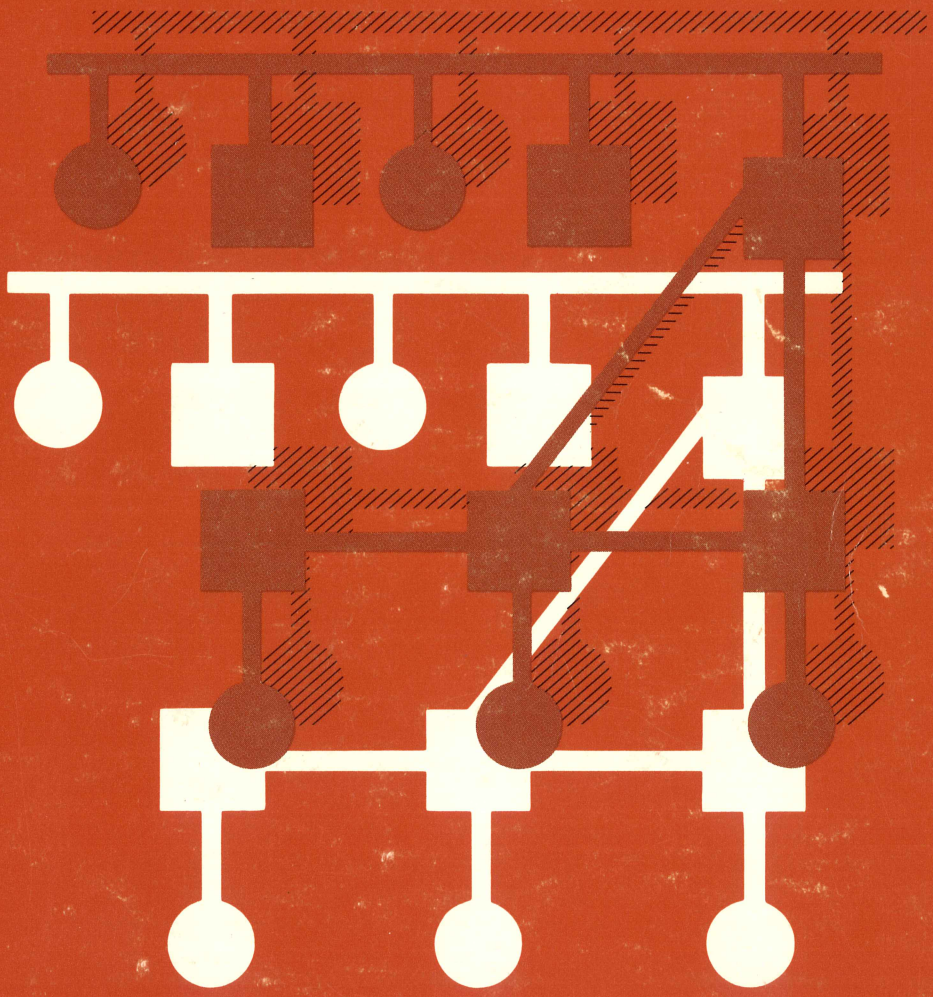
DECnet
DIGITAL Network Architecture (Phase IV)
General Description

Order No. AA-N149A-TC



digital

architecture

John Mann

DECnet
DIGITAL Network Architecture (Phase IV)
General Description

Order No. AA-N149A-TC

May 1982

This document describes the design of the DIGITAL Network Architecture that serves as a model for Phase IV DECnet implementations. It includes descriptions of functions, protocol messages, and operation.

To order additional copies of this document, contact your local Digital Equipment Corporation Sales Office.

digital equipment corporation · maynard, massachusetts

First Printing, May 1982

The postage-prepaid READER'S COMMENTS form on the last page of this document requests the user's critical evaluation to assist us in preparing future documentation.
The following are trademarks of Digital Equipment Corporation:

DIGITAL	DECsystem-10	MASSBUS
DEC	DECtape	OMNIBUS
PDP	DIBOL	OS/8
DECUS	EDUSYSTEM	PHA
UNIBUS	FLIP CHIP	RSTS
COMPUTER LABS	FLOCAL	RSX
COMTEX	INDAC	TYPESET-8
DDT	LAB-8	TYPESET-11
DECCOMM	DECSYSTEM-20	TMS-11
ASSIST-11	RTS-8	ITPS-10
VAX	VMS	SBI
DECnet	IAS	PDT
DATATRIEVE	TRAX	

Contents

Chapter 5 The Session Control Layer

Chapter 6 The Network Application Layer

Chapter 7 Network Management

Glossary

Figures

Tables

Preface

This document is an overview of the DIGITAL Network Architecture (DNA). DNA is a model of structure and function upon which DECnet implementations are based. DECnet is a family of communications software and hardware products that enable DIGITAL operating systems and computers to function in a DECnet network.

A DECnet network is a group of DIGITAL computer systems with associated operating systems, DECnet software, and communication hardware that are connected to each other by physical channels or *lines*. Each computer in a network containing a DECnet implementation is called a *node*. The *network* therefore consists of connected nodes and lines. DNA defines standard protocols, interfaces, and functions that enable DECnet network nodes to share data and access each others' resources, programs, and functions.

This document describes Phase IV DNA. Phase IV DECnet implementations will include DECnet-VAX, DECnet-11M, DECnet-11M-PLUS and DECnet-11S, Over time, Phase IV support will be provided for DECsystem-20 and the Professional 3XX Series. The previous version of DECnet, Phase III, is compatible with Phase IV. However, Phase III nodes can provide Phase III functions only.

DNA supports a broad range of applications and a variety of network topologies. (A network *topology* is a particular configuration of nodes and lines.) User documentation and marketing brochures describe in detail various types of applications and specific programming and network management information.

This document summarizes the design and structure of DNA and serves as an introduction to the DNA functional specifications. It is intended for readers with a knowledge of communications technology who desire an understanding of the overall DNA structure. A glossary at the end of the document defines many DNA terms. Additionally, many DNA terms are italicized and explained in the text at their first occurrence.

The DNA functional specifications, containing the architectural details of DNA, include the following:

DNA Data Access Protocol (DAP) Functional Specification, Version 5.6.0, Order No. AA-K177A-TK

DNA Digital Data Communications Message Protocol (DDCMP) Functional Specification, Version 4.1.0, Order No. AA-K175A-TK

DNA Session Control Functional Specification, Version 1.0.0, Order No. AA-K182A-TK

Functional specifications for the following layers of DNA Phase IV are not currently available, but will be available coincident with the the availability of DECnet-VAX Phase IV support:

DNA Routing Functional Specification, Phase IV

DNA Maintenance Operations Functional Specification, Phase IV

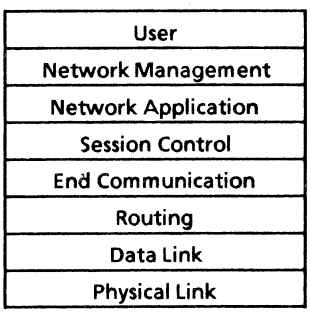
DNA Network Management Functional Specification, Phase IV

DNA End Communications Functional Specification, Phase IV

The following specifications describe functions supported by DNA Phase IV:

The Ethernet, A Local Area Network, Data Link Layer and Physical Layer Specification, Version 1.0, Order number AA-759A-TK

CCITT Recommendation X.25, Interface between Data Terminal Equipment and Data Circuit Terminating Equipment for Terminals Operating in the Packet Mode on Public Data Networks



Chapter 1
Introduction

A network architecture specifies common communication mechanisms and user interfaces that computer systems of different types must adhere to when passing data between systems. A network architecture may also be designed so that implementations meet other goals, such as cost effectivness and flexibility.

A network architecture is necessary for several reasons:

- Communications technology is changing continually. It is necessary to have a structure for networking that can adjust to these changes as effectively as possible.
- A variety of operating systems, communications devices, and computer hardware exists. A common standard of networking to which each operating system or communication hardware facility can adhere is necessary.
- A network, to be most useful and cost effective, should provide a broad range of user applications and functions. This requires very complex software. A network architecture forces such software to have a clean, well-structured design.
- Network management, error recording, and maintenance are easier when implementations provide standard procedures for error detection, recording, isolation, recovery, and repair. An architecture provides this standardization.
- Networks need to be adaptable to different communication situations. A hierarchical, modular architecture enables the substitution of modules of equivalent function to suit the specific needs of a particular network. For example, a data link protocol that ensures data integrity over a leased physical circuit can be replaced by a protocol that allows two systems to communicate over a public data network.
- A network architecture enables system-independent functions to be designed specifically enough to be implemented in hardware. This can improve system performance. For example, the DEUNA communications device implements the Ethernet data link, the DMR11, DMP11 and DMV11 implement the DDCMP data link, and the KMS11-BD implements the X.25 frame level. Basically, DIGITAL Network Architecture is the specification of a layered hierarchy in

which each layer contains modules that perform defined functions. The architecture specifies the functions of these modules and relationships between them.

Modules in a layer typically use the services of modules in the layer immediately below. Some layers may contain more than one type of module, but in this case the modules fall into a more general category of function. For example, user-written programs all operate in the top layer, or User layer. The bottom layer is the most basic layer. It manages the physical transmission of information over a channel.

Each module specified by DNA operates as a *black box*. That is, the operation within the black box is transparent to other layers and to equivalent modules in other nodes. The architecture does not define specific code for implementing modules. It only defines specific operations that the modules must perform. These definitions may take the form of algorithms written in a higher level programming language.

The architecture specifies two kinds of relationships between modules:

- **Interfaces.** Interfaces are the relationships between different modules that are usually in the same node. Typically, a module in one layer interfaces with a module in the layer immediately below to receive a service and with a module in the layer immediately above to provide a service. The architecture specifies the functions supplied by these interfaces as calls to subroutines. These specifications are functional: they need not be implemented as subroutine calls.
- **Protocols.** Protocols are the relationships between equivalent modules that are usually in different nodes. These protocols consists of messages with specific formats and rules for exchanging the messages. Such protocols are called *peer protocols*.

A state table is often specified to show the transitions occurring in a black box when protocol message exchanges or interface calls from higher layers take place.

Figure 1-1 shows the basic architectural structure of DNA.

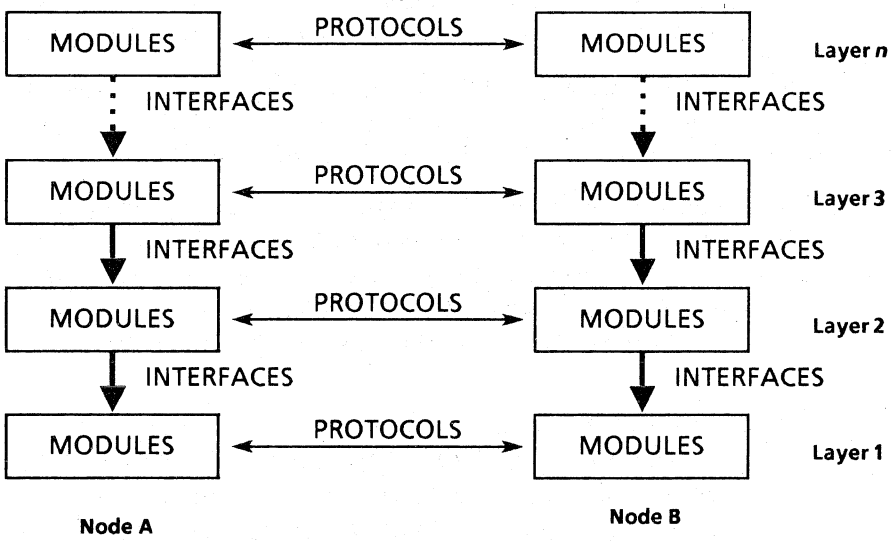


Figure 1-1: Basic DNA Structure

The architecture specifies common error reporting, operational *parameters*, and *counters* that certain layers must maintain. This standardization ensures that maintenance, error logging, and network management can take place consistently and, to some extent, across systems. For example, an operator could manage an unattended remote system from his local node. He could expect to observe or, in some cases, change values of parameters equivalent in function and name to those at his own node, even if the remote node was running under a different operating system.

DNA is an open-ended architecture. Future phases of DNA may model additional layers, additional modules within existing layers, or alternative models for certain layers.

1.1 Design Goals

DNA achieves the following design goals:

Create a common user interface. The application interface to the network is common across the varied implementations. The common interface hides the internal characteristics and topology of the network.

Support a broad spectrum of applications. DECnet networks can:

- Share resources.
- Distribute computation.
- Communicate efficiently with remote systems.

Resource-sharing activities include remote file access, inter-computer file transfer, distributed data base queries, task-to-task communication, and remote use of peripheral devices. Data moved across the network can be streams of related sequential data and short independent messages.

Distributed computation means cooperating programs executing in different computers in a network. Examples are real-time process control (such as a manufacturing system), specialized data base multiprocessor systems (such as order-processing or payroll systems), and distributed processing systems (such as a banking, airline reservation, or combined order-processing/inventory control system).

Remote communication facilities include remote public data network interfaces, Ethernet interactive terminals, and batch entry/exit stations.

Support a wide range of communication facilities. Such facilities include a variety of asynchronous and synchronous, full- and half-duplex communications devices, some with multipoint (multidrop) and/or multiple controller/multiplex capabilities. (Consult the glossary for definitions.) DECnet networks support a variety of communication channels, such as leased lines, satellite links, Ethernet local area networks, X.25-based packet switching networks, and local links.

Be cost effective. A DECnet network application costs about the same as a custom-designed network that achieves the same performance for the same application.

Support a wide range of topologies. DECnet networks support many

configurations of nodes and lines. These range from point-to-point, star, or bus topologies to hierarchical structures to topologies in which each node has equal control over network operation.

Be highly available. DECnet networks can be configured to maintain operation even if a subset of lines or nodes fail. Because maintenance functions are highly distributed, DECnet networks can recover from operator error unless the error is extreme.

Be extensible. DNA allows for incorporation of future technology changes in hardware and/or software. Moreover, DNA makes possible the movement of functions from software to hardware. DNA also permits subsets of modules.

Be highly distributed. The major functions of DNA are not centralized in one node in a network.

Implement network control and maintenance functions at a user level. This permits easier system management. For example, terminal commands can set, change, or display lower level parameters or counters.

Allow for security. The DNA design allows for security at several levels. User access control and the authentication of nodes is implemented in most DECnet products.

1.2 Layers, Modules, and Interfaces

DNA defines the following layers, described in order from the highest to the lowest:

User layer. The User layer contains most user-supplied functions. It also contains the Network Control Program, a Network Management module that gives system managers interactive terminal access to lower layers. Chapter 7 describes the function of the Network Control Program, the only DNA-defined User layer module.

Network Management layer. Modules in the Network Management layer provide user control of and access to operational parameters and counters in lower layers. Network Management also performs down-line loading, up-line dumping, remote system control, and test functions. In addition, Network Management performs event logging functions. This layer is the only one that has direct access to each lower layer. Chapter 7 describes Network Management.

Network Application layer. The Network Application layer provides generic services to the User layer. Services include remote file access, remote file transfer, remote interactive terminal access, gateway access to non-DNA systems, and resource managing programs. This layer contains both user- and DIGITAL-supplied modules. Modules execute simultaneously and independently in this layer. The Network Application layer contains functions of both the Application and Presentation layers of the ISO Reference Model. Chapter 6 describes the Network Application layer.

Session Control layer. The Session Control layer defines the system-dependent aspects of logical link communication. A *logical link* is a virtual circuit (as opposed to a physical one) on which information flows in two directions. Session Control functions include name to address translation, process addressing, and, in some

systems, process activation and access control. The Session Control layer corresponds to the Session layer of the ISO Reference Model. Chapter 5 describes the Session Control layer.

End Communication layer (previously called the Network Services layer)**.** The End Communication layer is responsible for the system-independent aspects of creating and managing logical links for network users. End Communication modules perform data flow control, end-to-end error control, and the segmentation and reassembly of user messages. The End Communication layer corresponds to the Transport layer of the ISO Referennce Model. Chapter 4 describes the End Communication layer.

Routing layer (previously called the Transport layer)**.** Modules in the Routing layer route user data, called a *datagram* and contained in a *packet*, to its destination. Routing modules also provide congestion control and packet lifetime control. The Routing layer corresponds to the Network layer of the ISO Reference Model. Chapter 3 describes the Routing layer.

Data Link layer. Modules in the Data Link layer create a communication path between adjacent nodes. Data Link modules ensure the integrity of data transferred across the path. Data link modules for Ethernet local area networks, X.25 public data networks, and synchronous or asynchronous lines execute simultaneously and independently in this layer. Chapter 2 describes the Data Link layer.

Physical Link layer. Modules in the Physcial Link layer manage the physical transmission of data over a channel. The functions of modules in this layer include monitoring channel signals, clocking on the channel, handling interrupts from the hardware, and informing the Data Link layer when a transmission is completed. Implementations of this layer encompass parts of device drivers for each communication device as well as the communication hardware itself. The hardware includes devices, modems, and lines. In this layer, industry standard electrical signal specifications such as *EIA RS-232C*, or *CCITT V.24*, the Ethernet physical layer, or CCITT X.25 level 1 operate rather than peer protocols. There is no chapter devoted to this layer since only its Network Management interface, counters, and events are DNA defined.

Figure 1-2 shows the relationships among the DNA layers in a single node. The three upper layers each interface directly with Session Control for logical link services. Each layer interfaces with the layer directly below to use its services. The User layer interfaces directly with the Network Application layer as well. In addition, Network Management modules interface directly with each lower layer for access and control purposes. Finally, Network Management interfaces directly with the Data Link layer for service functions that do not require logical links.

Figure 1-3 shows a typical DECnet node containing multiple modules in some layers.

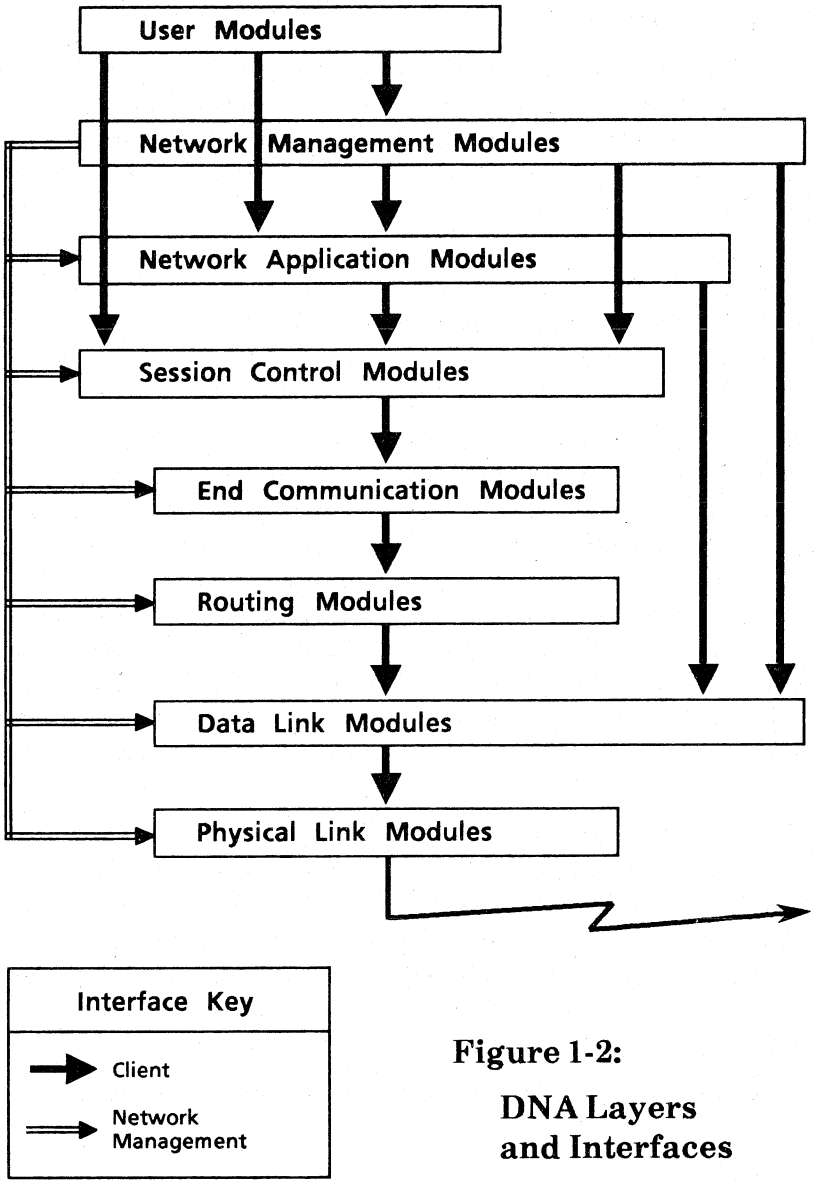


Figure 1-2:
DNA Layers and Interfaces

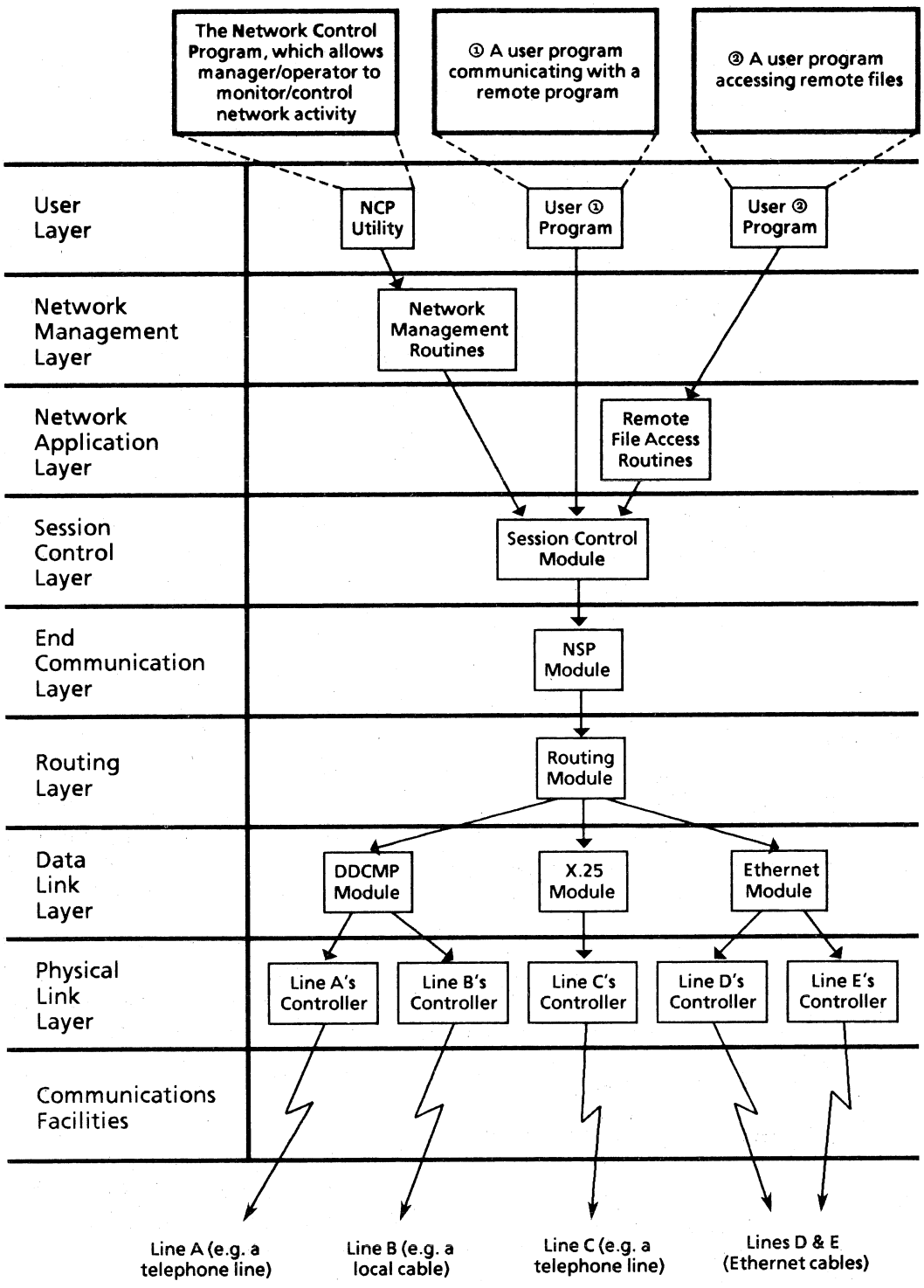


Figure 1-3: DNA Modules Resident in a Typical DECnet Node

1.3 User Services

As shown in Figures 1-2 and 1-3, there are different ways in which modules use the services of other modules in a system. DNA allows the following services:

- **User-to-user.** A user level process in one node communicates via a logical link with a user level process in another node. For example, a FORTRAN call from the user level might, using a direct interface to the Session Control module that communicates with a remote node, send data to a user level process in another node.
- **Remote application.** A user level process uses a Network Application module which, in turn, uses a logical link to perform a function at a remote node. For example, a user level command causes a Network Application module to transfer a file to a remote node. The Network Application module uses a protocol called DAP to communicate with the remote Network Application module.
- **Network Management.** A user level command or process uses a Network Management module to perform a Network Management service at a remote node. Network Management performs some services using a logical link provided by Session Control and other services by interfacing directly to the Data Link layer. An example of a Network Management operation is a command that displays a remote node's counters at a local terminal.

1.4 Protocols

Modules with equivalent functions in the same layer, but in different nodes, communicate using protocols. A protocol is both a set of messages and the rules for exchanging the messages. The architecture defines the message formats and rules very specifically.

Protocols are necessary for every DNA function that requires communication between nodes. For example, DDCMP, which resides in the Data Link layer, assigns numbers to transmitted messages and checks the numbers of received messages. In this way, the protocol ensures that transmitted data is received in the correct order.

The DNA-defined protocols, listed according to layer, are:

Network Management Layer

- **Network Information and Control Exchange (NICE) Protocol.** This is used for triggering down-line loading, up-line dumping, testing, reading parameters and counters, setting parameters, and zeroing counters.
- **Event Logger Protocol.** This is used for recording significant occurrences in lower layers. An event could result from a line coming up, a counter reaching a threshold, a node becoming unreachable, and so on.
- **Maintenance Operation Protocol.** This protocol performs data link level loopback tests, remote control of unattended systems, and down-line loading/up-

line dumping of computer systems without mass storage.

Network Application Layer

- **The Data Access Protocol (DAP).** This is used for remote file access and transfer.
- **The Network Virtual Terminal Protocols.** This family of protocols is used for terminal access through the network.
- **X.25 Gateway Access Protocol.** This protocol allows a node which is not connected directly to a public data network to access the facilities of that network through an intermediary gateway node.
- **SNA Gateway Access Protocol.** This protocol allows a node which is not connected directly to an IBM SNA network access to the facilities of the SNA network for terminal access and remote job entry.
- **The Loopback Mirror Protocol.** This is used for Network Management logical link loopback tests.

Session Control Layer

- **Session Control Protocol.** This is used for functions such as sending and receiving logical link data, and disconnecting and aborting logical links.

End Communication Layer

- **Network Services Protocol (NSP).** This handles all the system-independent aspects of managing logical links.

Routing Layer

- **Routing Protocol.** This handles routing and congestion control.

Data Link Layer

- **Digital Data Communications Message Protocol (DDCMP).** This ensures integrity and correct sequencing of messages between adjacent nodes.
- **X.25 Protocol.** This implements the X.25 packet level (level 3) and X.25 frame level (level 2) of the CCITT X.25 recommendation for public data network interfaces.
- **Ethernet Protocol.** This implements the Ethernet data link protocol for communication between adjacent nodes connected by an Ethernet local area network.

Figure 1-4 shows protocol communication between two nodes. Some implementations may use the Network Services Protocol (NSP) that creates and manages logical links for network communication within a single node, as well as for inter-node communication. For example, Network Management modules may use a logical link even when performing local functions.

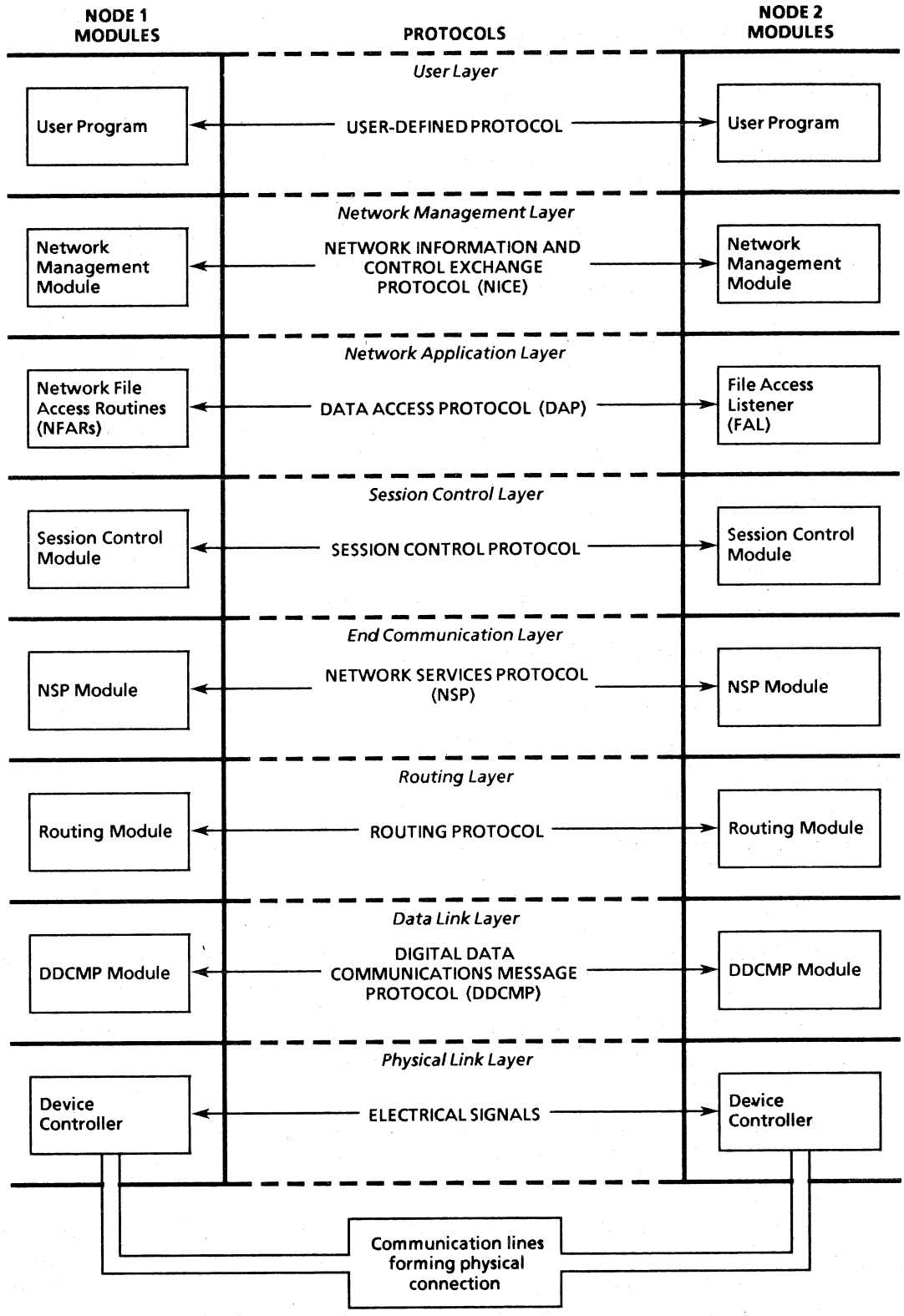


Figure 1-4: Protocol Communication between Two Nodes

1.5 Data Flow

The primary purpose of a network is to pass data from a *source* in one node to a *destination* in another. Because DNA is layered, it is important to understand how data flows through these layers and between nodes. Data traveling from one node in a network to another passes from a source process in the user layer down through each layer of the DNA hierarchy of the source node before being transmitted across a line. If the destination node is not adjacent, the data must then travel up to the Routing layer of the adjacent node, where it is *routed* (or *switched*), sent back down through the two lower layers, and transmitted across the next line in the path. The data keeps travelling in this manner until it reaches its destination node. At this node, the data passes up the hierarchy of layers to the destination process.

Figure 1-5 shows how information is built as data passes through the DNA layers at one node. In this example, Network Management is not involved.

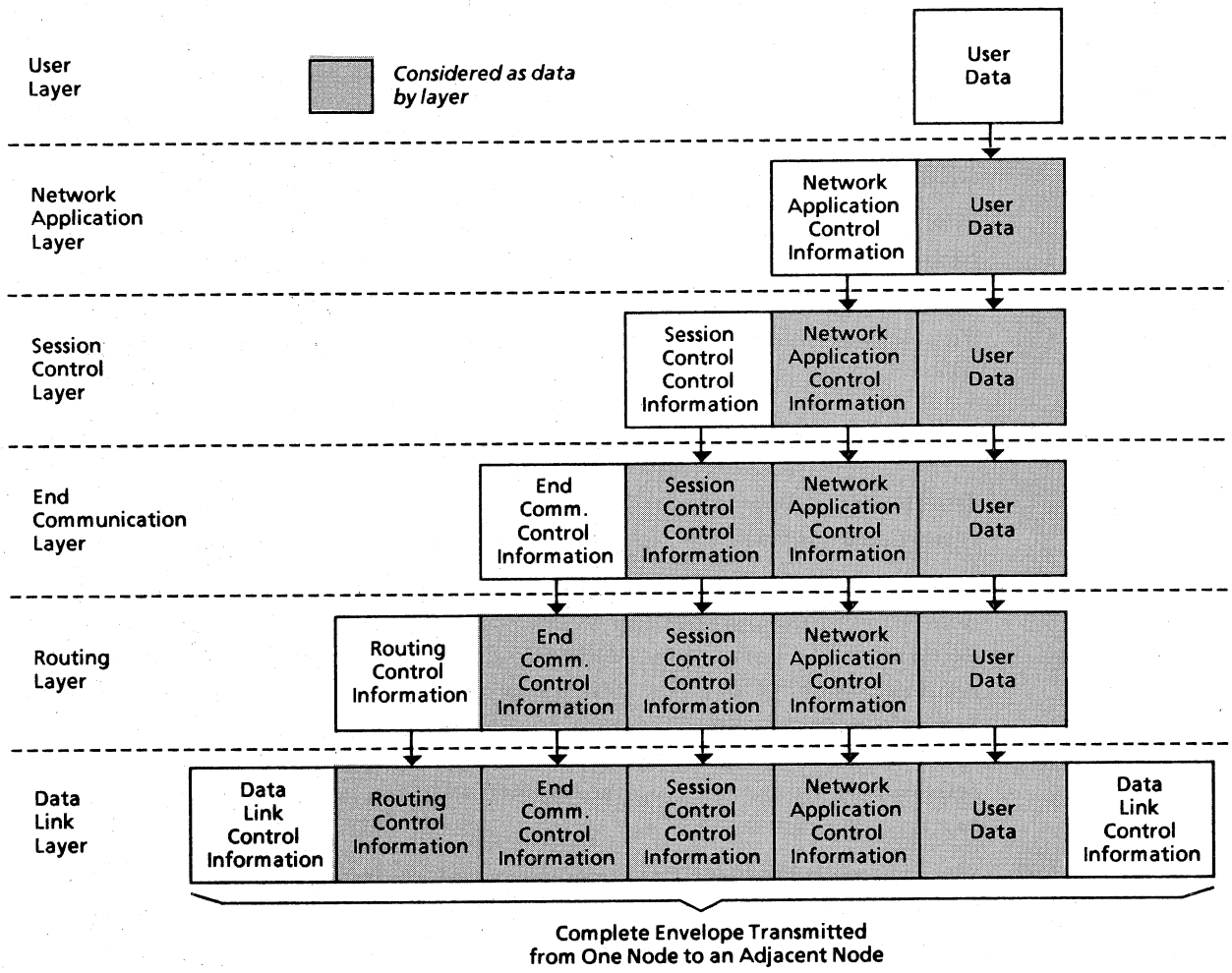


Figure 1-5: Information Building as Data Traverses DNA Layers

In the following scenario a user attempts to form a logical link with another user. The requesting user passes initial connection data to the destination user. The numbered steps correspond with numbers on Figure 1-6 which follows this explanation.

Data Flow at the Source Node

① The source user requests a connection to the destination user and passes connect data.

② The Session Control module receives the data, maps the destination node name to a numerical address, if necessary, and places the data in the next transmit buffer, adding control information to the message. The message then passes to the End Communication layer.

③ The End Communication module adds its control information (including a logical link identification) and passes the message (now called a datagram) to the Routing layer.

④ The Routing module adds a header consisting of the destination and source node addresses and selects an outgoing circuit for the message based on routing information. Routing then passes the message (now called a packet) to the Data Link layer, specifying the outgoing circuit address and, if necessary, the station address of the receiving node on the circuit.

⑤ The Data Link module adds its protocol header consisting of framing, synchronizing, addressing and control information, and its protocol trailer consisting of a cyclic redundancy check (CRC). The packet is now enveloped for transmission.

⑥ The Physical Link module transmits the enveloped message over the physical line.

Data Flow Across the Network to the Destination Node

⑦ The enveloped data message arrives at the next node. The Physical Link module receives the message and passes it to the Data Link layer.

⑧ The Data Link module checks the packet for bit errors in transmission. On links with a significant probability of transmission errors, the Data Link protocol performs error correction. DDCMP and X.25 provide error correction by retransmission. On links with a low probability of transmission error, for example Ethernet links, packets received in error are discarded, with error recovery provided by higher layers. The Data Link header and trailer are removed from a correctly received packet, which is then passed up to the Routing layer.

⑨ The Routing layer checks the destination address in the header. If the address is not this node, Routing selects the next outgoing circuit from its routing table, and passes the message back to the Data Link layer. The Routing layer has routed the message on to the next line in its path. The message proceeds as in steps 5 and 6 above.

⑩ The message proceeds through the network, switching at routing nodes, until it arrives at the Routing layer with the same address as the destination address in the message.

Data Flow at the Destination Node

⑪ The packet passes to the Routing layer of the destination node as described in 7 and 8 above. The destination Routing module removes the Routing header, and passes the datagram to End Communication.

⑫ The End Communication layer module examines the End Communication layer header on the datagram. If it has the resources to form a new logical link, it passes the connect data, without the End Communication layer control information, to Session Control.

⑬ The Session Control module performs any necessary access control functions and passes the message to the appropriate process in the user layer after removing Session Control header information.

⑭ The destination process interprets the data according to whatever higher level protocol is being used.

Figure 1-6 illustrates the data flow just described.

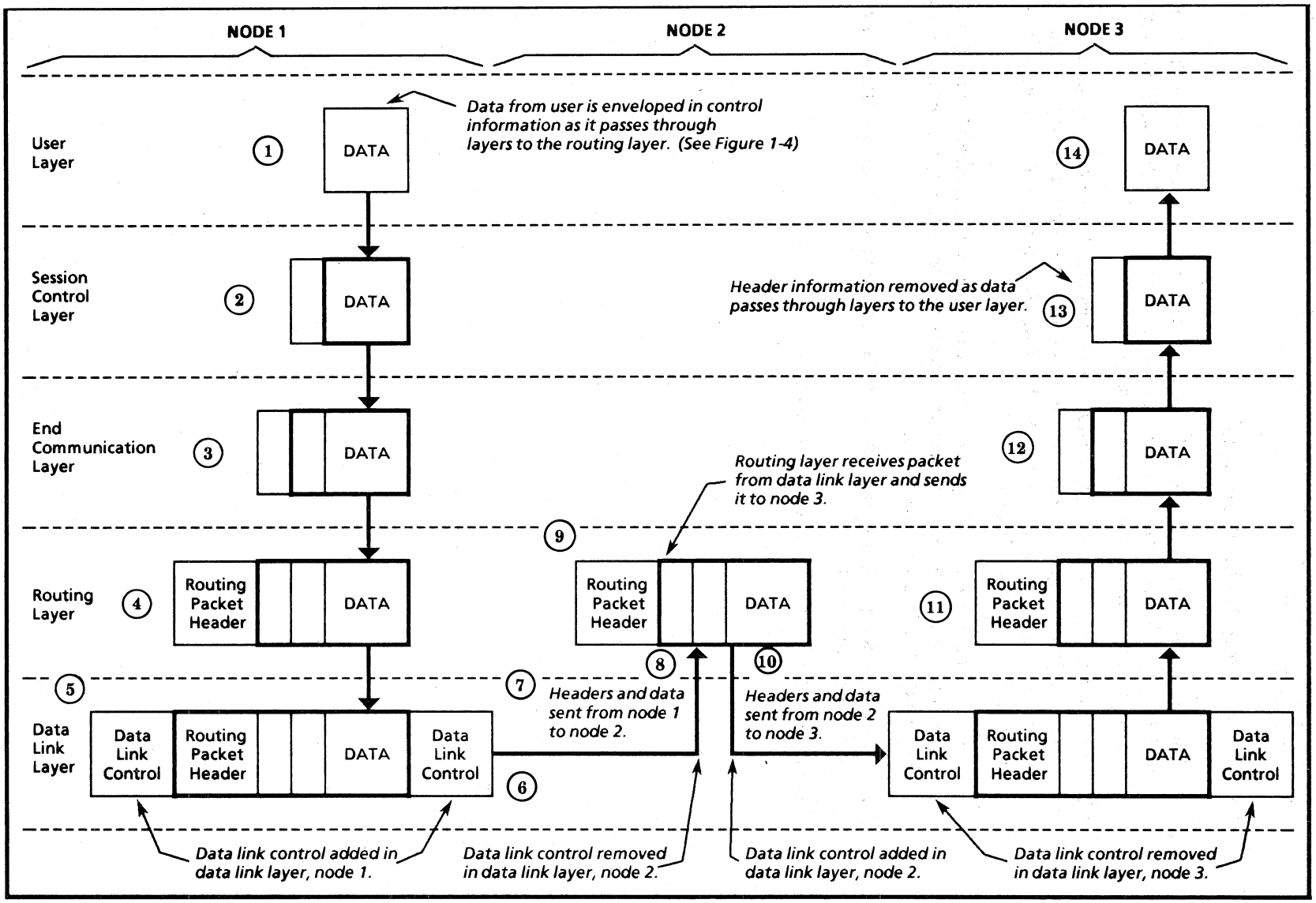
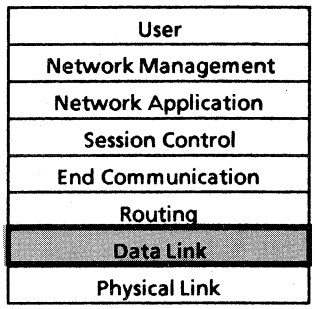


Figure 1-6: Data Flow



Chapter 2
The Data Link Layer

The Data Link layer, residing immediately above the Physical Link layer, is responsible for creating a communications path between adjacent nodes. The Data Link layer frames messages for transmisssion on the channel connecting the nodes, checks the integrity of received messages, manages the use of channel resources, and, when required, ensures the integrity and proper sequence of transmitted data.

Currently there are three protocols residing in the DNA Data Link layer:

- **Digital Data Communications Message Protocol (DDCMP).** DDCMP operates over synchronous or asynchronous communications links. It provides correct sequencing of data and error control to ensure data integrity. Section 2.1 describes DDCMP.
- **Ethernet Data Link.** The Ethernet Data Link operates over a coaxial cable based local area network. It provides a best-effort delivery service and includes link management procedures and error detection capability. Section 2.2 describes the Ethernet Data Link.
- **X.25 Levels 2 and 3.** Levels 2 and 3 operate over level 1 of the CCITT recommendation X.25. This recommendation defines a standard interface between data terminal equipment (DTEs) such as DECnet nodes, and a packet - switched data network. X.25 based data networks provide a virtual circuit service between pairs of DTEs. Section 2.3 describes X.25 levels 2 and 3.

2.1 DDCMP Functional Description

DDCMP is a *byte-oriented* protocol. There are three general types of data link protocols: byte-oriented, character-oriented, and bit-oriented. A byte-oriented protocol provides a count of the number of bytes that will be sent in the data portion of each data message sent. In contrast, a character-oriented protocol uses special ASCII characters to indicate the beginning of a message and the end of a block of text, while a bit-oriented protocol uses flags to frame data, sent in undefined lengths.

DDCMP was designed in 1974 specifically for DNA. DDCMP is functionally similar

to HDLC (High-level Data Link Control – the data link protocol adopted in 1975 by the International Standards Orgainization), although HDLC is a bit-oriented protocol. Another type of data link protocol, binary synchronous (BISYNC), is character-oriented.

DDCMP is a general-purpose protocol: it operates on a variety of communication systems from the very small to the very large. DDCMP makes maximum use of channel bandwidth and handles transparent data efficiently. (*Data transparency* is the capability of receiving, without misinterpretation, data containing bit patterns that resemble protocol control characters. Character-oriented protocols cannot handle transparent data as efficiently as byte- or bit-oriented protocols.) Other major goals of the DDCMP design include error recording, to warn of impending channel failure on degraded lines, and the provision of a simplified mode for bootstrapping and testing functions.

DDCMP transmits data grouped into physical blocks known as data messages. DDCMP provides a mechanism for exchanging error-free messages. A general description of how this mechanism works follows:

DDCMP assigns a number to each data message beginning with number one (after each initialization) and incremented by one (modulo 256) for each subsequent data message. DDCMP places a 16-bit cyclic redundancy check (CRC-16) error detection polynomial at the end of each data message transmitted.

The receiving DDCMP module checks for errors and, if there are none, returns the message number with a positive acknowledgment of message receipt.

The receiving DDCMP need not acknowledge each message sent; acknowledgement of data message *n* implies acknowledgment of all data messages sent up to and including data message *n*.

If an error is detected by the receiving DDCMP, it uses time-outs and control messages to resynchronize and trigger retransmission.

The principal DDCMP features are as follows:

- Obtains data from the Physical Link layer in blocks consisting of 8-bit bytes.
- Sequences data by message numbers.
- *Pipelines* up to 255 messages. That is, it sends messages without waiting for acknowledgment of each successive message.
- Operates independently of channel bit width (serial or parallel) and transmission characteristics (asynchronous or synchronous).
- Operates with a wide variety of communication hardware and modems.
- Detects errors by means of CRC-16 message trailers.
- Retransmits to correct errors.
- Achieves optimum performance with techniques such as pipelining, *piggybacking* (that is, sending an acknowledgment within a returned data message), and implying a positive acknowledgment of previous messages by negative acknowledgment of current message.
- Operates in both half-duplex and full-duplex modes.
- Supports point-to-point and multipoint communications.
- Synchronizes transmission and reception on byte and message level.

- Frames (envelops) data messages.
- Provides a maintenance mode for diagnostic testing and bootstrapping functions.
- Provides data transparency.
- Notifies the other end of the link when restarting or initializing.
- Maintains error counters.
- Records the occurrence of events for automatic error reporting to the user.

2.1.1 DDCMP Messages

There are three types of DDCMP messages:

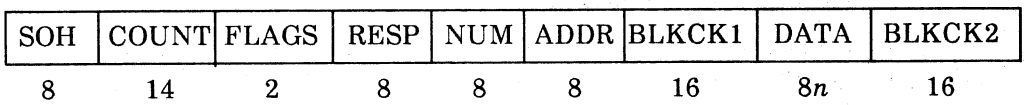
- Data (Section 2.1.1.1)
- Control (Section 2.1.1.2)
- Maintenance (Section 2.1.1.3)

Data messages send user data over a physical link. Control messages return acknowledgments and other control information. Maintenance messages consist of the basic DDCMP envelope but contain information for down-line loading, up-line dumping, link testing, or controlling a remote, adjacent system.

2.1.1.1 Data Messages

DDCMP formats all messages received from the Routing layer to be sent across the physical link into a data message format (Figure 2-1). The data message format ensures proper handling and error checking of both the header information and the data being sent. In the message format figures in this chapter, the numbers below each message field indicate its length in bits.

Data Message Format



SOH = the numbered data message identifier
COUNT = the byte count field
FLAGS = the link flags
RESP = the response number
NUM = the transmit number
ADDR = the station address field
BLKCK1 = the block check on the numbered message header
DATA = the n-byte data field, where $0 < n = \text{COUNT} < 2^{14}$
BLKCK2 = the block check on the data field

Figure 2-1: DDCMP Data Message Format

2.1.1.2 Control Messages

DDCMP has five control messages, which carry link control information, transmission status, and initialization notification between DDCMP modules. A brief description of each message follows.

Acknowledge Message (ACK). This message acknowledges the receipt of correctly numbered data messages that have passed the CRC-16 check. The ACK message is used when acknowledgments are required and when no numbered data messages are to be sent in the reverse direction. The ACK message conveys the same information as the RESP field in numbered data messages.

Negative Acknowledge Message (NAK). The NAK message passes error information from the DDCMP data-receiving module to the DDCMP data-sending module. The NAKTYPE field indicates the cause of the error. The NAK message serves two purposes:

- It acknowledges receipt of all previously transmitted messages with a number less than the current message number received (modulo 256).
- It notifies the sender of error conditions related to the current message.

Reply to Message Number (REP). The REP message requests received message status from the data receiver. The data sender sends a REP message under the following conditions:

- The data sender has sent a data message, and
- The data sender has not received an acknowledgment of that data message, and
- The time allocated for an acknowledgment has expired.

Start Message (STRT). The STRT message establishes initial contact and synchronization on a DDCMP link. The DDCMP module sends this message during link start-up or reinitialization.

Start Acknowledge Message (STACK). The STACK message is the response to a STRT message. It tells the receiving DDCMP that the transmitting node has completed initialization.

Figure 2-2 shows the formats of the DDCMP Control messages.

Acknowledge Message (ACK) Format

ENQ	ACKTYPE =1	ACKSUB =0	FLAGS	RESP	FILL =0	ADDR	BLKCK3
8	8	6	2	8	8	8	16

Negative Acknowledge Message (NAK) Format

ENQ	NAKTYPE =2	REASON	FLAGS	RESP	FILL =0	ADDR	BLKCK3

Reply to Message Number (REP) Format

ENQ	REPTYPE =3	REPSUB =0	FLAGS	FILL =0	NUM =0	ADDR	BLKCK3

Start Message (STRT) Format

ENQ	STRTTYPE =6	STRTSUB =0	FLAGS	FILL =0	FILL =0	ADDR	BLKCK3

Start Acknowledge Message (STACK) Format

ENQ	STCKTYPE =7	STCKSUB =0	FLAGS	FILL =0	FILL =0	ADDR	BLKCK3

ENQ	=	the control message identifier
ACKTYPE	=	the ACK message type with a value of 1
NAKTYPE	=	the NAK message type with a value of 2
REPTYPE	=	the REP message type with a value of 3
STRTTYPE	=	the STRT message type with a value of 6
STCKTYPE	=	the STACK message type with a value of 7
ACKSUB	=	the ACK subtype with a value of 0
REASON	=	the NAK error reason
REPSUB	=	the REP subtype with a value of 0
STRTSUB	=	the STRT subtype with a value of 0
STCKSUB	=	the STACK subtype with a value of 0
FLAGS	=	the link flags
RESP	=	the response number used to acknowledge received messages that checked out to be correct
FILL	=	a fill byte with a value of 0
ADDR	=	the tributary address field
BLKCK3	=	the control message block check

Figure 2-2: DDCMP Control Message Formats

2.1.1.3 Maintenance Messages

The Maintenance Message, used in DDCMP maintenance mode, has the format shown in Figure 2-3. This message is a DDCMP envelope for data controlling down-line loading, up-line dumping, link testing, and controlling an unattended computer system. The DNA protocol for performing these functions is the Maintenance Operation Protocol (MOP). MOP messages are sent within the DDCMP Maintenance Message. See Chapter 7 for a description of MOP.

Maintenance Message Format

DLE	COUNT	FLAGS	FILL	FILL	ADDR	BLKCK1	DATA	BLKCK2
8	14	2	8	8	8	16	8*n*	16

DLE	=	the maintenance message identifier
COUNT	=	the byte count field
FLAGS	=	the link flags
FILL	=	a fill byte with a value of 0
ADDR	=	the tributary address field
BLKCK1	=	the header block check on fields DLE through ADDR
DATA	=	the *n*-byte data field, where $0 < n = \text{COUNT} < 2^{14}$
BLKCK2	=	the block check on the DATA field

Figure 2-3: DDCMP Maintenance Message Format

2.1.2 DDCMP Operation

The DDCMP module has three functional components:

- Framing
- Link management
- Message exchange

Framing. The framing component locates the beginning and end of a message received from a transmitting DDCMP module. Framing involves locating a certain bit, byte, or message, and then receiving at the same rate as subsequent bits, bytes, or messages.

The modems and interfaces at the Physical Link level synchronize bits.

The DDCMP framing component synchronizes bytes by locating a certain 8-bit window in the bit stream. On asynchronous links DDCMP uses start-stop transmission techniques to synchronize bytes. On synchronous links DDCMP searches for a SYN character. Byte synchronization is inherent in 8-bit multiple parallel links. DDCMP synchronizes messages by searching for one of the three special starting bytes (SOH, ENQ, or DLE) after achieving byte synchronization. To maintain message synchronization, DDCMP:

- Counts out fixed length headers.
- When required, counts out variable length data based on the count field of the header.

Link management. The link management component controls transmission and reception on links connected to two or more transmitters and/or receivers in a given direction. Link Management controls the direction of data flow on half-duplex links and the selection of tributary stations on multipoint links, using link flags. In addition, link management uses selective addressing to control the receipt of data on multipoint links.

Message exchange. The message exchange component transfers the data correctly and in sequence over the link. Message exchange operates at the message level (after framing is accomplished), exchanging data and control messages.

2.1.3.1 Typical Message Exchange

DDCMP is a *positive-acknowledgment-with-retransmission* protocol. This means that for each data message correctly received and passed to the next level DDCMP returns a positive acknowledgment. Such acknowledgment is either an Acknowledge Message (ACK) or a piggy-backed acknowledgment in the response (RESP) field of a data message

If DDCMP receives a message out of sequence or with an error detected by the CRC, DDCMP does not pass the data to the client. Typically, DDCMP does not acknowledge this message. Eventually a time-out occurs, and the data-sending DDCMP retransmits the message. Alternatively, DDCMP may send a NAK to the data-sending DDCMP module.

The transmission of DDCMP Data messages works as follows:

1. The transmitter increments the message number (modulo 256), putting it, *n*, in the data message. The message is transmitted within the required framing envelope. A timer is started.
2. The receiver frames and receives the message, checks the received CRC value against a computed CRC value, and compares the message number with that expected. If the message checks out, the receiver returns a positive acknowledgment (ACK) with that number, passes the message to the client, and increments the next expected number to $n + 1$ (modulo 256). If the message does not check out, the receiver ignores the message (or sends a NAK).
3. The transmitter then follows one of three procedures:
   - If the transmitter receives a positive acknowledgment, it checks the number received to see if it is for an outstanding message. If so, the transmitting DDCMP notifies the client of successful transmission of that message as well as of any previous, lower-numbered outstanding messages. If all outstanding messages have been acknowledged, the timer is stopped. If one or more messages remain outstanding, the timer is restarted.
   - If the transmitter receives nothing, the timer expires. The transmitter sends a REP message to initiate error recovery.
   - If the transmitter receives a negative acknowledgment, it retransmits the message and all higher-numbered messages.

The transmitter may send several data messages before requiring acknowledgment of the first one. Acknowledgment of the highest numbered message implies acknowledgment of lower-numbered messages. A negative acknowledgment implies a positive acknowledgment of any previously transmitted lower-numbered messages.

Figure 2-4 shows a message exchange involving positive acknowledgment and pipelining. Figure 2-5 shows error recovery from a NAK.

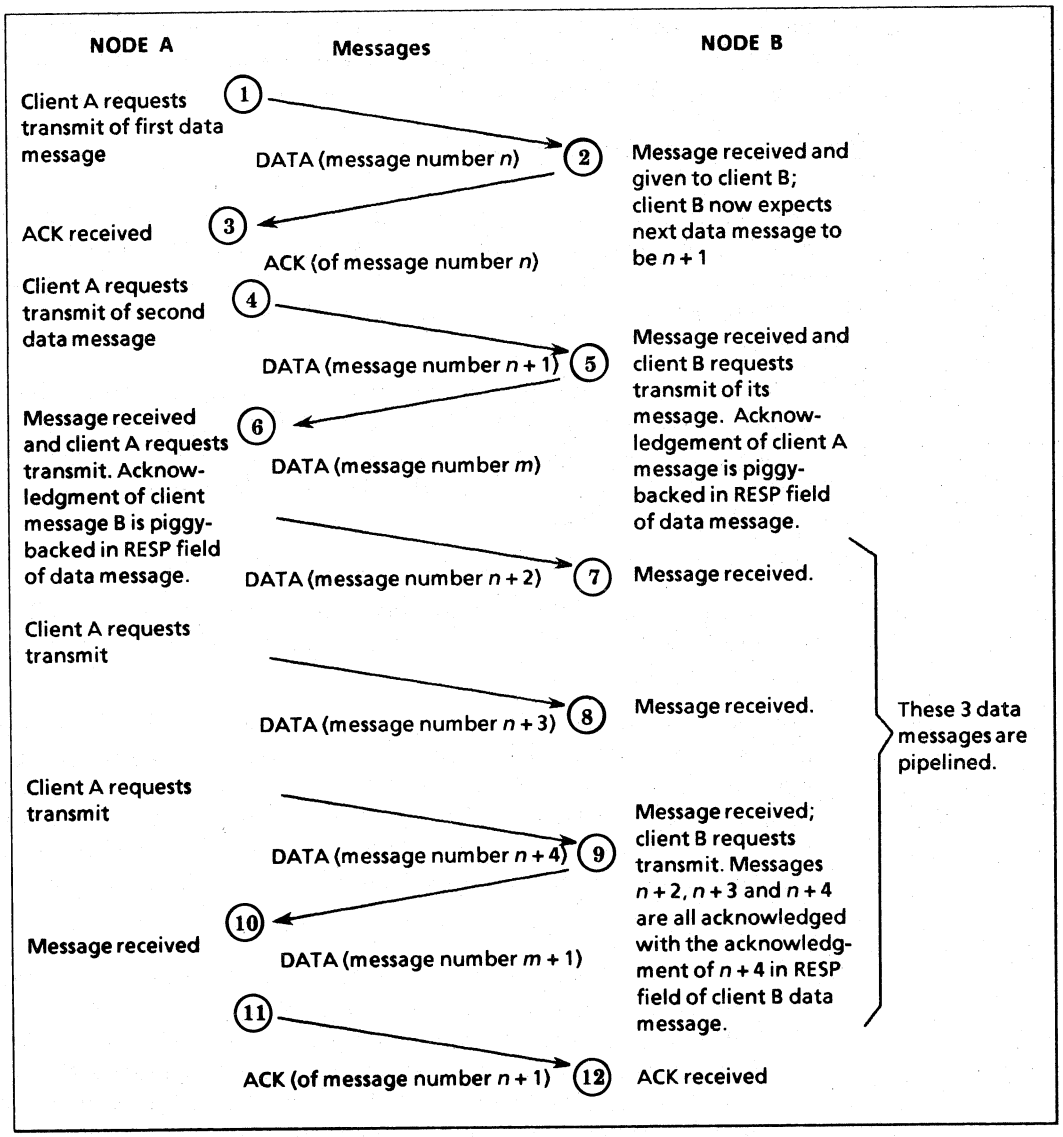


Figure 2-4: Typical DDCMP Message Exchange, Showing Positive Acknowledgment, Piggybacking, and Pipelining

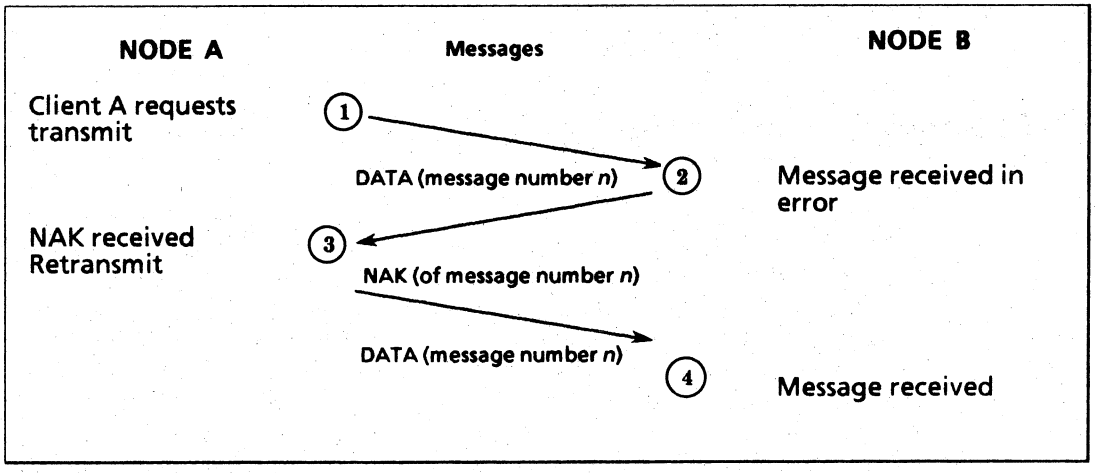


Figure 2-5: DDCMP Message Exchange, Showing Error Recovery

2.1.3.2 Maintenance Mode

Maintenance mode uses the DDCMP framing and link management components, but not the message exchange component. Sequencing or acknowledgment, if required, must be handled within the data fields of the maintenance messages and is part of the higher level protocol.

2.2 Ethernet Functional Description

The Ethernet local area network provides a communication facility for high speed data exchange among computers and other digital devices located within a moderate-size geographic area. It includes a Physical layer and a Data Link layer. The primary characteristics of the Physical layer are a data rate of 10 million bits per second, a maximum station separation of 2.5 kilometers, a maximum of 1024 stations, a shielded coaxial cable medium using base-band signalling, and support of a branching, non-rooted tree topology. The primary characteristics of the Data Link layer are a link control procedure using a fully distributed peer protocol with statistical contention resolution and a message protocol which supports variable size frames and offers best-effort service.

The Ethernet is a local area network developed jointly by Digital Equipment Corporation, Intel Corporation, and Xerox Corporation. The Ethernet specification arose from an extensive collaborative effort of the three corporations and several years of work at Xerox on a prototype Ethernet. The Ethernet specification provides a detailed specification of the two lowest layers of a network architecture. DNA Phase IV incorporates the Ethernet into the Physical Link and Data Link layers and includes support of the Ethernet in the Routing and Network Management layers.

The Ethernet Data Link layer defines a medium-independent link level communication facility, built on the medium-dependent physical channel provided by the Ethernet Physical layer. It is applicable to a general class of local area broadcast media suitable for use with the channel access discipline known as *Carrier-Sense Multiple-Access with Collision-Detection (CSMA/CD)*. The Ethernet Data Link layer provides the following functions:

- **Data Encapsulation**
  - Framing (frame boundary delimitation)
  - Addressing (handling of source and destination addresses)
  - Error detection (detection of physical channel transmission errors)
- **Link Management**
  - Channel allocation (collision avoidance)
  - Contention resolution (collision handling)

The split is reflected in the division of the Data Link layer into the Data Encapsulation sub-layer and the Link Management sub-layer, as shown in Figure 2-6.

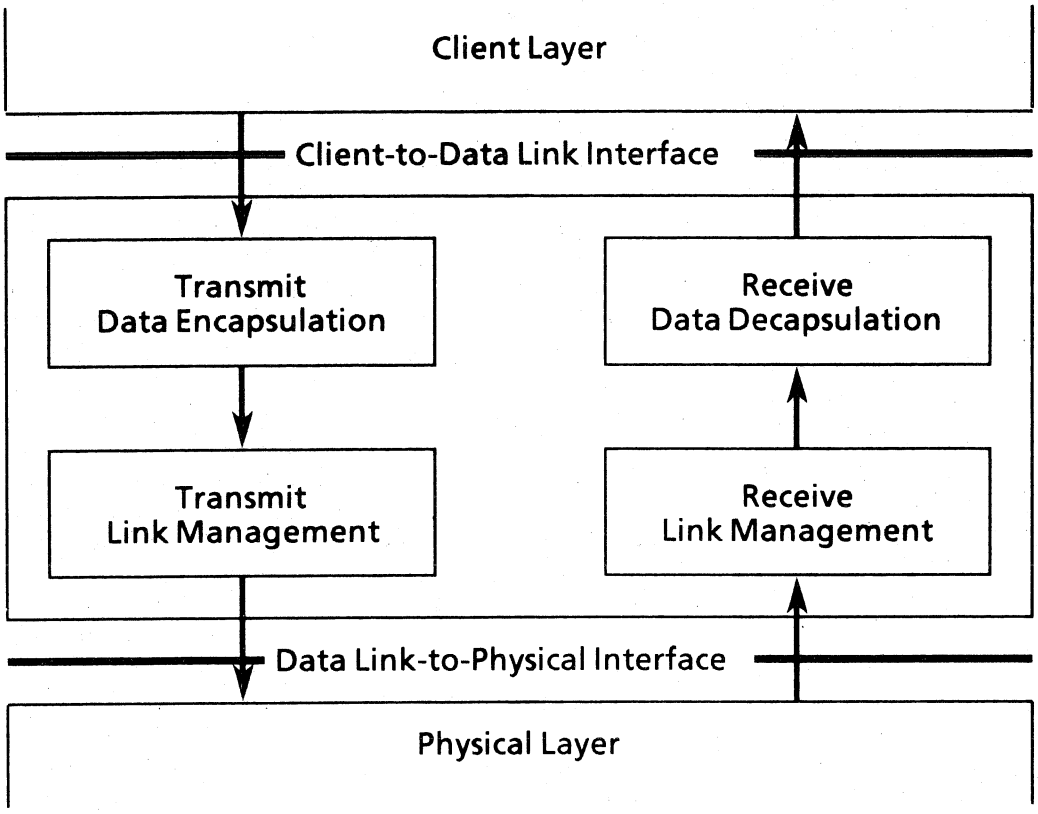


Figure 2-6: Ethernet Data Link Layer Functions

The Ethernet Data Link provides a best-effort delivery service. It does not provide an error control facility to recover from transmission errors. In DNA the error control functions necessary for reliable communications are provided by the End Communication layer, using the Network Services Protocol (NSP). Additional error control at the data link level is not required for Ethernet local networks, due to the inherently low error rate of the Ethernet physical channel.

2.2.1 Ethernet Messages

The Ethernet Data Link has one type of message, called a *frame*. The data encapsulation function of the Data Link layer comprises the construction and processing of frames. The sub-functions of framing, addressing, and error detection are reflected in the frame format as follows:

- **Framing.** No explicit framing information is needed since the necessary framing cues are present in the interface to the Physical layer.
- **Addressing.** Two address fields are provided to identify the source and destination stations for the frame.
- **Error Detection.** A Frame Check Sequence (FCS) field is provided for detection of transmission errors.

Figure 2-7 below illustrates the Ethernet Frame format. In the figure, the numbers below each field indicate its length in bits.

Ethernet Data Link Frame Format

DESTINATION	SOURCE	TYPE	DATA	FCS
48	48	16	$8n$	32

DESTINATION = the destination data link address. A data link address is one of three types:

- **Physical Address:** The unique address associated with a particular station on the Ethernet. The 48 bit field permits a station to have a unique address over all Ethernets. In DNA Phase IV, each network node has a 16 bit node address. The 48 bit Ethernet data link address of a DNA Phase IV node is derived by prefixing the 16 bit address with a 32 bit prefix assigned to DNA Phase IV nodes. Thus, DNA Phase IV addresses are unique over a single DNA network, which may include multiple Ethernets, DDCMP links, and X.25 links.
- **Multicast Address:** A multi-destination address associated by higher level convention with a group of logically related stations on an Ethernet. DNA assigns multicast addresses to the group of all Ethernet End nodes and to the group of all Ethernet Routers (see Section 3.2.1).
- **Broadcast Address:** A distinguished, predefined address which denotes the set of all stations on an Ethernet.

SOURCE = the source data link address. This field always contains the physical address of the station transmitting a frame on the Ethernet.

TYPE = the type field. The type field is reserved for use by higher level protocols to identify the higher level protocol associated with the frame, permitting multiple higher-level protocols to coexist in the same Ethernet. Ethernet type field values are assigned to the DNA Phase IV Routing protocol and to the DNA Maintenance Operation protocols.

DATA = the data field. The Ethernet data field contains higher level protocol data and is $8n$ bits long, where $46 \leq n \leq 1500$. Full transparency is provided, in the sense that any arbitrary sequence of 8 bit bytes may appear in the data field. The minimum length of the data field ensures that all frames occupy the channel long enough for reliable collision detection.

FCS = the frame check sequence. This field contains the CRC-32 polynomial check on the rest of the frame.

Figure 2-7: Ethernet Data Link Frame Format

DNA defines a standard convention for padding higher level protocols to the minimum Ethernet data field size. Higher level protocol data is preceded by a 16 bit byte count and followed by any required pad bytes. When this convention is in effect, Ethernet frames have the following format, and the COUNT, DATA, and PAD fields have the meaning described in Figure 2-8:

Ethernet Frame Format with Padding

DESTINATION	SOURCE	TYPE	COUNT	DATA	PAD	FCS
48	48	16	16	$8n$	$8m$	32

COUNT = the 16 bit count of DATA bytes, where COUNT = n

DATA = the data field. The Ethernet data field contains higher level protocol data and is $8n$ bits long, where $0 \leq n \leq 1498$. Full transparency is provided, in the sense that any arbitrary sequence of 8 bit bytes may appear in the data field.

PAD = zero or more bytes of zeros, where $m = \max(0,44-n)$.

Figure 2-8: Ethernet Frame Format with Padding

2.2.2 Ethernet Operation

This section provides an overview of the operation of the Ethernet Data Link layer when transmitting or receiving frames. In this overview, the higher level user of the Ethernet is referred to as the *client layer*. In DNA Phase IV Routing and Network Management are clients of the Ethernet.

2.2.2.1 Transmission without Contention

When the Client layer requests the transmission of a frame, the Transmit Data Encapsulation component of the Data Link layer constructs the frame from the client-supplied data, in the format specified by the client (with or without padding). The Data Link layer appends a frame check sequence to provide for error detection and hands the frame to the Transmit Link Management component for transmission.

Transmit Link Management attempts to avoid contention with other traffic on the Ethernet channel by monitoring the carrier sense signal provided by the Physical Link layer, and deferring to passing traffic. When the channel is clear, frame transmission is initiated. The Data Link layer provides a serial stream of bits to the Physical layer for transmission.

The Physical layer transmits the frame, monitoring the medium and generating the collision detect signal which, in the contention-free case under discussion, remains off for the duration of the frame's transmission.

When the frame has completed without contention, the Data Link layer so informs the Client layer and awaits the next request for frame transmission.

2.2.2.2 Reception without Contention

At the receiving station, the arrival of a frame is first detected by the Physical Link layer. As the encoded bits arrive from the medium they are decoded and passed to the Data Link layer.

Meanwhile, the Receive Link Management component of the Data Link layer, having seen carrier sense go on, has been waiting for the incoming bits to be delivered. Receive Link Management collects bits from the Physical layer as long as the carrier sense signal remains on. When the carrier sense signal goes off, the frame is passed to Receive Data Decapsulation for processing.

Receive Data Decapsulation checks the frame's destination address field to decide whether the frame should be received by the station. If so, the type field is checked to ascertain which client module should process the incoming frame (for example, Routing or Network Management). If the client module specified that padding is in use for this protocol type, the frame is decapsulated appropriately. The contents of the frame are then passed to the client with an appropriate status code. The status code is generated by inspecting the frame check sequence to detect any damage to the frame enroute and by checking for proper byte boundary alignment of the end of the frame.

2.2.2.3 Collisions: Handling of Contention

If multiple stations attempt to transmit at the same time, their transmitting data link controllers may interfere with each others' transmissions in spite of their attempts to avoid this by deferring. When two stations' transmissions overlap, the resulting contention is called a *collision*. A station can experience a collision only during the initial part of its transmission (the *collision window*), before its transmitted signal has had time to propagate to all parts of the Ethernet channel. Once the collision window has passed, the station is said to have acquired the channel; subsequent collisions are avoided, since all other stations can be assumed to have noticed the signal, via carrier sense, and to be deferring to it.

In the event of a collision, the Physical Link layer first notices the interference on the channel and turns on the collision detect signal. This is noticed in turn by the Transmit Link Management component of the Data Link layer, and collision handling begins. First, Transmit Link Management enforces the collision by transmitting a bit sequence called a *jam*. This ensures that the duration of the collision is sufficient to be noticed by the other transmitting station(s) involved in the collision. After the jam is sent, Transmit Link Management terminates the transmission and schedules a retransmission attempt for a randomly selected time in the near future. Retransmission is attempted repeatedly in the face of repeated collisions. Since repeated collisions indicate a busy channel, Transmit Link Management attempts to adjust to the channel load by voluntarily delaying its own retransmissions to reduce its load on the channel (*backing off*). This is accomplished by expanding the interval from which the random retransmission time is selected on each retransmission attempt. Eventually, either the transmission succeeds, or the attempt is abandoned on the assumption that the channel has failed or has become overloaded.

At the receiving end, the bits resulting from a collision are received and decoded by the Physical layer just as are the bits of a valid frame. The Data Link's Receive Link Management component recognizes collision fragments, which are always shorter than the shortest valid frame, and discards them.

2.3 X.25 Functional Description

CCITT Recommendation X.25 defines a standard interface between an intelligent system (*Data Terminal Equipment* or *DTE*) and an intelligent system that is an access point to a public data network (*Data Communications Equipment* or *DCE*) operating in the packet mode. Public data networks offering the X.25 interface are available in many countries and represent an economical alternative to leased lines and dial-up lines for many applications.

CCITT Recommendation X.25 is structured into three levels:

- Level 1, the *Physical* level, defines the mechanical, electrical, functional, and procedural characteristics of the physical link between the DTE and the DCE. In DNA, Level 1 resides in the Physical Link layer.
- Level 2, the *Frame* level, defines the link access procedure for data exchange over the link between the DTE and the DCE. In DNA, Level 2 resides in the Data Link layer.
- Level 3, the *Packet* level, defines the packet format and control procedures for the exchange of packets containing control information and client data between the DTE and the DCE. In DNA, Level 3 resides in the Data Link layer.

There are two independent uses for X.25 in a Digital system or a DECnet network: (1) as a Data Link module that allows two DECnet systems to communicate, and (2) as a gateway to non-Digital systems accessible via a public data network. A single X.25 link can support multiple *virtual circuits*. DNA Routing may use several virtual circuits for communicating with DECnet nodes (see Section 3.3.4) while DNA X.25 Gateway Acess may use other virtual circuits for communicating with non-Digital systems (see Section 6.3).

2.3.1 X.25 Frame Level

The X.25 specification defines two alternative link access procedures (LAP and LAPB) for Frame level communication between a DTE and a DCE. DNA supports the LAPB procedure. This procedure defines the control of full duplex communication over a synchronous line between a DTE and a DCE. LAPB performs the following functions:

- Provides an error free link by detecting transmission errors and recovering from such errors
- Provides synchronization to ensure that the Frame level entities in the DTE and DCE are in step
- Detects procedural errors and reports them to the user of the link.

2.3.1.1 X.25 Frame Level Messages

Frame level modules in the DTE and DCE communicate by sending and receiving frames to provide link control, data exchange, flow control, and error control. There are three types of frames:

Information Frames (I Frames). Information frames are used to transmit packet level information and are also used to acknowledge previous correctly received Information frames.

Supervisory Frames (S Frames). Supervisory frames are used to perform channel control functions such as acknowledging correctly received Information frames, requesting retransmission of Information frames, and indicating temporary inability to receive Information frames.

Unnumbered Frames (U Frames). Unnumbered frames are used to provide additional link control functions such as connecting and disconnecting the link.

Frames are divided into commands and responses. Table 2-1 summarizes the frame types.

Table 2-1: X.25 Level 2 Frame Types

Format	Commands	Responses
I	Information Transfer (I)	
S	Receive Ready (RR) Receive NOT Ready (RNR) Reject (REJ)	Receive Ready (RR) Receive NOT Ready (RNR) Reject (REJ)
U	Connect Link (SABM) Disconnect (DISC)	Unnumbered Acknowledgment (UA) Disconnected Mode (DM) Frame Reject (FRMR)

2.3.2 X.25 Packet Level

The X.25 Packet level provides for multiple virtual connections between a DTE and other DTEs connected to the public data network. A virtual connection between two DTEs is called a *virtual circuit.* Data transfer on a virtual circuit has the following properties:

- It is full duplex. Packets may be transmitted in both directions simultaneously.
- It is sequential. Packets are delivered by the network to the remote DTE in the same order as transmitted.

- It is reliable. Packets are delivered without corruption of the Data field and are not duplicated by the network. Packets may be lost, but loss will be reported by the network to the transmitting DTE (normally by resetting or clearing the circuit).

A *logical channel* is an association between a DTE and its DCE for a given virtual circuit. A logical channel is identified by a *logical channel number*. For each virtual circuit a logical channel is assigned at both DTE/DCE interfaces. This number is allocated independently at both DTE/DCE interfaces. Each DTE identifies a virtual circuit by its local logical channel number.

Two types of virtual circuits are supported:

Switched Virtual Circuits (SVC). A *switched virtual circuit* is a temporary association between two DTEs. In DNA, switched virtual circuits connecting DECnet nodes are established and destroyed using Network Management functions. Switched virtual circuits connecting DECnet nodes to non-Digital systems are established by users of X.25 Gateway Access.

Permanent Virtual Circuits (PVC). A *permanent virtual circuit* is a virtual circuit between two DTEs that is always established. A logical channel is permanently allocated at each DTE/DCE interface to a permanent virtual circuit.

2.3.2.1 X.25 Packet Level Messages

The DTEs attached to a virtual circuit access the circuit by exchanging X.25 level 3 packets with their local DCEs, using X.25 level 2 procedures. Table 2-2 summarizes the X.25 level 3 packet types and their functions.

Table 2-2: X.25 Level 3 Packet Types

Packet Type	Direction	Description
Call Request	DTE to DCE	Assigns a logical channel to the DCE and establishes a virtual circuit to the DTE addressed in the packet.
Incoming Call	DCE to DTE	Indicates that a remote DTE wishes to establish a virtual circuit with the receiving DTE and assigns a logical channel.
Call Accepted	DTE to DCE	Indicates that the DTE accepts the establishment of the virtual circuit.
Call Connected	DCE to DTE	Indicates that the remote DTE has accepted the establishment of the virtual circuit.
Clear Request	DTE to DCE	Indicates that the DTE wishes to deassign the logical channel and destroy the virtual circuit.
DCE Clear Confirmation	DCE to DTE	Informs the DTE that the logical channel is deassigned.
Clear Indication	DCE to DTE	Indicates that the remote DTE destroyed the virtual circuit.
DTE Clear Confirmation	DTE to DCE	Informs the DCE that the virtual circuit has been destroyed and deassigns the logical channel.
DTE Data	DTE to DCE	Carries user data from the DTE over the logical channel.
DCE Data	DCE to DTE	Carries data from the remote DTE over the logical channel
DCE RR	DCE to DTE	Updates the DTE transmit flow control information.
DCE RNR	DCE to DTE	Indicates a temporary inability of the DCE to accept data packets. This condition is cleared by a DCE RR packet.
DTE RR	DTE to DCE	Updates the DCE flow control information, authorizing the transmission of additional DCE Data packets.
DTE Interrupt	DTE to DCE	Carries user interrupt data over the logical channel.
DCE Interrupt Confirmation	DCE to DTE	Acknowledges receipt of a DTE Interrupt packet.
DCE Interrupt	DCE to DTE	Carries Interrupt data from the remote DTE.
DTE Interrupt Confirmation	DTE to DCE	Acknowledges receipt of a DCE Interrupt packet.
Reset Request	DTE to DCE	Requests that the logical channel and virtual circuit be set to an initial state.
DCE Reset Confirmation	DCE to DTE	Acknowledges that the logical channel has been reset.
Reset Indication	DCE to DTE	Indicates that the logical channel has been reset.
DTE Reset Confirmation	DTE to DCE	Acknowledges the resetting of the logical channel.
DTE Restart Request	DTE to DCE	Requests that all logical channels for SVCs be deassigned and that all PVCs be reset.
DCE Restart Confirmation	DCE to DTE	Acknowledges the Restart Request
Restart Indication	DCE to DTE	Indicates that all logical channels for SVCs should be deassigned and all logical channels for PVCs should be reset.
DTE Restart Confirmation	DTE to DCE	Acknowledges a Restart Indication
Diagnostic	DCE to DTE	Indicates to the DTE an error on one of its logical channels.

2.3.2.2 X.25 Packet Level Operation

In response to a user request, the Packet level initiates the establishment of a virtual circuit to the specified DTE. The Packet level assigns a logical channel number and transmits a Call Request packet, specifying the logical channel number and the address of the remote DTE.

The remote DCE to which the DTE addressed in the Call Request packet is connected assigns a logical channel number and transmits an Incoming Call packet over the logical channel. The called DTE accepts the call by transmitting a Call Accepted packet over the logical channel. In DNA, fields in the Incoming Call packet (such as the *Calling DTE Address, Called DTE Subaddress,* and the *Data* field) determine which DNA Module handles an incoming call. (The called DTE can reject an incoming call by transmitting a Clear Request packet.)

Once the virtual circuit has been established, data packets can be transferred. Each data packet is sequentialy numbered (modulo 8). A Send Sequence Number is carried in each data packet and specifies the position of the packet in the sequential stream. A Receive Sequence Number is used to authorize the transmission of additional packets (by acknowledging received packets) and is carried in Data Packets, Receive Ready packets, and Receive Not Ready packets.

Flow Control

Flow control takes place separately for each direction of transfer on a logical channel, making use of the packet sequence numbers. Flow control is based on the concept of a *window.* A window is a range of packets authorized to be transmitted across the DTE/DCE interface. The window size is selected independently for each direction of transfer on a logical channel. The range of packets in the window changes as packets are received and acknowledged by the destination DTE packet level.

Flow control can also utilize Receive Ready and Receive Not Ready packets. In DNA, this method of flow control is employed when transmitting data to a DCE. Receive Not Ready packets are never used when receiving data from a DCE.

Logical Messages

A DTE can construct a logical message from a number of consecutive packets, by using the *More Data* bit in the Data packets. The More Data bit is set in each packet except the last in a sequence of packets which compose a logical message. Data packets also contain another control bit, the *Q* bit, which allows a transmitting DTE to define two categories of data.

Interrupt Data

Each direction of transmission on a virtual circuit may have an outstanding *Interrupt.* This interrupt flow allows a user to transmit one byte of information not subject to the rules of flow control which apply to normal data packets. Interrupt data is transmitted via an Interrupt packet. Interrupt data is acknowledged by an Interrupt Confirmation packet.

Reset Procedure

A *Reset* procedure may be used to initialize a virtual circuit. Either DTE or the public data network can initiate a reset. All data and interrupt packets in transit at the time of a reset are discarded by the network. When the network wishes to indicate that the virtual circuit is being reset, it transmits a Reset Indication packet containing a reason for resetting. When a DTE wishes to reset a logical channel it transmits a Reset Request packet. A reset of one logical channel for a virtual circuit will cause a reset of the corresponding remote logical channel. A DCE confirms a Reset Request packet by transmitting a DCE Reset Confirmation packet. A DTE confirms a Reset Indication packet by transmitting a Reset Confirmation packet.

Clearing Procedure

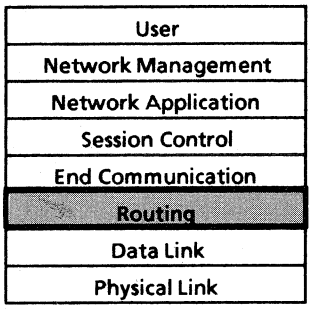
Either DTE associated with a switched virtual circuit can destroy the circuit by *Clearing*. The DTE transmits a Clear Request packet. The DCE returns a DCE Clear Confirmation packet which deassigns the logical channel. The Clear Request is reported to the remote DTE in a Clear Indication packet. The remote DTE replies with a DTE Clear Confirmation packet. The logical channel at the remote DTE is then deassigned.

Restart Procedure

The *Restart* procedure allows all switched virtual circuits associated with a DTE to be Cleared and all permanent virtual circuits to be reset after a catastrophic failure. A DTE issues a Restart Indication packet each time the DTE level 2 module connects to the DCE. A DCE can transmit a Restart Indication packet to force a DTE to clear all its virtual circuits.

Permanent Virtual Circuits (PVCs)

Permanent virtual circuits operate in the same manner as switched virtual circuits except that no call setup is required. Logical channel numbers are permanently assigned to PVCs. After a Restart, all PVCs are usable for data transfer. In DNA, Network Managment defines the circuit name and parameters of each PVC known to the packet level.



Chapter 3
The Routing Layer

The Routing layer provides a message delivery service. The Routing layer accepts messages (called packets within the context of Routing) from the End Communication layer in a source node, and forwards the packets, possibly through intermediate nodes, to a destination node. Routing implements a *Datagram* Service. A Datagram service delivers packets on a best-effort basis. That is, the Routing layer makes no absolute guarantees against packets being lost, duplicated, or delivered out of order. Rather, a higher layer of DNA provides such guarantees (see Chapter 4, The End Communication Layer).

The Routing layer selects routes based on network topology and operator-assigned circuit costs and adapts to changes in the network topology. For example, Routing can find an alternate path if a circuit or node fails. Routing does not adapt to traffic loading. The amount of traffic on a circuit does not affect the path selected by Routing for forwarding packets.

3.1 Routing Functional Description

The Routing layer provides the following functions:

- **Determines packet paths.** A path is the sequence of circuits and nodes) between a source node and a destination node. If more than one path exists for a packet, Routing determines the best path.
- **Forwards packets.** If a packet is addressed to the local node, Routing delivers it to the End Communication layer. If a packet is addressed to a remote node, Routing forwards it on the next adjacency in the path.
- **Adapts to topology changes.** If a circuit or node fails on a path, Routing finds an alternate path, if one exists.
- **Adapts to different kinds of circuits.** Routing supports paths consisting of multiple types of Data Link circuits, including point-to-point and multi-point lines, Ethernets, and X.25 virtual circuits.
- **Periodically updates other Routing Modules.** Routing modules in other nodes are periodically updated to be aware of any topology changes (such as circuits down or up).

- **Returns packets addressed to unreachable nodes.** Routing returns packets addressed to unreachable nodes to the End Communication layer, if requested to do so by the End Communication layer.
- **Manages buffers.** Routing manages the buffers at nodes that are capable of routing packets from a remote source to a remote destination.
- **Limits the number of nodes that a packet can visit.** Routing prevents old packets from cluttering up the network.
- **Performs node verification.** Routing will exchange verification passwords with an adjacent node if so requested by Network Management.
- **Monitors errors detected by the Data Link layer.**
- **Maintains counters and gathers event data for network management purposes.**

Routing allows for a functional subset, permitting two types of Phase IV Routing implementations:

1. **Routing.** Sometimes called *full routing*, this implementation type contains the full complement of Routing components. A Phase IV routing node can:
   - Send packets from itself to any other Phase IV node.
   - Receive packets from any other Phase IV node.
   - Route packets from other nodes through to other nodes. This is referred to as *route-through* or *packet switching*.
2. **Nonrouting.** This implementation type contains a subset of the Routing components. A nonrouting node can:
   - Send packets from itself to any other Phase IV node.
   - Receive packets addressed to itself from any other Phase IV node.

Nonrouting nodes can be placed only as *end nodes* in a network. An end node may be connected to a network by only one active circuit. An end node does not require the packet switching components. Such a node, therefore, has less routing overhead.

Phase IV nodes are compatible with Phase III nodes. Phase IV nodes differ from Phase III nodes in the following ways:

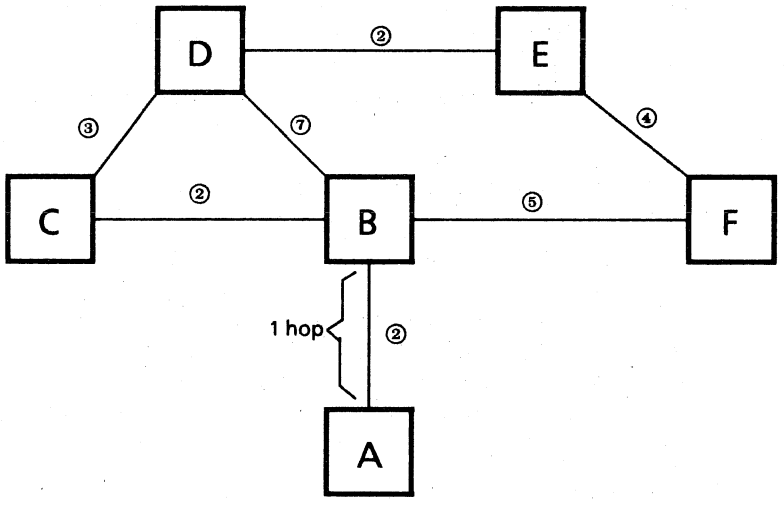
- Phase IV nodes can be attached to an Ethernet.
- Phase IV nodes can handle larger networks than Phase III nodes. A Phase III node cannot communicate with, nor be on the path to, nodes with addresses out of the range that the Phase III node can handle.
- Phase IV packet formats differ from Phase III in minor ways. Phase IV nodes translate packet formats into Phase III format when communicating with a Phase III node.

The logical distance from one node to another in a network is a *hop*. The complete distance that a packet travels from source to destination is the *path length*. Path length is measured in hops. The maximum number of hops the routing algorithm will

forward packets is called *maximum visits*. This number is limited to 63 by the Routing architecture.

Routing allows the network manager to assign a *cost* to each circuit connected to a node. Cost is an arbitary integer (within the size limits of the cost field). Cost is used in the routing algorithm that determines the best path for a packet. The Routing layer routes packets on the path of least cost, even if this is not the path with the fewest hops. The DNA Routing Functional Specification does not dictate how to assign circuit costs, but it does suggest a cost assignment algorithm based on circuit bandwidth. Both cost and maximum hops are values the system manager at each node assigns using Network Management software (Chapter 7).

Figure 3-1 illustrates some of the Routing terms. The glossary contains precise definitions of these and other Routing terms (including network diameter, maximum path length, maximum visits, maximum path cost, and maximum cost).



Node A wants to send a packet to Node D. There are three possible paths

PATH	PATH COST	PATH LENGTH
A to B, B to C, C to D	② + ② + ③ = 7*	3 hops
A to B, B to D	② + ⑦ = 9	2 hops
A to B, B to F, F to E, E to D	② + ⑤ + ④ + ② = 13	4 hops

*7 is the lowest path cost; Node A therefore routes the packet to Node D via this path.

Figure 3-1: Routing Terms

Routing does not automatically adjust to traffic flow, since packets always travel on the path having the lowest cost.

The Routing layer routes packets to nodes by using numerical addresses. Because users often prefer to refer to nodes by name rather than by address, network node names must be mapped to unique network node addresses. A node is known to other nodes by one (optional) name and one address, both of which are unique in the network. However, implementations can optionally provide local users with the capability of assigning additional node names that map to network-known names and/or addresss. This process occurs at the Session Control layer (Chapter 5).

3.2 Routing Messages

There are two types of Routing messages: data packets and control messages. Data packets carry data to and from the End Communication layer. Routing adds a packet route header to messages sent by the End Communication layer. Control messages exchange information between Routing modules in adjacent nodes to initialize the Routing layer, maintain routing data, and monitor the states of adjacencies.

3.2.1 Packet Route Header

There are two formats for data packet routing headers, depending on whether the adjacent node is on an Ethernet or some other type of circuit. The Ethernet Endnode Data Packet format is used when communicating directly with an Ethernet end node (a non-routing node directly connected to an Ethernet). The other format, the Phase IV Data Packet format, is used in all other cases. The Ethernet Endnode Data Packet Format has expanded addressing and reserved fields unused in Phase IV, for later expansion.

Figure 3-2 shows the Data packet routing header formats. The numbers under fields in this and the following Routing message formats indicate the lengths of the fields in bits.

Phase IV Data Packet Routing Header Format

RTFLG	DSTNODE	SRCNODE	FORWARD
8	16	16	8

Ethernet Endnode Data Packet Routing Header Format

RTFLG	DSTNODE	SRCNODE	RES	FORWARD	RES
8	64	64	8	8	16

RTFLG = the set of flags used by the routing nodes, including:
- Return to sender flag (indicates whether or not the packet is being returned)
- Return to sender request flag (indicates whether to discard or try to return packet)
- Intra-Ethernet Packet (indicates to the Ethernet Endnode receiving this packet that the source of the packet is on the same Ethernet, and can be communicated with directly).

DSTNODE = the destination node address
SRCNODE = the source node address
FORWARD = the number of nodes this packet can visit
RES = a reserved field

Figure 3-2: Packet Route Header Formats

3.2.2 Routing Control Messages

There are six types of Routing Control messages. One type, the Routing Message is used on all types of circuits. Two types of Routing Control messages are used only on Ethernets: the Ethernet Router Hello Message, and the Ethernet Endnode Hello Message. The final three types of Routing Control messages, the Hello and Test Message, the Initialization Message, and the Verification Message are used only on non-Ethernet circuits. The following paragraphs discuss each of these messages.

Routing message. The Routing message provides information necessary for updating the routing data base of an adjacent node. The information contained in the Routing message is the path cost and path length for a set of destinations.

Ethernet Router Hello message. The Ethernet Router Hello message is used for initialization and periodic monitoring of Routers on an Ethernet circuit. Each Ethernet router periodically broadcasts an Ethernet Router Hello message to all other nodes (both Routers and Endnodes) on the same Ethernet circuit using a multicast operation. The message contains a list of all Routers on the Ethernet circuit from which the sending Router has recently received Ethernet Router Hello messages. By exchanging Ethernet Router Hello messages, all Routers remain informed of the status of the other Routers on the Ethernet circuit. The message also provides input to an algorithm which selects a single Router on the Ethernet circuit to be the *Designated Router* for that Ethernet. The designated Router assists two end nodes in discovering that both are on the same Ethernet, by forwarding a packet from one to

the other as an Intra-Ethernet packet. Subsequent communications between the end nodes can be direct.

Ethernet Endnode Hello message. The Ethernet Endnode Hello message is used for initialization and periodic monitoring of Endnodes on an Ethernet circuit. Each Ethernet Endnode periodically broadcasts an Ethernet Endnode Hello message to all Routers on the Ethernet circuit using a multicast operation. The message is used by Ethernet Routers to maintain the status (up or down) of Endnodes on the Ethernet.

Hello and Test message. The Hello and Test message tests an adjacency to determine if it is still operational (that both the circuit and the adjacent node are up). Routing sends this message periodically on non-Ethernet circuits in the absence of other traffic. Upon receipt of this or any other valid message, Routing starts (or restarts) a timer. If the timer expires before another message is received from that node, Routing considers the adjacency down.

Initialization message. Routing sends this message when initializing a non-Ethernet circuit. The message contains information about the node type, required verification, maximum Data Link layer receive block size, and Routing version.

Verification message. This message is used for verification purposes on non-Ethernet circuits if the Initialization message indicates it is required.

Figure 3-3 summarizes the Routing control message formats.

Routing Message Format

CTLFLG	SRCNODE	RTGINFO	CHECKSUM
16	16	16*n*	16

Ethernet Router Hello Message Format

CTLFLG	VERS	ID	INFO	LIST
8	24	48	64	56*m*

Ethernet Endnode Hello Message Format

CTLFLG	VERS	ID	INFO
8	24	48	168

Hello and Test Message Format

CTLFLG	SRCNODE	TEST DATA
16	16	8-128

Initialization Message Format

CTLFLG	SRCNODE	INITFO
16	16	56

Verification Message Format

CTLFLG	SRCNODE	FCNVAL
16	16	8-64

CTLFLG = Routing control flag, with the following types :
- Initialization message
- Verification message
- Hello and Test message
- Routing message
- Ethernet Router Hello message
- Ethernet Endnode Hello message

SRCNODE = Identification of source node's Routing Module

RTGINFO = Path length and path cost to a set of *n* destinations

CHECKSUM = One's complement add check on routing information

VERS = Routing module version number

ID = Identification of node sending message

INFO = Node type (Router or Endnode), maximum Data Link layer receive block size, hello timer

LIST = list of known Routers on the Ethernet circuit (Each of the *m* entries consists of a 7 bit priority field for selecting the Designated Router, 1 bit to indicate two-way connectivity, and the 48 bit identification of a Router).

TEST DATA = sequence of up to 128 bytes of data to test the circuit

INITFO = node type, required Verification message, maximum Data Link layer receive block size, Routing version.

FCNVAL = type-dependent verification information; function value.

Figure 3-3: Routing Control Message Formats

3.3 Routing Operation

The Routing layer consists of the following two sublayers with associated functional components:

Routing Control Sublayer

- Routing (Section 3.3.1)
- Congestion control (Section 3.3.2)
- Packet lifetime control (Section 3.3.3)

Routing Initialization Sublayer

- Initialization (Section 3.3.4)

3.3.1 Routing

This component contains four processes described below:

- Decision
- Update
- Forwarding
- Receive

Decision. The decision process selects routes to each destination in the network. It consists of a connectivity algorithm that maintains path lengths, and a traffic assignment algorithm that maintains path costs. When a routing node receives a Routing message, the routing node executes the connectivity and traffic assignment algorithms. This execution results in updates to the data bases used to determine packet routes.

Update. The update process constructs and propagates Routing messages (see Section 3.2.2). The update process sends Routing messages to adjacent nodes as required by the decision process and periodically to ensure the integrity of the routing data bases.

Forwarding. The forwarding process supplies and manages the buffers necessary to support packet route-through to all destinations. It performs a table look-up to select the output circuit for the packet. If a destination is unreachable, this process either returns the packet to the sender or discards it, depending on the option flagged in the packet route header.

Receive. The receive process inspects a packet's route header, dispatching the packet to an appropriate Routing control component or to the End Communication layer.

3.3.2 Congestion Control

Congestion control consists of a single process, transmit management. This process manages buffers by limiting the maximum number of packets on a queue for a circuit. If a queue for a particular circuit reaches a predetermined threshold, additional

packets for that queue are discarded to prevent congestion. Transmit management also regulates the ratio of packets received directly from End Communication to route -through packets. This regulation prevents locally-generated packets from degrading route-through service.

3.3.3 Packet Lifetime Control

Packet lifetime control prevents excessive packet looping by discarding packets that have visited too many nodes.

3.3.4 Initialization and Circuit Monitor

Routing initialization is a start-up procedure and periodic procedure for adjacent nodes which ascertains neighbor node identities, and adapts to the characteristics of the circuit which provides communication to that neighbor (such as Data Link block size).

Initialization and monitoring operation are dependent on the type of circuit and the type of adjacent node (a circuit and the node at the other end of the circuit are collectively called an *adjacency*). The following paragraphs discuss these operations for the types of circuits supported by the Phase IV Routing Layer.

3.3.4.1 Ethernet Circuits

On an Ethernet circuit, initialization and monitoring are performed by the Ethernet Endnode Hello message and the Ethernet Router Hello message. The initialization and monitoring functions of these messages are described in Section 3.2.2.

3.3.4.2 DDCMP Circuits.

On DDCMP circuits (both point-to-point and multi-point), Initialization and monitoring are performed by the Initialization message, the Verification message, and the Hello and Test message. In addition to the functions described in Section 3.2.2 for these messages, the Initialization message adapts the Routing layer's operation to the block size of the physical line as managed by the Data Link Layer.

3.3.4.3 X.25 Circuits.

On X.25 virtual circuits (both Permanent and Switched) the Initialization and Verification messages are used as for DDCMP circuits, except that the Data Link block size is taken from the X.25 Data packet size selected when the virtual circuit is initialized.

X.25 circuits are initialized when the Routing layer is so commanded by Network Management. The Routing Initialization sub-layer contains a database, maintained by Network Management, which consists of a mapping between Routing layer circuits and X.25 Packet level Permanent Virtual Circuit identifiers in the case of Permanent Virtual Circuits, and between Routing layer circuits and virtual call parameters in the case of Switched Virtual Circuits. The database contains items such as *DTE subaddress range, maximum packet size*, number of times to reattempt failed calls, etc.

When placing a virtual call, the Routing Initialization sub-layer provides the necessary parameters to the X.25 Packet level component of the Data Link layer, which places the call to the remote Routing layer at the supplied foreign DTE address.

If the call is accepted (see below) Routing Initialization proceeds with circuit initialization as described for DDCMP.

The Routing Initialization layer listens for incoming virtual calls on circuits managed by the X.25 Packet level component of the Data Link layer. When informed of an incoming call by the X.25 Packet level, Routing Initialization executes an algorithm to determine whether to accept the call, reject the call, or leave the call pending in case some other module of DNA (such as the X.25 Gateway Server module - see Section 6.3.2) wishes to process the call. The algorithm uses the mapping database described above and makes its decision based on comparing the foreign DTE subaddress in the Incoming Call packet to the DTE subaddress range in its database. If the decision is to accept the call, initialization proceeds as described in the preceding paragraphs.

Once an X.25 circuit is established, the Routing Initialization sub-layer performs some additional functions not performed for Ethernet or DDCMP circuits, in order to maximize the performance and integrity of X.25 circuits. These functions are as follows:

- **Blocks and deblocks Routing layer messages.** Since communication over X.25 circuits is in the form of (usually small) packets, Routing Initialization will block and deblock DNA Datagrams into X.25 Packets in order to minimize the number of packets transmitted over the virtual circuit.
- **Fragments and Reassembles Routing layer messages.** If the X.25 virtual circuit packet size is too small to contain an entire Routing layer message, Routing Initialization will fragment and reassemble Routing layer messages so that they will fit in the available X.25 packet size.
- **Checksums X.25 packets.** The Routing Initialization sub-layer appends a checksum to every Data packet transmitted to insure the integrity of data transmitted over the virtual circuit.

As with other types of circuits, the Routing Initialization sub-layer monitors X.25 circuits for non-recoverable errors. These include X.25 Resets, Restarts, and Clears. If any of these events happen, Routing Initialization declares the circuit down, so informs the Routing Control sublayer, and attempts to re-initialize the circuit.

Figure 3-4 summarizes the operation of the Routing layer.

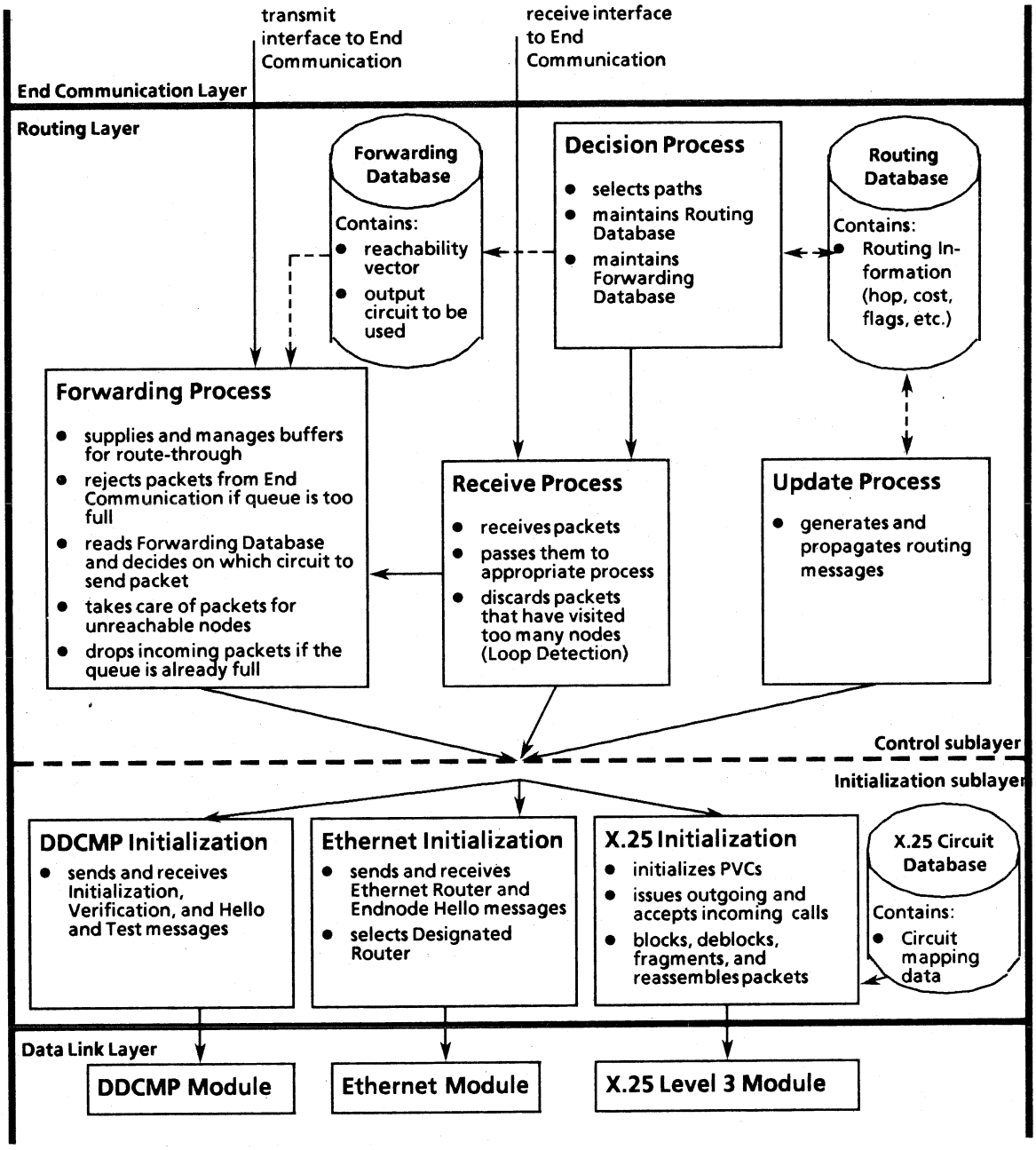
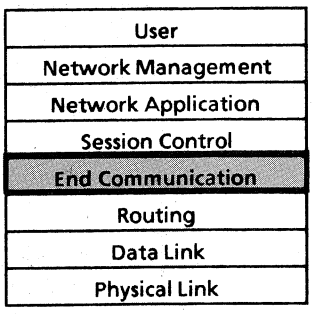


Figure 3-4: Routing Layer Components and their Functions



Chapter 4
The End Communication Layer

The End Communication layer, residing immediately above the Routing layer, provides a system-independent process-to-process communication service that allows two processes to exchange data reliably and sequentially, regardless of their locations in a network. The Network Services Protocol (NSP) is the protocol of the End Communication layer. Communication is on a connection basis. A connection between two processes is called a logical link. A logical link permits two-way simultaneous transmission of normal data messages and independent two-way simultaneous transmission of interrupt messages.

4.1 NSP Functional Description

NSP performs the following functions:

- Creates and destroys logical links.
- Guarantees the delivery of data and control messages in sequence to a specified destination by means of an error control mechanism.
- Manages the movement of interrupt and normal data from transmit buffers to receive buffers, using flow control mechanisms.
- Breaks up normal data messages into segments that can be transmitted individually, and reassembles these segments in proper sequence upon reception.

4.2 NSP Messages

NSP modules provide logical link service, flow control, error control, and other functions by sending and receiving NSP messages. There are three types of NSP messages:

- Data
- Acknowledgment
- Control

Table 4-1 summarizes the functions performed by each NSP message.

Table 4-1: NSP Messages

Type	Message	Description
Data	Data Segment	Carries a portion of a Session Control message. (This has been passed to Session Control from higher DNA layers and Session Control has added its own control information, if any.)
Data (also called Other Data)	Interrupt	Carries urgent data, originating from higher DNA layers. It also may contain an optional Data Segment acknowledgment.
	Data Request	Carries data flow control information and optionally a Data Segment acknowledgment (also called Link Service message).
	Interrupt Request	Carries interrupt flow control information and optionally a Data Segment acknowledgment (Link Service message).
Acknowledgment	Data Acknowledgment	Acknowledges receipt of either a Connect Confirm message or one or more Data Segment messages, and optionally an Other Data message.
	Other Data Acknowledgment	Acknowledges receipt of one or more Interrupt, Data Request or Interrupt Request messages, and optionally a Data Segment message.
	Connect	Acknowledges receipt of a Connect Initiate message.
Control	Connect Initiate and Retransmitted Connect Initiate	Carries a logical link connect request from a Session Control module.
	Connect Confirm	Carries logical link connect acceptance from a Session Control Module.
	Disconnect Initiate	Carries a logical link connect rejection or disconnect request from a Session Control module.
	No Resources	Sent when a Connect Initiate message is received and there are no resources to establish a new logical link (also called Disconnect Confirm message).
	Disconnect Complete	Acknowledges the receipt of a Disconnect Initiate message (also called Disconnect Confirm message).
	No Link	Sent when a message is received for a nonexistent logical link (also called Disconnect Confirm message).
	No Operation	Does nothing.

4.3 NSP Operation

NSP operation consists of four functions:

- Creation, maintenance, and destruction of logical links (Section 4.3.1)
- Segmentation and reassembly of data (Section 4.3.2)
- Error control (Section 4.3.3)
- Flow control (Section 4.3.4)

4.3.1 Logical Links

The primary functions of NSP are to create, operate, and destroy logical links at the request of the Session Control layer. A logical link may be thought of as a full-duplex logical channel between two users. The users are guaranteed that, in the absence of a network failure disconnecting them, data sent on a logical link by one user will be received by the other user in the order sent. An equivalent term for logical link that is often used is *virtual circuit*. A logical link enables a user process at one end of the link to send data to a user process at the other end of the link. The NSP mechanisms that set up a link, check data for errors, and manage data flow are transparent to the user processes.

There can be several logical links at any time, even between the same two NSP implementations. Any Phase III or IV node can establish a logical link with any other Phase III or IV node in the same network. Figure 4-1 illustrates typical connections.

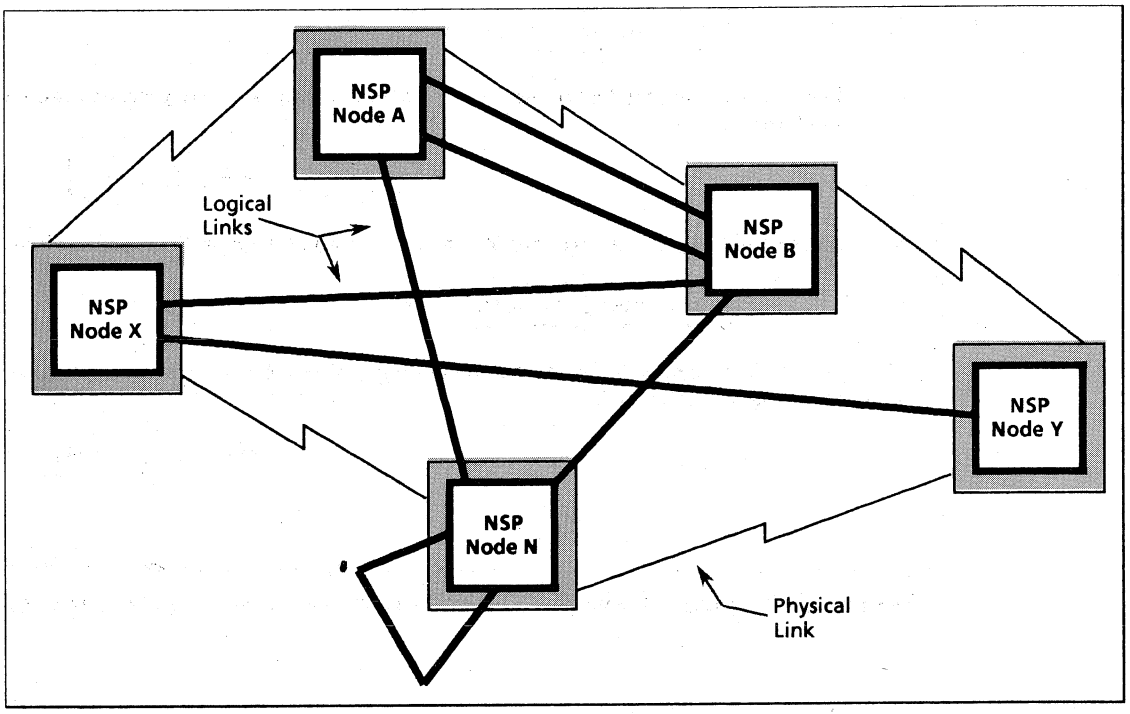


Figure 4-1: Logical Links

NSP establishes, maintains, and destroys logical links by exchanging control messages with other NSP modules or with itself. Figure 4-2 shows a typical message exchange. In this figure, an NSP module first initiates a connection, then sends data, and finally, disconnects the link, based on commands from Session Control. The messages that control data flow are not shown.

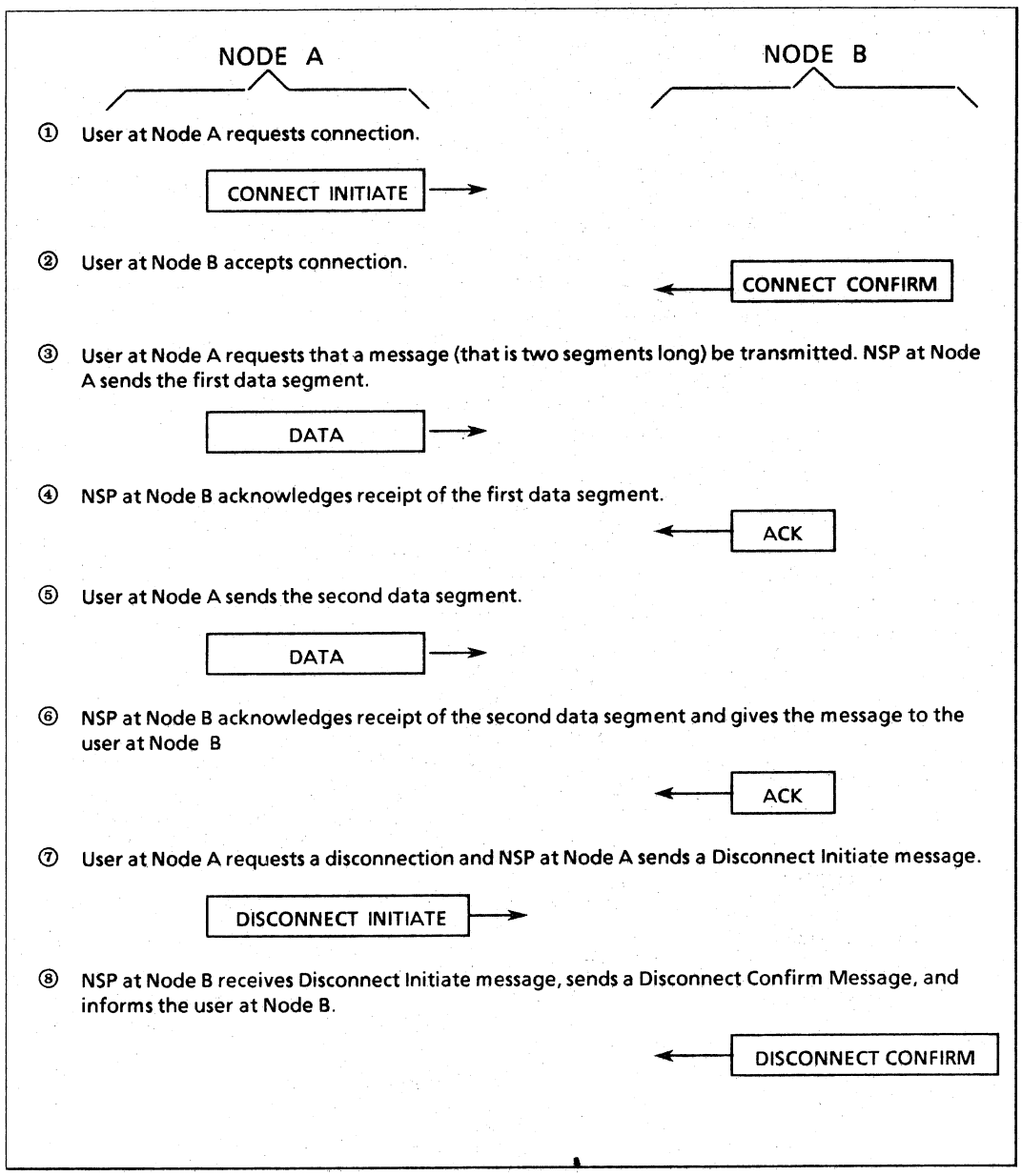


Figure 4-2: Typical Message Exchange Between Two Implementations of NSP

NSP operates concurrently in both directions on a logical link (full-duplex). The user process at either end of the link can initiate a disconnection at any time.

You can think of a logical link as made up of two separate data *subchannels*, each carrying messages in both directions:

Normal data subchannel. This subchannel carries Data Segment messages between two NSP modules.

Other Data Subchannel. This subchannel carries:

- Interrupt messages
- Data Request messages
- Interrupt Request messages

4.3.2 Segmentation and Reassembly of Data

The Routing layer limits the amount of user data that NSP can send in one datagram. Taking normal data from Session Control buffers, NSP breaks it up, if necessary, into segments. Each segment is numbered and sent along with its number and other control information in a Data Segment message to the receiving NSP module. The receiving NSP module uses the sequence numbers to reassemble the data segments in correct sequence in the receiving Session Control buffers.

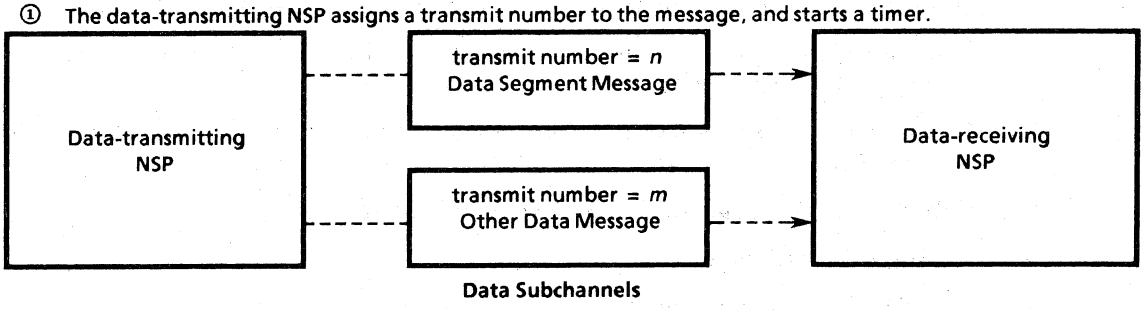
NSP segments normal data only: interrupts are not segmented because interrupt data is limited in size and will always fit in a single datagram.

4.3.3 Error Control

The NSPs at each end of a logical link positively acknowledge received data. Optionally, an NSP may negatively acknowledge received normal data that it must discard (for example, if the data is received too far out of order). If the transmitting NSP receives a negative acknowledgment or fails to receive a positive acknowledgment during a timeout interval, it retransmits the data.

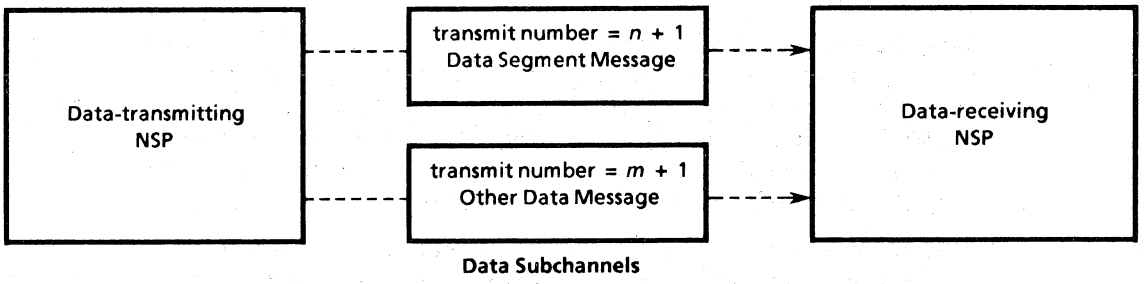
Figure 4-3 describes data segmentation and acknowledgment.

① The data-transmitting NSP assigns a transmit number to the message, and starts a timer.



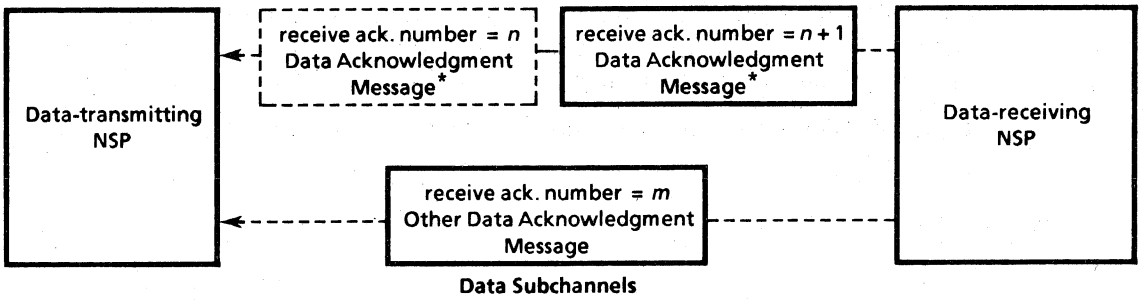
② If the timer times out, the message is retransmitted.

③ If the timer does not time out, and the flow control mechanism allows another message to be sent, the data-transmitting NSP assigns the transmit number plus one to the next data message transmitted in that subchannel.



④ When the message with the first transmit number is received by the data-receiving NSP, it returns that number as an acknowledgment number within the first acknowledgment.

⑤ If the next data message transmit number received is equal to the current acknowledgment number plus one, the data-receiving NSP accepts the data message, incrementing the acknowledgment number. It then sends the new receive acknowledgment number back to the data-transmitting NSP within an acknowledgment message.



*The data-receiving NSP might not send an acknowledgment for each data message received. The receive acknowledgment number implies that all previous numbers were received.

⑥ However, if the data-receiving NSP receives a data message transmit number less than or equal to the current receive acknowledgment number for that subchannel, the data segment is discarded. The data-receiving NSP sends an acknowledgment back to the data-transmitting NSP. The acknowledgment contains the receive acknowledgment number.

⑦ If the data-receiving NSP receives a data message transmit number greater than the current receive acknowledgment number plus one for that subchannel, the data segment may be held until the preceding segments are received or it may be discarded.

Figure 4-3: Acknowledgment Operation

4.3.4 Flow Control

NSP's flow control mechanisms ensure that data is not lost for lack of buffering capability and that deadlocks do not occur. Both normal and interrupt data are flow-controlled.

The data-receiving part of NSP controls data flow. When a logical link is formed, each NSP informs the other of the way it wants to control the flow of normal data as a data receiver. The receiving NSP chooses one of three types of normal data flow control:

- None.
- Segment. The receiver sends a request count of the number of segments it can accept. See Figure 4-4.
- Message. The receiver sends a request count of the number of Session Control messages it can accept. Note, message flow control is obsolescent and will be eliminated at a future time.

These schemes define the use of a request count that is set by the receiver and used by the transmitter to determine when data may be sent. In addition, the receiver can always tell the transmitter either to stop sending data unconditionally or to start sending data under the normal request count conditions. The receiver also controls interrupt data flow with an interrupt request count.

The Data Request Message, Interrupt Message, Interrupt Request Message, and the Other-Data-Ack Message may contain an acknowledgment for a Data-Segment Message. In addition, the Data-Segment Message and Data-Acknowledgment Message may contain an acknowledgment for an Other-Data Message. This ability, to combine acknowledgments with data and control messages, reduces the number of NSP messages required per user message.

Figure 4-4 illustrates NSP Flow control operation:

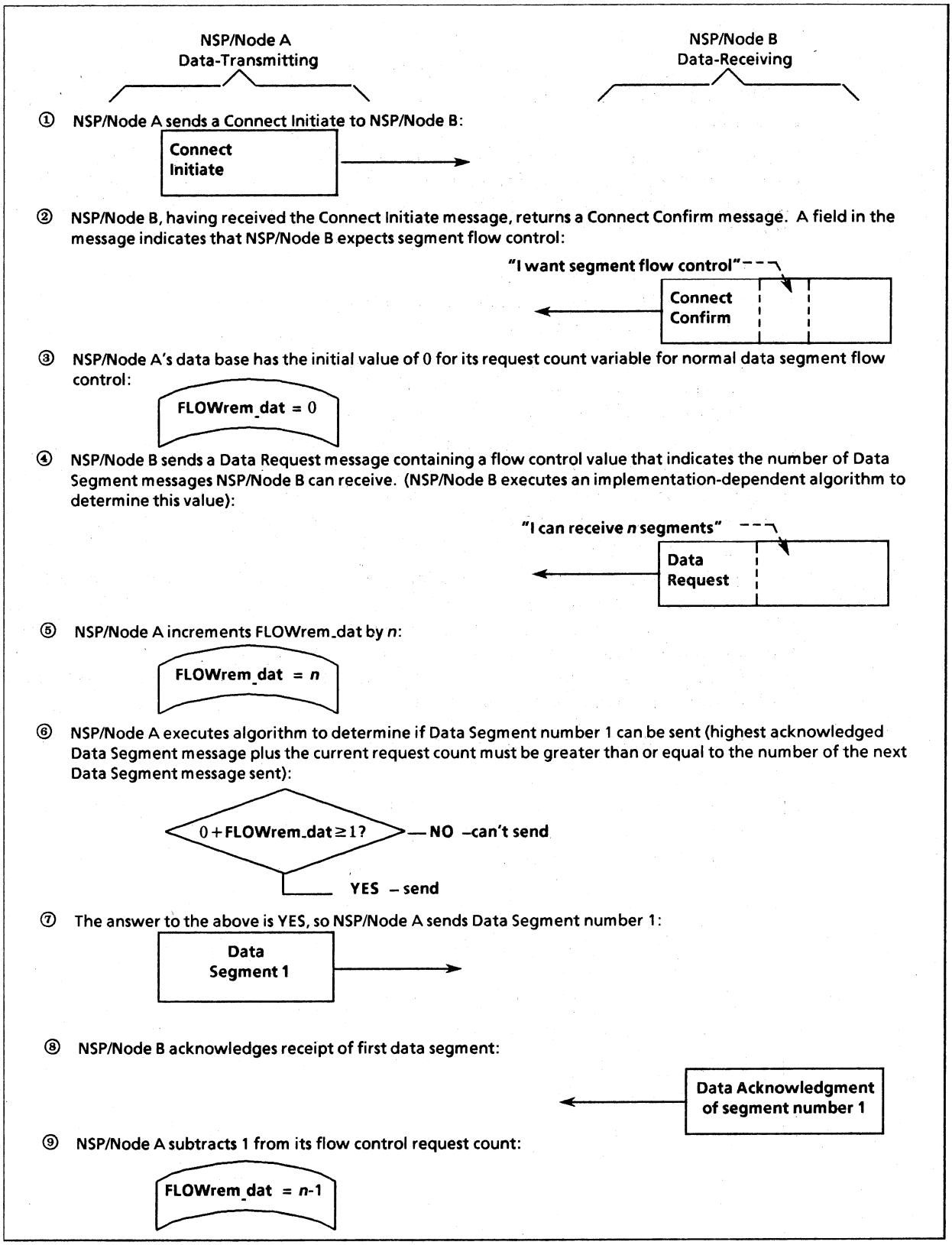
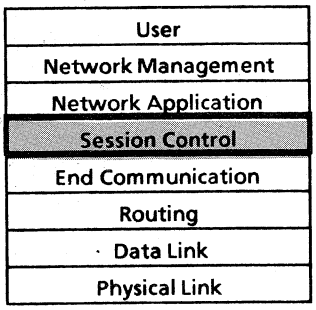


Figure 4-4: Segment Flow Control Shown in One Direction on a Logical Link



Chapter 5
The Session Control Layer

The Session Control layer, residing immediately above the End Communication layer, provides system-dependent, process-to-process communication functions. These functions bridge the gap between the End Communication layer and the logical link functions required by processes running under an operating system.

5.1 Session Control Functional Description

Session Control functions include:

- **Mapping node names to node addresses.** A Session Control module maintains a node name mapping table that defines the correspondence between a node name and either a node address or an adjacency. (An adjacency is used for loopback testing under the control of Network Management.) The table enables the Session Control module to select the destination node address or channel number for outgoing connect requests to the End Communication Layer. For incoming connect requests from the End Communication Layer, the Session Control module uses the table to identify the node from which the request originated.
- **Identifying end users.** A Session Control module executes a system dependent algorithm to determine if an existing end user process corresponds to the destination end user process specified in an incoming connect request. It also performs additional functions related to passing a connect request to an existing end user process.
- **Activating or creating processes.** A Session Control module may create a new process or activate an existing process to handle an incoming connect request.
- **Validating incoming connect requests.** A Session Control module uses access control information included in a incoming connect request to perform system-dependent validation functions.

Session Control defines two data bases that are not implementation-specific:

- A node-name mapping data base.
- A data base containing the states of Session Control and optional default connection timers.

Figure 5-1 shows a model of Session Control operating within a network node.

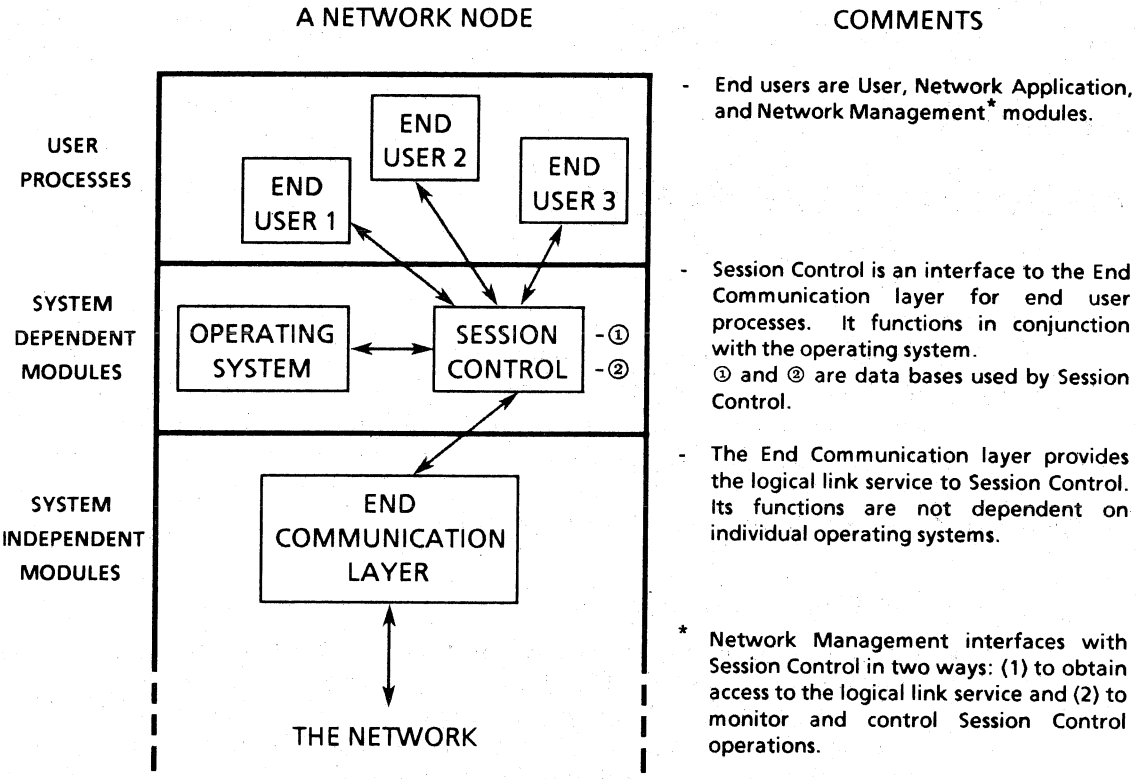


Figure 5-1: A Session Control Model

5.2 Session Control Messages

Session Control's message protocol defines messages sent on a logical link as connect data, reject data, and disconnect data.

Figure 5-2 shows the formats for the Session Control messages. The numbers below each message field indicate its maximum length in bytes.

Connect Data Message Format

DSTNAME	SRCNAME	MENUVER	RQSTRID	PASSWRD	ACCOUNT	USRDATA
19	19	1	39	39	39	39

Reject/Disconnect Data Message Format

REASON	DATACTL
1	n

DSTNAME = the destination end user name
SRCNAME = the source end user name
MENUVER = the field format and version format
RQSTRID = the source user identification for access verification
PASSWRD = the access verification password
ACCOUNT = the link or service account data
USRDATA = the end user process connect data
REASON = a reason code
DATACTL = user data (length of field determined by the total length of reject or disconnect data received from the End Communication layer)

Figure 5-2: Session Control Message Formats

5.3 Session Control Operation

Session Control performs the following basic operations:

- Requests logical links on behalf of end user processes (Section 5.3.1).
- Receives connect requests addressed to end user processes (Section 5.3.2).
- Sends and receives logical link data (Section 5.3.3).
- Disconnects and aborts logical links (Section 5.3.4).
- Optionally, monitors logical links (Section 5.3.5).

5.3.1 Requesting a Connection

Upon receipt of a logical link connect request from an end user process, Session Control performs four tasks:

- Identifies the destination node address or channel number for the End Communication layer by using the node name mapping table.
- Formats connect data for the End Communication layer.
- Issues a connect request to the End Communication layer.
- Optionally starts an outgoing connection timer. Expiration of the timer prior to an acceptance or rejection from the End Communication layer causes Session Control to disconnect the logical link for the source end user process.

5.3.2 Receiving a Connect Request

Upon detection of an incoming connect request from the End Communication layer, Session Control performs six tasks:

- Parses connect data to obtain such information as source and destination end user process names and access control information.
- Validates any access control information.
- Identifies, creates, or activates a destination end user process.
- Maps the source node's address or channel number to a node name, if there is one.
- Delivers the incoming connect request to the end user process.
- Optionally, starts an incoming timer when the connect request is delivered. Expiration of the timer before the end user process accepts the connect request causes Session Control to issue a reject to the End Communication layer.

5.3.3 Sending and Receiving Data

This is a system-dependent function that handles end user process requests to send and receive data. Basically, the functions are passed directly to the End Communication layer.

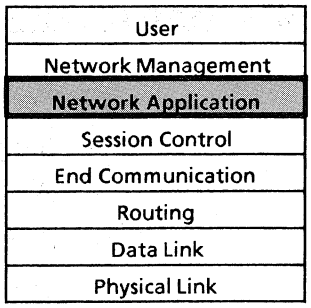
5.3.4 Disconnecting and Aborting a Logical Link

As in the case of sending and receiving data, Session Control passes end user process disconnect and abort requests directly to the End Communication layer. Similarly, notification of a logical link disconnect or abort is passed directly to the end user process.

5.3.5 Monitoring a Logical Link

This is an optional system-dependent function that may be used for the following purposes:

- Detecting probable network disconnection between the nodes at either end of the logical link.
- Detecting a failure, by the End Communication layer, to deliver transmitted data in a timely manner.



Chapter 6

The Network Application Layer

The Network Application Layer contains a number of independent, commonly used modules. These modules access data or provide communication services to users. Currently, five Phase IV DIGITAL-supplied protocols are specified for this layer:

- **The Data Access Protocol (DAP).** This protocol permits remote file access and file transfer in a manner that is independent of the I/O structure of the operating system being accessed. Section 6.1 describes DAP.
- **The Network Virtual Terminal Protocols.** These protocols permit terminals connected locally to a host DECnet system or to a Terminal Concentrator system to access remote DECnet host systems in order to issue interactive commands and run application programs. Section 6.2 describes Network Virtual Terminals.
- **The X.25 Gateway Access Protocol.** This protocol permits user-written modules in a host DECnet system to communicate with peer modules in a non-DECnet system across an X.25-based Public Packet Switching Network. The user module does not have to reside in the same DECnet system that is directly connected to the X.25 Network (the DTE - Data Terminal Equipment). Section 6.3 describes X.25 Gateway Access.
- **The SNA Gateway Access Protocol.** This protocol permits user-written modules (and a number of DIGITAL-supplied modules) in a host DECnet system to communicate with IBM host application programs and application subsystems in a Systems Network Architecture (SNA) network. The user modules do not have to reside in the same DECnet system that is directly connected to the SNA network (the Gateway system). Section 6.4 describes SNA Gateway Access.
- **The Loopback Mirror Protocol.** This protocol consists of one message used between the Network Management loopback access routines and loopback mirror. These modules test logical links. Section 7.2.3 describes this operation.

6.1 DAP Functional Description

DAP provides the following functions and features:

- Supports heterogeneous file systems.
- Retrieves a file from an input device (a disk file, a card reader, a terminal, etc.).
- Stores a file on an output device (a magtape, a line printer, a terminal, etc.).
- Transfers files between nodes.
- Supports deletion and renaming of remote files.
- Lists directories of remote files.
- Recovers from transient errors and reports fatal errors to the user.
- Allows multiple data streams to be sent over a logical link.
- Submits and executes remote command files.
- Permits sequential, random, and indexed (ISAM) access of records.
- Supports sequential, relative, and indexed file organizations.
- Supports wildcard file specification for sequential file retrieval, file deletion, file renaming, and command file execution.
- Permits an optional file checksum to ensure file integrity.

DAP is designed to minimize protocol overhead. For example, defaults are specified for fields wherever possible. In addition, a file transfer mode eliminates the need for DAP control messages once file data flow begins. Finally, relatively small file records can be blocked together and sent as one large message.

DAP is a set of messages and the rules governing their exchange between two cooperating processes. Section 6.1.1 describes DAP messages. Section 6.1.2 summarizes the operation of the cooperating DAP-speaking processes, which provide DECnet remote file access. Section 6.1.3 describes the most common DAP-speaking DECnet facilities.

6.1.1 DAP Messages

DAP-speaking processes use the messages listed in Table 6-1 to accomplish remote file access and transfer.

TABLE 6-1: DAP Messages

Message	Function
Configuration	Exchanges system capability and configuration information between DAP-speaking processes. Sent immediately after a logical link is established, this message contains information about the operating system, the file system, protocol version, and buffering capability.
Attributes	Provides information on how data is structured in the file being accessed. The message contains information on file organization, data type, format, record attributes, record length, size, and device characteristics.
Access	Specifies the file name and type of access requested.
Control	Sends Control information to a file system and establishes data streams.
Continue-Transfer	Allows recovery from errors. Used for retry, skip, and abort after an error is reported.
Acknowledge	Acknowledges access commands and Control messages used to establish data streams.
Access Complete	Denotes termination of access.
Data	Transfers file data over the logical link
Status	Returns status and information on error conditions
Key Definition Attributes Extension	Specifies key definitions for indexed files.
Allocation Attributes Extension	Specifies the character of the allocation when creating or explicitly extending a file.
Summary Attributes Extension	Returns summary information about a file.
Date and Time Attributes Extension	Specifies time-related information about a file.
Protection Attributes Extension	Specifies file protection codes.
Name	Sends name information when renaming a file or obtaining file directory data.

6.1.2 DAP Operation

Two cooperating processes exchange DAP messages: the user process and the server process that acts on the user's behalf at the remote node. User I/O commands accessing a remote file are mapped into equivalent DAP messages and transmitted via a logical link to the server at the remote node. The server interprets the DAP commands and actually performs the file I/O for the user. The server returns status and data to the user.

In a typical DAP dialog, the first message exchange is of Configuration messages that provide information about the operating and file systems, buffer size, and so on. Attributes messages then supply information about the file. The Access Request message typically follows to open a particular file. If data is to be transferred, a data

stream is then set up. For both sequential and random access file transfer, one control message sets up the data stream. After the completion of the file transfer, Access Complete messages terminate the data stream.

Figure 6-1 shows a DAP message exchange for sequential file retrieval.

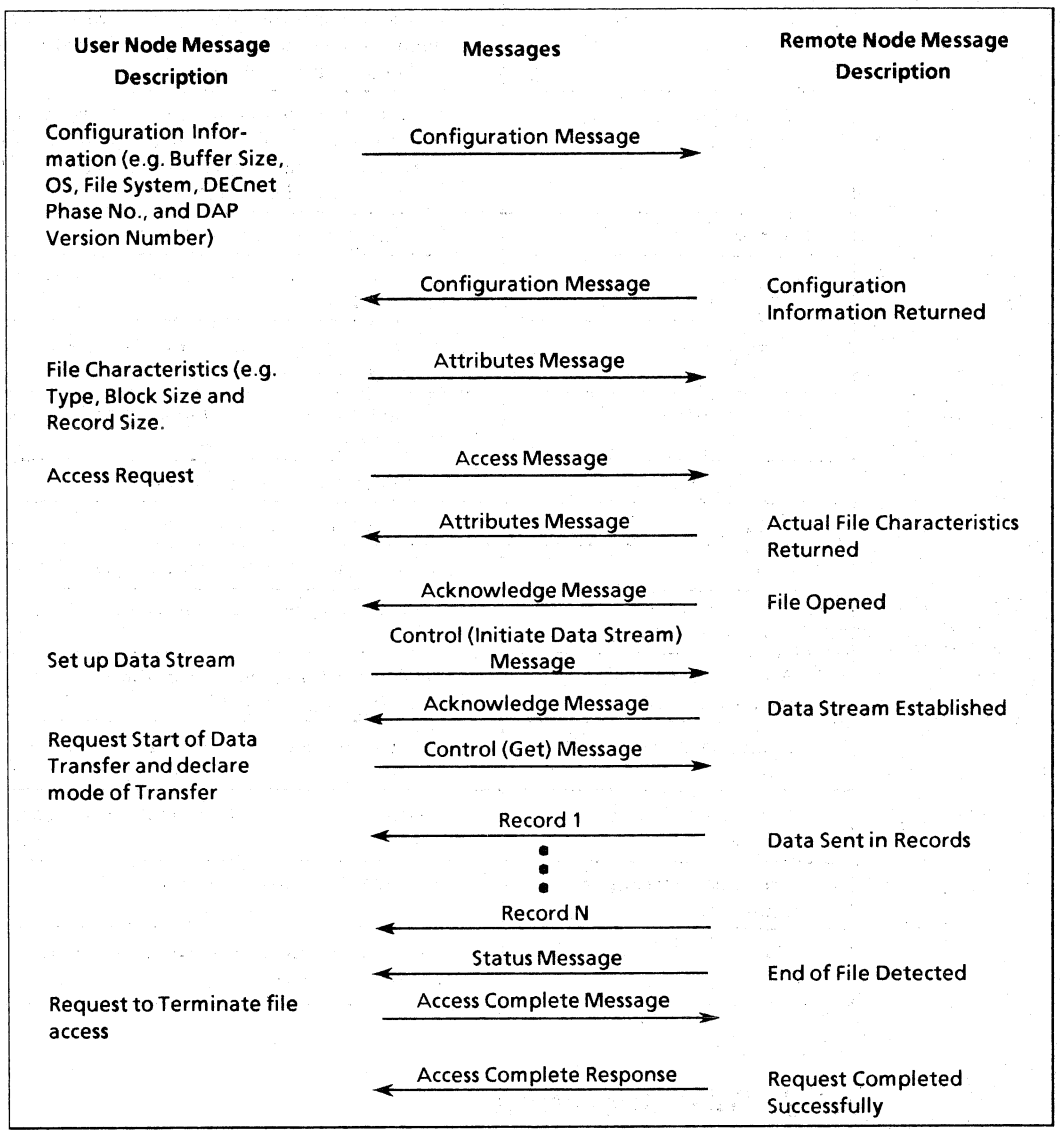


Figure 6-1: DAP Message Exchange (Sequential File Retrieval)

6.1.3 DECnet Remote File Access Facilities

DECnet implementations currently access remote files using the following facilities:

- **File Access Listener (FAL).** FAL receives user I/O requests at the remote node and acts on the user's behalf. This is the remote DAP-speaking server process.
- **Network File Transfer (NFT).** This interactive utility operates at the user level. It interfaces to a DAP-speaking accessing process to provide DAP functions. NFT provides network-wide file transfer and manipulation services..
- **Record Management Services (RMS).** RMS is the standard file system for many of DIGITAL's operating systems. In particular, DECnet-VAX transparently supports RMS. RMS can transmit and receive DAP messages over logical links. If an RMS file access request includes a node name in the file specification, RMS maps the access request into equivalent DAP messages. These DAP messages are sent to a remote FAL to complete the request. To the user, remote file access is handled identically to local file access, except that a remote node name and possibly access control information is necessary for remote file access.
- **Network File Access Routines (NFARs).** NFARs are a set of FORTRAN-callable subroutines. NFARs become a part of the user process; they cooperate with FAL, using DAP, to access remote files for user applications. RSX DECnet uses NFARs to provide DAP functions.
- **VAX/VMS Command Language Interpreter.** VMS commands pertaining to file access and manipulation interface with RMS to provide network-wide file access; hence, no separate NFT utility is required in VAX/VMS.
- **Network Management modules.** Network Management modules use DAP services to obtain remote files for down-line loading other remote nodes and to transfer up-line dumps for storage.

Figure 6-2 shows node-to-node file transfer using DAP.

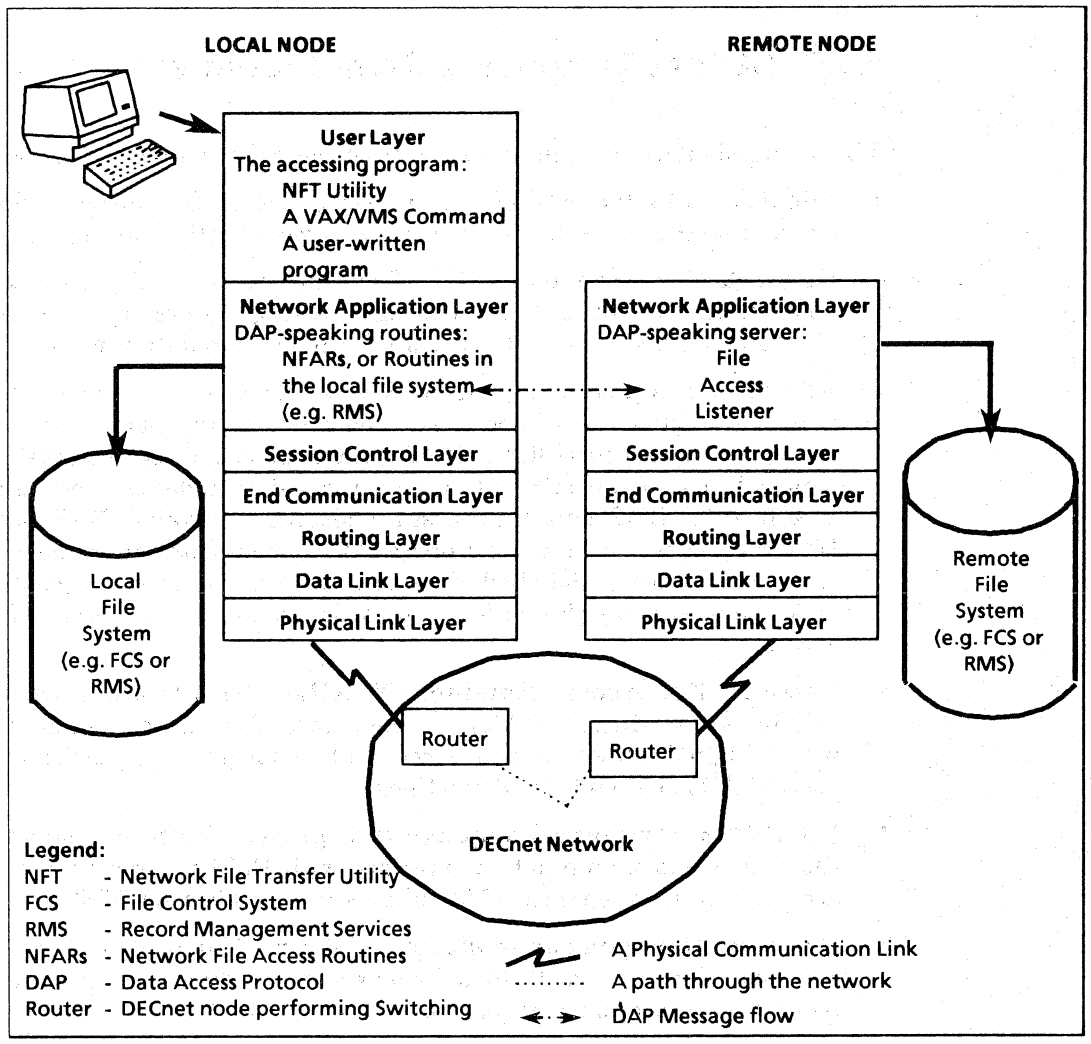


Figure 6-2: File Transfer Across a Network

6.2 Network Virtual Terminal Functional Description

The Network Virtual Terminal (NVT) service consists of the following:

- A functional model of a terminal (virtual terminal) for access through the network. This model is known as the network command terminal.
- A set of protocols for communication between a host node, with an application program running, and a server node, with a terminal directly attached.

The NVT is associated with the Terminal Software Architecture, which also defines nonnetwork aspects of terminal services.

The Network Virtual Terminal service provides these functions and features:

- Distributes terminal-handling functions between two systems.
- Supports heterogeneous host systems. Different operating systems can cooperate via common protocols, and a host operating system can manage a terminal in its own way, regardless of what operating system runs in the server.
- Allows a server to connect to a specified host or a host to a specified remote terminal.
- Provides terminal input/output and characteristics management functions at the operating system services level. The description of the command terminal protocol includes a list of these functions (Section 6.2.2).
- Offers standard terminal services, featuring good performance and device independence. Optionally, offers methods of controlling the terminal's behavior in considerable detail.
- Supports high availability implementations. The protocols contain functions that allow the restart of interrupted communication.

The virtual terminal protocols are organized into two sublayers of the Network Application layer, as a provision for future enhancement. Figure 6-3 depicts the organization of the Network Virtual Terminal service.

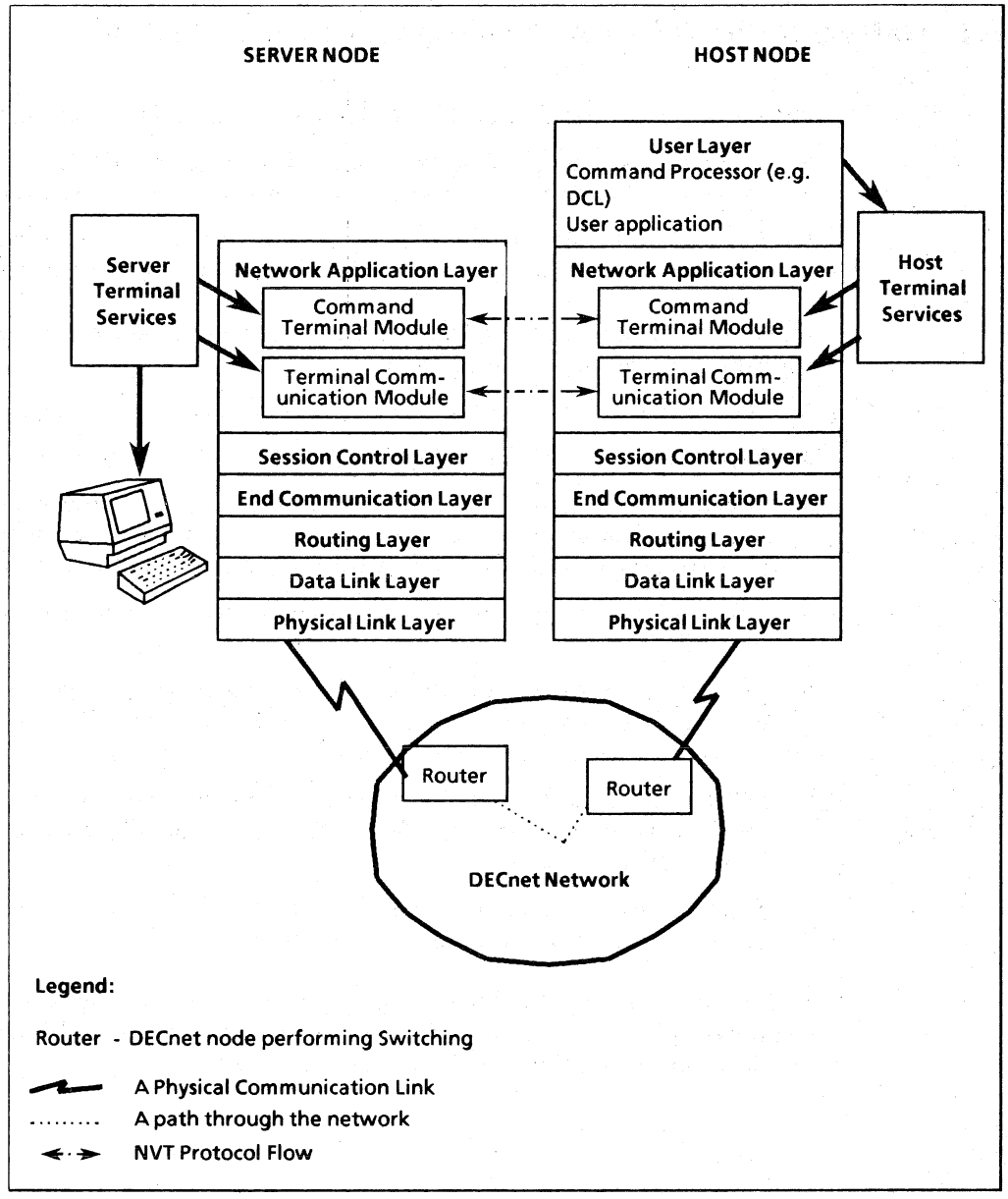


Figure 6-3: Network Virtual Terminal Service

Section 6.2.1 describes the Terminal Communication Protocol, which controls the connections between applications and terminals. Section 6.2.2 describes the Command Terminal Protocol, which provides functions for terminals to communicate with Command Language Processors and application programs. Section 6.2.3 describes the operation of the Network Virtual Terminal service.

6.2.1 The Terminal Communication Protocol

The base protocol for the Network Virtual Terminal service is the Terminal Communication Protocol. It is in the lower of the two sublayers of the Network Virtual Terminal service, known as the *foundation layer*.

The Terminal Communication Protocol has responsibility for making and breaking connections between applications and terminals. In effect, it extends DNA Session Control layer services by establishing logical links between endpoints that are specific to terminal services. The endpoint in the host system is called a *portal*; the other end, in the server system, is called a *logical terminal*.

A portal corresponds to a remote terminal identifier in the host, and a connection *binds* that identifier to an actual terminal. Thus, the NVT connection is called a *binding*.

In the future, it may be desirable to add new models, as alternatives to the network command terminal. Such new models might require their own protocols. To provide for this, the logical terminal, binding, and portal share a state called a *mode*. Through *mode management*, the terminal communication protocol is able to select the appropriate higher level Virtual Terminal protocol and model.

Table 6-2 shows the Terminal Communication Protocol messages types and their functions.

TABLE 6-2: Terminal Communication Protocol Messages

Message	Function
Bind Request	Requests a Binding; identifies version and type of sending system.
Rebind Request	Requests a rebinding (reestablishes broken communications, for high availability implementations).
Unbind	Requests that a binding be released.
Bind Accept	Accepts a Bind request.
Enter Mode	Requests entry of a new mode (the only mode currently defined is command mode). This selects the command terminal protocol as the higher level protocol.
Exit Mode	Requests that the current mode be exited.
Confirm Mode	Confirms the entry of a new mode.
No Mode	Indicates that the requested mode is not available or confirms an exit mode request.
Data	Carries Data (i.e. command terminal protocol information).

6.2.2 The Command Terminal Protocol

The Command Terminal Protocol provides access to the functions of the network command terminal model.

The network command terminal offers a set of functions oriented mainly toward command line input and output, but general enough to support a broad range of video and hardcopy terminal applications. These functions include:

- Read a line of input, with echoing and operator editing performed by the server system. Line terminators may be individually specified. The host software has considerable control over echoing and operator editing, including the enabling and disabling of each feature individually.

- Accept input even if the program has not issued a read request; save it until a read request arrives (*typeahead*).
- Input, or take action in response to, certain characters immediately as the keys are struck (*out-of-band* character processing). These characters may be individually specified.
- Write a string of output characters. Writing may proceed concurrently with reading, subject to certain synchronization options. The terminal operator can cause output to be discarded.
- Recognize ANSI standard escape sequences on input and output.
- Cancel a read request.
- Read and set terminal device characteristics.
- Determine how many characters are in the typeahead and input buffers combined.
- Clear the typeahead and input buffers.

Table 6-3 shows the Command Terminal Protocol message types and their functions.

TABLE 6-3: Command Terminal Protocol Messages

Message	Function
Initiate	Carries initialization information, as well as protocol and implementation version numbers.
Start Read	Requests that a READ be issued to the terminal.
Read Data	Carries input data from terminal on completion of a read request.
Out-of-Band	Carries out-of-band input data.
Unread	Cancels a prior read request.
Clear Input	Requests that the input and typeahead buffers be cleared.
Write	Requests the output of data to the terminal.
Write Complete	Carries write completion status.
Discard State	Carries a change to the output discard state due to a terminal operator request (via an entered output-discard character).
Read Characteristics	Requests terminal characteristics.
Characteristics	Carries terminal characteristics.
Check Input	Requests input count (number of characters in the typeahead and input buffers combined).
Input Count	Carries input count as requested with Check Input.
Input State	Indicates a change from zero to non-zero or vice-versa in the number of characters in the input and typeahead buffers combined.

6.2.3 NVT Operation

In a typical network command terminal session, a terminal management module in a server system with an active terminal requests a binding to some host system. It does this by invoking a terminal communication services function. (This typically takes place in response to a command such as Set Host from the terminal operator.)

The terminal communication services module initiates a logical link to its counterpart in the host system, using DNA Session Control layer services. On discovering the incoming logical link, the host module allocates a portal, thus beginning the formation of a binding. The host module then accepts the logical link. Once the logical link has been formed, the server module sends a Bind Request message to the host. A terminal management module in the host, perhaps acting on behalf of the host's login process, recognizes the binding request, and causes the terminal communication services module to send a Bind Accept message.

Once the binding has been formed, the host takes the initiative and readies the binding for the command terminal protocol (enters command mode) with a further exchange of messages. This takes place in connection with the first terminal I/O request from the Login process. The respective protocol modules can now begin speaking the command terminal protocol, by each sending an Initiate message to the other. After initialization, the dialog of remote terminal service requests and responses ensues.

Terminal service requests normally originate with an application program in the host system. The application program issues requests to host operating system terminal services. Host terminal services issue corresponding requests to the host protocol module. The server protocol module reproduces those requests remotely and reissues them to the server terminal services.

Termination of the session can come about in a number of ways. A typical orderly shutdown begins with a Logout by the application in the host. As a result, the host terminal communication services module sends an Unbind request. The server responds by releasing its resources. Finally, the host disconnects the logical link.

Figure 6-4 shows NVT message exchange .

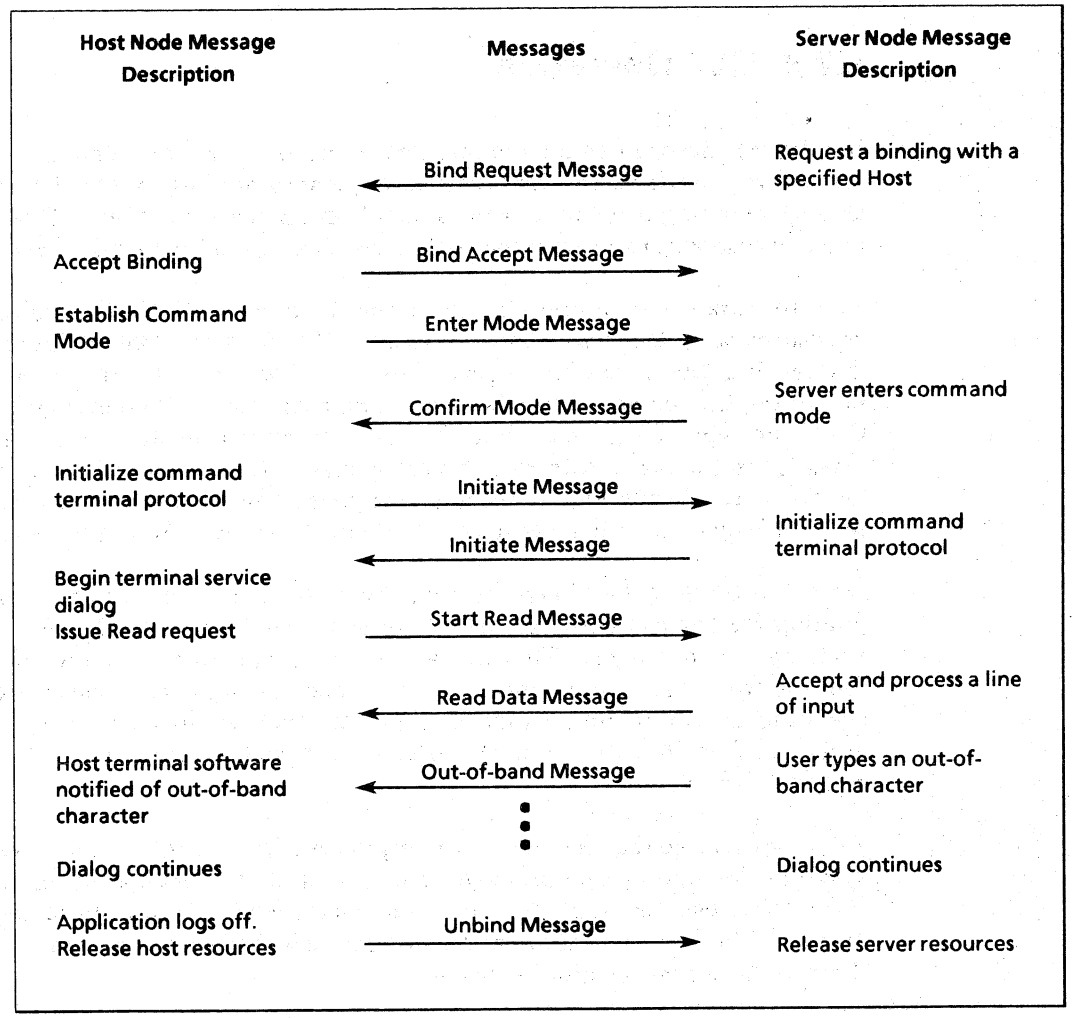


Figure 6-4: NVT Protocol Message Exchange

6.3 X.25 Gateway Access Functional Description

X.25 Gateway Access provides the following functions and features:

- Supports communication between user-written programs in a host DECnet system and modules in a non-DECnet system over an X.25-based public data network. (Two DECnet systems may communicate over an X.25 network using standard DNA protocols without recourse to this gateway function. See Section 3.3.4.3 for a description of how this is accomplished.)
- Supports all facilities and modes of operation defined in CCITT Recommendation X.25, with the exception of *Datagram* mode and the *Delivery Confirmation* bit.
- Supports user access to both Permanent Virtual Circuits (PVCs) and Switched Virtual Circuits (SVCs). Switched Virtual Circuits are also referred to as virtual calls.
- Permits user-written programs to access PVCs or issue virtual calls from any DECnet system containing an X.25 Gateway Access module. The user program does not have to reside in the DECnet system that is the DTE (Data Terminal Equipment) directly connected to the X.25 network.
- Permits incoming virtual calls to be directed to a user process residing at any node in the DECnet network. A Network Management function maps call data from incoming virtual calls to a DECnet process description so that the call may be directed to the correct process.
- Permits user access to all X.25 packet fields such as *more data, qualified data,* etc.
- Recovers from transient errors, and reports fatal errors to the user.

X.25 Gateway Access is designed to make the full capabilities of the CCITT X.25 Packet Level interface available to user programs residing anywhere in a DECnet network. To accomplish this, X.25 Gateway Access communicates with the DECnet system that is the DTE on the X.25 network over a DECnet logical link (the DTE system is often called the *Gateway system*). X.25 Gateway Access protocol messages are exchanged over the logical link to make the facilities of the Packet Level interface available remotely.

Section 6.3.1 describes the messages exchanged by the X.25 Gateway Access Protocol, Section 6.3.2 describes the operation of X.25 Gateway Access, and Section 6.3.3 describes the modules that cooperate to provide X.25 Gateway Access.

6.3.1 X.25 Gateway Access Messages

X.25 Gateway Access uses the messages listed in Table 6-4 to accomplish remote access to the X.25 network.

TABLE 6-4: X.25 Gateway Access Messages

Message	Function
Open	Requests the use of a permanent virtual circuit.
Open Accept	Accepts a permanent virtual circuit Open request and allocates the PVC to the Gateway Access user.
Open Reject	Rejects a Permanent Virtual Circuit Open request, refusing allocation of the PVC to the Gateway Access user.
Outgoing Call	Requests that an outgoing Virtual Call be placed by issuing a Call Request Packet to the X.25 network.
Call Reject	Indicates rejection of an outgoing or incoming virtual call for lack of resources.
Incoming Accept	Indicates acceptance of an outgoing virtual call.
Incoming Call	Indicates an incoming virtual call has been received.
Outgoing Accept	Accepts a previously received incoming virtual call.
Clear Request	Requests that a virtual call be Cleared.
Clear Indication	Indicates a clear request packet has been received from the X.25 network.
Clear Confirm	Indicates a clear confirmation packet has been received from the X.25 network.
Reset Request	Requests that a virtual circuit be reset.
Reset Indication	Indicates that a reset indication packet has been received from the X.25 network.
Reset Confirmation	Indicates a reset confirmation by either the user or the X.25 network.
Reset Marker	Marks the place in the stream of Data messages where a reset occurred.
Data	Contains Data packets outbound from or inbound to a user.
Interrupt	Contains Interrupt packet data outbound from or inbound to a user.
Interrupt Confirmation	Confirms the receipt of an Interrupt message (used for flow control).
No Com	Indicates that a PVC entered a no-communication state.
No Com Seen	Indicates that the user of a PVC acknowledges a No Com message, and thus has been notified that there have been one or more failures of the X.25 network (e.g. Restarts).

6.3.2 X.25 Gateway Access Operation

Three processes are involved in X.25 Gateway operation: the user process, the X.25 Gateway Server process, and the foreign X.25 DTE process. The user process accesses the X.25 Packet level functions by issuing calls on the X.25 Gateway Access module. These calls are translated into X.25 Gateway Access protocol messages and transmitted over a DECnet logical link to the X.25 Gateway Server process. The X.25 Gateway Server transmits and receives X.25 packets to and from the X.25 network. These packets in turn flow over an X.25 virtual circuit to the foreign X.25 DTE process.

In a typical X.25 Gateway Access Protocol dialogue, the first message exchange is either to open a Permanent Virtual Circuit via an Open message or to place a virtual call using an Outgoing Call message. In the case of a virtual call, when the foreign DTE process has accepted the call by issuing a Call Accept packet, the user process is informed by the receipt of an Incoming Accept message. The user process may then exchange data with the foreign DTE process by sending and receiving Data and Interrupt messages. When the user wishes to terminate the virtual call, a Clear Request message is sent to the X.25 Gateway Server, which clears the call by issuing a Clear request packet to the X.25 network. The user process is then informed of the termination of the call by receipt of a Clear Confirm message.

Figure 6-5 shows X.25 Gateway Access protocol exchange for Switched Virtual Circuit operation, where the virtual call is initiated by the user process on the DECnet system.

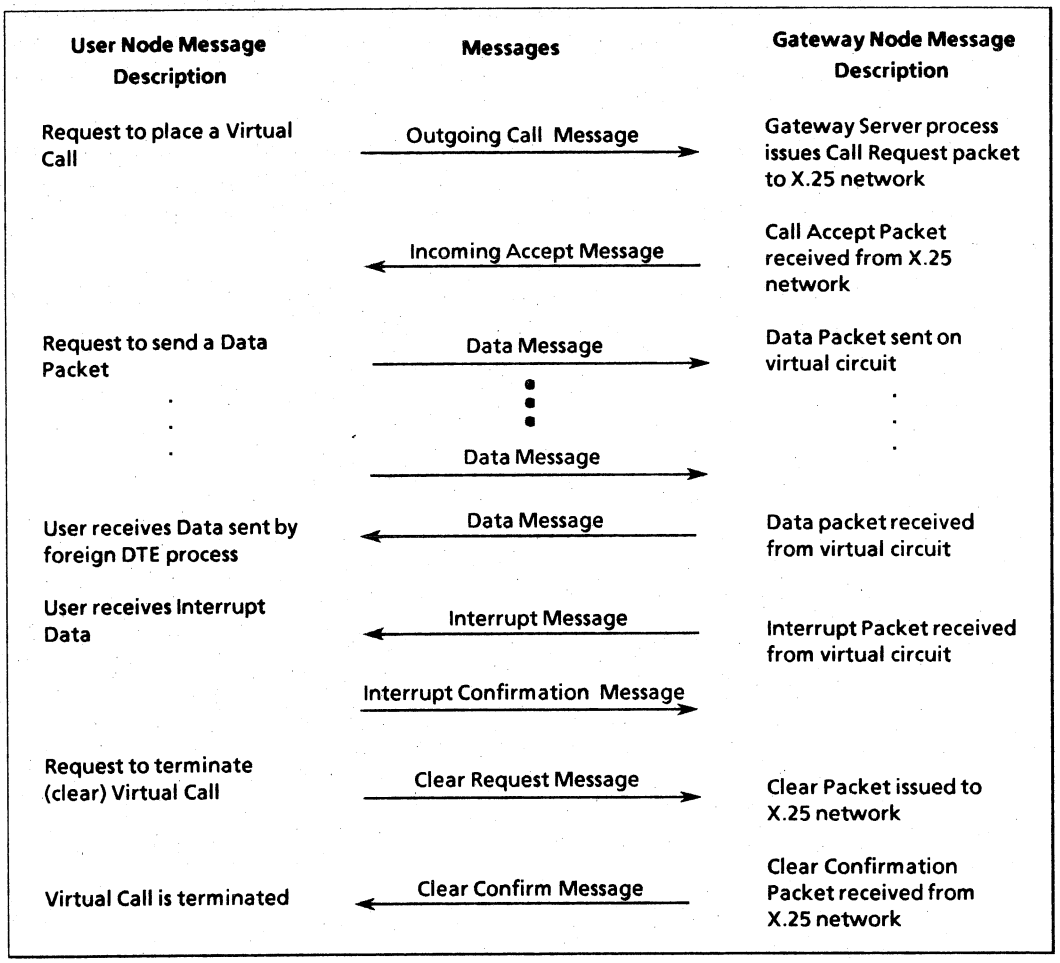


Figure 6-5: X.25 Gateway Access Message Exchange

6.3.3 X.25 Gateway Access Modules

DECnet implementations use the following facilities to provide the X.25 Gateway Access service:

- **X.25 Gateway Access Module.** This module receives user X.25 requests and communicates with the DECnet system that is the DTE directly connected to the X.25 public data Network (the Gateway system).
- **X.25 Gateway Server Module.** This module resides in the DECnet system that is the DTE on the X.25 network (the Gateway system). This module communicates with the X.25 Gateway Access module over a DECnet logical link and acts on requests made by the X.25 Gateway Access module and the X.25 network. It uses the facilities of the m.25 Packet Level Module to gain access to the X.25 network.
- **X.25 Packet Level Module.** This module resides in the Data Link layer of the Gateway system and acts as the DTE on the X.25 network. It uses the X.25 Frame level module for its connection to the X.25 DCE. This module is described in Section 2.3.2.
- **X.25 Frame Level Module.** This module also resides in the Data Link layer of the Gateway system. It provides the X.25 Frame level protocol in order to communicate with the X.25 network's DCE. This module is described in Section 2.3.1.

Figure 6-6 depicts the relationship of these modules to the X.25 network and to the foreign DTE:

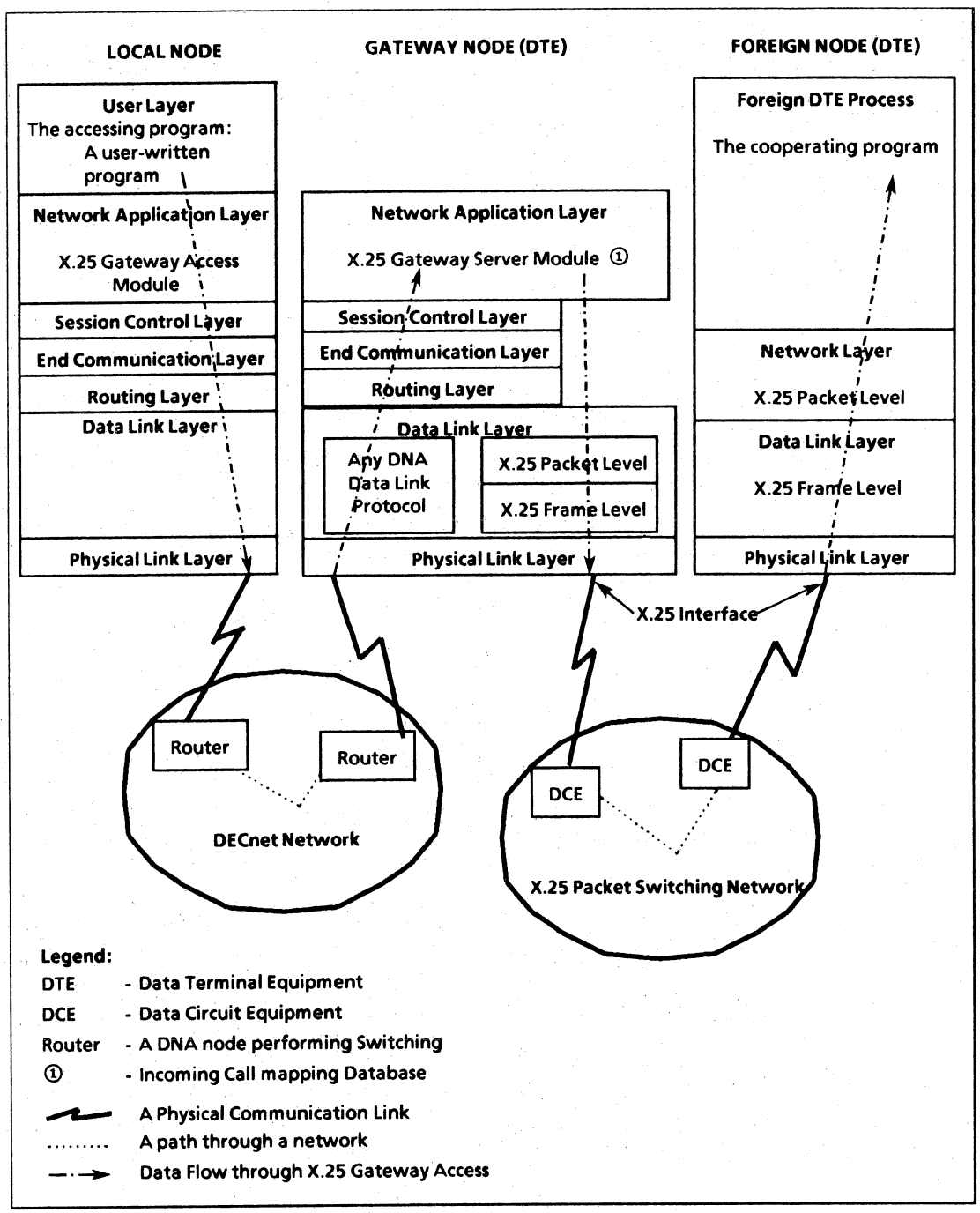


Figure 6-6: X.25 Gateway Access Operation

6.4 SNA (Systems Network Architecture) Gateway Access Functional Description

SNA Gateway Access provides the following functions and features:

- Supports communication between user-written (and a number of DIGITAL-written) programs in a host DECnet system and modules in an IBM host system over an SNA network.
- Supports programs in DECnet systems which act as SNA Secondary Logical Units (SLUs) of any of the defined SNA Logical Unit Types (LUs).
- Allows user programs full access to the facilities provided by the SNA Transmission Control and Data Flow Control layers.
- Communicates with the SNA network as a Physical Unit Type 2 (PU2).
- Permits user-written programs to participate in SNA *sessions* from any DECnet system containing an SNA Gateway Access module. The user program does not have to reside in the DECnet system that is acting as the PU2 directly connected to the SNA network.
- Manages the SNA session pacing automatically for the user program.
- Recovers from transient errors and reports fatal errors to the user

SNA Gateway Access is designed to make the full capabilities of the SNA Data Flow Control, Transmission Control, and Path Control layers available to user programs residing anywhere in a DECnet network. To accomplish this, SNA Gateway Access communicates with the DECnet system that is the PU2 on the SNA network over a DECnet logical link. SNA Gateway Access Protocol messages are exchanged over the logical link to make the facilities of SNA sessions available remotely.

Section 6.4.1 describes the messages exchanged by the SNA Gateway Access protocol, Section 6.4.2 describes the operation of SNA Gateway Access, and Section 6.4.3 describes the modules that cooperate to provide SNA Gateway Access.

6.4.1 SNA Gateway Access Protocol Messages.

SNA Gateway Access uses the messages listed in Table 6-5 to accomplish remote access to the SNA network.

TABLE 6-5: SNA Gateway Access Messages

Message	Function
Connect	Requests that a Session be established with a Primary Logical Unit (PLU) in an SNA host.
Listen	Waits for a Session to be established by a PLU.
Bind Data	Indicates that a BIND has been received for a session solicited with Connect or Listen.
Bind Accept	Requests that the session being established with the BIND be accepted and placed in the running state.
Normal Data	Carries data on the SNA session's normal flow to and from the PLU.
Interrupt Data	Carries data on the SNA session's expedited flow to and from the PLU.
Disconnect	Requests orderly session termination.
Disconnect Complete	Indicates normal session termination has completed successfully.
Abort	Requests abnormal termination of a session, or indicates an UNBIND has been received from the PLU.

6.4.2 SNA Gateway Access Operation

Three processes are involved in SNA Gateway Access operation: the user process, the SNA Gateway Server process, and the IBM host application process. (Note: IBM hosts are called PU5 nodes in SNA terminology.) The user process accesses the SNA SLU functions by issuing calls on the SNA Gateway Access Routines. These calls are translated into SNA Gateway Access protocol messages and transmitted over a DECnet logical link to the SNA Gateway Server process. The SNA Gateway Server transmits and receives SNA Basic Information Units (BIUs) by accessing the functions of an SNA protocol emulator, which connects directly to the SNA network as a PU2 and contains the functions of the SNA Path Control and Data Link Control (SDLC) layers.

In a typical SNA Gateway Access protocol dialog, the first message exchange is to establish a session with a PLU in the IBM host, by exchanging Connect and Bind Data messages. The session is completely established when the Bind Accept message is successfully sent. The user process may then exchange data with the PLU by sending and receiving Normal Data and Interrupt Data messages. When the user wishes to terminate the session, it informs the PLU by using the appropriate Data Flow Control Request Unit (RU). The PLU may then terminate the session by sending an UNBIND, which results in session termination with an appropriate reason code to the user process.

Figure 6-7 shows SNA Gateway Access Protocol exchange for a session requested by the user process (the SLU):

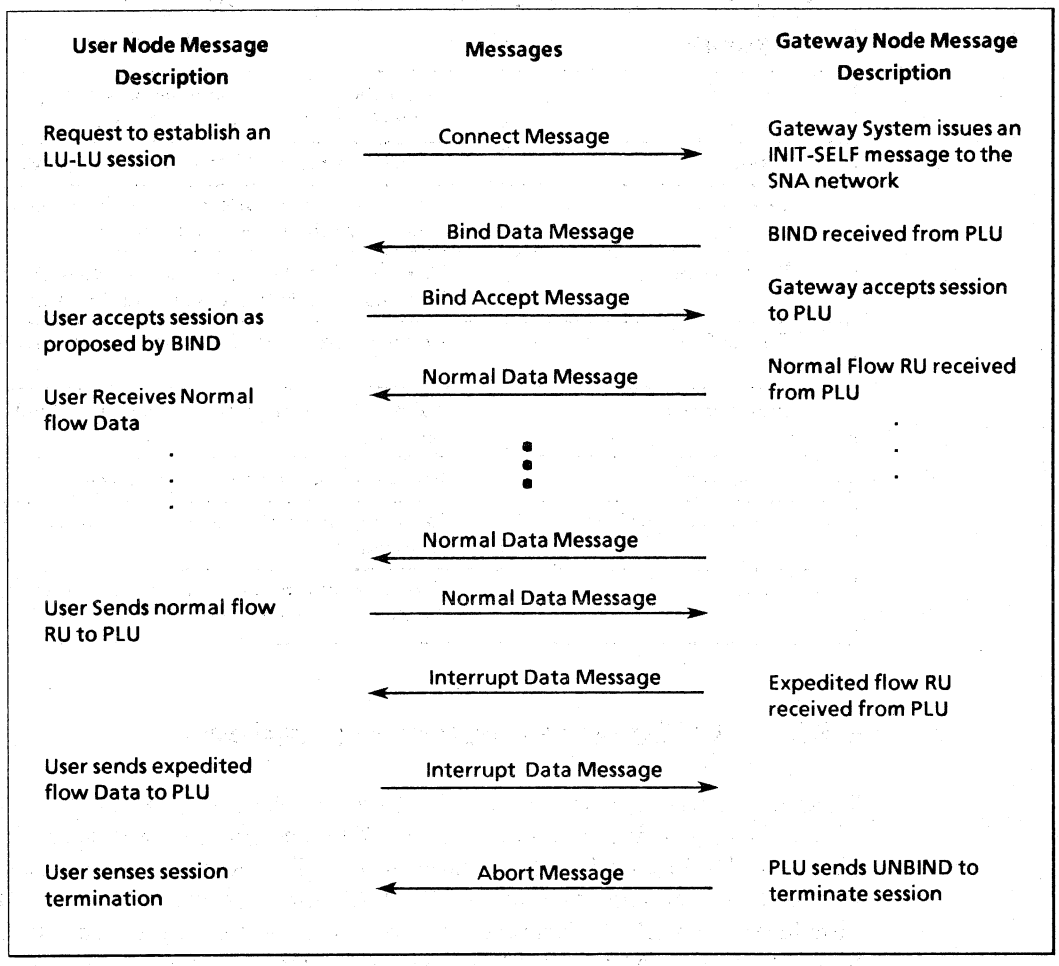


Figure 6-7: SNA Gateway Access Message Exchange

6.4.3 SNA Gateway Access Modules

DECnet implementations use the following facilities to provide the SNA Gateway Access service:

- **SNA Gateway Access Module.** This module receives user SNA session requests and communicates with the DECnet system that is the Physical Unit (PU) directly connected to the SNA network (the Gateway system).
- **SNA Gateway Server Module.** This module resides in the DECnet system that is the PU on the SNA network (the Gateway system). This module communicates with the SNA Gateway Access module over a DECnet logical link and acts on requests made by the SNA Gateway Access module and the PLU. It uses the facilities of an SNA protocol emulator to gain access to the SNA network.
- **SNA protocol emulation.** This set of modules resides in the Gateway system and performs the functions of SNA Path Control and Data Link Control. It connects to the SNA network as a Physical Unit Type 2 node. The SNA protocol emulation modules also perform the functions necessary to communicate with the SNA network's System Service Control Point (SSCP).

Figure 6-8 depicts the relationship of these modules to the SNA network and to the IBM host application (the PLU):

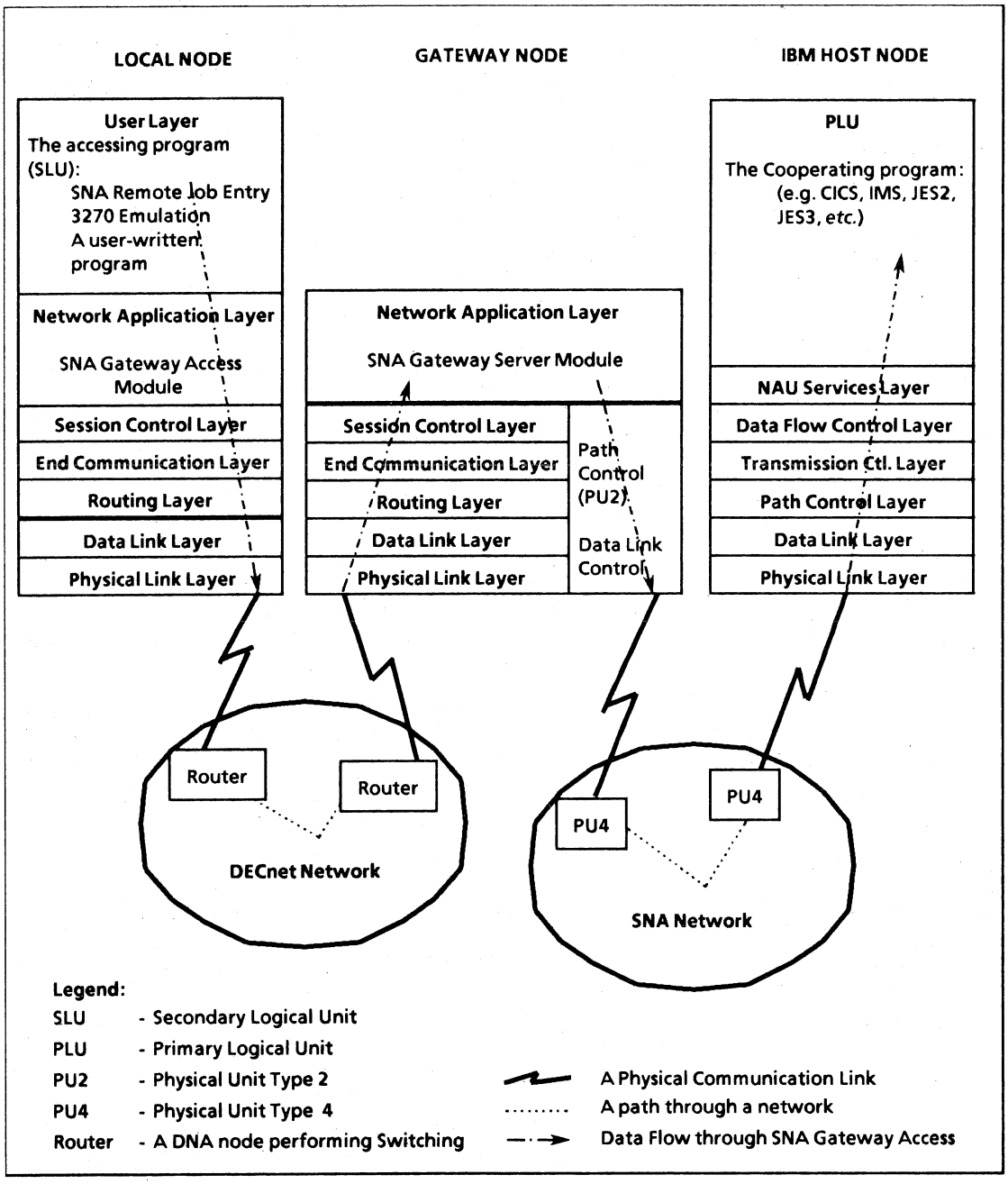
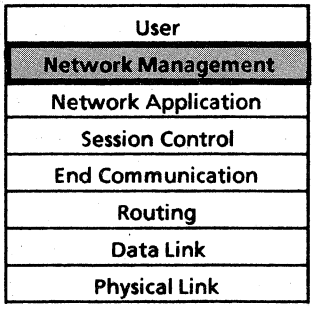


Figure 6-8: SNA Gateway Access Operation



Chapter 7
Network Management

Network Management allows system managers to control and monitor network operation. Network Management also provides information for use in planning the evolution of a network and correcting network problems.

The Network Management design has three outstanding characteristics:

- Both programs and terminals can access and control a DECnet network via a set of functionally discrete calls and commands.
- Control over a DECnet network can be either distributed or central. Distribution of control can be either partial or complete.
- Network Management is a set of primitive functions or "tools." The system or network manager can fashion them into a management system that meets his or her specific needs. This allows the managers of a network to construct their own network management philosophy.

Most of the Network Management modules reside in the Network Management layer. In addition to these, however, there are Network Management modules in the User and Network Application layers. Also, one Network Management module, the Event Logger, has a queue residing in each lower layer. This chapter discusses all Network Management modules, regardless of where they reside.

Since Network Management is modular, a DECnet system is not required to implement the architecture in its entirety.

7.1 Network Management Functional Description

Network Management performs the following functions:

- **Loading and dumping of remote systems.** For example, a system manager can down-line load an operating system into an unattended, remote system.

- **Changing and examining network parameters.** For example, an operator can change circuit costs or node names.
- **Examing network counters and events that indicate how the network is performing.** For example, the Event Logger automatically records significant network events.
- **Testing links at both the data link and logical link levels.** For example, a system manager can send a test message that loops back to its origin from a specific point in the hardware connection.
- **Setting and displaying the states of lines and nodes.** For example, an operator can reconfigure a network by turning nodes and links on or off.

7.2 Network Management Operation

This section summarizes the functions of each Network Management component and describes the operation of the major Network Management functions.

7.2.1 Components

The Network Management components are as follows:

User Layer

- **Network Control Program (NCP).** NCP is a utility at the user level that interfaces with lower level modules. NCP has a standard set of interactive terminal commands that each DECnet implementation uses.

Network Management Layer

- **Network Management Access Routines.** These routines provide generic Network Management functions. The routines communicate across logical links with the Network Management Listener using the Network Information and Control Exchange (NICE) protocol.
- **Network Management Listener.** The Network Management Listener receives Network Management commands from the Network Management level of remote nodes via the NICE protocol. In some implementations it also receives commands from the Network Management Access Routines via the NICE protocol. The Network Management Listener passes function requests to the Local Network Management Functions.
- **Local Network Management Functions.** This component takes function requests from the Access Routines, translating the requests into system-dependent calls.
- **Link Watcher.** Sensing service requests on a link from an adjacent node, the Link Watcher handles state changing for automatic remote load, dump, and trigger functions.

- **Maintenance Functions.** The Maintenance Functions provide the actual protocol operation to support system maintenance functions such as down-line load and data link loop testing.
- **Link Service Functions.** The higher level Network Management modules interface to the Link Service Functions for services that require a direct data link (bypassing the Session Control, End Communication, and Routing layers).
- **Event Logger.** The Event Logger records significant events occurring in the lower layers. DNA specifies event types in the Network Management interface to each layer. An event processor within the Event Logger takes *raw events* queued in each layer and records events of the types specified by the system and system manager. Using the Event Logger protocol, an event transmitter can inform event receivers at other nodes of event occurences. Events travel to a specified *sink node* for console, file or monitor output.

Figure 7-1 shows the Relationship among Network Management components in the User and Network Management layers.

Network Application Layer

- **Loopback Access Routines and Loopback Mirror.** For logical link loopback tests, the Local Network Management Functions can interface to the Loopback Access Routines, which use the Loopback Mirror protocol to loop test messages from a remote Loopback Mirror.

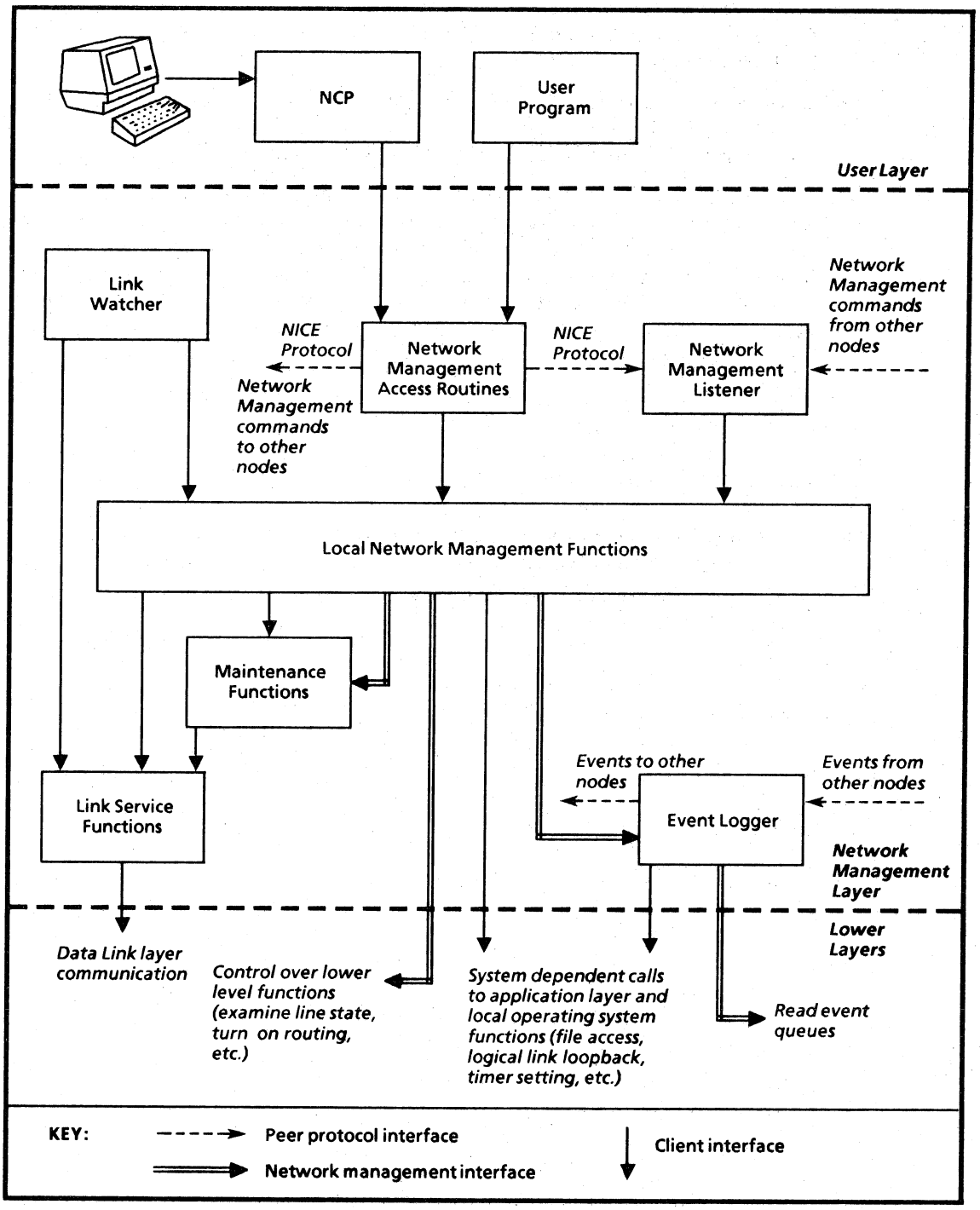


Figure 7-1: Relationship among Network Management Components at a Single Node

7.2.2 Remote Loading, Dumping, Controlling, and Link Loopback Testing Functions

Network Management uses the Maintenance Operation Protocol (MOP) to perform remote loading, dumping, controlling and testing. (Refer to Section 7.4). The *target* (the node being loaded, dumped, *etc.*) or the *executor* (the adjacent node executing the requests) or a remote *command* node can initiate the function. Figure 7-2 illustrates the down-line load request operation.

Link loopback testing is a procedure for isolating faults in a physical connection between two nodes. Link loopback tests consist of sending out test messages that are looped back at some point. By changing the loopback point, an operator can precisely locate problems.

There are three methods of performing link loopback tests:

1. Using NCP commands, the operator can set certain line devices to loopback mode, and then perform the loopback tests. This "automatic" method can be useful in testing unattended remote nodes.

2. To loop a message from another hardware point, such as a modem, the operator must set up a loopback device manually. This can be done by throwing a loopback switch on a modem or connecting a loopback plug in place of a modem. The operator can then execute the loopback test using NCP commands.

3. By not setting a hardware loopback device and using link loopback NCP commands, an operator can cause loopback test messages to loop back from the Maintenance Functions in the Network Management layer of the adjacent node.

Figure 7-3 illustrates link loopback tests.

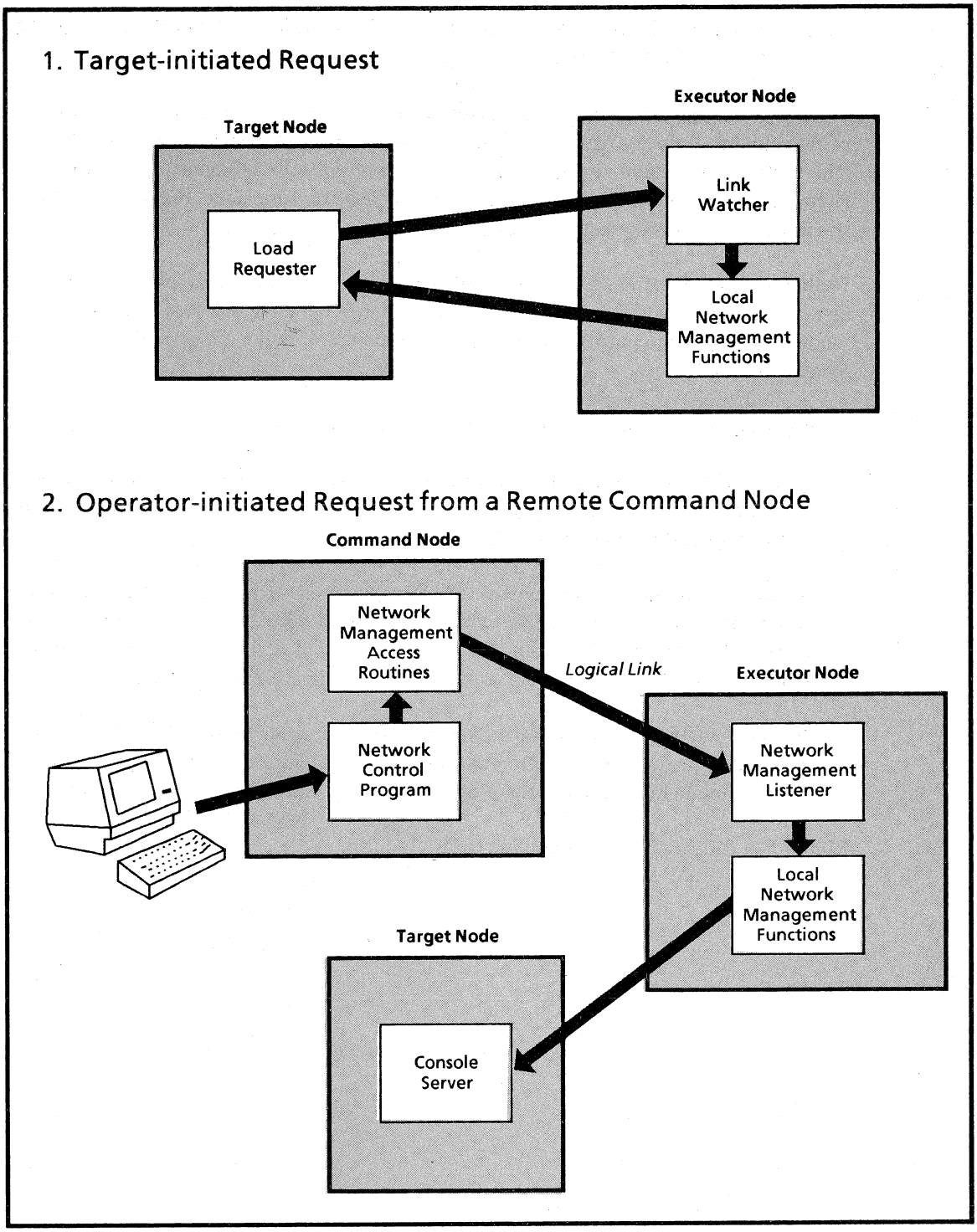


Figure 7-2: Down-Line Load Request Operation

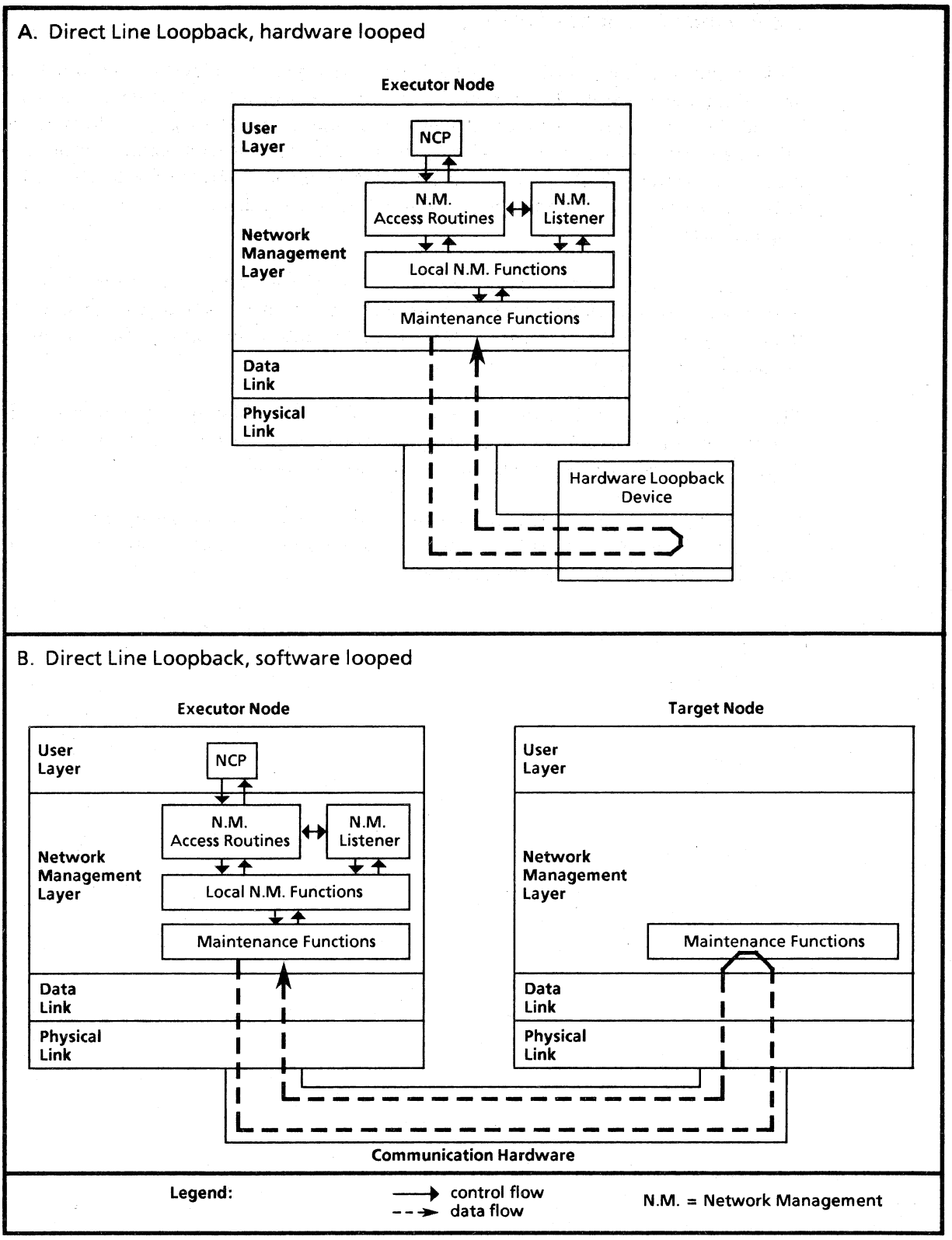


Figure 7-3: Line Loopback Tests

7.2.3 Node Loopback Testing

Node loopback testing consists of sending test messages over a logical link to be looped back at a specific point. The operator may set a hardware loopback device on a line, line device, or modem between the executor and adjacent target. Alternatively, software components can loop back test data. Different software components can be used. For example, FAL, a user program, or the Loopback Mirror can loop data back to their associated access routines. Figures 7-4 and 7-5 describe some types of node loopback tests.

One type of node loopback test uses the Network Management Loopback Access Routines and Loopback Mirror residing in the Network Application layer (Figure 7–4, B and Figure 7–5, C). These modules communicate over logical links using the Loopback Mirror Protocol.

Other loop tests use logical links, but do not use Network Management software. For example, Figure 7–5, B shows a test message travelling from one user task to another. Figure 7–5, D shows a file transfer used as a logical link test.

By using the NCP SET NODE LINE command, an operator can set up a special *loop node* path (Figure 7–4). In this case, if no hardware loopback device is set, the adjacent node's Routing layer loops back the test.

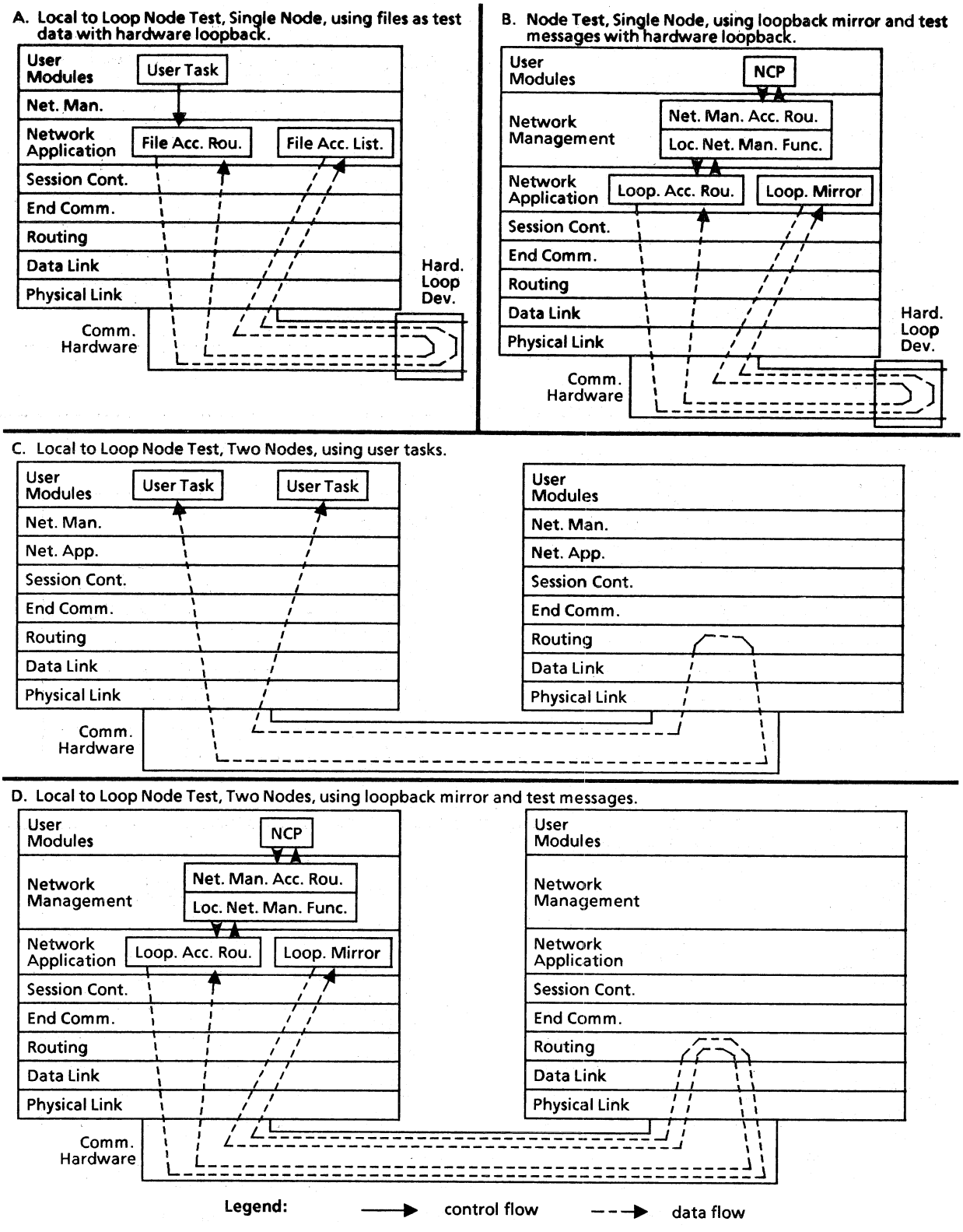


Figure 7-4: Examples of Node Level Testing Using a Loopback Node

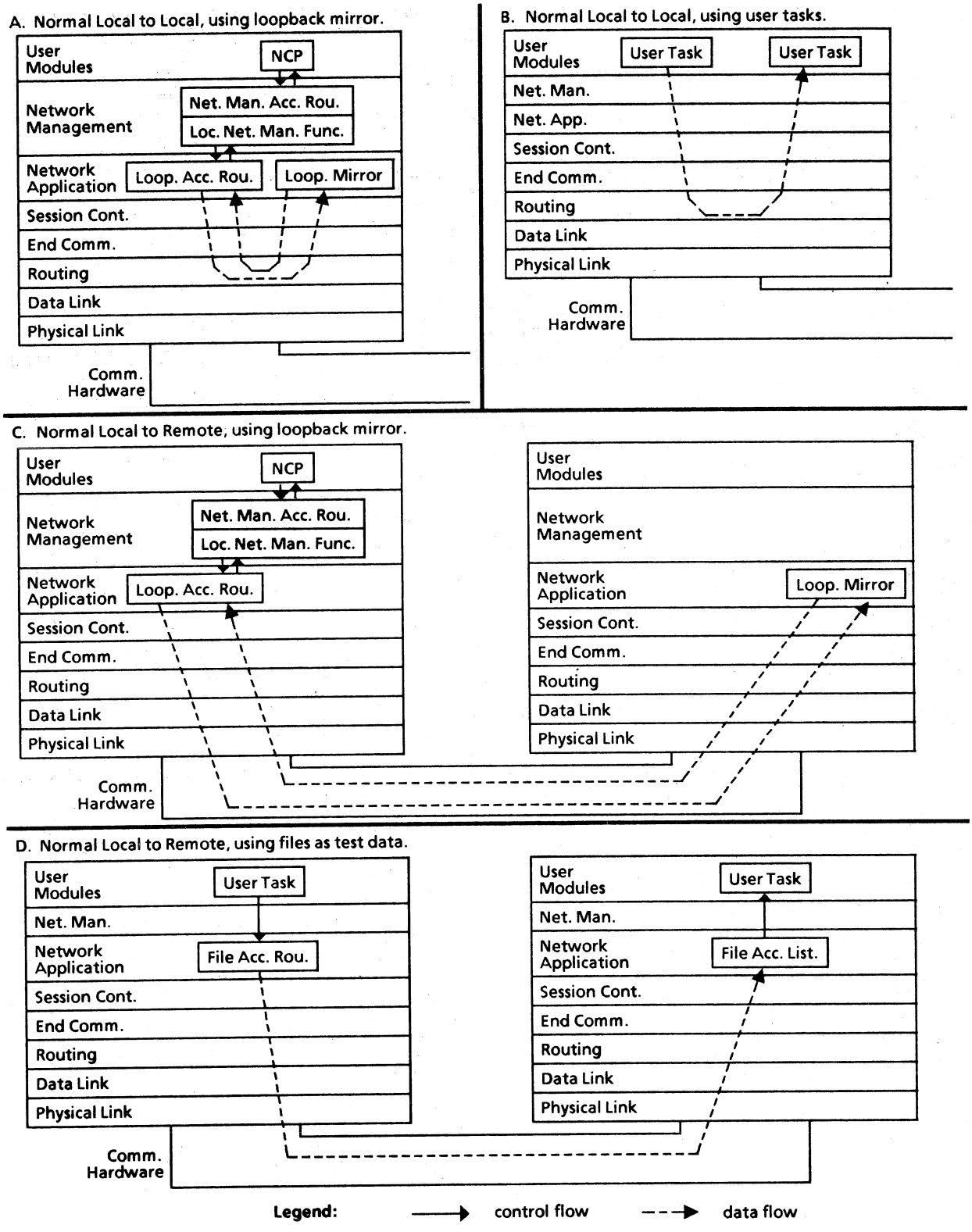


Figure 7-5: Examples of Node Level Logical Link Loopback Tests

7.2.4 Parameters, Counters, and Events

DNA specifies line, circuit, node, module, and logging parameters that Network Management can access. These are internal network values that are important enought to the system manager to examine, and, in some cases, change. These values are classified as characteristics or status. *Characteristics* generally remain constant until changed by a system manager. *Status* parameters reflect the condition of lines, circuits, nodes, modules, or loggers. For example, line state is a status parameter. The action of the network can change line, circuit, and node states. The logging state is really a system-manager-controlled on/off switch.

Each parameter has a unique type number. Some examples of parameters are:

- Circuit cost
- Node address
- Line state
- Node hops
- Logging sink node

DNA also specifies line, circuit, node and module counters and events. These are internal network variables that keep track of errors and network activity at each layer.

The Network Management design specifies only counters and events that are useful to the system manager. For example, certain Routing counters enable a system manager to determine if there are too many packets being discarded by the network due to congestion. In this case, the manager can adjust parameters that control how packets are routed in order to better balance traffic loads. Additionally, the manager can decide whether or not he needs to order more communication facilities.

The Network Management design limits the overhead to the network involved in gathering statistics and errors as much as possible. For example, there are no Routing counters that detect bugs in the Routing code: Network Management assumes the code is correct. The Routing layer algorithms themselves have been designed to detect and recover from most failures which would disrupt the network operation.

The Local Network Management Functions interface to network counters (see Figure 7-1). The Event Logger receives event information (Figure 7-1). Changes in counter values trigger many events. The Local Network Management Functions return counters in response to user level requests. On the other hand, the Event Logger, once set up, records events for user use automatically.

Each DNA functional specification contains, if applicable, its Network Management interface, including counters and events. The *DNA Network Management Functional Specification* contains tables of all Network Management parameters, counters, and events.

7.3 Network Management Messages

The three Network Management protocols are:

1. **Network Information and Control Exchange (NICE) Protocol.** This handles most Network Management functions.
2. **Event Logger Protocol.** This handles event logging from remote nodes.
3. **Loopback Mirror Protocol.** This handles the node loopback function.

Table 7-1 describes the NICE messages.

Table 7-1: NICE Messages

Message	Description
Request Down-line Load	Requests a specified executor node to down-line load a target node.
Request Up-line Dump	Requests a specified executor node to dump the memory of a target node.
Trigger Bootstrap	Requests a specified executor node to trigger the bootstrap loader of a target node.
Test	Requests a specified executor node to perform a node, circuit, or line loopback test.
Change Parameters	Requests a specified executor node to set or clear one or more Network Management parameters.
Read Information	Requests a specified executor node to read a specified group of parameters or counters.
Zero Counters	Requests a specified executor node to either read and zero or zero a specified group of counters.
System Specific	Requests a system-specific Network Management function.
Response	Provides request status and requested information in response to a NICE request.

There is only one event logger message, the Event message. It provides information when an event occurs.. This information includes:

- The event sink (in other words, where the event is to be logged -- one or more of console, file, or monitor)
- The event type and class
- The date and time the event occurred at the source node
- The name and address of the source node
- Whether the event related to a module, node, circuit, or line
- Specific data concerning the event.

Table 7-2 describes the Loopback Mirror Protocol messages.

Table 7-2: Loopback Mirror Messages

Message	Description
Command	Requests a loop test and sends the data to be looped.
Response	Returns status information and the looped back data.

7.4 Maintenance Operation Functional Description

Maintenance operations are special, primitive functions that must be available without the services of any DNA layers between Network Management and Data Link. This is because one of the nodes involved in the operation may be in a state where it cannot support more than a minimal Data Link. This could be, for example, a node that is being down-line loaded or up-line dumped.

At least some maintenance operation is supported in all of the DNA compatible Data Link protocols. DDCMP supports all functions in its maintenance mode. The X.25 frame level supports line loopback in a special loopback mode. Ethernet supports all functions as maintenance protocol types.

In most cases, maintenance operations are accomplished using the Maintenance Operation Protocol (MOP). In the case of the Ethernet, loopback testing is done with a standard protocol being incorporated into the Ethernet Functional Specification.

Maintenance operation includes the following functions:

- Down-line loading the memory of a computer system.
- Up-line dumping memory contents, usually upon a system failure.
- Loopback testing of the data link and/or its hardware components.
- System console control for a remote and possibly unattended computer system.

7.4.1 MOP Messages

Table 7-3, following, describes the MOP messages.

Table 7-3: MOP Messages

Message	Description
Memory Load with Transfer Address	Causes the contents of the image data to be loaded into memory at the load address, and the system to be started at the transfer address.
Memory Load without Transfer Address	Causes the contents of the image data to be loaded into memory at the load address.
Request Memory Dump	Requests a dump of a portion of memory to be returned in a Memory Dump Data message.
Request Program	Requests a program to be sent.
Request Memory Load	Requests the next segment of image data in a loading sequence and provides error status on the previous segment.
Request Dump Service	Requests a dump to be taken.
Memory Dump Data	Returns the requested memory image in response to a Request Memory Dump message.
Parameter Load with Transfer Address	Loads system parameters and transfers control to the loaded program.
Dump Complete	Signals completion of a requested dump.
Assistance Volunteer	Indicates willingness to perform dump or load service in response to a Request Program or Request Dump Service message.

(continued on next page)

Table 7-3 (Cont.): MOP Messages

Message	Description
Loop Data	Contains data to be sent back in a loopback test.
Looped Data	Returns the data from a Loop Data message.
Boot	Causes a system to reload itself if the verification code matches and the system is willing to do so. May result in a down-line load.
Request ID	Causes a system to send a System ID message.
System ID	Identifies the sending system by, for example, maintenance version, maintenance functions supported, processor type, *etc.*
Request Counters	Causes a system to send a Counters message.
Counters	Returns Data Link counter values in response to a Request Counters message.
Reserve Console	Reserves a system's console for dialog with the requester.
Release Console	Releases a console reserved with the Reserve Console message.
Console Command and Poll	Sends command data to a reserved console and/or polls for response data.
Console Response and Acknowledge	Returns response data and/or acknowledges receipt of a Console Command and Poll message.

7.4.2 MOP Operation

According to functional requests from higher level modules, or in response to MOP messages received from remote systems, MOP sends appropriate messages within the Data Link envelope. MOP messages and functions handle all message acknowledgment, time-out, and retransmission functions.

The node being serviced (being down-line loaded, up-line dumped, controlled, or tested) is called the *target* node. The node providing the services is called the *executor* node. MOP messages pass alternately between the executor and target.

Ideally, a target system would have all programs necessary to process MOP messages available in local main memory, Read Only Memory (ROM), or mass storage. Practically, this is not always the case and programs must be down-line loaded. This requirement to down-line load the desired function even extends to down-line load itself, which must usually be loaded through stages of progressively more capable loaders.

In all MOP message interchanges it is the responsibility of the sender of the request message to time-out and retransmit if a response is not received.

7.4.2.1 Down-line Loading

Either the target or the executor system can initiate a down-line load. In either case, the target sends a Program Request message to tell the executor what is needed and to indicate its willingness to cooperate. The major message exchange consists of Memory Load messages from the executor and Request Memory Load responses from the target. The sequence is completed when the host sends a Memory Load with Transfer Address or a Parameter Load with Transfer Address message.

On an Ethernet data link, a target can multicast its program request and select an executor from those responding with an Assistance Volunteer message.

7.4.2.2 Up-line Dumping

Up-line dumping is very similar to down-line loading. The target uses the Request Dump Service and Memory Dump Data messages. The executor controls the process with the Request Memory Dump and Dump Complete messages.

7.4.2.3 Link Testing

MOP tests the data link by looping a test message from a point in the physical connection. By moving the loopback point and isolating components, the user can diagnose problems. The active side (the executor module controlling the test) sends a Loop Data message out on the link and waits for a response. The passive side returns a Looped Data message if the message is looped at a computer. If the message is looped at an unintelligent point such as a loopback plug, modem, or hard-wired driver, the passive side returns the Loop Data message. The Looped Data message prevents infinite loops between intelligent processors.

MOP is not used for link testing on an Ethernet data link. Instead, the Ethernet standard loop protocol is used.

7.4.2.4 System Control

MOP can be used to control remote, possibly unattended, systems through what can be thought of as a virtual console. The remote system can simply be restarted, using the Boot message. Using the Reserve Console message, an executor can take control of a remote system console and proceed through a complete command dialog carried in the Console Command and Poll messages sent by the executor and the resulting Console Response and Acknowledge messages returned by the target. The session is ended by the executor sending a Release Console message or, in case the executor fails, a lack of executor messages within a time-out period at the target.

Glossary

access control

Screening inbound connect requests and verifying them against a local system account file. Access control is an optional Session Control function.

active side

With regard to MOP loopback tests, the node that controls a test.

adjacency

A (circuit, node) pair. An Ethernet with *n* attached nodes represents *n*-1 adjacencies to a Router on that Ethernet. On DDCMP and X.25 circuits, adjacencies and circuits are in one-to-one correspondence, since these circuits interconnect pairs of nodes.

adjacent node

A node removed from another node by a single physical line.

aged packet

A packet that has exceeded the maximum number of visits.

ASCII

American Standard Code for Information Interchange. This is a seven-bit-plus-parity code established by the American National Standards Institute to achieve compatibility between data services.

asynchronous line

A physical line on which the time intervals between transmitted characters may be of unequal length. Bit and byte framing are provided by the Physical Link layer on an asynchronous line, while message framing is provided by the Data Link layer.

back-off

The Ethernet Data Link procedure which ensures stable behavior on overloaded Ethernets by delaying retransmissions to reduce the load on the channel.

bandwidth

The range of frequencies assigned to a channel; the difference, expressed in Hertz, between the highest and lowest frequencies of a band. The higher the bandwidth, the more the data throughput.

binary synchronous protocol

A data link protocol that uses a defined set of control characters and control character sequences for synchronized transmission of binary coded data between stations in a data communications system.

carrier-sense multiple access with collision detection (CSMA/CD)

A distributed channel allocation procedure in which every station can receive all other stations' transmissions. Each station awaits an idle channel before transmitting, and each station can detect overlapping transmissions by other stations.

CCITT

The International Telegraph and Telephone Consultative Committee, the technical committee of the International Telecommunications Union (ITU), responsible for the development of recommendations regarding telecommunications, including data communications.

channel

The data path joining two or more stations, including the communications control capability of the associated stations.

characteristics

Parameters that are generally static values in volatile memory or permanent values in a permanent data base. Characteristic are a Network Management information type.

circuit

A logical connection between protocol modules at the Data Link layer. There is one circuit for each DDCMP point-to-point line and one circuit for each tributary station on a DDCMP multi-point line. There is one circuit for each X.25 permanent or switched virtual circuit. There is one circuit for each protocol type on an Ethernet.

circuit cost

A positive integer value associated with using a circuit. Messages are routed along the path between two nodes with the smallest total cost.

client

A module that requests service of another module.

client layer

A layer, typically the $n+1$ layer, which acts as the client of the n layer.

collision

Overlapping transmissions by two or more stations on an Ethernet. The Etherent Physical Link layer detects collisions and the Ethernet Data Link layer retransmits affected data after a random interval.

command node

The node where an NCP command originates.

congestion

The condition that arises when there are too many packets to be queued.

congestion control

The Routing component that manages buffers by limiting the maximum number of packets on a queue for a line. Also called transmit management.

control station

To the Data Link layer protocol, a station that has overall responsibility for orderly operation of a circuit.

controller

The control hardware for a line. For a multiple line controller device the controller is responsible for one or more units. The controller identification is part of a line identification.

cost

See circuit cost.

CSMA/CD

See carrier-sense multiple access with collision detection.

data flow

The movement of data from a source Session Control to a destination Session Control. NSP transforms data from session Control transmit buffers to a network form before sending it across a logical link. NSP retransforms the data at the destination from its network form to its receive buffer form. Data flows in both directions (full-duplex) on a logical link.

data communications equipment (DCE)

The equipment that provides the functions required to establish, maintain, and terminate a connection between DTEs using a physical circuit or a virtual circuit.

data link

A logical connection between two stations on the same circuit. In the case of a multipoint link there can be multiple circuits.

data termnal equipment (DTE)

The equipment, typically a computer system or terminal, comprising a data source and sink connected to common carrier communication facilities.

data transparency

The capability of receiving, without misinterpretation, data containing bit patterns that resemble protocol control characters.

datagram

A unit of data passed between the Routing Layer and the End Communication layer. When a route header is added, it becomes a packet.

DCE

See data communications equipment

designated router

The Router on an Ethernet chosen to perform additional duties, such as informing the endnodes on the Ethernet of the existence and identify of the Ethernet Routers. The Router chosen is the one with highest priority, with highest station address breaking ties.

down-line load

To send a copy of a system image or other file over a line to the memory of a target node.

DTE

See data terminal equipment.

EIA

Electronic Industries Association. A standards organization specializing in the electrical and functional characteristics of interface equipment.

end node

A topological description of a nonrouting node. Since a nonrouting node cannot perform route-through and supports only a single active line, it must be an end node. However, it is also possible for a routing node with a single line to be an end node.

end user module

A module that runs in a user process of a network node and communicates with Session Control to obtain logical link service.

error control

The protocol function that ensures the reliable delivery data. It typically consists of sequencing, acknowledgment, and retransmission mechanisms.

ethernet

A local area network using a Carrier-Sense Multiple Access with Collision Detect (CSMA/CD) scheme to arbitrate the use of a 10 megabit per second baseband coaxial cable.

events

Occurrences that are logged for recording by Network Management.

executor node

The node in which the active Local Network Management Function is running (that is, the node actually executing a Network Management command).

flow control

The protocol function that coordinates the flow of data between two protocol modules to ensure that data is not lost, to prevent buffer deadlock, and to minimize communication overhead.

frame

A Data Link layer message as sent or received by an Ethernet data link module or an X.25 Frame level module.

frame level

Level 2 of the CCITT X.25 recommendation which defines the link access procedure for data exchange over the link between the DTE and the DCE.

framing

The Physical or Data Link layer component that synchronizes data at the bit, byte, and message level.

full-duplex channel

A channel that provides concurrent communication in both directions (to and from a station).

full routing node

An implementation of the DNA Routing layer which contains the full complement of Routing components. A full routing node performs route-through functions.

gateway

A module or set of modules which transforms the conventions of one network into the conventions of another.

half-duplex channel

A channel that permits two-way communication, but in only one direction at any instant.

hierarchical network

A computer network in which processing control functions are performed at several levels by computers specially suited for the functions performed, for example, in a factory or laboratory automation.

hop

To the Routing layer, the logical distance between two adjacent nodes in a network.

host node

A node that provides services for another node.

interface

The relationship between different modules that are usually in the same node.

intra-Ethernet packet

A packet forwarded by a Router over an Ethernet to a destination end node which indicates that the source of the packet is on the same Ethernet.

ISO reference model

The International Standards Organization Reference Model for Open System Interconnection, ISO draft proposal DP7498. A proposed international standard for network architectures which defines a seven layer model, specifying services and protocols for each layer.

jam

A bit sequence transmitted by an Ethernet Data Link module upon detecting a collision to ensure that all affected stations detect the collision.

leased line

A nonswitched circuit leased from a public utility company (common carrier) for exclusive use.

line

A physical path which provides direct communication among some number of stations.

link level loopback

Testing a specific data link by sending a message directly to the data link layer and over a circuit or line to a device that returns the message to the source.

link management

The DDCMP component that controls transmission and reception on links connected to two or more transmitters and/or receivers in a given direction. Also, the Ethernet data link component which is responsible for channel allocation (collision avoidance) and contention resolution (collision handling).

logging

Recording information from an occurrence that has potential significance in the operation and/or maintenance of a network in a potential permanent form where it can be accessed by persons and/or programs to aid them in making real-time or long-term decisions.

logging sink node

A node to which logging information is directed.

logical channel

An association between an X.25 DTE and its DCE for a given virtual circuit.

logical link

A virtual circuit between two end user processes in the same node or in separate nodes. Session Control acts as an interface between an end user process requiring logical link service and the End Communication layer, which actually creates, maintains, and destroys logical links.

loop node

A special name for a node that is associated with an adjacency for loop testing purposes. The NCP SET NODE CIRCUIT command sets the loopback node name.

master station

A station that has control of a channel at a given instant, for the purpose of sending data messages to a slave station (whether or not it actually does).

maximum cost

An operator-controllable Routing Layer parameter that defines the point where the routing decision algorithm in a node declares another node unreachable because the cost of the least costly path to the other node is excessive. For correct operation this parameter must not be less than the maximum path cost of the network.

maximum hops

An operator-controllable Routing Layer parameter that defines the point where the routing decision algorithm in a node declares another node unreachable because the length of the shortest path between the two nodes is too long. For correct operation this parameter must not be less than the network diameter.

maximum path cost

The routing cost between the two nodes of the network having the greatest routing cost, where routing cost is the cost of the least cost path between a given pair of nodes. In Figure 3-1, the maximum path cost is 9.

maximum path length

The routing distance between the two nodes of the network having the greatest routing distance, where routing distance is the length of the least cost path between a given pair of nodes. In Figure 3-1, the maximum path cost is 9.

maximum visits

An operator-controllable Routing Layer parameter that defines the point where the packet lifetime control algorithm discards a packet which has traversed too many nodes. For correct operation, this parameter must not be less than twice the maximum path length of the network.

message exchange

The DDCMP component that transfers data correctly and in sequence over a link.

monitor

A logging sink that is to receive a machine-readable record of events for possible real-time decision-making.

multidrop line

See multipoint line

multiple line controller

A controller that can manage more than one unit. (DIGITAL multiple line controllers are also called multiplexers.)

multiplex

To simultaneously transmit or receive two or more data streams on a single channel.

multipoint line

A line connecting more than two stations, where one station is responsible for channel control. Also referred to as multidrop.

network

A collection of nodes interconnected by lines.

network diameter

The reachability distance between the two nodes of the network having the greatest reachability distance, where reachability distance is the length of the shortest path between a given pair of nodes. In Figure 3-1 the network diameter is 3.

node

An implementation of a computer system that supports the Routing Layer, the End Communication layer and Session Control layer. Each node has a unique node address.

node address

The unique numeric identification of a specific node.

node level loopback

Testing a logical link using messages that flow with normal data traffic through the Session Control, End Communication, and Routing layers within one node or from one node to another and back. In some cases node level loopback involves using a loopback node name associated with a particular circuit.

node name

An optional alphanumeric identification associated with a node address in a strict one-to-one mapping. No name may be used more than once in a node. The node name must contain at least one alphabetic character.

node name mapping table

A table that defines the correspondence between node names and node addresses or adjacencies. Session Control uses the table to identify destination nodes for outgoing connect requests and source nodes for incoming connect requests.

nonrouting node

A Phase III or Phase IV DECnet node that contains a subset of routing modules and can forward and receive packets but not perform route-through. It is connected to the network by a single active circuit.

object type

A numeric value that may be used for process or task addressing by DECnet processes instead of a process name. An object type is normally used to identify a generic service, such as file access.

OSI

See ISO reference model.

Other Data

The NSP Data Request, Interrupt Request, and Interrupt messages. These are all the NSP data messages other than Data Segment. Because all Other Data messages move in the same data subchannel, it is sometimes useful to group them together.

packet

A unit of data to be routed from a source node to a destination node. When stripped of its route header and passed to the End Communication Layer, it becomes a datagram.

packet level

Level 3 of the CCITT X.25 recommendation which defines the packet format and control procedures for the exchange of packets.

packet lifetime control

The Routing component that monitors adjacencies to detect if an adjacency has gone down, and prevents excessive looping of packets by discarding packets that have exceeded the maximum visit limit.

packet switching

See route-through

parallel data transmission

A data communication technique in which more than one code element (for example, bit) of each byte is sent or received simultaneously.

parameters

DNA values to which Network Management has access for controlling and monitoring purposes.

passive side

With regard to MOP loopback tests, the node that loops back the test messages.

path

A possible route for a packet from source node to destination node. This can be a sequence of connected nodes between the two nodes.

path cost

The sum of the circuit costs along a path between two nodes.

path length

The number of hops along a path between two nodes.

peer protocol

A protocol for communication between modules in the same layer.

permanent virtual circuit

A virtual circuit between two V.25 DTEs that is always established. A logical channel is permanently allocated at each DTE/DCE interface to a permanent virtual circuit.

physical link

An individually hardware addressable communication path.

piggybacking

Sending an acknowledgment within a returned data or control message.

pipelining

Sending messages without waiting for individual acknowledgment of each successive message.

point-to-point circuit

A link connecting only two stations.

protocol

A set of messages with specific formats and rules for exchanging the messages.

raw event

A logging event as recorded by the source process, incomplete in terms of total information required.

reachable node

A node to which a routing node believes it can direct a packet.

reassembly

The placing of multiple, received data segments by NSP into a single Session Control receive buffer.

remote node

To one node, any other network node.

request count

This term has two definitions in NSP: (1) Variables that NSP uses to determine when to send data, and (2) values sent in Link Service (Data Request and Interrupt Request) messages.

The flow control mechanism adds the request counts received in Data Request and Interrupt Request messages to the request counts it maintains to determine when to send data.

retransmission

The resending of messages that have not been acknowledged within a certain period of time. This is part of a protocol's error control mechanisms.

RMS

Record Management Services. This file system will be used on all major DIGITAL systems except where space is limited (for example, RT-11). In addition to access modes provided by previous file systems. RMS provides random access for direct and indexed files and ISAM.

router

See full routing node.

route-through

The directing of packets from source nodes to destination nodes by one or more intermediary nodes. Routing nodes permit route-through. Also called packet switching.

routing

Directing data message packets from source nodes to destination nodes.

routing node

See full routing node

segment

The data carried in a Data Segment message. NSP divides the data from Session Control transmit buffers into numbered segments for transmission over logical links.

segmentation

The division of normal data from Session Control transmit buffers into numbered segments for transmission over logical links.

server

A module or set of modules in a layer that perform a well defined service, such as remote file access or gateway communication on behalf of another module.

server system

A node which contains one or more servers. A server system is often dedicated to server functions.

sink node

A node which receives and records Network Management events.

slave station

A tributary station that can send data only when polled or requested to by a master.control station.

star topology

A network configuration in which one central node is connected to more than one adjacent end node. A star can be a subset of a larger network.

station

With regard to the Data Link layer protocol, a termination on a data link. A station is a combination of the physical link (communication hardware) and the data link protocol implementation.

station address

An address assigned at the data link level to a station.

status

Dynamic information relating to a network such as a line state. A Network Management information type.

subchannel

A logical communication path within a logical link that handles a defined category of NSP data messages. Because Data Segment messages are handled differently from Other Data messages, the two types of messages can be thought of as travelling in two different subchannels.

switched virtual circuit

A temporary association between twoX.25 DTEs.

synchronous line

A physical line on which data characters and bits are transmitted at a fixed rate with transmitter and receiver synchronized. Bit framing is provided by the Physical Link layer on a synchronous line, while byte and message framing are provided by the Data Link layer.

target node

The node that receives a memory image during a down-line load, generates an up-line dump, or loops back a test message.

topology

The physical arrangement and relationships of interconnected nodes and lines in a network. A legal topology satisfies the requirements of the Routing Layer specification.

transparent data

Binary data transmitted with the recognition of most control characters suppressed. For example, DDCMP provides data transparency because it can receive, without misinterpretation, data containing bit patterns that resemble DDCMP control characters.

tributary station

A station on a multipoint line that is not a control station.

unit

The hardware controlling one circuit on a multiple line controller. A unit, a controller, and associated Data Link modules form a station. See Figure G-1.

unreachable node

A node to which a routing node has determined that the cost of the least costly path exceeds the maximum cost of the network, or the length of the least costly path exceeds the maximum hops of the network.

up-line dump

To send a copy of a target node's memory image up a line to a file at the host node.

user

A person or module which requests service from a layer. *Client* is the preferred terminology when referring to modules.

virtual call

See switched virtual circuit.

virtual circuit

A connection between a data source and a sink in a network. They may be realized by different circuit configurations. Virtual circuits typically provide guaranteed delivery and sequentiality of client data.

wild card

With regard to DAP, an asterisk (*) that replaces an element in a file specification. The asterisk specifies all known items in the range indicated by its position. For example, FILE.*;* specifies all known type, and versions of all files named FILE.

window

A range of packets authorized for transmission across an X.25 DTE/DCE interface.

DECnet Architecture
General Description
AA-N149A-TC

READER'S COMMENTS

NOTE: This form is for document comments only. DIGITAL will use comments submitted on this form at the company's discretion. If you require a written reply and are eligible to receive one under Software Performance Report (SPR) service, submit your comments on an SPR form.

Did you find this manual understandable, usable, and well-organized? Please make suggestions for improvement.

Did you find errors in this manual? If so, specify the error and the page number.

Please indicate the type of user/reader that you most nearly represent.

- ☐ Assembly language programmer
- ☐ Higher-level language programmer
- ☐ Occasional programmer (experienced)
- ☐ User with little programming experience
- ☐ Student programmer
- ☐ Other (please specify)_____

Name_____ Date_____

Organization_____

Street_____

City_____ State_____ Zip Code_____
or
Country

—Do Not Tear - Fold Here and Tape —

digital

No Postage
Necessary
if Mailed in the
United States

BUSINESS REPLY MAIL
FIRST CLASS PERMIT NO.33 MAYNARD MASS.

POSTAGE WILL BE PAID BY ADDRESSEE

SOFTWARE DOCUMENTATION
1925 ANDOVER STREET TW/E07
TEWKSBURY, MASSACHUSETTS 01876

— — — Do Not Tear - Fold Here and Tape — — — — — — — — — — — — — — — — — —

Cut Along Dotted Line

digital