NPSI DATE function. A certain number of commands are sent, and the CTCP verifies that the received responses are the expected ones.

  Because this CTCP is prepared to handle only one MCH, the call requests cause incoming calls on other VCs of the same MCH. Similar logic applies for the other control or data packets.

- Passive part: This part receives incoming calls, and does not serve to test the X.25 NPSI DATE function. This part has the normal DATE CTCP reactions.

Receive Exit Routine



Check for any asynchronous request command of DATE

The RESID or the CNID identifies the session between application and a VC-LU

The session exists and is not controlled

The session exists and is under control

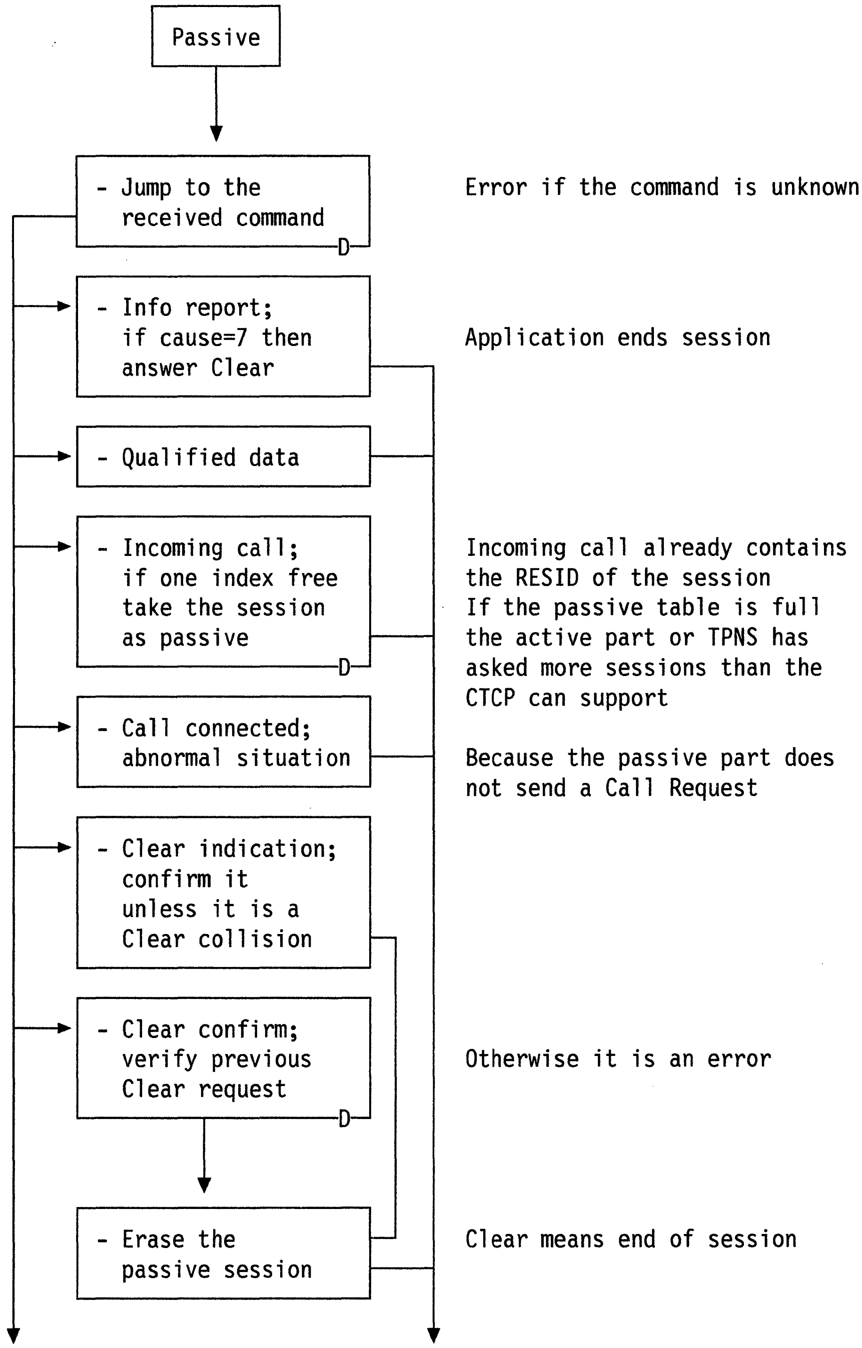
This command prevents the session from beginning; thus, it refers to a CNID

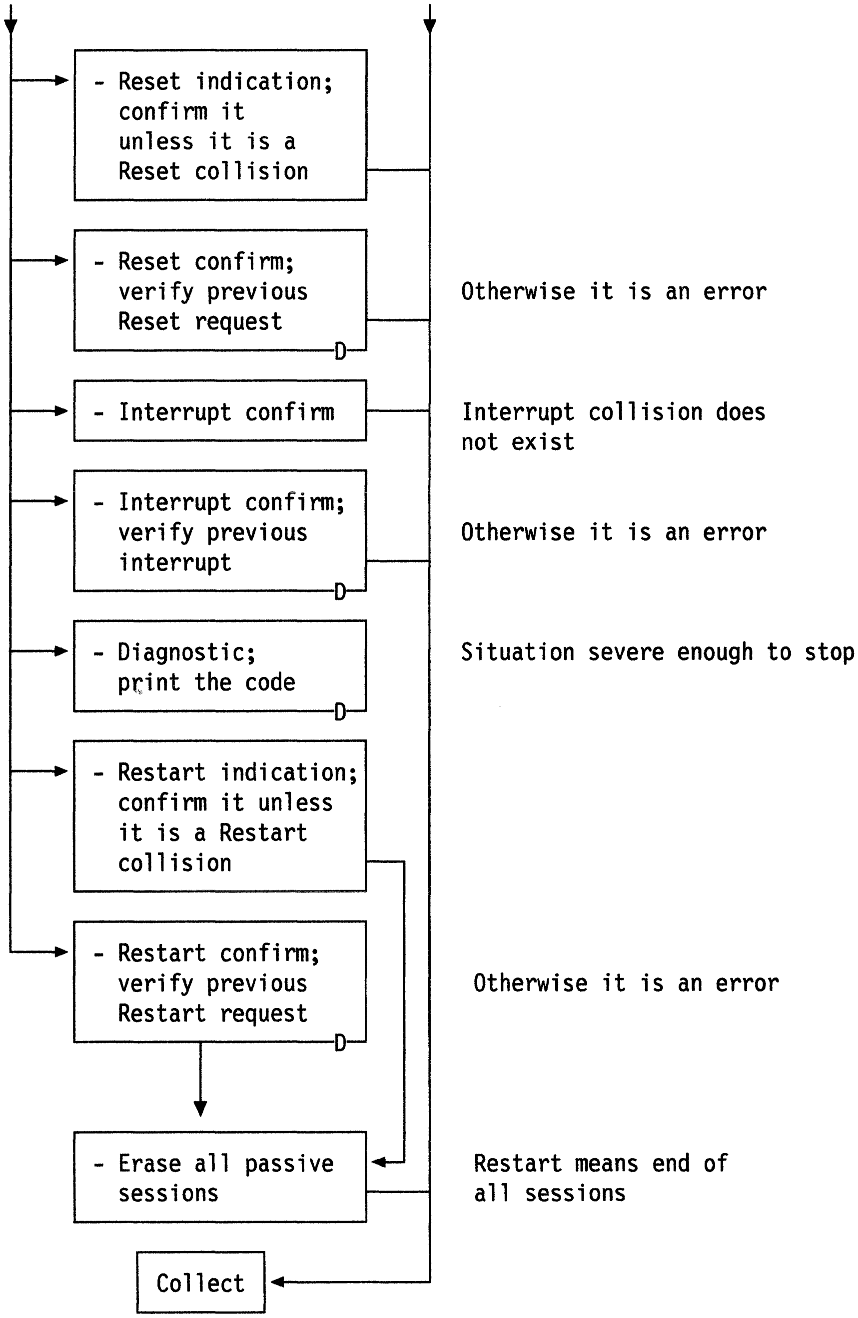
These commands do not deal with a specific session (RESTART DIAGNOSTIC...)

Abnormal command

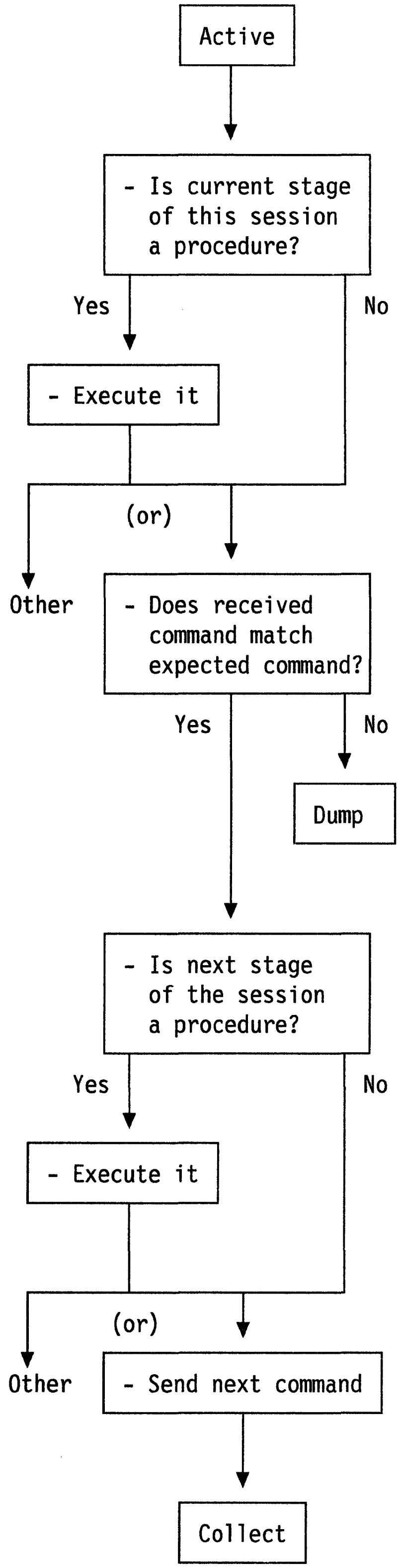
Passive Part of the Receive Routine

The D in the bottom right corner of some frames means Dump if the stated conditions are not true.




- Reset indication; confirm it unless it is a Reset collision
- Reset confirm; verify previous Reset request
D
- Interrupt confirm
- Interrupt confirm; verify previous interrupt
D
- Diagnostic; print the code
D
- Restart indication; confirm it unless it is a Restart collision
- Restart confirm; verify previous Restart request
D
- Erase all passive sessions
Collect
Otherwise it is an error
Interrupt collision does not exist
Otherwise it is an error
Situation severe enough to stop
Otherwise it is an error
Restart means end of all sessions

Active Part of the Receive Routine



The current stage is a pointer excerpt of a table (the script)

The procedure gives control to any point or gives the next command

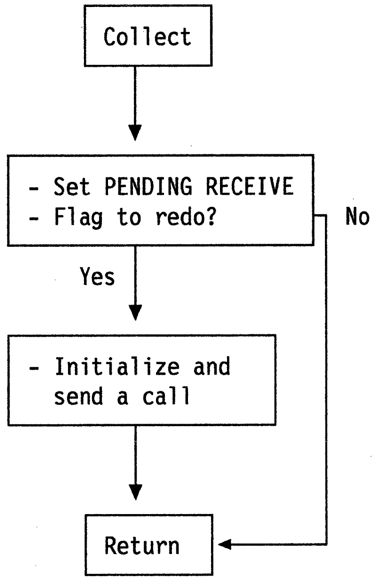
Allow full control of the session

Session is progressing abnormally

The next stage is a pointer excerpt of a table (the script)

The procedure gives control to any point or gives the next command

End of the Receive Routine



This second send allows the program to loop

Return to VTAM

Sample Output Assembly of a CTCP Program

The following is a sample of output assembly generated by a CTCP program using the DATE function of X.25 NPSI.

```
                                           EXTERNAL SYMBOL DICTIONARY                                          PAGE    1
SYMBOL   TYPE  ID  ADDR  LENGTH  LD ID FLAGS                                                         ASM H V 02 16.17 09/27/88
CTCPSTR   SD 0001 000000 0013D4        00
                                                                                                                PAGE    2
  LOC  OBJECT CODE      ADDR1 ADDR2  STMT   SOURCE STATEMENT                                     ASM H V 02 16.17 09/27/88
000000                               517+CTCPSTR  CSECT                               * CODE SECTION NAME           01-00055
000000                               518+         DS    0H                                                          02-SAVE
000000 90EC D00C             0000C   519+         STM   14,12,12(13)                     SAVE REGISTERS             02-SAVE
000004 0530                          520+         BALR  R3,R0                             * SET FIRST BASE              01-00057
                             00006   521+         USING *,R3,R4,R5                        * ASSEMBLER DIRECTIVE         01-00058
000006 5840 300E             00014   522+EXTBASES L     R4,B10000                         * SET SECOND BASE             01-00059
00000A 5850 3012             00018   523+         L     R5,B20000                         * SET THIRD BASE              01-00060
00000E 47F0 3016             0001C   524+         B     LB0000                            * SKIP BASES VALUES           01-00061
000012 0000
000014 00001006                      525+B10000   DC    A(EXTBASES+4096)                  * VALUE OF SECOND BASE        01-00062
000018 00002006                      526+B20000   DC    A(EXTBASES+8192)                  * VALUE OF THIRD BASE         01-00063
00001C 18CD                          527+LB0000   LR    R12,R13                           * OLD SAVE AREA IN R12        01-00064
00001E 41D0 33D6             003DC   528+         LA    R13,MAINSAVE                      * NEW SAVE AREA IN R13        01-00065
000022 50DC 0008             00008   529+         ST    R13,8(R12)                        * NEW POINTER IN OLD AREA     01-00066
000026 50C0 33DA             003E0   530+         ST    R12,MAINSAVE+4                    * OLD POINTER IN NEW AREA     01-00067
                                     531+         PRINT NOGEN                                                           02-00042
                                     532 * OPEN ACB MCH ******************************************************** CTC51700
                                     533          OPEN  ACBMCH                                                          CTC51800
                                     539          MVERIFY                                 * VERIFY OPEN COMPLETION      CTC51900
                                     540+         PRINT GEN                                                             02-00033
000036 12FF                          541+         LTR   R15,R15                           * TEST RETURN CODE            01-00282
000038 4780 303A             00040   542+         BZ    MV0001                            * NO PROBLEM                  01-00283
00003C 45E0 3446             0044C   543+         BAL   R14,FAILEXIT                      * DUMP                        01-00284
                             00040   544+MV0001   EQU   *                                 * RESUME STATEMENTS           01-00285
                                     545+         PRINT NOGEN                                                           02-00042
                                     546 * GET MCH LU NAME ***************************************************** CTC52000
000040 D703 3432 3432 00438 00438    547          XC    WTORECB(4),WTORECB                * ERASE WTORECB FOR USE       CTC52100
000046 D207 341E 3426 00424 0042C    548          MVC   REPLY(8),BLANK8                   * INITIALISE REPLY            CTC52200
                                     549          WTOR  ' CTCP ||| ENTER MCH LU NAME (DU3) :',REPLY,8,WTORECB          CTC52300
                                     560          WAIT  1,ECB=WTORECB                     * WAIT FOR OPERATOR REPLY     CTC52400
00008C D507 341E 3426 00424 0042C    564          CLC   REPLY(8),BLANK8                   * COMPARE TO NO ANSWER        CTC52500
000092 4770 3096             0009C   565          BNE   UPCASE                            * THERE IS AN ANSWER          CTC52600
000096 D202 341E 3F88 00424 00F8E    566          MVC   REPLY(3),=C'DU3'                  * TAKE DEFAULT ANSWER         CTC52700
00009C D607 341E 3426 00424 0042C    567 UPCASE   OC    REPLY(8),BLANK8                   * CONVERT INTO UPPER CASE     CTC52800
0000A2 D207 30B9 341E 000BF 00424    568          MVC   WTONAME+23(8),REPLY               * TRANSFERT TO STRING         CTC52900
                                     569 WTONAME  WTO   ' MCH LU NAME :           '       * CONFIRM NAME                CTC53000
                                     577          MODCB NIB=NIBMCH,NAME=(*,REPLY)         * PUT MCH LU NAME IN NIB      CTC53100
                                     597          MVERIFY                                 * VERIFY MODCB COMPLETION     CTC53200
                                     598+         PRINT GEN                                                             02-00033
                                                                                                                PAGE   13
  LOC  OBJECT CODE      ADDR1 ADDR2  STMT   SOURCE STATEMENT                                     ASM H V 02 16.17 09/27/88
000104 12FF                          599+         LTR   R15,R15                           * TEST RETURN CODE            01-00282
000106 4780 3108             0010E   600+         BZ    MV0002                            * NO PROBLEM                  01-00283
00010A 45E0 3446             0044C   601+         BAL   R14,FAILEXIT                      * DUMP                        01-00284
                             0010E   602+MV0002   EQU   *                                 * RESUME STATEMENTS           01-00285
                                     603+         PRINT NOGEN                                                           02-00042
                                     604 * ESTABLISH SESSION BETWEEN CTCP AND MCH, INITIALISE SEND RPL ********* CTC53300
                                     605          OPNDST RPL=RPLMCH,OPTCD=(SYN,ACQUIRE,START)                           CTC53400
                                     664          MVERIFY                                 * VERIFY OPNDST COMPLETION    CTC53500
                                     665+         PRINT GEN                                                             02-00033
00015E 12FF                          666+         LTR   R15,R15                           * TEST RETURN CODE            01-00282
000160 4780 3162             00168   667+         BZ    MV0003                            * NO PROBLEM                  01-00283
000164 45E0 3446             0044C   668+         BAL   R14,FAILEXIT                      * DUMP                        01-00284
                             00168   669+MV0003   EQU   *                                 * RESUME STATEMENTS           01-00285
                                     670+         PRINT NOGEN                                                           02-00042
                                     671          SHOWCB AM=VTAM,RPL=RPLMCH,FIELDS=ARG,AREA=NIBARG,LENGTH=4             CTC53600
                                     689          MVERIFY                                 * VERIFY SHOWCB COMPLETION    CTC53700
                                     690+         PRINT GEN                                                             02-00033
000198 12FF                          691+         LTR   R15,R15                           * TEST RETURN CODE            01-00282
00019A 4780 319C             001A2   692+         BZ    MV0004                            * NO PROBLEM                  01-00283
00019E 45E0 3446             0044C   693+         BAL   R14,FAILEXIT                      * DUMP                        01-00284
                             001A2   694+MV0004   EQU   *                                 * RESUME STATEMENTS           01-00285
                                     695+         PRINT NOGEN                                                           02-00042
                                     696          MODCB AM=VTAM,RPL=RPLMCHS,ARG=(*,NIBARG)                              CTC53800
```

```
                                        715          MVERIFY                            * VERIFY MODCB  COMPLETION  CTC53900
                                        716+         PRINT GEN                                                      02-00033
0001D8 12FF                             717+         LTR   R15,R15                      * TEST RETURN CODE          01-00282
0001DA 4780 31DC              001E2     718+         BZ    MV0005                       * NO PROBLEM                01-00283
0001DE 45E0 3446              0044C     719+         BAL   R14,FAILEXIT                 * DUMP                      01-00284
                              001E2     720+MV0005   EQU   *                            * RESUME STATEMENTS         01-00285
                                        721+         PRINT NOGEN                                                    02-00042
                                        722 * SEND RESTART REQUEST ************************************************** CTC54000
0001E2 D701 3F9E 3F9E 00FA4 00FA4       723          XC    RESIVAL(2),RESIVAL           * NULLIFY RESID FOR RESTART CTC54100
0001E8 92FF 3F8C       00F92            724          MVI   RSARBOOL,TRUE                * SET RESTART BOOLEAN       CTC54200
                                        725          SETSEND RSARREQ                    * SEND RESTART REQUEST      CTC54300
                                        726+         PRINT GEN                                                      02-00033
0001EC 4170 4073              01079     727+         LA    R7,RSARREQ                   * GET ADRESS OF DATA        01-00262
0001F0 4187 0001              00001     728+         LA    R8,1(R7)                     * GET ADRESS OF STRING      01-00266
0001F4 1799                             729+         XR    R9,R9                        * NULLIFY R9                01-00267
0001F6 4397 0000              00000     730+         IC    R9,0(R7)                     * GET LENGTH OF STRING      01-00268
0001FA 45E0 3B06              00B0C     731+         BAL   R14,SBRSEND                  * CALL SEND PROCEDURE       01-00269
                                        732+         PRINT NOGEN                                                    02-00042
                                        733 * SET PENDING RECEIVE MCH EXIT ROUTINE ********************************* CTC54400
0001FE D200 42CA 3F8B 012D0 00F91       734          MVC   AREAMCHR(1),=C' '            * ERASE AREAMCHR            CTC54500
000204 D2FE 42CB 42CA 012D1 012D0       735          MVC   AREAMCHR+1(255),AREAMCHR     * ERASE AREAMCHR            CTC54600
00020A D203 43CA 43C9 013D0 013CF       736          MVC   AREAMCHR+256(4),AREAMCHR+255 * ERASE AREAMCHR            CTC54700
                                        737          RECEIVE RPL=RPLMCH,AREA=AREAMCHR,RECLEN=260,              CCTC54800
                                                           OPTCD=(ASY,SPEC),RTYPE=(DFSYN,DFASY,NRESP),EXIT=RCVMCH  CTC54900
                                        785          MVERIFY                            * VERIFY RECEIVE COMPLETION CTC55000
                                        786+         PRINT GEN                                                      02-00033
000274 12FF                             787+         LTR   R15,R15                      * TEST RETURN CODE          01-00282
000276 4780 3278              0027E     788+         BZ    MV0006                       * NO PROBLEM                01-00283
00027A 45E0 3446              0044C     789+         BAL   R14,FAILEXIT                 * DUMP                      01-00284
```

Glossary, Bibliography, and Index

Glossary

This glossary contains terms and abbreviations related to X.25, X.25 NPSI, SNA, and telecommunications. It includes information from:

- The *American National Dictionary of Information Processing Systems*, copyright 1982 by the Computer and Business Equipment Manufacturers Association (CBEMA). Copies can be purchased from the American National Standards Institute at 1430 Broadway, New York, New York 10018. These definitions are identified by an asterisk (*).
- The *ISO Vocabulary — Information Processing*, developed by the International Organization for Standardization, Technical Committee 97, Subcommittee 1. Definitions from published sections of this vocabulary are identified by the symbol "(ISO)" following the definition. Definitions from draft international standards, draft proposals, and working papers in development by the ISO/TC97/SC1 vocabulary subcommittee are identified by the symbol "(TC97)," indicating that final agreement has not yet been reached among participating members.
- The *CCITT Eighth Plenary Assembly Red Book, Terms and Definitions*, and working documents published by the International Telegraph and Telephone Consultative Committee of the International Telecommunication Union, Geneva, 1985. These are identified by the symbol "(CCITT/ITU)" following the definition.

For abbreviations, the definition usually consists only of the words represented by the letters; for complete definitions, see the entries for the words.

A

ABM. Asynchronous balanced mode.

ACB. (1) Application control block. (2) In VTAM, access method control block. (3) In NCP, adapter control block.

access barred. In data communication, a condition in which a data terminal equipment (DTE) cannot call the DTE identified by the selection signals.

access method. A technique for moving data between main storage and input/output devices.

adapter control block (ACB). In NCP, a control block that contains line control information and the states of I/O operations for BSC lines, SS lines, or SDLC links.

alert. (1) In SNA, a record sent to a system problem management focal point to communicate the existence of an alert condition. (2) In the NetView program, a high priority event that warrants immediate attention. This data base record is generated for certain event types that are defined by user-constructed filters.

API. Application program interface.

application program interface (API). (1) The formally defined programming language interface between an IBM system control program or licensed program and its user. (2) The interface through which an application program interacts with an access method. In VTAM, it is the language structure used in control blocks so that application programs can reference them and be identified to VTAM.

ASCII. American National Standard Code for Information Interchange.

asynchronous balanced mode (ABM). An operational mode of a balanced data link in which either combined station can send commands at any time and can initiate transmission of response frames without explicit permission from the other combined station. See also *normal response mode (NRM), asynchronous response mode (ARM).*

asynchronous response mode (ARM). An operational mode of an unbalanced data link in which a secondary station may initiate transmission without explicit permission from the primary station. See also *asynchronous balanced mode (ABM), normal response mode (NRM).*

B

balanced data link. In data communication, a data link between two participating combined stations; for transmissions it originates, each station can transmit both command frames and response frames, organize its data flow, and perform error recovery operations at the data link level. Contrast with *unbalanced data link.*

balanced station. Synonym for *combined station.*

basic information unit (BIU). In SNA, the unit of data and control information that is passed between half-sessions. It consists of a request/response header (RH) followed by a request/response unit (RU).

begin bracket. In SNA, the value (binary 1) of the begin-bracket indicator in the request header (RH) of the first request in the first chain of a bracket; the value denotes the start of a bracket. Contrast with *end bracket.* See also *bracket.*

bidder. In SNA, the LU-LU half-session defined at session activation as having to request and receive permission from the other LU-LU half-session to begin a bracket. Contrast with *first speaker*. See also *bracket protocol*.

billing function. An optional function of X.25 NPSI GATE Fast Connect that provides the CTCP with billing information.

binary synchronous communication (BSC). (1) Communication using binary synchronous line discipline. (2) A uniform procedure, using a standardized set of control characters and control character sequences, for synchronous transmission of binary-coded data between stations.

bind. In SNA, a request to activate a session between two logical units (LUs).

BIU. Basic information unit.

boundary function. (1) A capability of a subarea node to provide protocol support for attached peripheral nodes, such as: (a) interconnecting subarea path control and peripheral path control elements, (b) performing session sequence numbering for low-function peripheral nodes, and (c) providing session-level pacing support. (2) The component that provides these capabilities. See also *boundary node, intermediate routing function, subarea node*.

boundary node. (1) A subarea node with boundary function. See also *boundary function*. (2) The programming component that performs FID2 (format identification type 2) conversion, channel data link control, pacing, and channel or device error recovery procedures for a locally attached station. These functions are similar to those performed by a network control program for an NCP-attached station.

bracket. In SNA, one or more chains of request units (RUs) and their responses that are exchanged between the two LU-LU half-sessions and that represent a transaction between them. A bracket must be completed before another bracket can be started. Examples of brackets are data base inquiries/replies, update transactions, and remote job entry output sequences to work stations. See also *begin bracket, end bracket*.

bracket protocol. In SNA, a data flow control protocol in which exchanges between the two LU-LU half-sessions are achieved through the use of brackets, with one LU designated at session activation as the first speaker and the other as the bidder. The bracket protocol involves bracket initiation and termination rules. See also *bidder, first speaker*.

BSC. Binary synchronous communication.

C

call. (1) A transmission for the purpose of identifying the transmitting station for which the transmission is intended. (2) An attempt to reach a user, whether or not successful. (CCITT/ITU)

call accepted packet. A call supervision packet transmitted by a called data terminal equipment (DTE) to inform the data circuit-terminating equipment (DCE) of the acceptance of the call. (CCITT/ITU)

call accepted signal. A call control signal that is sent by the called data terminal equipment (DTE) to indicate that it accepts the incoming call. (TC97)

call collision. A condition that occurs when a data terminal equipment (DTE) transmits a call request signal and a data circuit-terminating equipment (DCE) simultaneously transmits an incoming call signal; neither the DTE nor the DCE receives the expected response. See also *clear collision, reset collision*.

call connected packet. A call supervision packet transmitted by a data circuit-terminating equipment (DCE) to inform a calling data terminal equipment (DTE) of the complete establishment of a call. (CCITT/ITU)

called party. On a switched line, the location to which a connection is established.

call establishment. The sequence of events for the establishment of a data connection. (CCITT/ITU)

calling. The process of transmitting selection signals in order to establish a connection between data stations. (TC97)

calling party. On a switched line the location that originates a connection.

call-not-accepted signal. A call control signal sent by the called data terminal equipment (DTE) to indicate that it does not accept the incoming call. (TC97)

call request packet. A call supervision packet transmitted by a data terminal equipment (DTE) to ask for a call establishment through the network. (CCITT/ITU)

call request signal. A signal in the call establishment phase which alerts the data circuit-terminating equipment (DCE) that the data terminal equipment (DTE) wishes to make a call. (CCITT/ITU)

call supervision packet. A packet used for the establishment or the clearing of a call at the DTE/DCE interface. (CCITT/ITU)

Casual Connection. Two VTAMs, a VTAM and an NCP, or two NCPs connected as SNA type 2.1 nodes.

CCITT. International Telegraph and Telephone Consultative Committee.

CCU. Central control unit.

central control unit (CCU). The communication controller hardware unit that contains the circuits and data flow paths needed to execute instructions and to control controller storage and the attached adapters.

CPU. Central processing unit.

central processing unit (CPU). The part of a computer that includes the circuits that control the interpretation and execution of instructions.

chaining. (1) A method of storing records in which each record belongs to a list or group of records and has a linking field for tracing the chain. (2) In VSE, a logical connection of sublibraries to be searched by the system for members of the same type, for example, phase or object modules.

channel. See *data communication channel*.

CICS. Customer Information Control System.

circuit. See *data circuit*.

circuit switched data network (CSDN). A process that, on demand, connects two or more data terminal equipment (DTE) and permits the exclusive use of a data circuit between them until the connection is released. Synonymous with *line switching*. See also *message switching, packet switching*.

circuit switched data transmission service. A service using circuit switching to establish and maintain a connection before data can be transferred between data terminal equipments (DTEs). (TC97) See also *packet switched data transmission service*.

circuit switching. A process that, on demand, connects two or more data terminal equipments (DTEs) and permits the exclusive use of a data circuit between them until the connection is released. * (ISO) Synonymous with *line switching*. See also *message switching, packet switching*.

class of service (COS). In SNA, a designation of the path control network characteristics, such as path security, transmission priority, and bandwidth, that apply to a particular session. The end user designates class of service at session initiation by using a symbolic name that is mapped into a list of virtual routes, any one of which can be selected for the session to provide the requested level of service. See also *user class of service*.

clear collision. A condition that occurs when a data terminal equipment (DTE) and a data circuit-terminating equipment (DCE) simultaneously transmit a clear request packet and a clear indication packet over the same logical channel. See also *call collision, reset collision*.

clear indication packet. A call supervision packet transmitted by a data circuit-terminating equipment (DCE) to inform a data terminal equipment (DTE) of the clearing of a call. (CCITT/ITU)

clear request packet. A call supervision packet transmitted by a data terminal equipment (DTE) to ask for clearing a call. (CCITT/ITU)

closed user group. In a group of users, a subgroup that is assigned a facility that enables a member of one subgroup to communicate only with other members of the subgroup. (TC97) A data terminal equipment (DTE) can belong to more than one closed user group.

closed user group with outgoing access. A closed user group that has a user assigned facility which enables that user to communicate with other users of a public data network transmission service, where appropriate, or with users having a data terminal equipment (DTE) connected to any other public switched network to which interworking facilities are available. (CCITT/ITU)

CNM. Communication network management.

combined station. (1) In high-level data link control (HDLC), the part of a data station that supports the combined control functions of the data link, generates commands and responses for transmission, and interprets received commands and responses. (ISO)

Note: Specific responsibilities assigned to a combined station include initialization of control signal interchange, organization of data flow, interpretation of received commands, and generation of appropriate responses and actions regarding error control and error recovery functions at the data link level. (2) A data station that generates commands and responses for transmission over a data link and interprets received commands and responses. (3) Synonymous with *balanced station*. See also *primary station, secondary station*.

command frame. A frame transmitted by a primary station or a frame transmitted by a combined station that contains the address of the other combined stations. (TC97)

command list (CLIST). A list of commands and statements designed to perform a specific function for the user. Command lists can be written in REXX or in NetView Command List Language.

communication and transmission control program (CTCP). A user-written or IBM-supplied program used in conjunction with the DATE or GATE function of X.25 NPSI to manage virtual circuits. It executes in the host processor. See also *DATE CTCP, fast connect GATE CTCP, GATE CTCP*.

communication common carrier. In the USA and Canada, a public data transmission service that provides the general public with transmission service facilities; for example, a telephone or telegraph company. See also *Post Telephone and Telegraph Administration, public network*.

communication controller. A type of communication control unit whose operations are controlled by one or more programs stored and executed in the unit; for example, the IBM 3725 Communication Controller. It manages the details of line control and the routing of data through a network.

communication line. Deprecated term for *telecommunication line*.

communication network management (CNM). The process of designing, installing, operating, and managing the distribution of information and controls among end users of communication systems.

communication scanner processor (CSP). A processor in the 3725 Communication Controller that contains a microprocessor with control code. The code controls transmission of data over links attached to the CSP.

contention mode. In data communication, a mode of transmission in which any station may transmit whenever the line is available. If stations transmit simultaneously, protocols determine who wins the contention.

complete packet sequence (CPS). A complete packet sequence contains contiguous full data packets, with the M bit set to 1 and the D bit set to 0, followed by any other data packet.

control block. (ISO) A storage area used by a computer program to hold control information.

control point (CP). (1) A system services control point (SSCP) that provides hierarchical control of a group of nodes in a network. (2) A control point (CP) local to a specific node that provides control of that node, either in the absence of SSCP control (for type 2.1 nodes engaged in peer to peer communication) or to supplement SSCP control.

COS. Class of service.

CP. (1) Control program. (2) Control point.

CPS. Complete packet sequence.

cross-domain. In SNA, pertaining to control of resources involving more than one domain.

cross-network. In SNA, pertaining to control or resources involving more than one SNA network.

cross-network session. AN LU-LU or SSCP-SSCP session whose path traverses more than one SNA network.

cryptographic. Pertaining to transformation of data to conceal meaning.

CSDN. Circuit-switched data network.

CSP. Communication scanner processor.

CTCP. Communication and transmission control program.

CUD. Call user data field.

CUG. Closed user group.

Customer Information Control System (CICS). An IBM licensed program that enables transactions entered at remote terminals to be processed concurrently by user-written application programs. It also includes facilities for building, using, and maintaining data bases.

CV. Control vector.

CWALL. An NCP threshold of buffer availability, below which the NCP will accept only high-priority path information units (PIUs).

D

data channel. A device that connects a processor and main storage with I/O control units. Synonymous with *input/output channel*. Contrast with *data communication channel*.

data circuit. (1) Associated transmit and receive channels that provide a means of two-way data communication. (ISO) (2) See also *physical circuit, virtual circuit*.

Notes:

1. Between data switching exchanges (DSEs), the data circuit may or may not include data circuit-terminating equipment (DCE), depending on the type of interface used at the data switching exchange.
2. Between a data station and a data switching exchange or data concentrator, the data circuit includes the data circuit-terminating equipment at the data station end, and may also include equipment similar to a DCE at the data switching exchange or data concentrator location.

data circuit-terminating equipment (DCE). The equipment installed at the user's premises that provides all the functions required to establish, maintain, and terminate a connection, and the signal conversion and coding between the data terminal equipment (DTE) and the line. (TC97) The DCE may be separate equipment or an integral part of other equipment.

data communication channel. (1) A means of one-way transmission. * (ISO) (2) Contrast with *data channel*. A channel may be provided by frequency- or time-division multiplexing. In CCITT terminology, a channel (data communication channel) provides one-way (simplex) transmission; data circuits and "logical channels" provide two-way (duplex) transmission. In data processing terminology, a channel (an I/O channel or data channel), provides two-way transfers of data. This distinction must be kept in mind when documenting the interface.

data flow control (DFC). In SNA, a request/response unit (RU) category used for requests and responses exchanged between the data flow control layer in one half-session and the data flow control layer in the session partner.

datagram. A self-contained, independent entity of data carrying sufficient information to be routed from the source data terminal equipment (DTE) to the destination DTE without relying on earlier exchanges between the source or destination DTE and the transporting network. (CCITT/ITU)

data link. (1) The assembly of parts of two data terminal equipments that are controlled by a link protocol, and the interconnecting data circuit, that enable data to be transferred from a data source to a data sink. (ISO) (2) The interconnecting data circuit and the link protocol between two or more equipments; it does not include the data source or the data sink. (3) In SNA, synonym for *link*. (4) Contrast with *telecommunication line*.

data link level. The conceptual level of control or processing logic existing in the hierarchical structure of a data station (primary, secondary, or combined station) that is responsible for maintaining control of the data link. The data link level functions provide an interface between the data station high level logic and the data link. These functions include transmit bit insertion and receive bit deletion; address/control field interpretation; command/response generation, transmission, and interpretation; and frame check sequence computation and interpretation. See also *packet level* and *physical level*. (TC97)

data packet. A packet used for the transmission of user data on a virtual circuit at the DTE/DCE interface. (CCITT/ITU)

data station. The data terminal equipment (DTE), the data circuit-terminating equipment (DCE), and any intermediate equipment. * (ISO) Synonymous with *data terminal installation*.

data switching exchange (DSE). The equipment installed at a single location to provide switching functions, such as circuit switching, message switching, and packet switching. (ISO)

data terminal equipment (DTE). That part of a data station that serves as a data source, data sink, or both, and provides for the data communication control function according to protocols. (TC97)

data terminal installation. Synonym for *data station*.

data transfer. The movement, or copying, of data from one location and the storage of the data at another location.

data transfer phase. The phase of a data call during which data signals can be transferred between data terminal equipments (DTEs) connected through the network. See also *network control phase*.

data transfer rate. The average number of bits, characters, or blocks per unit time passing between corresponding equipment in a data transmission system.

data transmission line. Synonym for *telecommunication line*.

DATE. Dedicated Access to X.25 Transport Extension.

DATE CTCP. A CTCP that is used in conjunction with the DATE function of X.25 NPSI to manage virtual circuits.

D bit. Delivery confirmation bit.

DCE. Data circuit-terminating equipment.

DCE clear confirmation packet. A call supervision packet transmitted by a data circuit-terminating equipment (DCE) to confirm the clearing of a call. (CCITT/ITU)

DCE/DTE interface. See *DTE/DCE interface*.

deadlock. (1) Unresolved contention for use of a resource. (2) An error condition in which processing cannot continue because each of two elements of the process is waiting for an action by or a response from the other. (3) An impasse that occurs when multiple processes are waiting for the availability of a resource that will not become available because it is being held by another process that is in a similar wait state.

Dedicated Access to X.25 Transport Extension (DATE). A function of X.25 NPSI that allows a communication and transmission control program (CTCP) to manage virtual circuits to SNA and non-SNA DTEs by processing qualified data, Interrupt, Call, Clear, and Reset packets. The contents of nonqualified data packets are transferred on the LU-LU session between the application program LU and the virtual circuit LU. Control and qualified data packets are transferred on the LU-LU session between the CTCP LU that manages virtual circuits and the multichannel link (MCH) LU.

dedicated channel. A channel that is not switched.

dedicated circuit. A circuit that is not switched.

definite response (DR). In SNA, a value in the form-of-response-requested field of the request header. The value directs the receiver of the request to return a response unconditionally, whether positive or negative, to that request. Contrast with *exception response, no response*.

definite response mode. A mode of operation in which an LU requires a response to its request.

definition statement. (1) In VTAM, the statement that describes an element of the network. (2) In NCP, a type of instruction that defines a resource to the NCP.

DFC. Data flow control.

dial-in. Refers to the direction in which a switched connection is requested by any node or terminal other than the receiving host or an NCP.

dial-out. Refers to the direction in which a switched connection is requested by a host or an NCP.

direct call. A facility which enables the establishment of a call without the need to convey address signals to the network. (CCITT/ITU)

discarded packet. A packet which is destroyed intentionally or by default while being transmitted through the network. (CCITT/ITU)

disconnected mode. Synonym for *disconnected phase*.

disconnected phase. A phase entered by a data circuit-terminating equipment (DCE) when it detects error conditions, recovers from a temporary internal malfunction, or receives a DISC command from a data terminal equipment (DTE). In the disconnected phase, the DCE can initiate link setup but can transmit only DM responses to received frames. See also *information transfer phase*.

DR. (1) In NCP and CCP, dynamic reconfiguration. (2) In SNA, definite response.

DSE. Data switching exchange.

DTE. Data terminal equipment.

DTE busy. Status of a DTE which is unavailable because it cannot accept an additional call. (ISO)

DTE clear confirmation packet. A call supervision packet transmitted by data terminal equipment (DTE) to confirm the clearing of a call. (CCITT/ITU)

DTE/DCE interface. The physical interface elements and the link access procedures between data terminal equipment (DTE) and data circuit-terminating equipment (DCE). (CCITT/ITU)

duplex. In data communication, pertaining to a simultaneous two-way independent transmission in both directions. * Synonymous with *full-duplex*.

E

EBCDIC. Extended binary-coded decimal interchange code.

echoplex mode. In data communication, a mode in which characters are automatically returned to the transmitting data terminal equipment (DTE).

Emulation Program. An IBM control program that allows a channel-attached 3705 or 3725 communication controller to emulate the functions of an IBM 2701 Data Adapter Unit, an IBM 2702 Transmission Control, or an IBM 2703 Transmission Control. See also *network control program*.

ENA. Extended network addressing.

enable presentation (ENP) character. A control character that enables presentation of the following characters to resume after having been stopped by an inhibit presentation (INP) character.

end bracket. In SNA, the value (binary 1) of the end bracket indicator in the request header (RH) of the first request of the last chain of a bracket; the value denotes the end of the bracket. Contrast with *begin bracket*. See also *bracket*.

end-to-end control. A means whereby during the data phase of a call, interconnected data terminal equipment (DTE) may exchange control signals without loss of data bit sequence independence. (CCITT/ITU)

ENP. Enable presentation character.

EP. Emulation Program.

ER. (1) Explicit route. (2) Exception response.

exception request (EXR). In SNA, a request that replaces another message unit in which an error has been detected.

exception response (ER). In SNA, a value in the form-of-response-requested field of a request header (RH).

An exception response is sent only if a request is unacceptable as received or cannot be processed. Contrast with *definite response (DR), no response.*

EXR. Exception request.

extended binary-coded decimal interchange code (EBCDIC). A set of 256 characters, each represented by eight bits.

extended network addressing. The network addressing system that splits the address into an 8-bit subarea and a 15-bit element portion. The subarea portion of the address is used to address host processors or communication controllers. The element portion is used to permit processors or controllers to address resources.

F

fallback. On an IBM 3745 with twin CCUs, the action of switching the lines attached to one CCU to the other CCU.

fast connect. An optional extension of the X.25 NPSI GATE function that preestablishes the SNA sessions between the host logical unit (LU) and the simulated LUs in X.25 NPSI.

fast connect GATE CTCP. A CTCP that is used in conjunction with the fast connect GATE function of X.25 NPSI to manage virtual circuits. See also *GATE CTCP.*

fast select. An option of a virtual call facility that allows inclusion of data in call-setup and call-clearing packets. (ISO)

FCS. Frame check sequence.

FIC. first-in-chain.

FID. Format identification.

first-in-chain (FIC). A request unit (RU) whose request header (RH) begin chain indicator is on and whose RH end chain indicator is off. See also *RU chain.*

first speaker. In SNA, the LU-LU half-session defined at session activation as: (1) able to begin a bracket without requesting permission from the other LU-LU half-session to do so, and (2) winning contention if both half-sessions attempt to begin a bracket simultaneously. Contrast with *bidder*. See also *bracket protocol.*

flag (F) sequence. The unique sequence of eight bits (01111110) employed to delimit the opening and closing of a frame. (TC97)

flow control. (1) The procedure for controlling the data transfer rate. (TC97) (2) In SNA, the process of managing the rate at which data traffic passes between components of the network. The purpose of flow control is to optimize the rate of flow of message units with minimum congestion in the network; that is, to neither overflow the buffers at the receiver or at intermediate routing nodes, nor leave the receiver waiting for more message units.

FMD. Function management data.

format identification (FID) field. In SNA, a field in each transmission header (TH) that indicates the format of the TH; that is, the presence or absence of certain fields. TH formats differ in accordance with the types of nodes between which they pass.

The six FID types are:

- FID0, used for traffic involving non-SNA devices between adjacent subarea nodes when either or both nodes do not support explicit route and virtual route protocols.
- FID1, used for transmission between the host, local NCP, and remote NCP.
- FID2, used for traffic between a subarea node and an adjacent type 2 peripheral node.
- FID3, used for traffic between a subarea note and an adjacent type 1 peripheral node.
- FID4, used for traffic between adjacent subarea nodes when both nodes support explicit route and virtual route protocols.
- FIDF, used for certain commands (for example, for transmission group control) sent between adjacent subarea nodes when both nodes support explicit route and virtual route protocols.

formatted system services (FSS). A facility that provides certain system services as a result of receiving a field-formatted command, such as an INITIATE or TERMINATE command. Contrast with *unformatted system services (USS).*

frame. (1) In high-level data link control (HDLC), the sequence of contiguous bits bracketed by and including opening and closing flag (01111110) sequences. (2) A set of consecutive digit time slots in which the position of each digit time slot can be identified by reference to a frame alignment signal. (CCITT/ITU)

frame check sequence (FCS). (1) A field immediately preceding the closing flag sequence of a frame that contains a bit sequence checked by the receiver to detect transmission errors. (2) In SDLC, 16 bits in a frame that contain transmission-checking information.

frame-level interface. The level of the DTE/DCE interface in packet mode operation relating to the exchange of packets with local error control, where packets are contained in frames. (CCITT/ITU) See also *packet level interface.*

FSS. Formatted system services.

FTAM. File transfer access method.

function management data (FMD). In SNA, a request unit (RU) category used for end-user data exchanged between logical units (LUs) and for requests and responses exchanged between network services components of LUs, physical units (PUs), and system services control points (SSCPs).

full-duplex. Synonym for *duplex*.

G

GATE. General Access to X.25 Transport Extension.

GATE CTCP. A CTCP that is used in conjunction with the GATE function of X.25 NPSI to manage virtual circuits. In addition to managing virtual circuits, a GATE CTCP can be used to relay user data to and from subsystems such as CICS, IMS, and TSO.

gateway. The combination of machines and programs that provide address translation, name translation, and system services control point (SSCP) rerouting between independent SNA networks to allow those networks to communicate. A gateway consists of one gateway NCP and at least one gateway SSCP.

General Access to X.25 Transport Extension (GATE). A function of X.25 NPSI that allows a communication and transmission control program (CTCP) to manage virtual circuits to non-SNA DTEs by processing data, qualified data, Interrupt, Call, Clear, and Reset packets.

H

half-duplex. In data communication, pertaining to an alternate, one way at a time, independent transmission. Contrast with *duplex*.

HDLC. High-level data link control.

high-level data link control (HDLC). Control of data links by use of a specified series of bits rather than by the control characters of the ISO Standard 7-bit character set for information processing interchange. (CCITT/ITU)

host node. A node providing an application program interface (API) and a common application interface. See *boundary node, node, peripheral node, subarea node*. See also *boundary function, node type*.

I

ICA. integrated communication adapter.

I format. Information format.

I frame. Information frame.

IMS. Information Management System.

incoming call packet. A call supervision packet transmitted by a data circuit-terminating equipment (DCE) to inform a called data terminal equipment (DTE) of a call requested by another DTE. (CCITT/ITU)

information (I) format. A format used for information transfer.

information (I) frame. A frame in I format used for numbered information transfer. See also *supervisory frame, unnumbered frame*.

Information Management System (IMS). A general purpose system whose full name is Information Management System/Virtual Storage (IMS/VS). It enhances the capabilities of OS/VS for batch processing and telecommunication and allows users to access a computer-maintained data base through remote terminals.

information transfer phase. A phase in which a data circuit-terminating equipment (DCE) can accept and transmit information (I) frames and supervisory (S) frames. See also *disconnected phase*.

inhibit presentation (INP) character. A control character that causes presentation of the following characters to be stopped.

INP. Inhibit presentation character.

input/output channel. Synonymous with *data channel*.

integrated communication adapter. An integrated adapter that allows connection of one or more telecommunication lines to a processing unit.

integrated services digital network (ISDN). A digital end-to-end telecommunication network that supports multiple services including, but not limited to, voice and data.

Note: ISDNs are used in public and private network architectures.

intermediate routing function. In SNA, a path control capability in a subarea node that receives and routes path information units (PIUs) that neither originate in nor are destined for network addressable units (NAUs) in that subarea node. See also *boundary function*.

intermediate routing node. In SNA, a subarea node with an intermediate routing function. A subarea node may be a boundary node, intermediate routing node, both, or neither, depending on how it is used in a network.

International Organization for Standardization (ISO). An organization of national standards bodies from various countries established to promote development of standards to facilitate international exchange of goods and services, and develop cooperation in intellectual, scientific, technological and economic activity.

ISDN. Integrated services digital network.

ISO. International Organization for Standardization.

ITU. International Telecommunication Union.

K

keyword. (1) (TC97) A lexical unit that, in certain contexts, characterizes some language construction. (2) * One of the predefined words of an artificial language. (3) One of the significant and informative words in a title or document that describes the content of that document. (4) A name or symbol that identifies a parameter. (5) A part of a command operand that consists of a specific character string (such as DSNAME=). See also *definition statement*.

L

LAP. Link access procedure.

LAPB. Link access procedure balanced. See *link access procedures (LAP, LAPB)*.

last-in-chain (LIC). A request unit (RU) whose request header (RH) end chain indicator is on and whose RH begin chain indicator is off. See also *RU chain*.

leased line. Synonym for *nonswitched line*.

LIC. (1) Last-in-chain (2) In NCP, line interface coupler.

line speed. The number of binary digits that can be sent over a telecommunication line in one second, expressed in bits per second (bps).

line switching. Synonym for *circuit switching*.

link access procedures (LAP, LAPB). The link level elements used for data interchange between a data circuit-terminating equipment (DCE) and a data terminal equipment (DTE) operating in user classes of service 8 to 11, as specified in CCITT Recommendation X.1.

link level. (1) A part of Recommendation X.25 that defines the link protocol used to get data into and out of the network across the full-duplex link connecting the subscriber's machine to the network node. LAP and LAPB are the link access protocols recommended by the CCITT. (2) See *data link level*.

link station. (1) In SNA, the combination of hardware and software that allows a node to attach to and provide control for a link. (2) In VTAM, a named resource within a subarea node that represents another subarea node that is attached by a subarea link. In the resource hierarchy, the link station is subordinate to the subarea link.

LLC. Logical link control.

LLU. Logical link unit.

load module. A program unit that is suitable for loading into main storage for execution; it is usually the output of a linkage editor. (ISO)

logical channel. In packet mode operation, a means of two-way simultaneous transmission across a data link, comprising associated send and receive channels. A logical channel can represent the path that data travels from its origin to the network or from the network to its destination. (CCITT/ITU)

logical circuits. In packet mode operation, a means of duplex transmission across a data link comprising associated send and receive channels. A number of logical circuits can be derived from a data link by packet interleaving. Several logical circuits can exist on the same data link.

Logical unit (LU). In SNA, a port through which an end user accesses the SNA network and the functions provided by system services control points (SSCPs). An LU can support at least two sessions—one with an SSCP and one with another LU—and may be capable of supporting many sessions with other LUs. See also *peripheral LU, physical unit (PU), primary logical unit (PLU), secondary logical unit (SLU), system services control point (SSCP)*.

lower window edge. The lowest sequence number in a window. (CCITT/ITU)

LU. Logical unit.

LUSIM. LU simulator.

LU simulator (LUSIM). A function of X.25 NPSI that simulates a logical unit (LU) for a non-SNA DTE so that the application LU or CTCP LU acts as though it is in session with an SNA DTE rather than with a non-SNA DTE. The LU-LU session between the application or

CTCP LU and the simulated LU uses LU type 1 protocols.

M

maintenance and operator subsystem (MOSS). A subsystem of an IBM communication controller, such as the 3725 or the 3720, that contains a processor and operates independently of the rest of the controller. It loads and supervises the controller, runs problem determination procedures, and assists in maintaining both hardware and software.

M bit. More data bit.

MBS. A series of complete packet sequences where each packet has the M bit set to 1 except the last packet of the last complete packet sequence.

MCH. Multichannel link.

message switching. (1) In a data network, the process of routing messages by receiving, storing, and forwarding complete messages. (2) The technique of receiving a complete message, storing, and then forwarding it to its destination unaltered. (TC97)

MIC. middle-in-chain.

middle-in-chain (MIC). A request unit (RU) whose request header (RH) begin chain indicator and RH end chain indicator are both off. See also *RU chain*.

migration. Installing a new version or release of a program when an earlier version or release is already in place.

MOSS. Maintenance and operator subsystem.

multichannel link (MCH). A means of enabling a data terminal equipment (DTE) to have several access channels to the data network over a single circuit. Three likely methods have been identified: packet interleaving, byte interleaving, and bit interleaving. (CCITT/ITU)

multilink procedure. A procedure for controlling the operation of an MCH that consists of several physical links running in parallel.

Multiple Virtual Storage (MVS). An IBM licensed program whose full name is the Operating System/Virtual Storage (OS/VS) with Multiple Virtual Storage/System Product for System/370. It is a software operating system controlling the execution of programs.

MVS. Multiple Virtual Storage operating system.

N

NAS. Network Action Scheduler.

NAU. Network addressable unit.

NCP. Network Control Program.

NCP/EP definition facility (NDF). A program that is part of System Support Programs (SSP) and is used to generate a partitioned emulation program (PEP) load module or a load module for a Network Control Program (NCP) or for an Emulation Program (EP).

NDF. NCP/EP definition facility.

NDM. Normal disconnected mode.

NEO. Network expansion option.

NetView program. A System/370-based IBM licensed program used to monitor a network, manage it, and diagnose its problems.

NPM. Netview Performance Monitor.

Netview Performance Monitor (NPM). An IBM licensed program that collects, monitors, analyzes, and displays data relevant to the performance of a VTAM telecommunication network. It runs as an online VTAM application program.

network addressable unit (NAU). In SNA, a logical unit, a physical unit, or a system services control point. The NAU is the origin or the destination of information transmitted by the path control network. See also *logical unit (LU), path control (PC) network, physical unit (PU), system services control point (SSCP)*.

network control phase. That phase of a data call during which network control signals are exchanged between a DTE and the network for the purpose of call establishment, call disconnection, or for control signaling during the data phase. (ISO)

Network Control Program (NCP). An IBM licensed program that provides communication controller support for single-domain, multiple-domain, and interconnected network capability. Its full name is Advanced Communications Function for the Network Control Program.

network failure. In a network, any condition that makes a service unavailable because the network or one of its essential components is not functioning correctly.

Network Routing Facility (NRF). An IBM licensed program that resides in the NCP, which provides a path for messages between terminals, and routes messages over this path without going through the host processor.

Network Terminal Option (NTO). An IBM licensed program used in conjunction with NCP that allows certain non-SNA devices to participate in sessions with SNA application programs in the host processor. NTO converts non-SNA protocol to SNA protocol when data is sent to the host from a non-SNA device and reconverts SNA protocol to non-SNA protocol when data is sent back to the device.

NIA. IBM 5973-LO2 Network Interface Adapter.

node. (1) In a network, a point at which one or more functional units connect channels or data circuits. (ISO) (2) In SNA, an endpoint of a link or a junction common to two or more links in a network. Nodes can be distributed to host processors, communication controllers, cluster controllers, or terminals. Nodes can vary in routing and other functional capabilities. (3) In ACF/VTAM, a point in a network defined by a symbolic name.

node type. In SNA, a designation of a node according to the protocols it supports and the network addressable units (NAUs) that it can contain. Five types are defined: 1, 2.0, 2.1, 4, and 5. Type 1, type 2.0, and type 2.1 nodes are peripheral nodes; type 4 and type 5 nodes are subarea nodes.

nonqualified data packet. A data packet in which the Q-bit is set off.

Non-SNA Interconnect (NSI). An IBM licensed program that provides format identification (FID1/4) support for selected non-SNA facilities. Thus, it allows SNA and non-SNA facilities to share SDLC links. It also allows the remote concentration of selected non-SNA devices along with SNA devices.

nonswitched connection. A connection that does not have to be established by dialing. Contrast with *switched connection.*

nonswitched line. A telecommunication line on which connections do not have to be established by dialing. Contrast with *switched line*. Synonymous with *leased line*.

no response. In SNA, a value in the form-of-response-requested field of the request header (RH) indicating that no response is to be returned to the request, whether or not the request is received and processed successfully. Contrast with *definite response (DR), exception response (ER).*

normal response mode (NRM). An operational mode of an unbalanced data link in which the secondary station initiates transmission only as the result of receiving explicit permission from the primary station. See also *asynchronous balanced mode (ABM), asynchronous response mode (ARM).*

NPSI. X.25 NCP Packet Switching Interface.

NRF. Network routing facility.

NRM. normal response mode.

NSI. Non-SNA Interconnection.

NTO. Network terminal option.

O

octet. A byte composed of eight binary elements.

OIC. only-in-chain.

only-in-chain. A request unit for which the request header (RH) begin chain indicator and RH end chain indicator are both on. See also *RU chain.*

Open Systems Interconnection (OSI). (1) The interconnection of open systems in accordance with specific ISO standards. (2) The use of standardized procedures to enable the interconnection of data processing systems.

operating system (OS). Software that controls the execution of programs. An operating system may provide services such as resource allocation, scheduling, input/output control, and data management.

Note: Although operating systems are predominantly software, partial or complete hardware implementations are possible.

optional network facilities. Facilities that a user of a packet switching data network can request when establishing a virtual circuit. See also *closed user group and throughput class negotiation.*

OS. Operating system.

OSI. Open Systems Interconnection.

OSICS. Open systems interconnection communication system.

OUFT. Optional user facility table.

P

pacing. In SNA, a technique by which a receiving component controls the rate of transmission of a sending component to prevent overrun or congestion. See also *session-level pacing, virtual route (VR) pacing.*

packet. A sequence of binary digits, including data and call control signals, that is transmitted and switched as a composite whole. (ISO) The data, call control signals, and error control information are arranged in a specific format. See also *call accepted packet, call connected packet, call request packet, call supervision packet, clear indication packet, clear request packet, data packet, DCE clear confirmation packet, discarded packet, DTE clear confirmation packet, incoming call packet, nonqualified data packet, permit packet, qualified data packet, reset packet, RNR packet, RR packet*.

packet assembler/disassembler (PAD). A user facility which permits non-packet mode terminals to exchange data in the packet mode. (CCITT/ITU)

packet level. The packet format and control procedures for the exchange of packets containing control information and user data between the data terminal equipment (DTE) and the data circuit-terminating equipment (DCE). See also *data link level, physical level*.

packet level interface. The level of the DTE/DCE interface in packet mode operation relating to the exchange of data and signaling, where this information is contained in packets. (CCITT/ITU) See also *frame-level interface*.

packet level processor (PLP).. The part of X.25 NPSI that handles X.25 level 3.

packet mode operation. Synonym for *packet switching*.

packet mode terminal. Data terminal equipment that can control, format, transmit, and receive packets. (TC97)

packet sequencing. A process of ensuring that packets are delivered to the receiving data terminal equipment (DTE) in the same sequence as they were transmitted by the sending DTE. (TC97)

packet switched data network (PSDN). A network that uses packet switching as a means of transmitting data.

packet switched data transmission service. A user service involving the transmission and, if necessary, the assembly and disassembly of data in the form of packets. (CCITT/ITU)

packet switching. (1) The process of routing and transferring data by means of addressed packets so that a channel is occupied only during the transmission of a packet. On completion of the transmission, the channel is made available for transfer of other packets. (ISO) (2) Synonymous with *packet mode operation*. Contrast with *circuit switching*.

packet window. The maximum number of consecutive data packets that are allowed to flow between a data terminal equipment (DTE) and a data circuit-terminating equipment (DCE) before an acknowledgment is received for a given logical channel.

PAD. Packet assembler/disassembler.

path control (PC) network. In SNA, the part of the SNA network that includes the data link control and path control layers. See *SNA network, user-application network*. See also *boundary function*.

path information unit (PIU). In SNA, a message unit consisting of a transmission header (TH) alone, or of a TH followed by a basic information unit (BIU) or a BIU segment. See also *transmission header*.

PCNE. Protocol converter for non-SNA equipment.

PDN. Public data network.

peripheral link. In SNA, a link that connects a peripheral node to a subarea node. See also *subarea link*.

peripheral LU. In SNA, a logical unit representing a peripheral node.

peripheral node. In SNA, a node that uses local addresses for routing and therefore is not affected by changes in network addresses. A peripheral node requires boundary function assistance from an adjacent subarea node. See also *intermediate routing node, node type, peripheral link, subarea node*.

permanent virtual circuit (PVC). A virtual circuit that has a logical channel permanently assigned to it at each data terminal equipment (DTE). A call establishment protocol is not required.

permit packet. A packet used for the transmission of permits for a virtual circuit at the DTE/DCE interface. (CCITT/ITU)

PH. packet header.

physical circuit. A circuit created with hardware rather than by multiplexing. See also *data circuit*. Contrast with *virtual circuit*.

physical level. The mechanical, electrical, functional, and procedural media used to activate, maintain, and deactivate the physical link between the data terminal equipment (DTE) and the data circuit-terminating equipment (DCE). See also *data link level, packet level*.

physical unit (PU). In SNA, a type of network addressable unit (NAU). A physical unit (PU) manages and monitors the resources (such as attached links) of a node, as requested by a system services control point (SSCP) through an SSCP-PU session. An SSCP activates a session with the physical unit in order to indirectly manage, through the PU, resources of the node such as attached links.

piggybacking. Act of acknowledging a received frame or packet within the next transmittal.

PIU. Path information unit.

PLP. Packet level processor.

PLU. Primary logical unit.

port. An access point for data entry or exit.

port swap. A function of NCP/X.25 NPSI that allows you to install spare ports to be used as backup in case of failure of the original port.

Post Telephone and Telegraph Administration (PTT). A generic term for the government-operated common carriers in countries other than the USA and Canada. Examples of the PTT are the Post Office in the United Kingdom, the Bundespost in Germany, and the Nippon Telephone and Telegraph Public Corporation in Japan.

primary logical unit (PLU). In SNA, the logical unit (LU) that contains the primary half-session for a particular LU-LU session. Each session must have a PLU and secondary logical unit (SLU). The PLU is the unit responsible for the bind and is the controlling LU for the session. A particular LU can contain both primary and secondary half-sessions for different active LU-LU sessions. Contrast with *secondary logical unit (SLU)*.

primary station. (1) In high-level data link control (HDLC), the part of a data station that supports the primary control functions of the data link, generates commands for transmission, and interprets received responses. (ISO) (2) In SNA, the station on an SDLC data link that is responsible for the control of the data link. There must be only one primary station on a data link. All traffic over the data link is between a primary station and a secondary station. (3) Contrast with *secondary station*. See also *combined station*.

Note: Specific responsibilities assigned to the primary station include initialization of control signal interchange, organization of data flow, and actions regarding error control and error recovery functions at the data link level.

problem determination. The process of identifying the source of a problem; for example, a program component, a machine failure, telecommunication facilities, user or contractor-installed programs or equipment, an environment failure such as a power loss, or a user error.

program temporary fix (PTF). A temporary solution or bypass of a problem diagnosed by IBM in a current unaltered release of the program.

protocol. (1) A specification for the format and relative timing of information exchanged between communicating parties. (CCITT/ITU) (2) The set of rules governing the operation of functional units of a communication system that must be followed if communication is to be achieved. (TC97) (3) In SNA, the meanings of, and the sequencing rules for, requests and responses used for managing the network, transferring data, and synchronizing the states of network components. See also *bracket protocol*.

protocol converter for non-SNA equipment (PCNE). A function of X.25 NPSI that allows attachment of non-SNA X.25 DTEs without the use of a packet assembler/disassembler (PAD). PCNE replaces the packet headers used to receive data from non-SNA X.25 DTEs with the SNA headers used to pass the data to an application LU, and vice versa. The PCNE function uses an LU simulator.

PSDN. Packet switched data network.

PTF. Program temporary fix.

PTT. Post Telephone and Telegraph Administration.

PU. Physical unit.

public data network (PDN). See *public network*.

public network. A network established and operated by an administration for the specific purpose of providing data transmission services to the public. Circuit switched, packet switched, and leased-circuit services are feasible. Contrast with *user-application network*.

PVC. Permanent virtual circuit.

Q

Q bit. Qualified data bit.

qualified data packet. A data packet in which the Q bit is set on.

QLLC. Qualified logical link control.

R

receive leg. The side of a duplex line that is receiving. Contrast with *transmit leg*.

receive not ready packet. See *RNR packet*.

receive ready packet. See *RR packet*.

RECFMS. Record formatted maintenance statistics. See also *packet switching*.

Recommendation X.21 (Geneva 1980). A Consultative Committee on International Telegraph and Telephone

(CCITT) recommendation for a general purpose interface between data terminal equipment and data circuit-terminating equipment for synchronous operations on a public data network.

Recommendation X.25 (Geneva 1980). A Consultative Committee on International Telegraph and Telephone (CCITT) recommendation for the interface between data terminal equipment and packet-switched data networks. See also *packet switching.*

Recommendation X.28. A Consultative Committee on International Telegraph and Telephone (CCITT) recommendation for the DTE/DCE interface for a start-stop mode data terminal equipment (DTE) accessing the packet assembly/disassembly (PAD) facility in a public data network situated in the same country.

Recommendation X.29. A Consultative Committee on International Telegraph and Telephone (CCITT) recommendation for procedures for the exchange of control information and user data between a packet assembly/disassembly (PAD) facility and a packet mode data terminal equipment (DTE) or another PAD facility.

Recommendation X.3. A Consultative Committee on International Telegraph and Telephone (CCITT) recommendation for packet assembly/disassembly (PAD) in a public data network.

record formatted maintenance statistics (RECFMS). A statistical record built by an SNA controller and usually solicited by the host.

REJ. Rejected message.

request header (RH). In SNA, control information preceding a request unit (RU). See also *request/response header (RH).*

request/response header (RH). In SNA, control information, preceding a request/response unit (RU), that specifies the type of RU (request unit or response unit) and contains control information associated with that RU.

request/response unit (RU). In SNA, a generic term for a request unit or a response unit. See also *request unit (RU), response unit (RU).*

request unit (RU). In SNA, a message unit that contains control information, end-user data, or both.

reset collision. A condition that occurs when a data terminal equipment (DTE) and a data circuit-terminating equipment (DCE) simultaneously transmit a reset request packet and a reset indication packet over the same logical channel. See also *call collision, clear collision.*

reset (of a virtual circuit). Reinitializing of flow control on a virtual circuit, which eliminates all data that may be in transit for the virtual circuit at the time of resetting. (CCITT/ITU)

reset packet. A packet used for the resetting of a virtual circuit at the DTE/DCE interface. (CCITT/ITU)

response. In data communication, a reply represented in the control field of a response frame. It advises the primary/combined station with respect to the action taken by the secondary/combined station to one or more commands. (TC97)

response frame. A frame transmitted by a secondary station or a frame transmitted by a combined station that contains the address of the transmitting combined station. (TC97)

response unit (RU). In SNA, a message unit that acknowledges a request unit; it may contain prefix information received in a request unit. If positive, the response unit may contain additional information (such as session parameters in response to Bind Session), or if negative, contains sense data defining the exception condition.

reverse charging acceptance. A facility that enables a data terminal equipment (DTE) to receive incoming packets that request reverse charging.

RH. Request/response header.

RNR. Receive not ready.

RNR packet. A packet used by a data terminal equipment (DTE) or by a data circuit-terminating equipment (DCE) to indicate a temporary inability to accept additional packets for a given virtual call or permanent virtual circuit.

RPOA. Recognized private operating authority.

RR. Receive ready.

RR packet. A packet used by a data terminal equipment (DTE) or by a data circuit-terminating equipment (DCE) to indicate that it is ready to receive data packets within the window.

RU. Request/response unit.

RU chain. In SNA, a set of related request/response units (RUs) consecutively transmitted on a particular normal or expedited data flow. The request RU chain is the unit of recovery. If one RU in the chain cannot be processed, the entire chain must be discarded.

Note: Each request unit belongs to only one chain, which has a beginning and an end indicated through control bits in request/response headers within the RU chain. Each RU can be designated as first-in-chain

(FIC), last-in-chain (LIC), middle-in-chain (MIC), or only-in-chain (OIC). Response units and expedited-flow request units are always sent as only-in-chain.

S

SDLC. Synchronous Data Link Control.

SDT. Start data traffic.

secondary logical unit (SLU). In SNA, the logical unit (LU) that contains the secondary half-session for a particular LU-LU session. Contrast with *primary logical unit (PLU)*.

secondary station. (1) In high-level data link control (HDLC), the part of a data station that executes data link control functions as instructed by the primary station and that interprets received commands and generates responses for transmission. (ISO) (2) A data station that executes data link control functions as instructed by the primary station. A secondary station interprets received commands and generates responses for transmission. Contrast with *primary station*. See also *combined station*.

sequence number. A number assigned to a particular frame or packet to control the transmission flow and receipt of data.

session-level pacing. In SNA, a flow control technique that permits a receiving session to control the data transfer rate (the rate which it receives request units) on the normal flow. It is used to prevent overloading a receiver with unprocessed requests when the sender can generate requests faster than the receiver can process them. See also *pacing, virtual route (VR) pacing*.

S frame. Supervisory frame

SHM. Short hold mode.

short hold mode (SHM). A function of X.25 NPSI that allows a virtual connection to be cleared if no traffic is present on the connection for a time interval specified by the user. When traffic resumes, the connection is automatically reestablished.

shutdown. The process of ending operation of a system or a subsystem, following a defined procedure.

SLU. Secondary logical unit.

SMN. Switched major node.

SNA. Systems Network Architecture.

SNA network. The part of a user-application network that conforms to the formats and protocols of Systems Network Architecture. It enables reliable transfer of data among end users and provides protocols for controlling the resources of various network configurations. The SNA network consists of network addressable units (NAUs), boundary function components, and the path control network.

SNA network interconnect (SNI). A facility that allows users to connect an SNA network with other SNA or non-SNA networks.

SNA network interconnection. The connection, by gateways, of two or more independent SNA networks to allow communication between logical units in those networks. The individual SNA networks retain their independence.

SNI. SNA network interconnect.

SSCP. System services control point.

SSP. System Support Programs (IBM licensed program.) Its full name is Advanced Communications Function for System Support Programs. Synonymous with *ACF/SSP*.

subaddressing. The mechanism by which the X.25 NPSI logical link control (LLC) or the communication and transmission control program (CTCP) is selected by the value of the last digit of the called DTE address in the incoming call packet.

subarea. A portion of the SNA network consisting of a subarea node, any attached peripheral nodes, and their associated resources. Within a subarea node, all network addressable units, links, and adjacent link stations (in attached peripheral or subarea nodes) that are addressable within the subarea share a common subarea address and have distinct element addresses.

subarea link. In SNA, a link that connects two subarea nodes. See also *peripheral link*.

subarea node. In SNA, a node that uses network addresses for routing and whose routing tables are therefore affected by changes in the configuration of the network. Subarea nodes can provide gateway function, and boundary function support for peripheral nodes. Type 4 and type 5 nodes are subarea nodes. See *boundary node, host node, node, peripheral node*. See also *boundary function, node type*.

supervisory (S) format. A format used to perform data link supervisory control functions, such as acknowledge I frames, request retransmission of I frames, and request temporary suspension of transmission of I frames. See also *information format, unnumbered format*.

supervisory (S) frame. A frame in supervisory format used to transmit supervisory control functions.

SVC. Switched virtual circuit.

SVCSC. Switched virtual circuit subarea communication.

switchback. On an IBM 3745 with twin CCUs, the action of switching the lines currently attached to a CCU, as the result of a fallback, back to the original CCU.

switched connection. (1) A mode of operating a data link in which a circuit or channel is established to switching facilities as, for example, in a public switched network. (ISO) (2) A connection established by dialing. (3) Contrast with *nonswitched connection.*

switched line. A telecommunication line in which the connection is established by dialing. Contrast with *nonswitched line*.

switched major node. In VTAM, a major node whose minor nodes are physical units and logical units attached by switched SDLC links.

switched network. Any network in which connections are established by closing switches, for example, by dialing.

switched virtual circuit (SVC). A virtual circuit that is requested by a virtual call. It is released when the virtual circuit is cleared.

switched virtual circuit (SVC) short hold mode. See *short hold mode (SHM)*.

switched virtual circuit subarea communication (SVCSC). A function of X.25 NPSI that, together with appropriate VTAM functions, allows communication over a switched virtual circuit (SVC) between (1) two communication controllers or (2) a communication controller and certain host processors equipped with appropriate hardware and software.

Synchronous Data Link Control (SDLC). A discipline conforming to subsets of the Advanced Data Communication Control Procedures (ADCCP) of the American National Standards Institute (ANSI) and High-level Data Link Control (HDLC) of the International Organization for Standardization (ISO), for managing synchronous, code-transparent, serial-by-bit information transfer over a link connection. Transmission exchanges may be duplex or half-duplex over switched or nonswitched links. The configuration of the link connection may be point-to-point, multipoint, or loop. See also *binary synchronous communications*.

system services control point (SSCP). In SNA, the focal point within an SNA network for managing the configuration, coordinating network operator and problem determination requests, and providing directory support and other session services for end users of the network. Multiple SSCPs, cooperating as peers, can divide the network into domains of control, with each SSCP having a hierarchical control relationship to the physical units and logical units within its domain.

Systems Network Architecture (SNA). The description of the logical structure, formats, protocols, and operational sequences for transmitting information units through and controlling the configuration and operation of networks.

System Support Programs (SSP). An IBM licensed program, made up of a collection of utilities and small programs, that supports the operation of the NCP.

T

TAP. Trace analysis program. Synonymous with *ACF/TAP*.

telecommunication line. (1) The portion of a data circuit external to a data-circuit terminating equipment (DCE) that connects the DCE to a data switching exchange (DSE), that connects a DCE to one or more other DCEs, or that connects a DSE to another DSE. (TC97) (2) Any physical medium, such as a wire or microwave beam, that is used to transmit data. (3) Synonymous with *data transmission line, transmission line*. (4) Contrast with *data link*.

Note: A telecommunication line is the physical medium; for example, a telephone wire or a microwave beam. A data link includes the physical medium of transmission, the protocol, and associated devices and programs—it is both logical and physical.

TH. Transmission header.

time-out. (1) An event that occurs at the end of a predetermined period of time that began at the occurrence of another specified event. (ISO) (2) A time interval allotted for certain operations to occur; for example, response to polling or addressing before system operation is interrupted and must be restarted.

time sharing control task (TSC). In TSO, a system task that handles system initialization, allocation of time shared regions, swapping, and general control of the time sharing operation.

Time Sharing Option (TSO). An optional configuration of the operating system that provides conversational time sharing from remote stations.

trace analysis program (TAP). An SSP program service aid that assists in analyzing trace data produced by VTAM, TCAM, and NCP and provides network data traffic and network error reports.

transmission header (TH). In SNA, control information, optionally followed by a basic information unit (BIU) or a BIU segment, that is created and used by path control to route message units and to control their flow within the network. See also *path information unit (PIU).*

transmission line. Synonym for *telecommunication line*.

transmission subsystem (TSS). The part of the controller that controls the data transfers over low- and medium-speed, switched and nonswitched transmission interfaces.

The TSS consists of:

- Up to 32 low-speed scanners (LSSs) associated with
- LIC units (LIUs), through
- Serial links (SLs).

transmission subsystem component (TSC). The component of VTAM that comprises the transmission control, path control, and data link control layers of SNA.

transmit leg. The side of a duplex line that is transmitting. Contrast with *receive leg*.

TSC. Transmission subsystem component.

TSO. Time Sharing Option.

type 2.1 node (T2.1 node). A node that can attach to an SNA network as a peripheral node using the same protocols as type 2.1 nodes. Type 2.1 nodes can be directly attached to one another using peer to peer protocols. See *end node, node*, and *subarea node*. See also *node type*.

U

U frame. Unnumbered frame.

unbalanced data link. A data link between a primary station and one or more participating secondary stations. The primary station assumes responsibility for the organization of data flow and for data link level error recovery operations and transmits command frames to the secondary stations. The secondary stations transmit response frames. Contrast with *balanced data link*. (TC97)

UNBIND. In SNA, a request to deactivate a session between two logical units (LUs). See also *session deactivation request*. Contrast with *BIND*.

unformatted system services (USS). In SNA products, a system services control point (SSCP) facility that translates a character-coded request, such as a logon or logoff request into a field-formatted request for processing by formatted system services and translates field-formatted replies and responses into character-coded requests for processing by a logical unit. Contrast with *formatted system services*.

unnumbered (U) format. A format used to provide additional data link control functions and unnumbered information transfer. See also *information format, supervisory format*.

unnumbered (U) frame. A frame in unnumbered format, used to transfer unnumbered control functions. See also *information frame, supervisory frame*.

USA. Upstream address.

user-application network. A configuration of data processing products, such as processors, controllers, and terminals, established and operated by users for the purpose of data processing or information exchange, which may use services offered by communication common carriers or telecommunication administrations. Contrast with *public network*.

user class of service. A category of data transmission service provided by a data network in which the data signaling rate, the data terminal equipment operating mode, and the code structure, if any, are standardized. (TC97) See also *class of service (COS)*.

USS. Unformatted system services.

V

VC. Virtual circuit.

VCCPT. Virtual circuit control parameter table.

VCM. Virtual circuit manager.

virtual call. See *virtual call facility*.

virtual call facility. A user facility in which a call setup procedure and a call clearing procedure will determine a period of communication between two data terminal equipments (DTEs) in which user's data will be transferred in the network in the packet mode of operation. All the user's data is delivered from the network in the same order in which it is received by the network. (CCITT/ITU)

virtual circuit. In packet switching, those facilities provided by a network that give the appearance to the user of an actual connection. (TC97) See also *data circuit*. Contrast with *physical circuit*.

virtual circuit LU. An LU that controls the flow of data over a virtual circuit between X.25 NPSI and a remote DTE. If the DTE is an SNA DTE, the virtual circuit LU is in that DTE. If the DTE is a non-SNA DTE, the virtual circuit LU is in the communication controller that runs

X.25 NPSI; it is a simulated LU. See also *LU simulator (LUSIM)*.

Virtual Machine/System Product (VM/SP). An IBM-licensed program that manages the resources of a single computer so that multiple computing systems appear to exist. Each virtual machine is the functional equivalent of a "real" machine.

virtual route (VR) pacing. In SNA, a flow control technique used by the virtual route control component of path control at each end of a virtual route to control the rate at which path information units (PIUs) flow over the virtual route. VR pacing can be adjusted according to traffic congestion in any of the nodes along the route. See also *pacing, session-level pacing*.

Virtual Telecommunications Access Method (VTAM). An IBM licensed program that controls communication and the flow of data in an SNA network. It provides single-domain, multiple-domain, and interconnected network capability.

VM. Virtual machine.

VSE. Virtual Storage Extended operating system. Synonymous with *VSE/AF*.

W

window. An ordered set of consecutive packet send sequence numbers of the data packets authorized to cross a DTE/DCE interface on a logical channel used for a virtual call or as a permanent virtual circuit.

window edge. The lowest sequence number in a window.

window size. The specified number of frames of information that can be sent before receiving an acknowledgment response.

X

XI. X.25 SNA Interconnection.

Bibliography

X.25 NCP Packet Switching Interface Publications

The following paragraphs describe the library of books associated with *X.25 NCP Packet Switching Interface* Version 3.

X.25 NCP Packet Switching Interface General Information Version 3 (GC30-3469)

This book introduces managers, system designers, programmers, and other data processing personnel to the basic concepts of packet-switching, X.25, and IBM's X.25 NCP Packet Switching Interface licensed program.

X.25 NCP Packet Switching Interface Planning and Installation Version 3 (SC30-3470)

This book provides system programmers and analysts with the information required to plan and implement the installation of NPSI. The topics include hardware/software requirements, preinstallation system performance considerations, instructions for defining and generating NPSI, and installation examples.

X.25 NCP Packet Switching Interface Host Programming Version 3 (SC30-3502)

This book is written for application and system programmers to assist them in writing application programs that use the X.25 NCP Packet Switching Interface. Application programmers should have some knowledge of DATE and GATE functions and the operating systems that support them. System programmers should be knowledgeable in SNA architecture.

X.25 NCP Packet Switching Interface Diagnosis, Customization, and Tuning Version 3 (LY30-5610)

This book is written for system programmers to assist them in trouble-shooting and diagnosing problems with the X.25 NCP Packet Switching Interface. It helps programmers to diagnose problems, resolve common errors, and describe problems to and interface with the IBM Support Center.

NCP and Related Products Directory of Programming Interfaces for Customers (GC31-6202)

This book provides a directory to other documents, or sections of documents, that contain the detailed descriptions of programming interfaces. It specifies files or data sets created by NCP and related products and indicates which macros are intended to be used as, or as part of, a programming interface.

Other Network Program Products Publications

For more information about the books listed in this section, see *Network Program Products Bibliography and Master Index for NetView, NCP, and VTAM.*

Network Program Products Bibliography and Master Index for NetView, NCP, and VTAM. (GC31-6815)

Network Program Products Planning and Reference for NetView, NCP, and VTAM (SC31-6811)

Network Program Products Planning (SC30-3351)

Network Program Products Samples (SC30-3352)

Network Program Products Storage Estimates (SC30-3403)

VTAM Publications

The following list shows the publications for VTAM V3R2.

VTAM Installation and Resource Definition (SC23-0111)

VTAM Customization (LY30-5614)

VTAM Operation (SC23-0113)

VTAM Messages and Codes (SC23-0114)

VTAM Programming (SC23-0115)

VTAM Programming for LU 6.2 (SC30-3400)

VTAM Diagnosis (LY30-5601)

VTAM Data Areas for MVS (LY30-5592)

VTAM Data Areas for VM (LY30-5593)

VTAM Data Areas for VSE (LY30-5594)

VTAM Reference Summary (LY30-5600)

VTAM Directory of Programming Interfaces for Customers (GC31-6403)

SNA Publications

The following publications contain information on SNA:

Systems Network Architecture Concepts and Products (GC30-3072)

Systems Network Architecture Technical Overview (GC30-3073)

Systems Network Architecture Format and Protocol Reference Manual: Management Services (SC30-3346)

Systems Network Architecture Formats (GA27-3136)

NCP Publications

The following publications apply to the libraries of NCP, SSP, and EP.

NCP and EP Reference Summary and Data Areas (LY30-5603)

SSP Customization Guide (LY43-0021)

NCP, SSP, and EP Diagnosis Guide (LY30-5591)

NCP, SSP, and EP Generation and Loading Guide (SC30-3348)

NCP, SSP, and EP Messages and Codes (SC30-3169)

NCP, SSP, and EP Resource Definition Guide (SC30-3447)

NCP, SSP, and EP Resource Definition Reference (SC30-3448)

NCP Migration Guide (SC30-3440)

NCP and EP Reference (LY30-5605)

NCP Customization Guide (LY30-5606)

NCP Customization Reference (LY30-5607)

3174 Publications

The following list shows a selected publication for the IBM 3174.

3174 Subsystem Control Unit; Customizing Guide (GA23-0214)

3745 Publications

The following list shows the publications for the IBM 3745.

IBM 3745 Communication Controller Introduction (GA33-0092)

IBM 3745 Communication Controller Configuration Program (GA33-0093)

IBM 3745 Principles of Operation (SA33-0102)

372x Publications

The following list shows selected publications for the IBM 3720.

3720/3721 Communication Controllers Introduction (GA33-0060)

3720/3721 Communication Controllers Configuration Guide (GA33-0063)

3720/3725 Communication Controllers Principles of Operation (GA33-0013)

Index

H

I

L

M

N

P

Q

R

S

T

U

Reader's Comments

X.25 Network Control Program
Packet Switching Interface
Host Programming
Version 3

Publication No. SC30-3502-2

Use this form to tell us what you think about this manual. If you have found errors in it, or if you want to express your opinion about it (such as organization, subject matter, appearance) or make suggestions for improvement, this is the form to use.

To request additional publications, or to ask questions or make comments about the functions of IBM products or systems, you should talk to your IBM representative or to your IBM authorized remarketer. This form is provided for comments about the information and the way it is presented.

When you send information to IBM, you grant IBM a nonexclusive right to use or distribute the information in any way it believes appropriate without incurring any obligation to you.

Be sure to print your name and address below if you would like a reply.

Name

Address

Company or Organization

Phone No.

IBM ®

Cut or
Along

Fold and Tape | **Please do not staple** | Fold and Tape

NO POSTAGE
NECESSARY
IF MAILED IN THE
UNITED STATES

BUSINESS REPLY MAIL

FIRST CLASS MAIL PERMIT NO. 40 ARMONK, NEW YORK

POSTAGE WILL BE PAID BY ADDRESSEE

International Business Machines Corporation
Information Development
Department E15
PO BOX 12195
RESEARCH TRIANGLE PARK NORTH CAROLINA 27709-9990

Fold and Tape | **Please do not staple** | Fold and Tape

Cut or
Along

IBM

Publication Number
SC30-3502-2

File Number
S370/4300/30xx-50

Program Number
5688-035

Printed in USA

SC30-3502-2