INTERNATIONAL ATOMIC ENERGY AGENCY
UNITED NATIONS EDUCATIONAL, SCIENTIFIC AND CULTURAL ORGANIZATION
INTERNATIONAL CENTRE FOR THEORETICAL PHYSICS
I.C.T.P., P.O. BOX 586, 34100 TRIESTE, ITALY. CABLE: CENTRATOM TRIESTE

H4.SMR/585-2

FIRST INTERNATIONAL SCHOOL ON COMPUTER NETWORK ANALYSIS AND MANAGEMENT

(3 - 14 December 1990)

Digital Network Architecture

Domenico Gianmarini

DIGITAL Equipment S.P.A.
Via Giambellino, 7
34127 Padova

EY-1379E-SG-0002

DIGITAL Network Architecture

A Self-Paced Course

Student Workbook

Prepared by Educational Services
of
Digital Equipment Corporation

Printed in U S A

The manuscript for this book was created using DIGITAL Standard Runoff. Book production was done by Educational Services Development and Publishing in Nashua, NH.

CONTENTS

SG STUDENT GUIDE

1 DNA OVERVIEW

2 DATA LINK CONTROL

3 ROUTING LAYER

4 END COMMUNICATION LAYER

5 SESSION CONTROL LAYER

6 NETWORK APPLICATION LAYER

7 NETWORK MANAGEMENT LAYER

8 MESSAGE EXCHANGE

AP APPENDIX

FIGURES

TABLES

EXAMPLES

STUDENT GUIDE

PREFACE

DIGITAL Network Architecture is a self-paced course for Software Specialists and Customers who need to use DECnet for technical support of Distributed Data Processing Networks or in applications environments. It is also for Managers who need a basic operational understanding of the DIGITAL Network Architecture in order to make network-related business decisions. This basic understanding will also allow Managers to obtain the most effective performance from their Distributed Data Processing Network.

This course provides a basic understanding of DECnet layers, protocols, message exchanges, and protocol operations. This course alone will not enable you to write system and network applications or troubleshoot network problems. For these more complex tasks, you must read the DECnet Functional Specifications.

INTRODUCTION

This course is a Self-Paced Instruction (SPI), designed for independent study. To complete the course you need the materials in the STUDENT STUDY PACKAGE. If additional information beyond the scope of this course is desired, order the reference manuals listed in the Course Resources section of this module. Copies of the reference manuals can be ordered from the nearest Digital Equipment Corporation Sales Office.

COURSE PREREQUISITE

Before taking this course, you must successfully complete the Network Concepts (SPI) course, EY-0149E-PS-0001.

COURSE GOALS

1. Discuss the functions and layers of the DIGITAL Network Architecture (DNA) on a technical level.
2. Attend any of the DECnet product-related courses. Note that this course is a prerequisite to all other DECnet product-related courses.
3. Relate the need for a network architecture that provides standard protocols.

MAJOR TOPICS

This course covers the following major topics:

1. Introduction to DNA

   - DNA Philosophy
   - DNA Architectural Layers
   - DNA Layers, Modules, and Interfaces
   - DNA User Services
   - DNA Protocols
   - DNA Message Data Flow
   - Physical Link Layer Functions

2. Functional Descriptions and Message Formatting performed by the DNA Layers for:

   - Data Link Control Layer
   - Routing Layer
   - End Communication Layer
   - Session Control Layer
   - Network Application Layer
   - Network Management Layer

3. Message Exchanges between the DNA layers and corresponding protocols for:

   - Data Link Layer Protocols
   - Routing Layer Protocol
   - End Communication Layer Protocol
   - Session Control Layer Protocol
   - Network Application Layer Protocols
   - Network Management Layer Protocols

COURSE MODULES

The course is made up of eight modules, each requiring approximately three hours of study time. It is primarily designed for students with no previous knowledge of DIGITAL Network Architecture (DNA). This material covers DIGITAL Network Architecture, Phase IV.

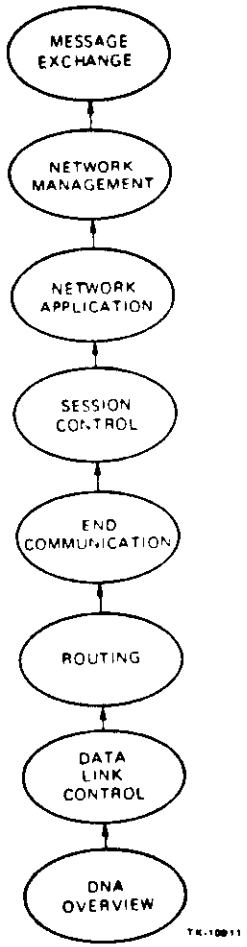
Listed below are the eight modules with their subsections:

COURSE MAP DESCRIPTION

The course map shows each module and how it is related to the other modules of the course.

Before beginning to study a particular module, complete all prerequisites for that module by simply following the arrows on the course map. The location of a module on the map suggests the point in the course where it will be most meaningful.

COURSE MAP



MODULE CONTENTS

Each module consists of the following:

- An introduction
- A list of objectives
- A list of learning activities explaining how to study the module
- A list of resources showing the required reading for the module
- The module text
- A module test

MODULE TESTS

The test at the end of each module has been designed to evaluate your knowledge of the module material. When you feel you have achieved the module objectives, you may take the test. The answers to each test are found in the accompanying Tests and Answers booklet. There are no grades, and you will not be compared to others taking the course. In each module test, you are merely asked to demonstrate an adequate understanding of the material before proceeding.

Your work need not be checked by someone else. However, discussing your answers with someone who has completed the module test may be mutually beneficial.

STUDY HINTS

You should first carefully study the objectives of a module. Next, read the list of assignments so that you fully understand how you are going to achieve the goals stated in the objectives. In particular, consider the wording of the assignments; words such as "study," "read," "look over," and "glance over" indicate the depth of study recommended in each particular assignment.

If after reading the objectives you feel that you already understand the material, take the module test first. If you have trouble answering the module test questions, return to the learning activities and perform all of the tasks indicated.

COURSE RESOURCES

For additional information on writing applications and system troubleshooting, refer to the following manuals. They may be ordered through your nearest Digital Equipment Corporation Sales Office.

1. DECnet DIGITAL Network Architecture (Phase IV) General Description (Order No. AA-N149A-TC)
2. DNA Data Access Protocol (DAP) Functional Specification, Version 7.0 (Order No. AA-K177B-TK)
3. DNA Digital Data Communications Message Protocol (DDCMP) Functional Specification, Version 4.1 (Order No. AA-K175A-TK)
4. DNA Session Control Functional Specification, Phase IV, Version 1.0 (Order No. AA-K182A-TK)
5. DNA Routing Layer Functional Specification, Phase IV, Version 2.0 (Order No. AA-X435A-TK)
6. DNA Low-Level Maintenance Operations Architectural Functional Specification, Version 3.0 (Order No. AA-X436-TK)
7. DNA Network Management Functional Specification, Version 4.0 (Order No. AA-X437A-TK)
8. DNA End Communications (NSP) Functional Specification, Phase IV, Version 4.0 (Order No. AA-X439A-TK)
9. DNA NI Data Link Architectural Functional Specification, Version 1.0, (Order No. AA-Y298A-TK)
10. DNA NI Node Product Architectural Functional Specification, Version 1.0 (Order No. AA-X440A-TK)
11. DNA X.25 Frame Level Functional Specification

12. DNA X.25 Packet Level Functional Specification
13. DNA X.25 Gateway Access Functional Specification
14. DNA SNA Gateway Access Functional Specification
15. DNA Network Virtual Terminal Functional Specification

DNA OVERVIEW

INTRODUCTION

This module introduces the basic concepts, characteristics, and goals of DIGITAL Network Architecture (DNA). The other modules in the course expand upon what is introduced in this module.

It is assumed, as stated by the course prerequisites, that you successfully completed Network Concepts, the self-paced course. The Network Concepts course covers basic network terms. It also illustrates and defines the various types of network topologies possible using the DIGITAL Network Architecture as a foundation.

OBJECTIVES

To use DECnet in technical support of applications environments, Software Specialists and Customer Personnel must understand the basic concepts of the DIGITAL Network Architecture (DNA) including its:

- Layering system
- Modules and interfaces
- Protocols
- Message data flow
- Available user services

RESOURCE

DECnet DIGITAL Network Architecture (Phase IV) General Description

LEARNING ACTIVITIES

1. Study the information in this module.
2. Read Chapter 1, Introduction to DECnet (Phase IV), in the DECnet DIGITAL Network Architecture (Phase IV) General Description.
3. Take the module test at the end of this module.
4. Correct the test using the answer sheet provided in the Tests and Answers booklet. Review the material on any questions you may have missed before moving on to the next module.

1.1 DNA ARCHITECTURAL LAYERS

DECnet DIGITAL Network Architecture (Phase IV) is made up of eight different software layers. Seven of these layers use predefined modules and protocols to perform specific functions. The eighth layer, the User layer, does not define any specific protocol (refer to Table 1-1). It is the responsibility of the user to define a protocol that will best serve his needs of intertask and/or program communications.

1.1.1 DNA Layers And Modules

Table 1-1 briefly describes each DNA Layer. Figure 1-1 shows the relative placement of the different layers within the DIGITAL Network Architecture (DNA). Figure 1-1 shows the various modules and protocols within the different layers of the DNA.

Table 1-1 DNA Layers

DNA Layer	Description
User Layer	Contains most user-supplied functions and the Network Control Program (NCP). NCP provides system managers and users with interactive terminal access to lower layers of the DNA. NCP is the only predefined DECnet module in the User layer.
Network Management Layer	Controls the lower layers of the DNA using the Network Information and Control Exchange Protocol (NICE). This is the only layer with direct access to all other lower layers. It allows the system manager and privileged users to control the network from any terminal in the network.
Network Application Layer	Performs generic services to the User layer using specific protocol modules. Contains both user-defined and DIGITAL-supplied software modules that can execute simultaneously or independently. It allows remote network functions to be performed. Tasks at this level perform specific functions (e.g., a file transfer from one node to another).

Table 1-1 DNA Layers (Cont)

DNA Layer	Description
Session Control Layer	Defines system-dependent aspects of the logical link communication. Provides: - Access control protection - Logical Link management - Guaranteed message delivery - Segmentation of message into packets
End Communication Layer	Controls the system-independent aspects of creating and managing logical links. Previously called the Network Services layer.
Routing Layer	Routes the user datagrams in packets to their destination (also called a "packet delivery service"). This layer was previously called the Transport layer.
Data Link Layer	Creates a communication path between adjacent nodes. Ensures data integrity and delivery of packets to the adjacent node. This layer consists of several different software modules. Supports X.25, Ethernet, and DDCMP links for both asynchronous and synchronous communications. Modules in the layer may execute simultaneously or independently.
Physical Link Control Layer	Manages the physical transmission and reception of user data over the physical network lines. This layer consists of parts of the Device Driver for each communications device and the communications device itself.

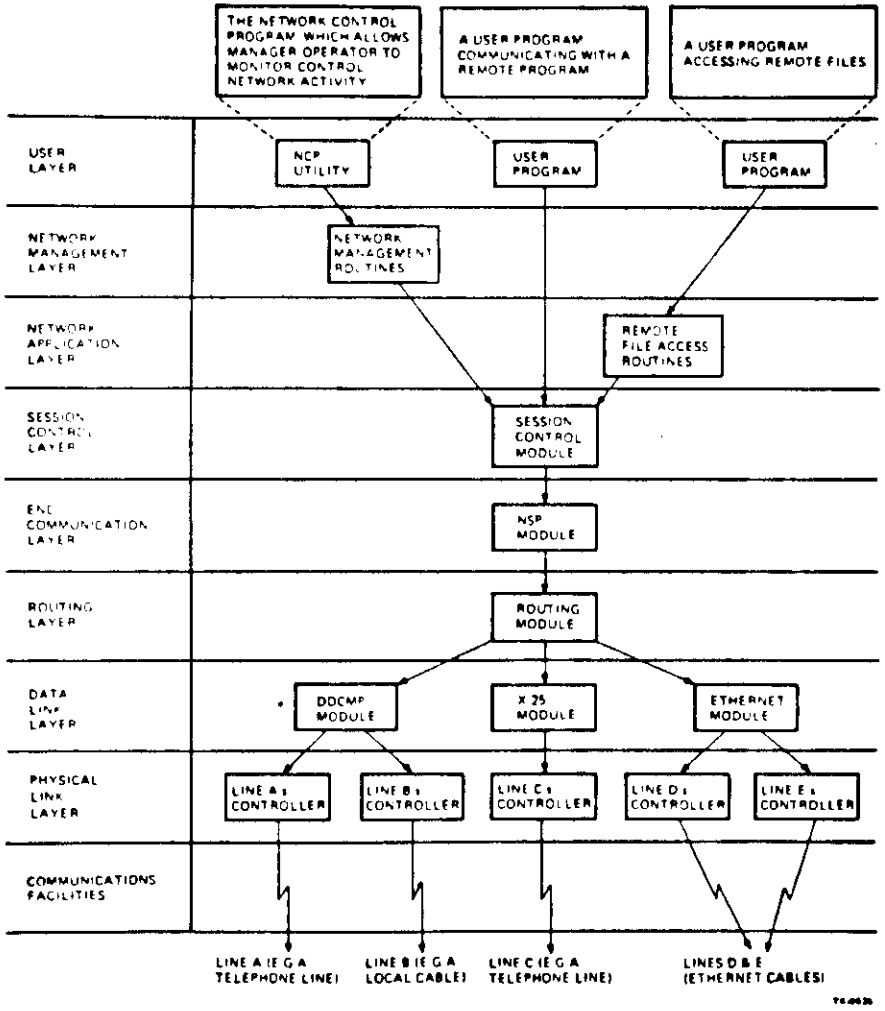


Figure 1-1 DNA Layers and Modules Resident in a Typical DECnet Node

1.1.2 DNA Layer Interfacing

The different DNA layers in a single DECnet node have specific paths between them. The path used depends upon whether the operation to be performed is for normal user data traffic, network management, or network fault isolation. These paths connect the different DNA layers together to accomplish the user's goal.

These paths change depending upon the user's goal, thus interfacing the DNA layers differently for specific user applications. For example, the user's goal could be to send or receive data messages or files (normal user data), or to send and receive test messages or files (test data).

Figure 1-2 shows the different paths for normal and management applications of the DNA.

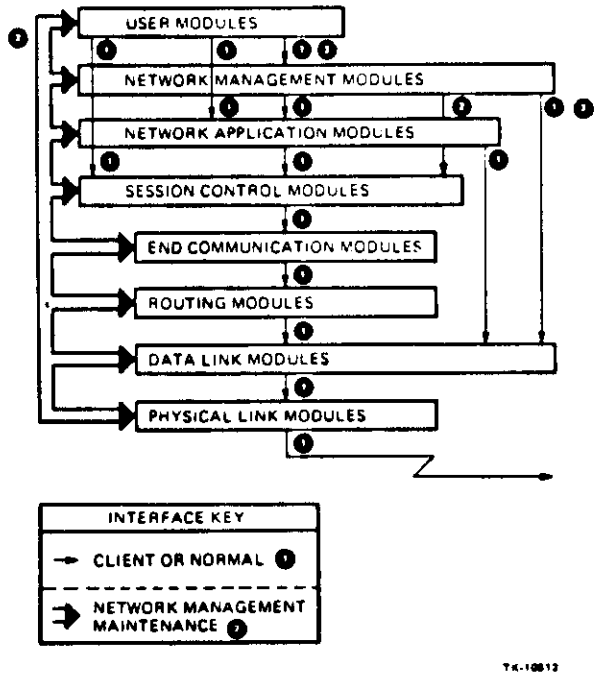


Figure 1-2 DNA Layers and Interfacing Paths

1.2 DNA USER SERVICES

As shown in Figures 1-1 and 1-2, there are many different ways the DNA layers can be accessed by the other layers within the DNA. This versatility provides the following User Services to the network user:

- User-to-User: A user level task or process in one node can communicate via a logical link with a user level task or process in another node.
- Remote Application: A user level task or process that uses a Network Application module at the local node to perform a function at a remote node via a logical link.
- Network Management: A user level command or process that uses a Network Management module at the local node to perform a Network Management service at a remote node. This user service uses a logical link created by the Session Control and other lower layers to interface directly to the Data Link layer.

1.3 DNA PROTOCOLS

A Protocol is both a set of messages and the rules for exchanging messages. DNA defines the message formats and rules very specifically. This allows communication between software modules with equivalent functions on different nodes.

The DNA protocol of any one software module is transparent to other DNA software modules. Only protocols of modules that are on different nodes but that are functionally equivalent can recognize each other's header information. They can tell the difference between protocol header information and user data.

Table 1-2 lists the DNA-defined protocols, the layers with which they are associated, and a brief description of each.

Table 1-2 DNA Protocols

Protocol	DNA Layer	Description
None	USER	Reserved for user-specific processes
Network Information and Control Exchange (NICE)	NETWORK MANAGEMENT	Triggers down-line loading, dumping, testing, reading parameters and counters, setting parameters, and zeroing counters.
Event Logger		Records significant occurrences such as changes or inconsistencies in lower DNA layers.
Maintenance Operation		Provides for data link level loopback testing, remote control of unattended systems, and down-line loading or up-line dumping of computer systems without mass storage.
Data Application (DAP)	NETWORK APPLICATION	Used for remote file access and transfer.
Network Virtual Terminal Protocols		Used for terminal access through-out the network.
X.25 Gateway Access		Provides an interface for a node not directly connected to a public data network. Allows the node to access the facilities of that network through an intermediary gateway node.
SNA Gateway Access		Provides interfacing for a node not connected directly to an IBM SNA network access to the facilities of that network for terminal access and remote job entry.

Table 1-2 DNA Protocols (Cont)

Protocol	DNA Layer	Description
Loopback Mirror	NETWORK APPLICATION	Used for network management logical link loopback tests.
Session Control	SESSION CONTROL	Controls network functions such as: - sending logical link data - receiving logical link data - disconnecting logical links - aborting logical links
End Communication	END COMMUNICATION	Controls all system-independent aspects of managing logical links.
Routing	ROUTING	Manages all message routing and congestion control.
DIGITAL Data Communications Message Protocol (DDCMP)	DATA LINK CONTROL	Ensures the integrity and correct sequencing of messages between two adjacent nodes.
X.25		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		Allows the user's node to connect to an adjacent node via an Ethernet local area network.

Figure 1-3 shows the protocol communication between two adjacent nodes. Notice there is only one physical link between the two nodes, and that each module's specific protocol will only communicate with the other node's equivalent module. Both modules on each node must use the same protocol.

The purpose for equivalent module-to-module communication is to make the other modules and layers in the nodes transparent to each other. The lower layers and modules of the DNA see the upper layers and their protocol messages as user data.

In a sense, the protocol messages of the various layers and modules build upon each other. Although invisible to the user, this protocol building effect is what accomplishes the user's goal.

This concept of transparent software layering with protocol connections between the various software modules and layers provides the Network user with a network that is:

- Flexible to upward change
    1. DNA allows incorporation of future technology changes in both hardware and software.
    2. DNA allows the movement of function responsibilities from software to hardware as more sophisticated hardware line controllers are introduced.
    3. DNA allows subsets of software modules to reside in any one layer, providing unlimited versatility to network nodes.

- Cost effective
    1. A DECnet network application costs about the same as a customer-designed network that achieves the same performance for the same application.

- Secure
    1. DNA layering allows for security at several levels (layers).
    2. User access and authenticity is implemented in most DECnet products.

- Highly available
    1. Networks can be configured to maintain operation even if a subset of lines or nodes fail. This is because the DNA Maintenance functions are highly distributed; DECnet networks can recover from operator error (unless the error is extreme).

- Highly distributed
    1. The major functions of DNA are not centralized in one node in a network. This prevents the network from relying heavily on any one specific node for network management and control.

- Able to implement network control and maintenance functions at a user level
    1. Allows easier system management
    2. Allows terminal commands to set, change, or display lower level parameters or counters

- Able to support a wide range of topologies, such as:
    1. Point-to-point
    2. Star
    3. Bus
    4. Multipoint
    5. Multiple controller/Multiplex
    6. Hierarchical structures
    7. Topologies in which each node has equal control over the network operation
    8. Networks tailored to maximize efficiency for a customer's specific application, no matter how large or small

- Able to support a wide range of communication facilities, such as:
    1. Leased telephone lines
    2. Private telephone lines
    3. Switched telephone lines
    4. Satellite links
    5. Ethernet local area networks
    6. X.25-based packet-switching networks
    7. Local coaxial cable links
    8. Fiber optics links

- Also able to support many different methods of communication, such as:
    1. Asynchronous
    2. Synchronous
    3. Full-Duplex
    4. Half-Duplex
    5. Simplex

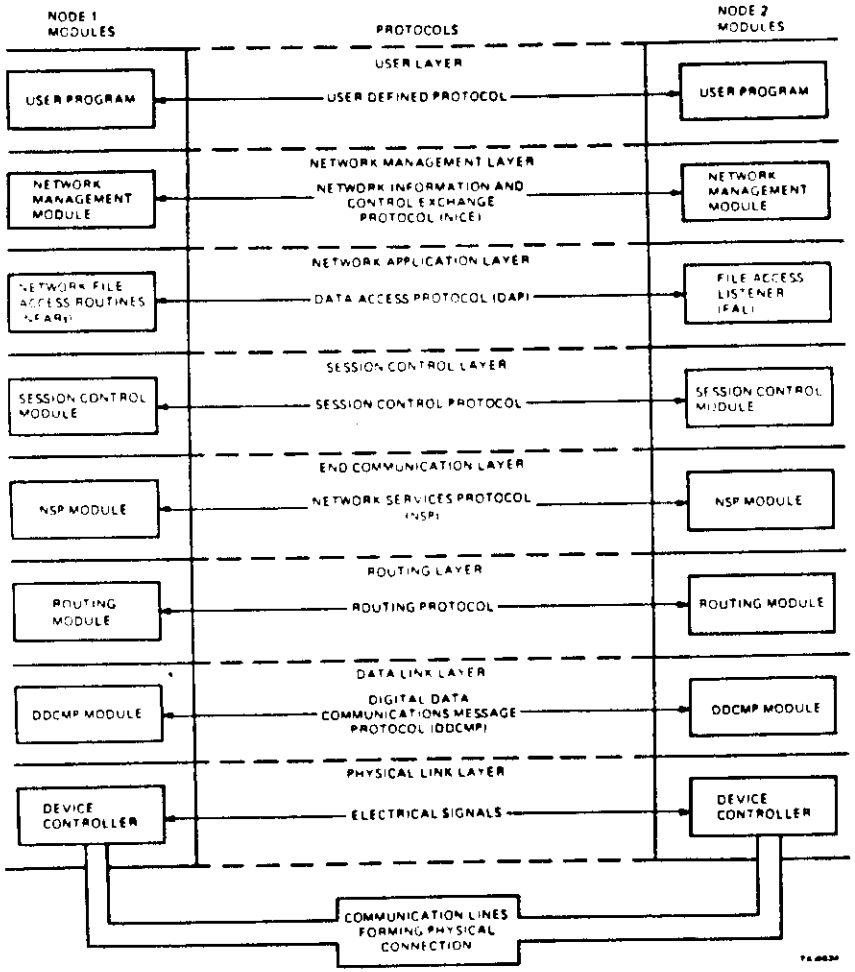


Figure 1-3 Protocol Communication Between Two Adjacent Nodes

1.4 DNA MESSAGE DATA FLOW

A Distributed Data Processing Network shares computer resources, performs distributed computation, and performs remote computer system communication. These network functional goals are accomplished under DNA by passing data from a source in one network node to a destination in another network node.

1.4.1 Message Building

It is important to understand how the original user's data is passed through the DNA layers in and between nodes in the network. Since DNA is a layered protocol structure that builds the data into more than just the original message, it must be looked at one piece at a time.

Figure 1-4 shows data building and stripping as the original user data is passed from layer to layer in a DECnet node. For sending messages (transmitting), start from the top and work down. For receiving messages, start at the bottom and work up. In this example, Network Management is not involved.

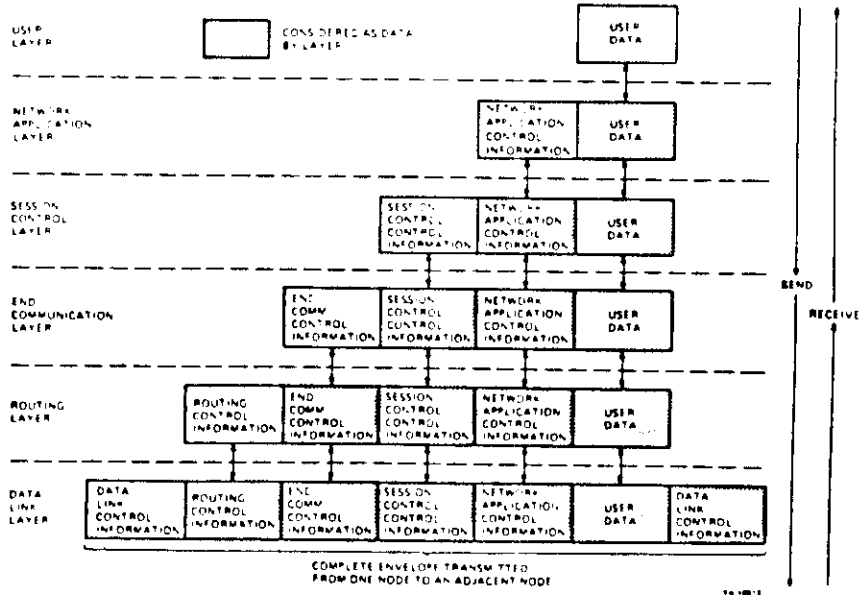


Figure 1-4 Data Message Passing Through DNA Layers

1.4.2 Message Flow From Node To Node

Messages traveling from one node in a network to another node, pass from a source process in the transmitting node to a destination process in the receiving node. To travel from node to node, the message must be directed along the way. This message direction is provided by the DNA protocols.

Figure 1-5 illustrates the message steering that must take place to route any data from one network node to another. This example shows a data message traveling from Node 1 to Node 3 using the Routing layer protocols of Node 2. In this case, Nodes 1 and 3 are not adjacent nodes.

The data from Node 1 originates at the User layer. It is then passed down through the DNA hierarchy where each layer's protocol adds its own specific piece of information for message control. Finally, the Data Link and Physical Link layers acting together transmit the data across the network line.

When the data is received by Node 2, it passes the message up to its Routing layer. Node 2's Routing layer then checks the Routing protocol information added by Node 1's Routing layer and determines that the message is not for it. Node 2 then checks to see if it knows of a path to the intended node.

If Node 2 does not know the way to the intended destination node, it sends a message back to Node 1 telling it that the destination node is currently unreachable.

In the figure, Node 2 does know the way to Node 3. Node 2 uses its own routing layer information (routing data base) to forward the message to Node 3.

Node 3 receives the message from Node 2 and passes the data up to its Routing layer. Node 3's Routing layer checks the Routing protocol information added by Node 1's Routing layer. Node 3's Routing layer protocol determines that the message is for it and then strips off the Routing layer protocol header information. Node 3's Routing layer then passes the message up to the next higher DNA layer. Each layer must strip off its own specific piece of protocol information from the message. Finally, the message from Node 1's User layer is passed up to the originator's intended destination, Node 3's User layer.

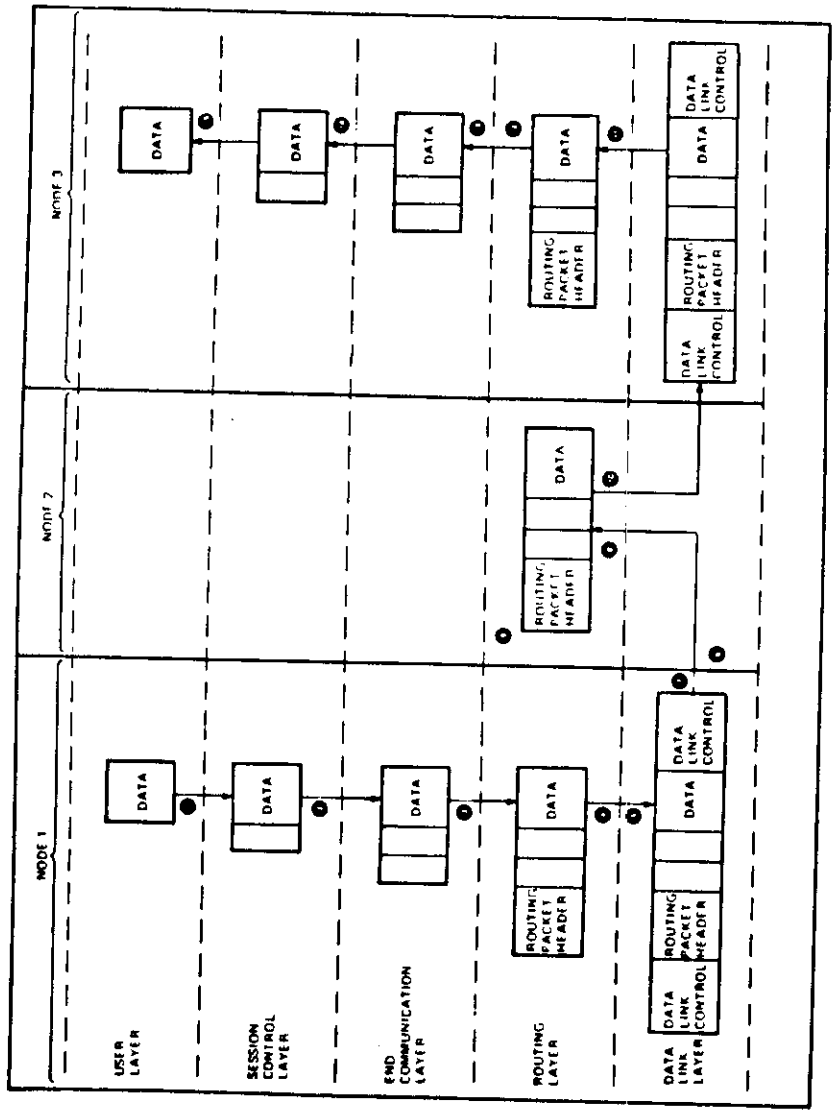


Figure 1-5 Message Flow from Node to Node

Figure 1-5 assumes that the user on Node 1 requests to send data to a user on Node 3. The following numbered steps correspond to the circled numbers found on Figure 1-5. The numbered steps are in order of occurrence as the original user's data message is processed, transmitted, and received by the user at Node 3.

Data Flow at the Source Node (Node 1)

❶ Node 1 user requests a logical connection to the Destination node (Node 3) user by passing the connect control data.

❷ The Session Control layer receives the user data, maps the destination node name, Node 3, to a numerical address, if necessary. It then places the data and its control information into the next transmit buffer. The message, or transmit buffer, is then passed down to the End Communication layer for further processing.

❸ The End Communication layer adds its control information, which includes a logical link identification number, to the message. It then passes the message, now called a datagram, down to the Routing layer.

❹ The Routing layer adds a header made up of the destination and source node addresses to the message. It then selects an outgoing circuit for the message based upon the routing information stored in the Routing layer. Routing then passes the message, now called a packet, to the Data Link layer.

❺ The Data Link layer adds its protocol header which consists of framing, synchronizing, addressing and control information to the packet. It also adds its protocol trailer, which is a cyclic redundancy check (CRC) character, to the packet. The packet is now "enveloped" for transmission. The Data Link layer then passes the packet to the Physical Link layer for transmission.

❻ The Physical Link layer transmits the enveloped message over the physical line to the next node on the physical transmission line.

Data Flow Across the Network to the Destination Node (Node 3)

❼ The enveloped data message arrives at Node 2, the next node on the physical line. The physical Link layer of Node 2 receives the message and passes it up to the Data Link layer.

❽ The Data Link layer checks the packet for any bit errors that may have been caused by the transmission medium.

On non-Ethernet links, the Data Link layer protocol performs error correction procedures. DDCMP and X.25 Links provide error correction by requesting the source node to retransmit the message.

For Ethernet links, packets received in error are discarded. Error recovery is provided by higher level DNA layers.

The Data Link layer strips the header and trailer information from correctly received packets. The data message is then passed up to the Routing layer.

❾ The Routing layer of Node 2 checks the Routing layer protocol information on the packet. The Routing layer information added by Node 1 is decoded to determine the intended destination of the packet. If this packet were intended for Node 2, the Routing layer protocol information would be stripped from the packet (it would now be called a datagram).

The datagram would then be passed up to the End Communication layer for further processing (protocol stripping). However, since this packet is not destined for Node 2, Node 2's Routing layer protocol must steer the message back out onto the proper transmission circuit to send this packet to Node 3. Once the proper transmission circuit is determined, it passes the packet down to its newly calculated transmission circuit's Data Link layer for further protocol processing.

❿ The message proceeds as in steps 5 and 6 above.

⓫ The message proceeds through the network, switching at routing nodes, until it arrives at a node whose address, as defined by its Routing layer, is the same as the destination address in the Routing protocol header.

Data Flow at the Destination Node (Node 3)

⑫ The packet passes to the Routing layer of the destination node, Node 3, as described in steps 7 and 8 above. The destination Routing layer checks the Routing header (step 9), but this time passes the datagram up to the next DNA layer, the End Communication layer.

⑬ The End Communication layer strips the original End Communication layer's protocol header information from the datagram. If the End Communication layer has the resources to form a new logical link, it will pass the connect data to the Session Control layer.

⑭ The Session Control layer then removes the Session Control protocol header information and performs any necessary access control functions. It then passes the message to the appropriate process in the User layer.

⑮ The destination process interprets the data according to whatever level protocol is being used by the destination node user.

1.5 NOTES ON THE PHYSICAL LINK LAYER

The Physical Link layer includes parts of the Device Driver software for each communications line controller device, and the communications hardware. The communications hardware includes line interface devices (line controllers), modems, and the physical communications line.

In this, the lowest of DNA layers, industry standard electrical signal specifications such as EIA RS-232-C, CCITT V.24, the Ethernet physical layer, or CCITT X.25 level 1 operate, rather than predefined DNA layer protocols. DNA only defines the following functions for the Physical Link layer:

- Interface between the Network Management and Physical Link layers
- Type of information to be stored by physical line counters
- Which hardware events to pass up to the higher DNA layers for interpretation, storage, or action
- Monitor physical line signals
- Clock data to and from the physical line
- Handle interrupts from the communications hardware device
- Inform the Data Link layer when a transmission or reception of data is completed

MODULE TEST

Answer the following questions by circling the letter next to the best possible solution. After you have finished the test, check your answers against the Answer Sheet provided in your Tests and Answers booklet. Do not proceed to the next module until you have correctly answered all of the following questions.

1. DNA:

   a. Provides the detailed functional specifications for DIGITAL computer networks.

   b. Defines a network model that permits all DECnet products to share the same data communications network.

   c. Defines an industry-wide model of a data communications network.

   d. Provides the product specifications for connecting to DEC systems.

2. Protocol is:

   a. The DNA interface between adjacent layers at the same node.

   b. A special message sent over the network to control data transmission.

   c. A special node in the network.

   d. The DNA interfaces between the same layer in different nodes in the network.

3. The DNA is layered because:

   a. It could not be modeled any other way.

   b. It permits the model flexibility to change with the evolution of new technology.

   c. It permits the model to be fixed and inflexible. Thus Digital Equipment Corporation can sell entire DNA structures even when small changes in technology occur.

   d. No other network architecture uses it.

4. A major DNA design goal is to:

   a. Provide a network that costs less than any custom-developed network.

   b. Support only high-speed data communications devices.

   c. Permit user-defined network configurations. Thus networks can be structured to meet the user's needs.

   d. Provide services for limited size networks. The architecture limits the number and types of nodes that can be placed in the network.

5. Which of the following characterizes DNA/DECnet (Phase IV) supported networks?

   a. They must be ring-structured networks.

   b. They do not have to be formally structured.

   c. They must be structured only as hierarchical-tree networks.

   d. They are limited to circuit-switching communication operations.

6. Which of the following best describes the DNA Data Link layer?

   a. Provides the network with logical links.

   b. Provides the functions used to plan, control, and maintain the operation of the network.

   c. Provides error-free physical links between adjacent nodes in the network.

   d. None of the above.

7. The Routing layer interfaces with which of the following DNA layers?

   a. End Communication, Network Management

   b. Session Control, Data Link, and Network Management

   c. End Communication, Data Link, and Network Management

   d. Data Link, Session Control

8. Which of the following protocols resides in the DNA User layer?

   a. DAP

   b. DDCMP

   c. NSP

   d. None of the above

. The User Level directly interfaces to which of the following DNA layers?

a. Session Control, Network Management

b. Session Control, Network Application, and Network Management

c. Session Control, Routing, and Network Application

d. Network Management, Network Application, and End Communication

). The NICE protocol resides in which of the following DNA layers?

a. Network Management

b. Session Control

c. Data Link

d. Network Application

1. What is Data Message enveloping?

a. Appending protocol header information to the message as it passes through the DNA layers.

b. Stripping protocol header information from the message as it passes through the Physical Link layer of adjacent nodes in the network.

c. Appending and stripping protocol header information from the message as it passes through the DNA layers.

d. Appending and stripping protocol header information from the message as it passes through the Physical Link layer of adjacent nodes in the network.

12. Assume Nodes 1 and 3 are adjacent nodes in a DECnet (Phase IV) network. Node 1's End Communication layer sends a control message to Node 3's End Communication layer. Which of the following DNA layers must this message pass through to be received successfully by Node 3?

a. Node 1, Data Link layer
Node 1, Physical Link layer
Node 3, Physical Link layer
Node 3, Data Link layer
Node 3, Routing layer

b. Node 1, Routing layer
Node 1, Data Link layer
Node 1, Physical Link layer
Node 3, Physical Link layer
Node 3, Session Control layer

c. Node 1, Routing layer
Node 1, Data Link layer
Node 1, Physical Link layer
Node 3, Physical Link layer
Node 3, Data Link layer
Node 3, Routing layer

d. Node 1, User layer
Node 1, Network Management layer
Node 1, Routing layer
Node 1, Data Link layer
Node 1, Physical Link layer
Node 3, Physical Link layer
Node 3, Data Link layer
Node 3, Routing layer
Node 3, End Communication layer
Node 3, Session Control layer
Node 3, Network Management layer
Node 3, User layer

13. What is the primary function of the Network Application layer?

    a. To provide a layer in the DNA model for generalized application software packages that can be used by all users.

    b. To provide the user the ability to plan, control, and maintain the network.

    c. To allow two user programs to exchange data over a logical link.

    d. To allow two users to exchange data over a physical link.

Match the following questions with the correct answer from the list below.

14. Which DNA layers are made transparent to the Network user by the DNA User layer?

15. Which DNA layers provide the user with an interface to the other DNA layers?

    a. User Layer
       Network Management Layer
       Network Application Layer

    b. Session Control Layer
       End Communication Layer
       Routing Layer
       Data Link Layer
       Physical Link Layer

    c. Network Management Layer
       End Communication Layer
       User Layer

    d. End Communication Layer
       Network Application Layer
       Session Control Layer
       User Layer

DATA LINK CONTROL

DATA LINK CONTROL

INTRODUCTION

This module introduces, describes, and illustrates the operations performed, and message formats used, by the Data Link Control layer of the DNA.

OBJECTIVES

To use DECnet in technical support of applications environments, Software Services and Customer Personnel must be able to:

1. Identify the Message Formats used by the DNA's Data Link Control layer.

2. Describe the functional operations performed by the DNA's Data Link Control layer, and the different Data Link modules that make up the Data Link Control layer.

LEARNING ACTIVITIES

1. Study the information in this module.

2. Read Chapter 2, The Data Link layer (Phase IV), in the DECnet DIGITAL Network Architecture (Phase IV) General Description.

3. Take the module test at the end of this module.

4. Correct the test using the Answer Sheet provided in the Test and Answers booklet. Review the material on any questions you may have missed before going on to the next module.

RESOURCES

1. DECnet DIGITAL Network Architecture (Phase IV) General Description
2. DNA Digital Data Communications Message Protocol (DDCMP) Functional Specification, Version 4.1
3. DNA NI Data Link Architectural Functional Specification, Version 1.0
4. DNA NI Node Product Architectural Functional Specification, Version 1.0
5. DNA X.25 Frame Level Functional Specification
6. DNA X.25 Packet Level Functional Specification

2.1 LAYER PURPOSE

The Data Link Control layer has two major purposes:

1. To create an error-free physical link between two adjacent network nodes.
2. To pass data over the physical link.

2.2 LAYER FUNCTIONS

The Data Link Control layer provides DNA with the following functions:

- Creates the communication path between two adjacent nodes.
- Frames messages for transmission on the channel connecting the two adjacent nodes.
- Checks the integrity of received messages.
- Manages the use of channel resources.
- When required, ensures the integrity and the proper sequence of transmitted data.

2.3 LAYER INTERFACES

There are five layer interfaces defined between the Data Link Control layer and the other DNA layers. Two of these interfaces are to its adjacent layers, the Physical Link layer and the Routing layer. Two other interfaces are to the Network Management layer, one for network management functions and the other for network maintenance functions. The fifth interface is to the Network Application layer for Systems Network Architecture (SNA) Gateway functions. Refer to Module 1, DNA Overview, Figure 1-2, for a depiction of these interfaces. The commands and responses that perform the layer interfacing differ depending upon the Data Link Control layer protocol module being used. There are three Protocol Modules currently residing within the Data Link Control layer:

1. DIGITAL Data Communications Message Protocol (DDCMP)
2. CCITT X.25 Levels 2 and 3
3. Ethernet Data Link

2.4 DDCMP MODULE

For a more indepth study of the DDCMP module and its interactions with DNA, read:

1. Chapter 2, Section 2.1, in the DECnet DIGITAL Network Architecture (Phase IV) General Description.
2. Chapters 1-7, Appendices A-G, in the DNA DIGITAL Data Communications Message Protocol (DDCMP) Functional Specification, Version 4.1.

2.4.1 DDCMP Functional Description

DDCMP is a general-purpose protocol that operates on a variety of communication systems from the very large to the very small. 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. DDCMP can handle this transparent data efficiently because it is a byte-oriented protocol. A byte-oriented protocol provides a count of the number of bytes that will be sent in the data portion of each message sent.

DDCMP controls the Physical Link layer operation. It transmits data grouped into physical blocks known as Data Messages. The primary function of DDCMP is to ensure data integrity and the sequentiality of data transmitted over a single physical link. Physical links are called Channels or DDCMP Circuits. Computers using this protocol are able to correctly exchange data over a physical network line. DDCMP ensures that the data is exchanged without errors caused by the physical network line. The DNA layers located above the DDCMP module interpret this data once it is correctly exchanged.

Users or Programs wishing to communicate using DDCMP must agree on the syntax of the data transmitted over the channel. DDCMP may be viewed, as shown in Figure 2-1, as an envelope that encloses the data message. The message to be transmitted is enclosed by a message header and a message trailer. The terms message header and trailer are discussed later in this section.

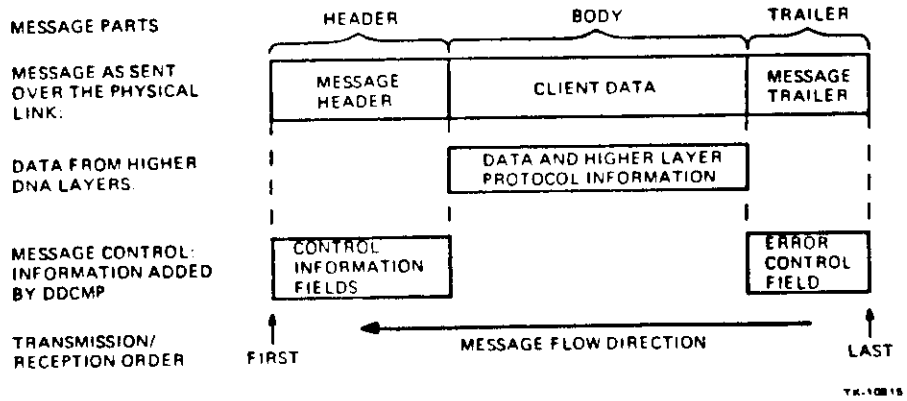


Figure 2-1 DDCMP Message Envelope

2.4.2 DDCMP Functional Operations

DDCMP provides a mechanism for exchanging error-free messages over the network line. This mechanism works as follows:

DDCMP assigns a number to each data message beginning with number zero, after each initialization, and increments by a count of one for each subsequent data message sent. The message numbers start at 0 and sequentially count up until a decimal count of 255 is reached. Once this count of 255 is reached, the next count causes a reset back to 0 so that the next data message sent will have a decimal count of 0. This numbering method permits DDCMP to have up to 256 outstanding unacknowledged messages. This is a useful feature of DDCMP when working on high delay circuits such as those using satellite links.

DDCMP places a 16-bit cyclic redundancy check (CRC-16) character at the end of each DDCMP header and at the end of the data portion of each message transmitted. This CRC-16 character is an error detection polynomial that performs the same basic functions as vertical and horizontal parity check circuits. The CRC-16 method of error detection is much more accurate than either the vertical or horizontal parity error checking methods.

The receiving DDCMP module checks for errors and, if there are none:

1. Passes the received message to the next higher DNA layer
2. Returns the message number with a Positive Acknowledgement (ACK) indicating proper message reception to the message transmitting station.

The receiving DDCMP module need not acknowledge each message received. Acknowledgement of data message (n) implies acknowledgement of all data messages up to and including data message (n).

If an error is detected by the receiving DDCMP module, one of the following happens:

1. time-outs occur
2. control messages are sent
3. data messages resynchronize and trigger automatic retransmission of the message or sequence of messages that contained errors.

These DDCMP functional operations can be grouped into three basic areas:

1. Framing
2. Link Management
3. Message Exchange

2.4.2.1 Framing - Framing is the process of locating the beginning and ending bytes of a message at the receiving end of a physical link. Framing is accomplished by synchronizing the receiving station's communications device receiver circuits with the message sent by the transmitting station. This synchronization must occur at the bit, byte, and message levels before framing is complete. DDCMP uses a special 8-bit byte called a synchronization character (SYN); the character used is 96 hexadecimal. This SYN character precedes all transmitted messages to allow the receiver time to synchronize with the bytes in the transmitted message.

2.4.2.2 Link Management - DDCMP manages the Physical Link between network nodes using the selection flag in the DDCMP message header. This link management is vital to half-duplex, point-to-point, and multipoint network links. Full-duplex, point-to-point network links do not require line management transmit and receive coordination. Full-duplex topologies provide each station's transmitter its own line to transmit messages. Figures 2-2 through 2-5 show the basic network link types.

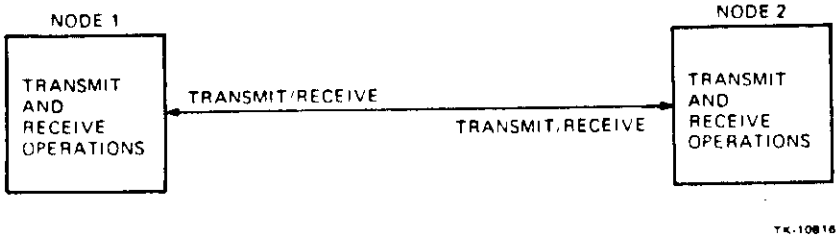


Figure 2-2 Half-Duplex, Point-to-Point Link

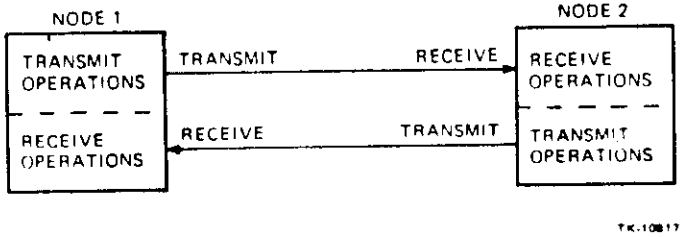


Figure 2-3 Full-Duplex, Point-to-Point Link

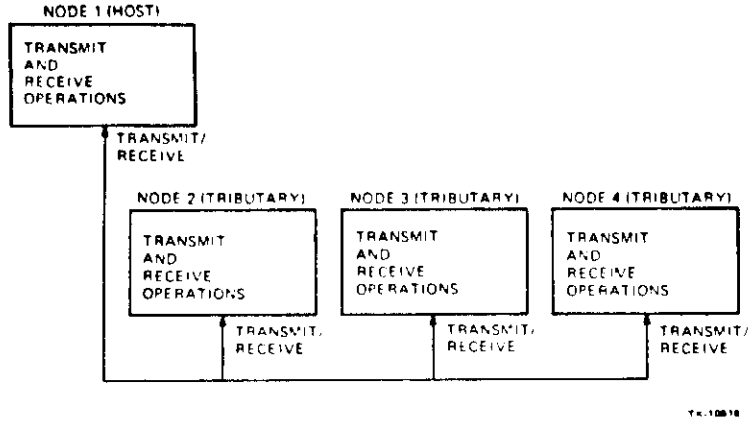


Figure 2-4 Half-Duplex Multipoint Link

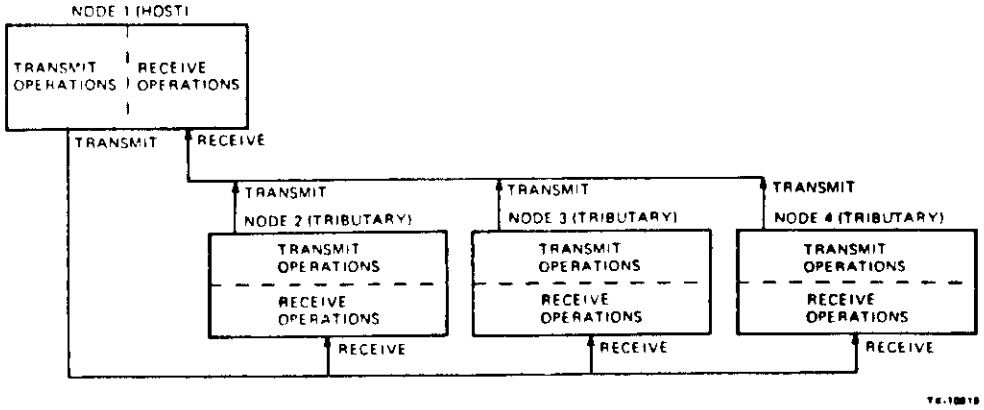


Figure 2-5 Full-Duplex Multipoint Link

In half-duplex, point-to-point links, the two adjacent nodes cannot transmit and receive at the same time. They must share the same physical line, and control who is transmitting at any given time over the line. Line control is accomplished via the selection flag. The selection flag is contained within the DDCMP message header for all DDCMP message types. The station currently transmitting must set the flag in the last transmitted message to relinquish control of the line. Receiving the selection flag informs the receiving station that this message is the last message from the transmitting station, and that the receiving station now has control of the line. Figure 2-6 shows an example of line management on a half-duplex, point-to-point link.

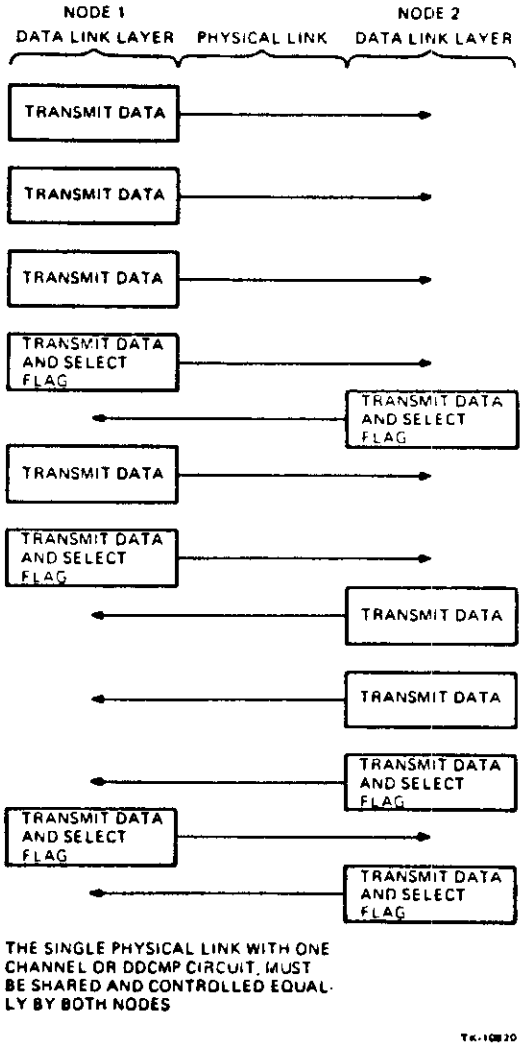


Figure 2-6 Line Management for Half-Duplex, Point-to-Point Links

In multipoint links there are more than two stations attached to the physical line. Thus, there is more than one channel, or DDCMP circuit, on the physical network line. To control the use of the physical line, one of these stations must be designated as the Network Controller, or Host Station. This station is responsible for all network line usage. The Host station uses the selection flag much in the same way as the half-duplex, point-to-point stations to control line access and use. The major differences between point-to-point and multipoint line control are:

- Only the Host station can select another station (a Tributary)
- Tributaries can only transmit on the line when selected by the Host station
- All tributaries are in the receive mode until selected for transmitting by the Host station
- All tributaries have a unique address, or list of addresses to which they respond
- Tributaries only receive messages that are addressed to them
- Tributaries must use their own unique address when transmitting to let the host know which tributary sent the message
- When the selected tributary is finished transmitting, it must set the selection flag in its last message to return control of the line back to the Host station
- Transmitting or receiving messages directly between tributary stations is not permitted; messages must first be routed through the Host station.

Figure 2-7 shows an example of line management on a multipoint link.

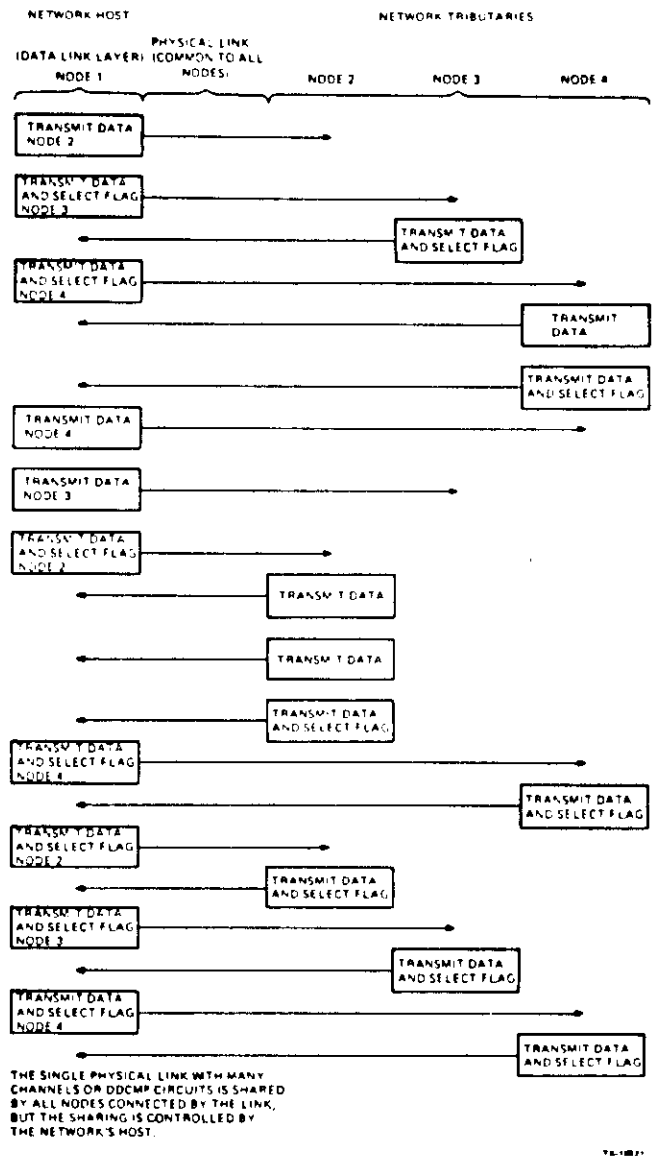


Figure 2-7 Line Management for Multipoint Links

In both half-duplex, point-to-point and multipoint links, a timer is used to handle a lost selection flag. The Selection Interval Timer is started when the transmitting (Host) station sends a message with the selection flag set to the receiving (tributary) station.

If this timer expires before the other station begins to transmit, it is assumed by the first station that the selection flag was received in error. The station whose timer expires then continues to operate in the transmit mode as if it had received a valid selection flag from the other station.

Figure 2-8 shows an example of a selection interval timer expiring in a half-duplex, point-to-point link.

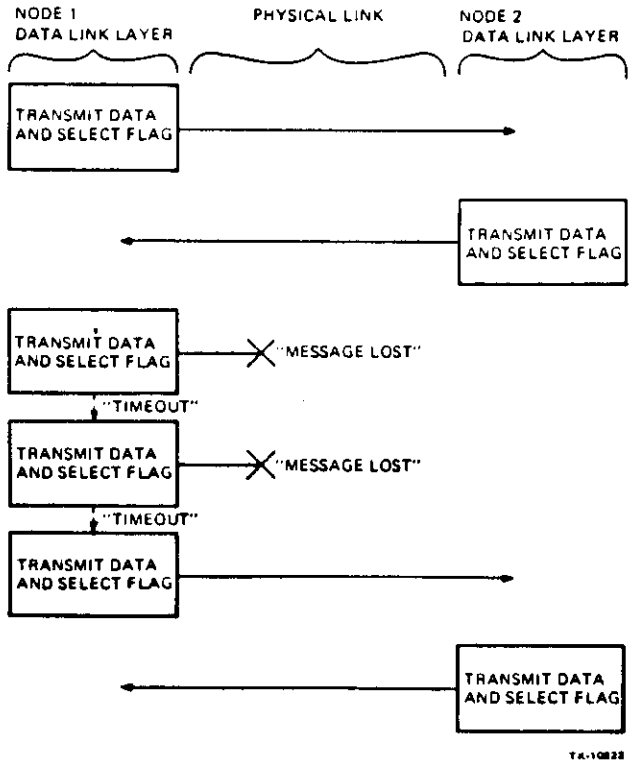


Figure 2-8 Expired Selection Interval Timer

2.4.2.3 Message Exchange - Message Exchange is the process of sending sequential error-free messages over the physical line. This process exchanges data and control messages after message framing is achieved.

DDCMP is a positive acknowledgement with retransmission protocol. For each correctly received data message it:

- Passes the message up to the next higher DNA layer
- Returns a Positive Acknowledgement to the message sender

The acknowledgement tells the message sender that the message was received correctly.

A message is acknowledged by an Acknowledge Message (ACK), or by a piggy-backed acknowledgement in the DDCMP Data Message Header. Piggy-backing is a technique used by DDCMP that allows a station to acknowledge a message or list of messages by sending the ACK within the message header of the next data message transmitted.

Instead of using valuable link time to send a separate ACK message, the receiving station may use its next normally transmitted data message to convey the acknowledgement. Control messages (ACKs) are overhead to the network user; they cannot carry any user data. The network that uses the least amount of overhead for data transfers is the most efficient. Figure 2-9 shows an example of data message acknowledgement using ACKs and piggy-backed ACKs within a returned data message.

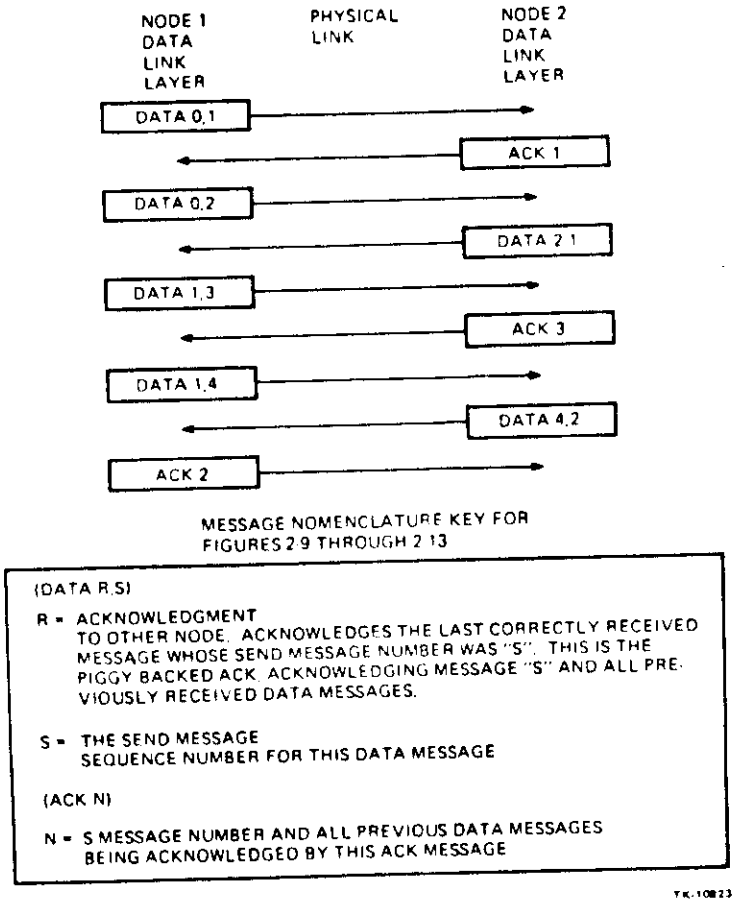


Figure 2-9 DDCMP Data Transfer and Acknowledgement Without Errors

Another technique used by DDCMP to improve link performance is Pipelining. Pipelining sends more than one data message without waiting for ACKs to each successive message. ACKs confirm the proper receipt of the specified message number in the ACK message. They also confirm the proper receipt of all previous messages between the message currently acknowledged and the last message acknowledged. Pipelined messages can be acknowledged by ACKs or by piggy-backed ACKs. Figure 2-10 shows an example of pipelined data messages being acknowledged by both ACK and piggy-backed ACK messages.

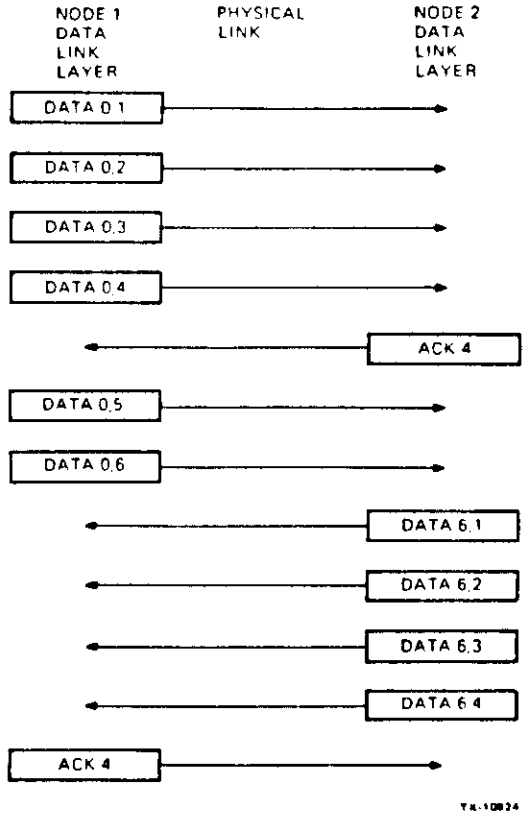


Figure 2-10 DDCMP Pipelining Without Errors

If the message is received in error or not received at all, (a lost message), it cannot be passed up to the next higher DNA layer, and is not acknowledged. Eventually, an error recovery mechanism is invoked and the message is retransmitted over the link.

DDCMP uses both timers and control messages for error recovery. When a station transmits a message, it starts a timer known as the Reply Timer. Until this timer expires the receiving station must either Acknowledge (ACK) or Negative Acknowledge (NAK) the transmitted message.

If the transmitted message is lost (not received), the receiving station cannot send either the ACK or the NAK to the sender. In this case, the transmitting station's reply timer will expire and cause the transmission of a control message called Reply. The receiving station can then ACK or NAK the Reply message. If NAKed the transmitting station will retransmit the original message or messages. The NAK message informs the transmitting station of the last message number received correctly via an implied ACK within the NAK message header. If the receiving station ACKs the Reply, the transmitting station assumes that the message(s) was/were received correctly.

If a message is received in error, the receiving station sends a NAK to the transmitting station. The message transmitting station automatically retransmits the message(s) in error. NAKs provide immediate notification of some error conditions and, therefore, eliminate waste and inefficiencies encountered by waiting for time-outs. NAKs are used to report errors if the receiving station recognizes that a message was sent but for some reason cannot correctly process the message. Figure 2-11 shows an example of automatic message retransmission using both time-outs and NAK messages.

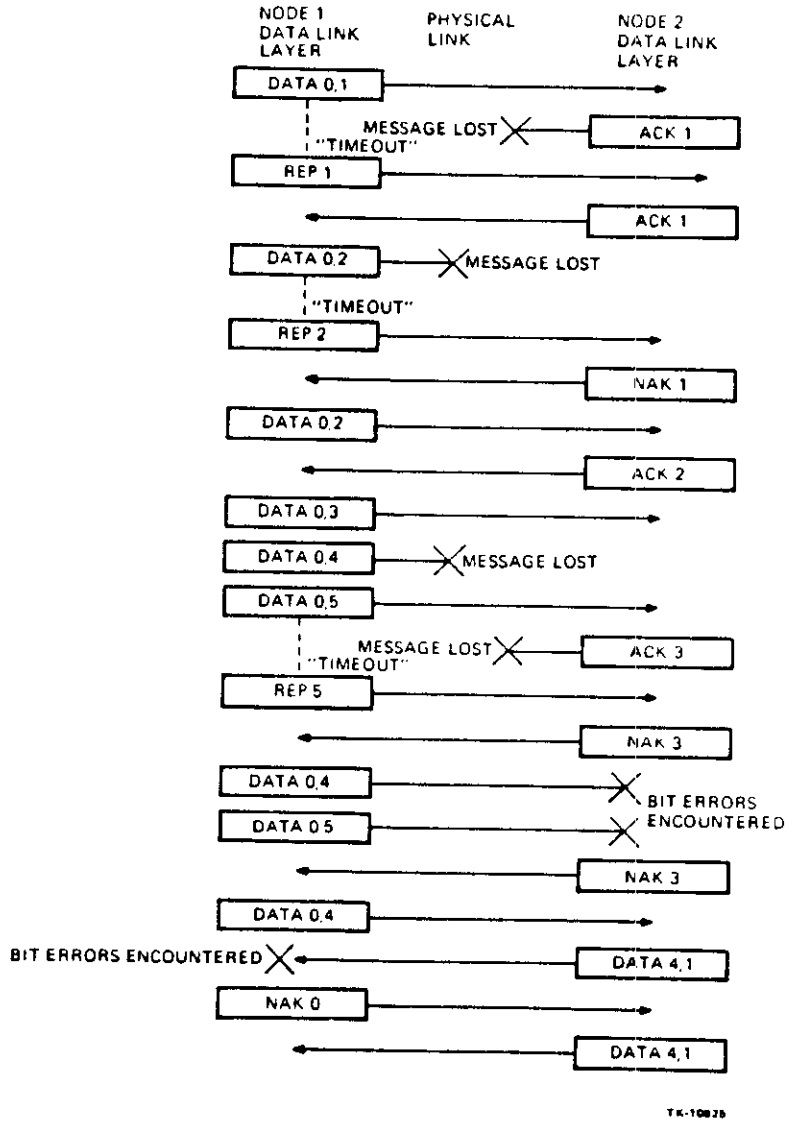


Figure 2-11 DDCMP Data Transfer with Retransmission Due to Errors

The DDCMP message exchange component is also responsible for the Link Initialization. DDCMP incorporates two control messages specifically for link initialization:

- Start DDCMP (STRT)
- Start Acknowledgement (STACK)

Link initialization is the process of obtaining synchronization between two adjacent stations on a line. Link initialization is used after a failure to reset message number values at both stations by error recovery procedures, or when the channel or circuit is first established on the line. When synchronization occurs message numbers on both stations are set to zero.

Figures 2-12 and 2-13 show the sequence of events necessary to initialize a channel or circuit between two adjacent nodes with and without errors caused by the physical line.

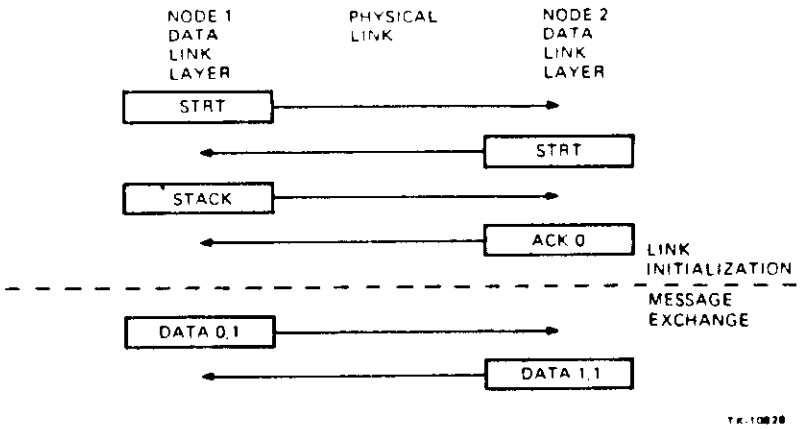


Figure 2-12 DDCMP Startup Without Errors

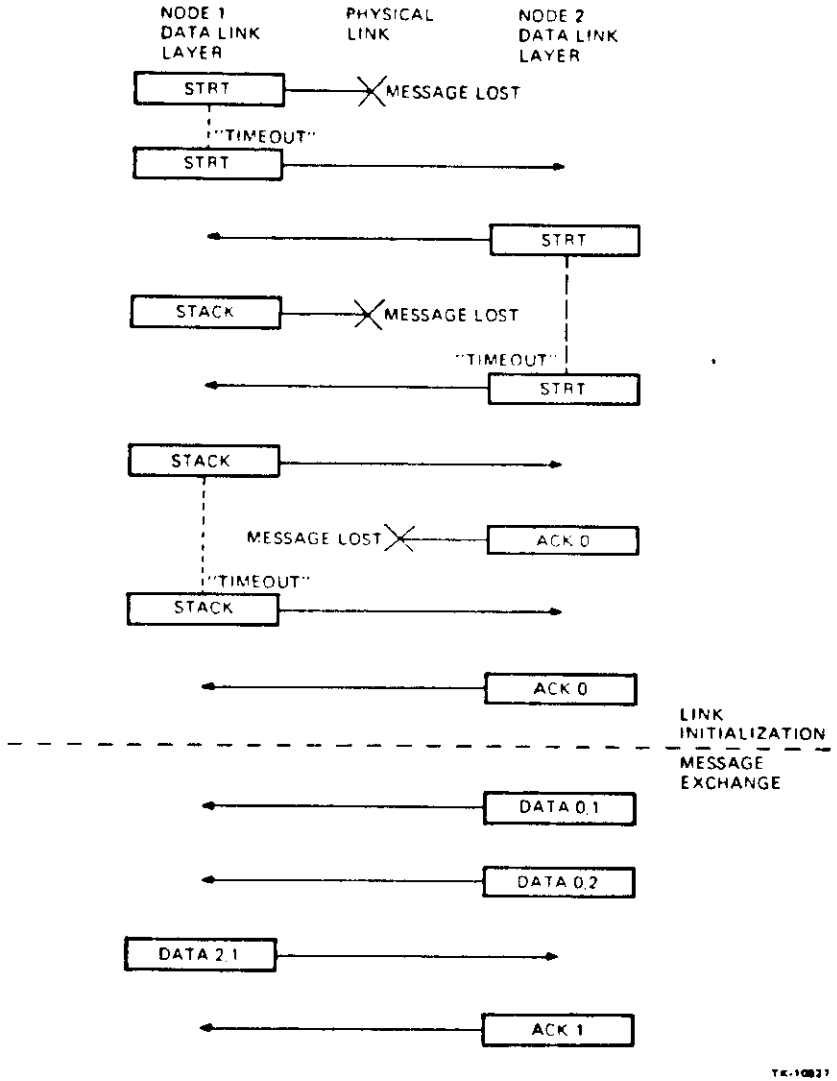


Figure 2-13 DDCMP Startup with Errors

2.4.3 DDCMP Message Formats

There are three types of DDCMP Messages:

1. Data messages
2. Control messages
3. Maintenance messages

Figure 2-14 shows the DDCMP message enveloping used by the three different DDCMP message types.

The message header in all three types of DDCMP messages:

1. Classifies each message sent into one of three categories
   - Data message (SOH) - Hex 81
   - Control message (ENQ) - Hex 05, and

     Positive Acknowledgement (ACK) - Hex 0501
     Negative Acknowledgement (NAK) - Hex 0502
     Reply for packet request (REP) - Hex 0503
     Start DDCMP (STRT) - Hex 0506
     Start Acknowledgement (STACK) - Hex 0507

   - Maintenance Message (DLE) - Hex 90
2. Provides an octal count equal to the number of data bytes in the data portion of the message, if a data or maintenance message
3. Controls transmission on the physical line for multipoint and point-to-point, half-duplex links
4. Sequentially numbers each correctly received message
5. Sequentially numbers each transmitted message
6. Provides a destination address for each message sent on point-to-point and multipoint links
7. Ensures that the message header is not received in error or out of sequence

The DDCMP message trailer used with Data and Maintenance messages ensures that the data portion, which follows the message header, is received without errors.

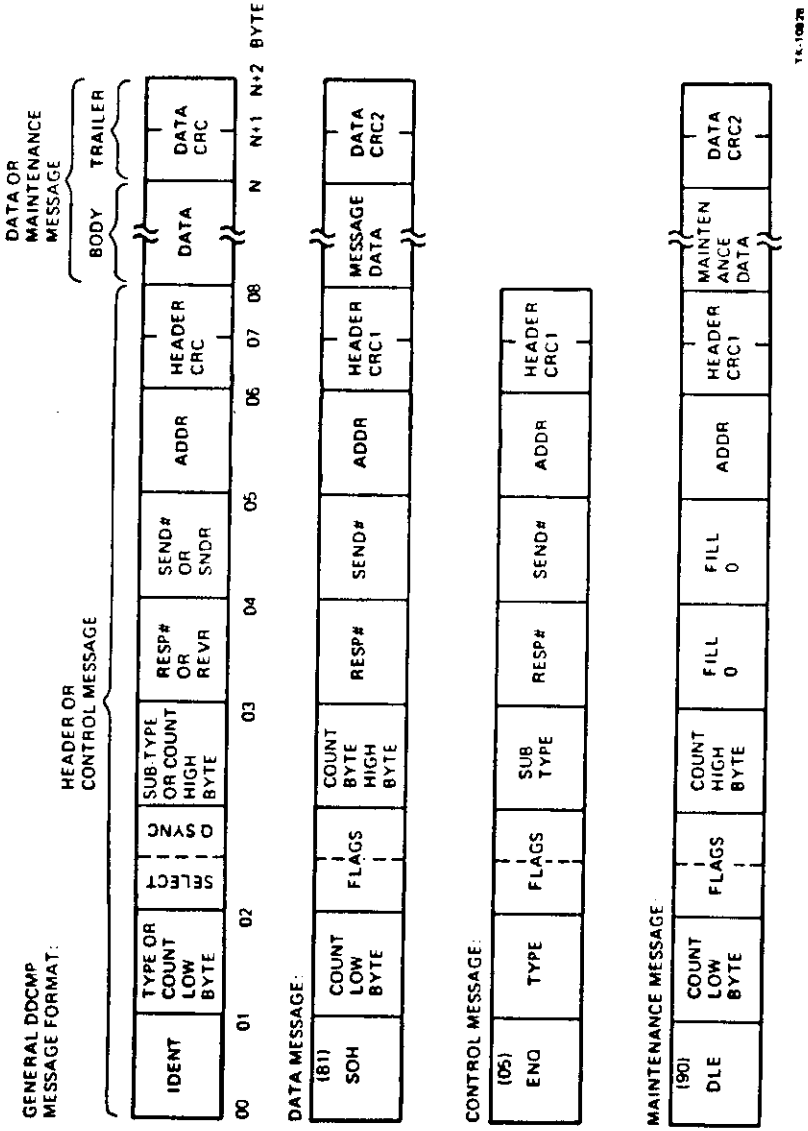


Figure 2-14 DDCMP Message Types

2.5 X.25 MODULE

For details concerning the X.25 Module and its interactions with DNA, read:

1. Chapter 2, Section 2.3, in the DECnet DIGITAL Network Architecture (Phase IV) General Description
2. Chapters 1-7, Appendices A-C, in the DNA X.25 Frame Level Functional Specification
3. Chapters 1-8, Appendices A-E, in the DNA Packet Level Functional Specification

2.5.1 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-Switching Mode. The DTE is a programmable device that is the User Side of the "user to network" interface. The DCE is the Network Side of the "user to network" interface.

The CCITT X.25 Packet-Switching Interface recommendation is structured into three different levels:

- Level 1 - The Physical level defines the interconnection characteristics between the DTE and the DCE. These characteristics are the mechanical, functional, and procedural rules for transmitting data between the DTE and the DCE. This level of X.25 is similar to the EIA recommendation RS-232-C in that they both define the rules by which the Physical Link layer is to operate. Level 1 resides in the Physical Link layer of DNA.

- Level 2 - The Frame level defines the rules for transmitting data between the DTE and the DCE. This level envelopes the packet in control information, prescribes the procedures necessary to ensure a high degree of transmission accuracy, and provides the detection of lost frames during transmission. Level 2 defines the link access procedure for the data exchange over the physical link between the DTE and the DCE. Level 2 resides in the Data Link layer of the DNA.

- Level 3 - The Packet level defines the rules for transmitting and controlling the packet between the DTE and the DCE. This level describes the packet format and length as well as the control procedures for exchanging packets between the DTE and the DCE. Packet level actions at one DTE affect actions at a corresponding DTE over the Public Data Network. Packets can be made up of user (client) data, or control information (administrative messages). Level 3 also resides in the Data Link layer of the DNA.

Figure 2-15 shows a typical CCITT X.25 Interface Standard DTE/DCE configuration. Figure 2-16 shows the logical and physical link connections between the DNA X.25 Physical, Frame, and Packet level modules resident in a DTE and DCE station. It also shows the available software interface paths to the Frame and Packet levels for user data exchange, network testing, and network management functions.

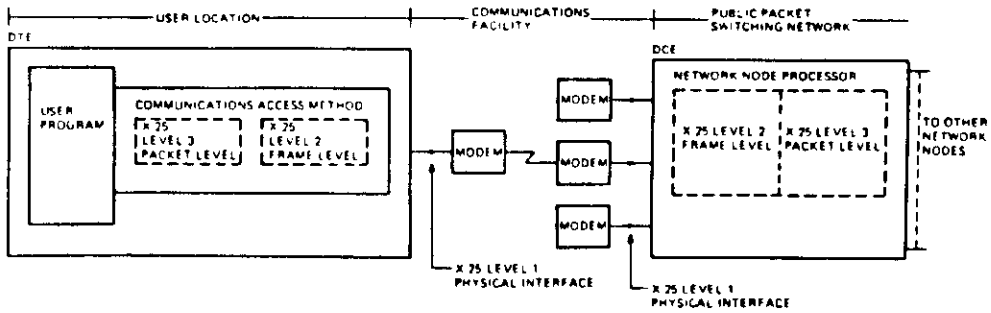


Figure 2-15 Typical CCITT X.25 DTE/DCE Configuration

Figure 2-16 shows that several client modules may simultaneously use the X.25 packet level interface, each module being generally unaware of the existence of the other modules sharing the interface. In this model, the X.25 packet level may have multiple frame level links into a single public data network, each link, or channel, associated with a single packet level DTE, and appearing as one or more DTEs to the public data network.

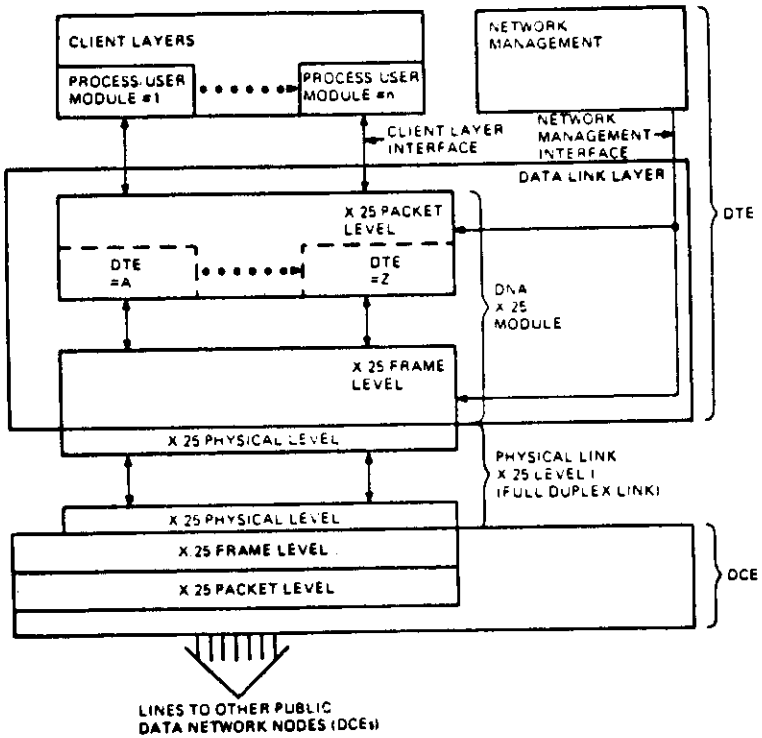


Figure 2-16 X.25 Physical, Frame, and Packet Level Relationships

CCITT X.25 is recommended for use with Packet-Switching networks. Packet-switching is the transfer of data by means of addressed data packets of a length independent of the original user's data message length. Because addresses are carried in each data packet, or data segment, circuit connections can be handled dynamically for each packet. A fixed connection need not be maintained for any particular call or calls. This eliminates wasting network bandwidth when waiting for replies to messages. The public packet-switching network (PPSN), called the Public Data network, handles all DCE to DCE circuit establishment and clearing. The user need only deliver the packets to the Public Data network. Management and control of message delivery is provided by the network.

In a DECnet network (built around the DNA structure) there are two independent uses for X.25 modules in the Data Link layer:

1. To link two DECnet systems for data communications at the Data Link layer level

2. To be a gateway to non-DIGITAL systems that are accessible via the Public Data Network for data communications.

A single X.25 link can support multiple virtual circuits. DNA Routing may use several virtual circuits for communicating between DECnet nodes (see Module 3), while DNA X.25 Gateway Access may use other virtual circuits for communicating with non-DIGITAL systems (see Module 6).

A virtual circuit (VC) or Switched Virtual Circuit (SVC) is a logical circuit connection between two nodes, (DTEs) for bidirectional transmission of multiple, contiguous packets of user, or client data. The Physical path over which the packets are transmitted may vary from message to message. The logical circuit concept of the virtual circuit assures presentation of each in the proper order to the receiving DTE node. The logical circuits used for this method of communication are established dynamically by the network as each message is presented to the network for transmission to its destination DTE.

DNA X.25 also supports Permanent Virtual Circuits (PVCs), which are logical circuits with fixed channel numbers. These channel numbers identify a permanent logical connection between your station, DTE, and another station, DTE, through the network. The transmission circuits for PVCs can be dynamically established between the DTEs as are the virtual circuit (VC) channels. However, PVCs do not need to exchange any call connection messages prior to exchanging data as do SVCs. This is because the logical channel numbers that identify the logical circuit connection are fixed.

2.5.2 X.25 Functional Operations

Together, the DNA X.25 and DCE X.25 modules are responsible for the bidirectional error-free exchange of user data between the user's DTE and the Public Data Network, DCE. The operations performed by the Frame and Packet level modules are, for ease of understanding, described separately in this course.

2.5.2.1 X.25 Frame Level - CCITT X.25 Frame level modules on the DTE and DCE are connected by a physical link. This link can be either through telephone lines and modems or through coaxial cable connections. This link must be a full-duplex line that connects a Frame level module resident in the DTE with a Frame level module resident in the DCE.

The logical, Frame level link channel between the DTE and the DCE can be viewed as two independent channels connecting a primary station in one Frame level module with a secondary station in the other Frame level module. Frames transmitted from the primaries are called commands, and frames transmitted from the secondaries are called responses. Frames that are transmitted between the DTE primary and the DCE secondary are addressed as "b". Frames transmitted between the DTE secondary and the DCE primary are addressed as "a". Figure 2-17 illustrates the logical channel connections created on the physical link when connecting the DTE and DCE Frame level modules.

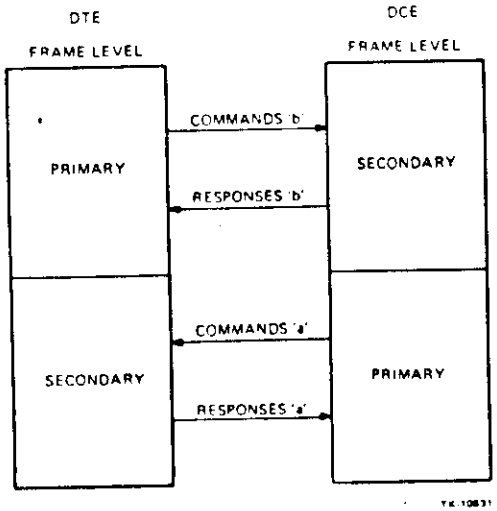


Figure 2-17 Frame Level Link Connections

Frame level modules resident in the DTE and DCE communicate via Frame messages. Frame messages provide data exchange, link, flow, and error control. There are three types of Frame messages used for DTE and DCE communications:

1. **Information Frames (I Frames)** - Used to transfer sequentially numbered frames with an information field, called a Packet, over the channel between the DTE and the DCE. Information frames are also used to acknowledge previous correctly received Information frames. This acknowledgement is done in a manner similar to that of piggy-backing used by the DDCMP module. The Information frame contains two sequence numbers; one for the current I frame being transmitted, and another that indicates the next I frame message sequence number expected from the other station. The next expected I frame number is an implied acknowledgement of the last correctly received I frame at this station.

2. **Supervisory Frames (S Frames)** - Supervisory frames are used to perform channel control functions. All S frames acknowledge the last correctly received I frame. The primary functions performed by the S frame are:

   Acknowledge received I frames

   Indicate the state of the transmitting station

   Request the state of the receiving station

   Station states that can be transmitted or requested by S frames are:

   1. Secondary is ready to receive I frames
   2. Secondary is busy and temporarily unable to receive I frames
   3. Transmitting station requires the retransmission of a particular, or a group of I frames. (This S frame message is similar to the DDCMP NAK message.)

   For more detailed information concerning the Link and Station states, refer to sections 4 and 5 in the DNA X.25 Frame Level Functional Specification.

3. Unnumbered Frames (U Frames) - Unnumbered frames are used to change the state of the link, such as connecting or disconnecting the link, between the DTE and the DCE. U frames are also used to acknowledge the proper receipt of U frame command messages. They do not acknowledge the proper receipt of I or S frames. However, they do report error conditions that are not recoverable by the retransmission of the I frame or frames received in error.

The three Frame message types are each broken down into Command and Response subtypes. Command messages are used to initiate an action or function, or to transfer data packets from DTE to DCE or DCE to DTE. Response messages are used to answer a command type message. Table 2-1 lists the three Frame message types, and gives the command and response message functions performed by each type of Frame message.

Table 2-1 X.25 Frame Level Message Types

Message Type	Command Message	Response Message
Information (I)	Information Transfer, User Data	None
Supervisory (S)		Receive Ready (RR) Receive NOT Ready (RNR) Reject (REJ)
	Receive Ready (RR) Receive NOT Ready (RNR) Reject (REJ)	
Unnumbered (U)	Connect Link (SABM) Disconnect (DICS)	
		Unnumbered Acknowledgement (UA)
		Disconnected Mode (DM) Frame Reject (FRMR)

CCITT X.25 Frame level defines two link access procedure protocols for Frame level communication between the DTE and the DCE:

1. Link Access Procedure (LAP)- Defines the DTE/DCE interface as operating in the two-way simultaneous Asynchronous Response Mode (ARM). That is, both DTE and DCE perform primary and secondary functions. LAP is not offered or supported by DNA.

2. Balanced Link Access Procedure (LAPB)- Defines the DTE/DCE interface as operating in two-way Asynchronous Balanced Mode (ABM). LAPB is supported by DNA, and is described in this section.

LAPB at the Frame level:

- Provides an error-free link between the DTE and the DCE by detecting transmission errors and automatically recovering from those errors.
- Provides synchronization to ensure that the frame levels of the DTE and DCE are in step.
- Detects procedural errors and reports them to the User of the link.

These LAPB responsibilities can be grouped into three basic areas:

1. Message Framing
2. Link Management (between the DTE and DCE station)
3. Message Exchange

Message Framing - Consists of locating the beginning and ending bytes of a message, at the receiving end of a physical link. Framing under X.25 is accomplished by appending certain control information fields to both ends of the user data packet. The fields that are appended conform to the High-Level Data Link Control Procedure (HDLC). The appended packets are called frames. Channel synchronization under HDLC is accomplished by synchronizing the receiving station's communications device receiver circuits with the message sent by the transmitting station. This synchronization must occur at the bit, byte, and message levels before framing is complete. CCITT X.25, following HDLC specifications, uses a special byte called a Flag character. The character used is 01111110 (binary) to delimit the starting and ending points of messages.

The flag character precedes and follows all transmitted messages to allow the receiver time to synchronize with the bytes in the transmitted message. The flag can be shared by two consecutive frame level messages. It can be the message ending flag for the first message and at the same time be the message starting flag for the second message transmitted.

Message framing is done in both the Packet and Framing modules of the DTE and DCE. Packet level message framing is discussed later in this module. Frame level message framing, as specified by HDLC, is similar to DDMCP message framing. A user data message, called a data packet, is enveloped by control information by the Frame level prior to transmission, then stripped off by the other station as a function of reception.

The control information appended to the packet by the Frame level is used to control transmissions and perform error checking. This information is broken down into fields similar to the header and trailer fields appended to the user data under the DNA DDCMP Protocol. The fields appended under X.25 are:

- Before the user's data packet (message body):
  1. Flag Field
  2. Address Field
  3. Control Field
- After the user's data packet (message body):
  4. Check Sequence Field (FCS), similar to the BCC CRC-16 in DDCMP
  5. Flag Field

Figure 2-18 illustrates the user's data packet and the fields appended by the Frame level module.

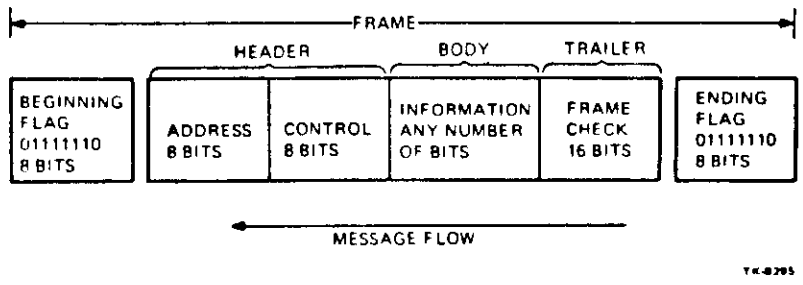


Figure 2-18 X.25 Frame level Information Fields

Link Management - CCITT X.25 manages the link between the DTE and the DCE by connecting and disconnecting the logical channel between the two stations. Link management beyond this at the Frame level is not necessary since the physical link between the DTE and DCE is full-duplex. The logical link, or channel connection, is only connected to, and disconnected from, the DCE in response to a command issued by the user of the link. If the link is disconnected by the DCE, the DTE Frame level will not reconnect the link until requested to do so by the user of the link. The link is established by first entering the disconnect sequence and then, once finished, entering the connect sequence.

The disconnect sequence is the process of transmitting an unnumbered command frame message, Disconnect (DISC). DISC is transmitted at predetermined time intervals until an unnumbered response frame message, Acknowledgement (UA), is received. This action ensures that the link is initially cleared. The connect sequence is the process of transmitting an unnumbered command frame message, Connect (SABM). SABM is transmitted at predetermined time intervals until a UA is received. The predetermined time interval is user-definable, and is called the T1 timer. The T1 timer is used to determine how long the DTE or DCE must wait before retransmitting a frame message if it has not yet received an acknowledgement of that message. Figure 2-19 shows the events that take place during the connect and disconnect sequences.

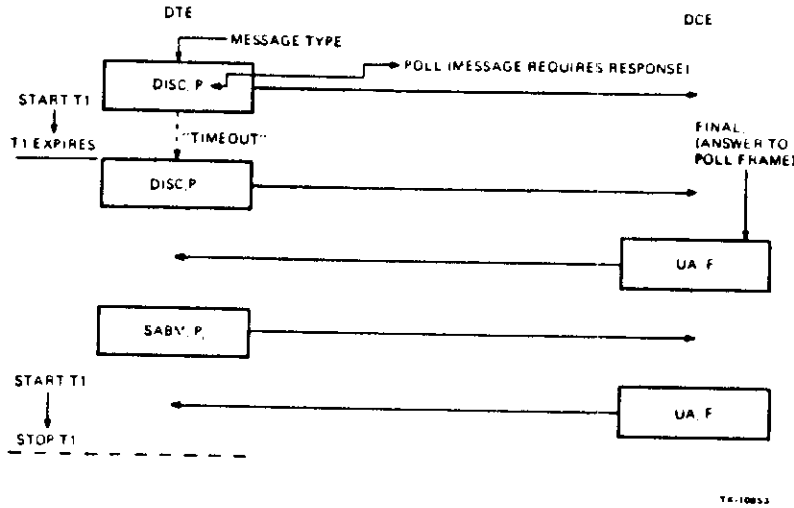


Figure 2-19 X.25 DTE/DCE Channel Disconnect and Connect Sequences

essage Exchange - CCITT X.25 message exchange is the process of ending user data from one network DTE to another. The user data s broken down and formatted by the X.25 Packet level protocol, hen passed to the LAPB for framing and transmission.

nce a connection is established, the Frame level transmits user ata packets over the channel using I frame messages. The eceiving station acknowledges the I frame message once it is eceived correctly. When the I frame is transmitted, the ransmitting station starts its T1 timer. The transmitting tation then expects the other station to either Reject (REJ) or cknowledge proper receipt of the message prior to its T1 timer xpiration. If the timer expires before the message is rejected r acknowledged, the transmitting station automatically etransmits the original message.

and S frame messages have their own frame sequence number. equence numbers can be either modulo 8 or modulo 128, depending pon the PPSN the user has subscribed to. The Sequence numbers re modulo 8 or 128, in that they both start at 0 and count up to or 127 (depending upon the modulo type) and then reset back to 0 or the next message number. DTEs and DCEs use the frame sequence umber to keep track of message frames exchanged over the channel etween each other. The X.25 I and S frame sequence numbers are imilar to the DDCMP message sequence numbers as far as their unctionality.

essages received in error are rejected by the receiving station. hey are not passed up to the Packet level for processing, and the eceiving station's LAPB causes the transmission of the S response rame message REJ. The REJ is transmitted, when one of the ollowing errors is detected:

- Invalid FCS
- Frames received out of sequence
- A busy condition (RNR) cleared, but correctly received I frames were discarded

Figure 2-20 shows the I and S frame messages exchanged during a ormal I frame message exchange sequence with and without errors letected, at the frame level.

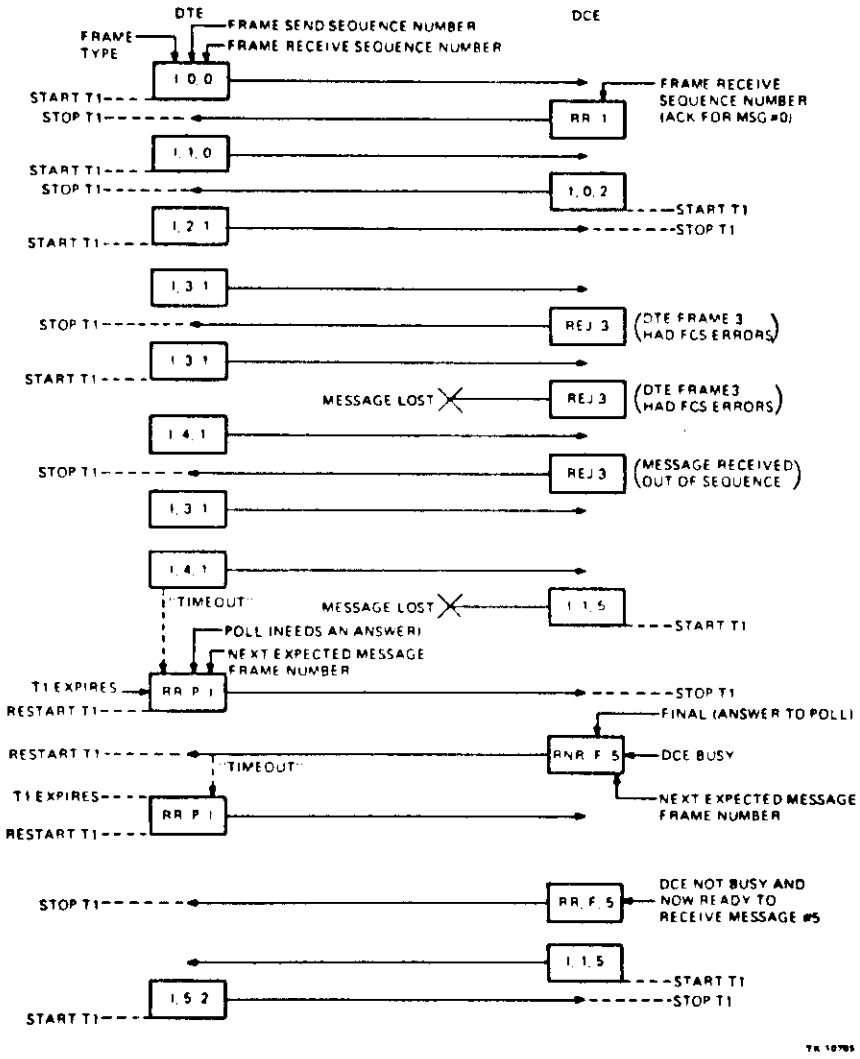


Figure 2-20 X.25 I and S Frame Message Exchange

Other error conditions are possible. These conditions cause the transmission of the unnumbered response frame message Reject (FRMR). FRMR is transmitted when the receiving station detects one of the following types of errors that are not recoverable by the retransmission of the original message.

- The receipt of a command or response message that is invalid or not implemented
- The receipt of an I frame message with an information field that exceeds the maximum size allowed for this DCE/DTE pair
- The acknowledgement of an I frame that has already been acknowledged or has not been transmitted, and is not the next sequential I frame to be transmitted.

When an FRMR message is received, the Frame level records the type of error reported by the FRMR. Transmission of the FRMR is basically a request to reset the channel. The FRMR causes a channel reset. To reset the channel, the receiver must respond with a link Disconnect (DISC) or link Connect (SABM) command message. Any other message is ignored and another FRMR will be transmitted. Figure 2-21 shows the use of an FRMR message sequence to restart the link after receipt of an invalid I frame sequence number.

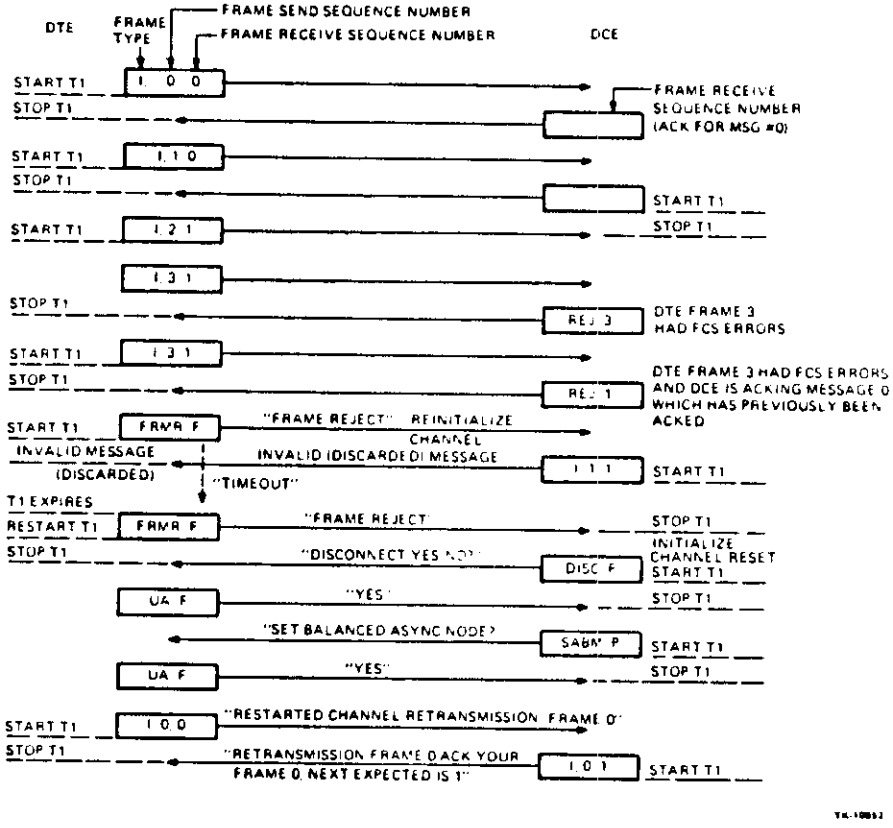


Figure 2-21 X.25 Unnumbered Frame Reject Request (FRMR) Message

2.5.2.2 X.25 Packet Level - The X.25 Packet level provides the network user with multiple virtual circuits between a DTE and other DTEs connected to the Public Data Network. The X.25 Packet level protocol specifies the logical interface between a DTE and a DCE.

Packet level actions at one DTE affect actions at the destination DTE through the Public Data Network via a temporary connection between the user's network DCEs. Activity at the user's DTE affects the local DCE. The local DCE in turn affects the remote DCE, which in turn affects the destination DTE. Packet level messages are only acknowledged by the local DCE; an acknowledgement does not imply that the packet was received by the remote DCE or DTE. Some, but not all, Public Data Networks do provide end-to-end acknowledgement. DNA's X.25 Packet Level module assumes that the acknowledgement of a packet originated at the local DCE.

User DTEs are logically connected over the Public Data Network by a temporary full-duplex, virtual connection. The virtual connection between two DTEs is called a virtual circuit. DTEs are logically connected by a virtual circuit, a temporary connection through the Public Data Network from DTE to DTE. Remember that the DTEs are connected to the network via the Network DCEs. The network DCE is connected to the user's DTE and establishes the connection over the network to another DCE, which in turn establishes a connection to the destination DTE. The logical connection between the DCE and DTE is called a logical channel.

A logical channel is the logical connection between a DTE and a DCE for a given virtual circuit. A logical channel number is assigned at both DTE/DCE interfaces. Each virtual circuit has a logical channel number assigned at both DTE/DCE interfaces. The channel numbers are allocated independently at both DTE/DCE interfaces. Each DTE identifies a virtual circuit by its logical channel number. A DTE may have many virtual circuits and logical channels established at any given time to many different destination DTEs. Figure 2-22 shows the relationship between virtual circuits and logical channels. Figure 2-23 shows how a DTE communicates simultaneously with three other DTEs (B, C, and D) over the public data network.

DNA X.25 supports two types of virtual circuits:

1. Switched Virtual Circuit (SVC)

2. Permanent Virtual Circuit (PVC)

DTEs attached to a virtual circuit access the circuit by exchanging X.25 Packet level messages with their associated DCEs, using the Level 2, Frame level, procedures.

The following figures are used to illustrate Packet level functions and operations, assuming that there are no Frame level errors. It is also assumed that Frame level communication to connect, disconnect, accept, and reject Frame level messages on the physical link are occurring normally and are transparent to the packet level protocols involved. Table 2-2 summarizes the X.25 Packet level message types (the services that they support). Table 2-3 lists the different packet types and summarizes the functions they perform.

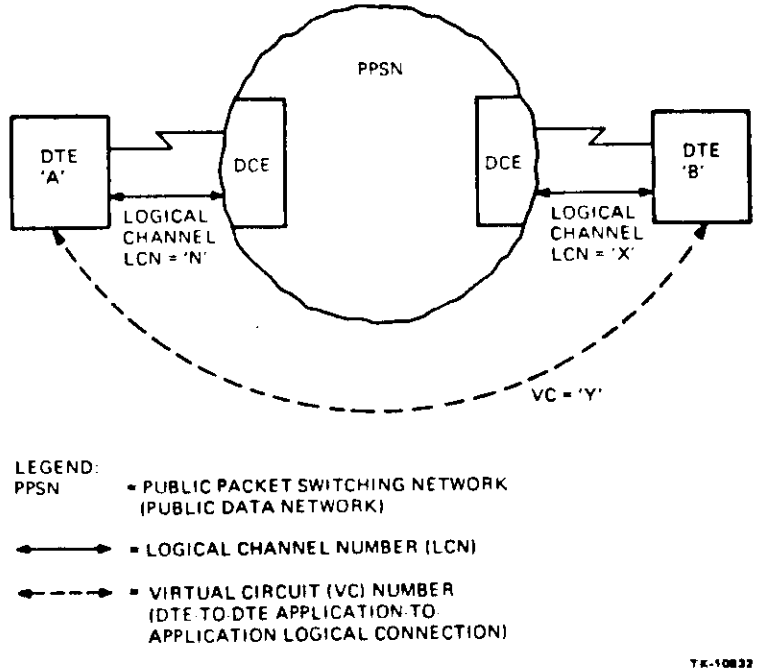


Figure 2-22 Logical Channels and Virtual Circuits

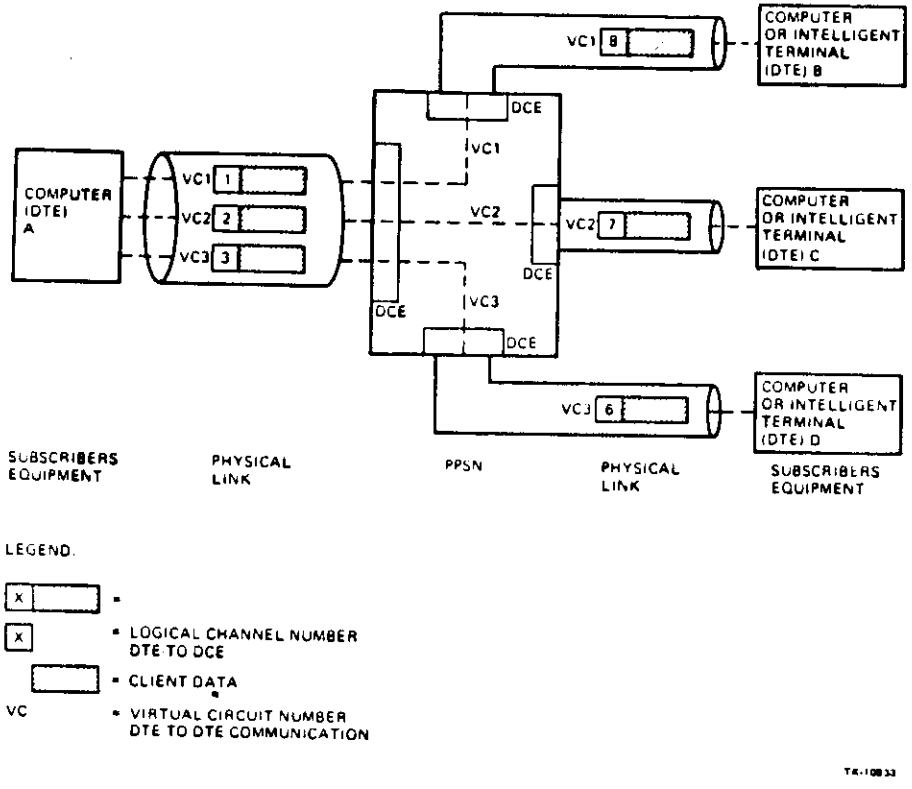


Figure 2-23 DTE to DTE Communication

Table 2-2 X.25 Packet Types and Services Supported

Packet Type		Service Supported		
From DCE to DTE	From DTE to DCE	SVC	PVC	DG**
Call Setup and Clearing				
Incoming Call	Call Request	X		
Call Connected	Call Accepted	X		
Clear Indication	Clear Request	X		
DCE Clear Confirm	DTE Clear Confirmation	X		
Data and Interrupt				
DCE Data	DTE Data	X	X	
DCE Interrupt	DTE Interrupt	X	X	
DCE INT Confirmation	DTE INT Confirmation	X	X	
Datagram				
DCE Datagram	DTE Datagram			X
Datagram Service Signal				X
Flow Control and Reset				
DCE RR (Mod. 8)	DTE RR (Mod. 8)	X	X	X
DCE RNR (Mod. 8)	DTE RNR (Mod. 8)	X	X	X
	DTE REJ (Mod. 8)*	X	X	X
DCE RR (Mod. 128)*	DTE RR (Mod. 128)*	X	X	X
DCE RNR (Mod. 128)*	DTE RNR (Mod. 128)*	X	X	X
	DTE REJ (Mod. 128)*	X	X	X
Reset Indication	Reset Request	X	X	X
DCE Reset Confirmation	DTE Reset Confirmation	X	X	X
Restart				
Restart Indication	Restart Request	X	X	X
DCE Restart Confirmation	DTE Restart Confirmation	X	X	X
Diagnostic				
Diagnostic*		X	X	X

* Indicates that this packet is not necessarily available on every network.
** Not supported by DNA X.25 modules

SVC=Switched Virtual Circuit
PVC=Permanent Virtual Circuit
DG=Datagram

Table 2-3 X.25 Packet Types and Functions

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 the remote DTE, specified in the packet, wishes to establish a virtual circuit with the receiving DTE, and assign a logical channel.
Call Accepted	DTE to DCE	Indicates that the DTE accepts the establishement of the virtual circuit.
Call Connected	DCE to DTE	Indicates that the remote DTE 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's transmit flow control information by transmitted P(R) value (P(R) is also "piggybacked" on data packets).

Table 2-3 X.25 Packet Types and Functions (Cont)

Packet Type	Direction	Description
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 RNR	DTE to DCE	Indicates a temporary inability of the DTE to accept data packets.
DTE REJ	DTE to DCE	Requests retransmission of data packets.
DTE RR	DTE to DCE	Authorizes the transmission of additional DCE Data packets by transmitting a P(R) value to update the DCE's transmit flow control information.
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 the state when the call is established (sequence numbers zero, no data in transit).
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.

Table 2-3 X.25 Packet Types and Functions (Cont)

Packet Type	Direction	Description
DTE Restart Request	DTE to DCE	Request that all logical channels for SVCs for the DTE be deassigned and that all PVCs be Reset.
DCE Restart Confirmation	DCE to DTE	Acknowledges the Restart request.
Restart Indication	DCE to DTE	Indicates to the DTE that all logical channels for SVCs should be deassigned and logical channels for PVC 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 that could not be indicated by retransmission of a packet on that channel (e.g., timeout).

The X.25 Packet level is responsible for the following operations:

- Establishing and destroying Virtual Circuits
- Avoiding call collision
- Transferring user data
- Appending packet sequence numbers to each data packet transferred
- Controlling packet flow
- Delimiting user data messages
- Controlling interrupt data flow
- Resetting Virtual Circuits
- Restarting a Virtual Circuit after catastrophic failures
- Managing permanent Virtual Circuits

These operations are grouped into two major categories:

1. Virtual Circuit Management
2. Data Transfer

Virtual Circuit Management - In response to a user request, the DNA X.25 Packet level module establishes a Virtual Circuit to a specified destination DTE by performing the following events. (Figure 2-24 depicts these events as they occur.)

❶ The user's local X.25 packet level assigns a logical channel number and transmits a Call Request packet. The Call Request contains the channel number to be used between the local DTE and DCE; it also contains the DTE destination address.

❷ The local DCE receives the Call Request and forms the virtual circuit to the remote DCE, forwarding the call request packet information.

❸ The remote DCE assigns a logical channel number and transmits an Incoming Call packet to the remote DTE.

❹ The remote DTE accepts the call by transmitting a Call Accepted packet over the assigned logical channel to its local DCE. (To reject the incoming call, the remote DTE transmits a Clear Request packet as in Figure 2-25.)

❺ The remote DCE forwards the Call Accepted information, or Clear Request information, to the user's local DCE.

❻ The user's local DCE informs the user's local DTE that the call was accepted by transmitting a Call Connected packet on the assigned channel. If the call was rejected the local DCE sends a Clear Indication packet to the local DTE over the assigned channel.

Note that in Figure 2-24, the remote DTE that accepts the call sends data packets to the local DTE before the local DTE can send any data. The local DTE must wait for the Call Connected packet before it can send any data to ensure that the call is accepted by the remote DTE.

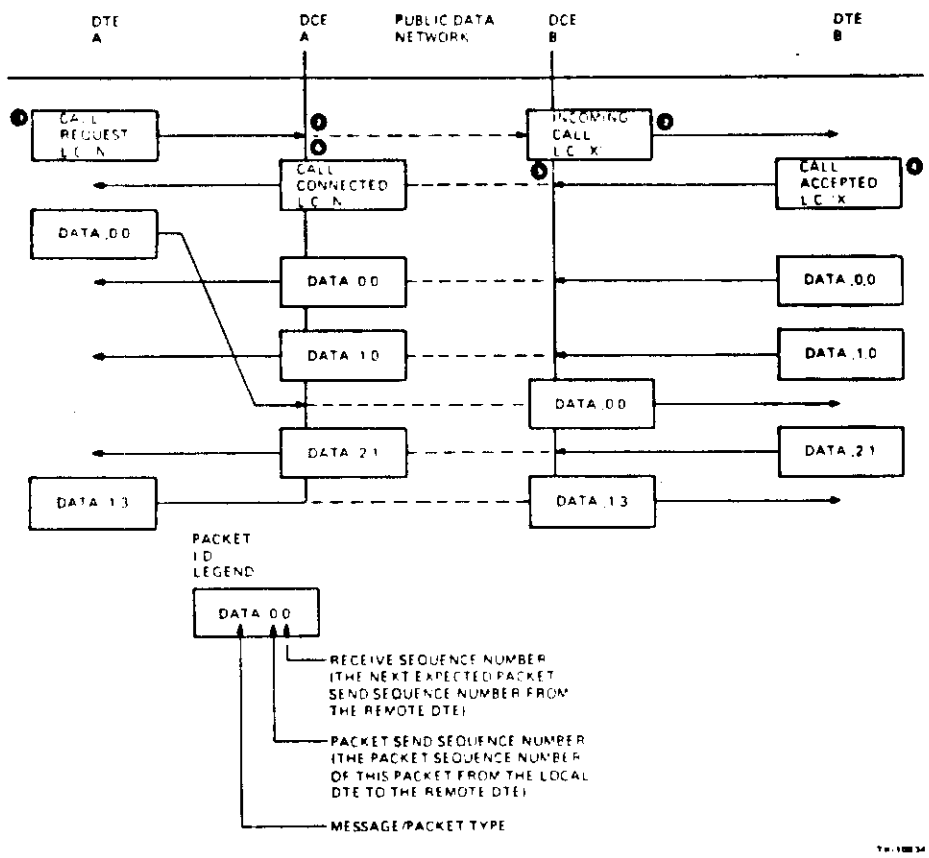


Figure 2-24 Establishing a Virtual Circuit

In Figure 2-25, when the remote DCE receives the Clear Request packet, it forwards the request information and transmits a Clear Confirmation packet to the remote DTE to deassign the channel between it and the DTE. The local DTE in turn receives the Clear Indication packet and transmits a Clear Confirmation packet to its local DCE to deassign the channel between it and its local DCE.

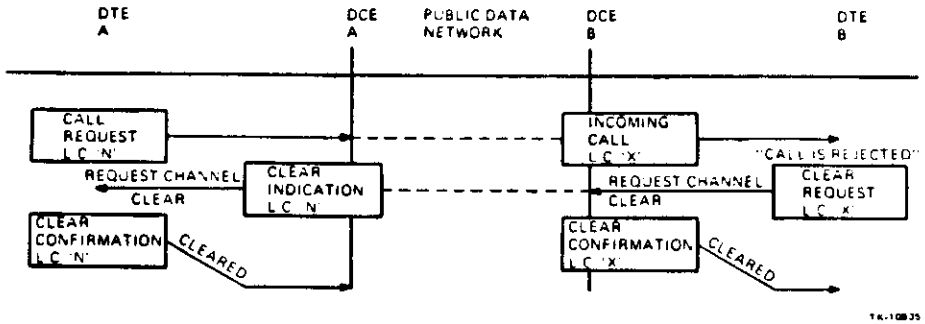


Figure 2-25 Rejecting a Call Request

A call may also be rejected when the local DCE cannot transmit, or forward, the Call Request information to the remote DCE/DTE pair. The local DCE then returns a Clear Indication packet to the local DTE reporting the reason for the failure (as in Figure 2-26).

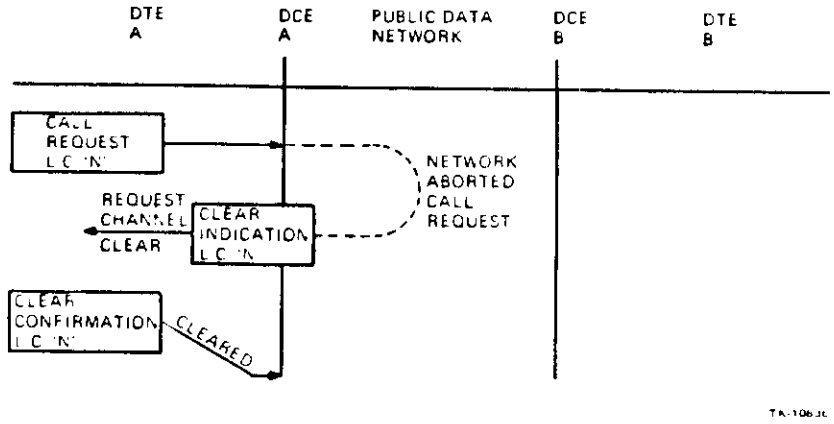


Figure 2-26 Call Request Rejected by the Public Data Network

Once the virtual circuit is established, the user can transmit and receive, in full-duplex, data packets between the local DTE and remote DTE (Figure 2-24). This data transfer continues until completed, reset, or aborted. When completed, either DTE can destroy the virtual circuit (if it is an SVC) by clearing it.

Figure 2-27 shows the sequence of events necessary to clear a switched virtual circuit once the transfer of user data is complete.

1. The DTE transmits a Clear Request packet to its local DCE.
2. The DCE forwards the information to the remote DCE, and returns a Clear Confirmation packet to its DTE.
3. The remote DCE transmits a Clear Indication packet to its DTE to deassign the logical channel.

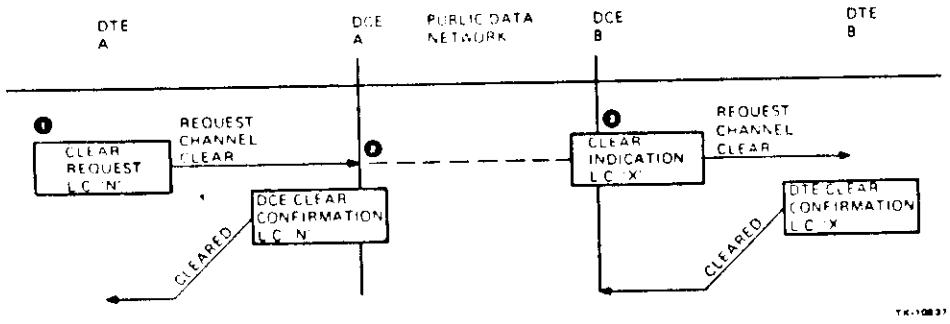


Figure 2-27 Clearing a Virtual Circuit

If a procedural error for the logical channel, such as a violation of the flow control rule, is detected while transferring data on an established virtual circuit, a virtual circuit reset sequence is automatically started. The virtual circuit reset is a Virtual Circuit Management error control mechanism.

The virtual circuit reset reinitializes the specified virtual circuit to a known state. Resetting a logical channel for a virtual circuit causes a reset of the corresponding remote logical channel. The reset, or initialization state of a virtual circuit occurs when:

- All packet sequence numbers are set to zero.
- No data or interrupt packets are in transit over the public data network. If any data or interrupt packets are in transit at the time of the reset, they are lost. The network discards all data and interrupt packets in transit when a reset condition is initiated.

Figure 2-28 shows a virtual circuit reset operation caused by a violation of the flow control rule. (Notice that DTE A's second data packet, (DATA, 8, 5) is discarded by the network.)

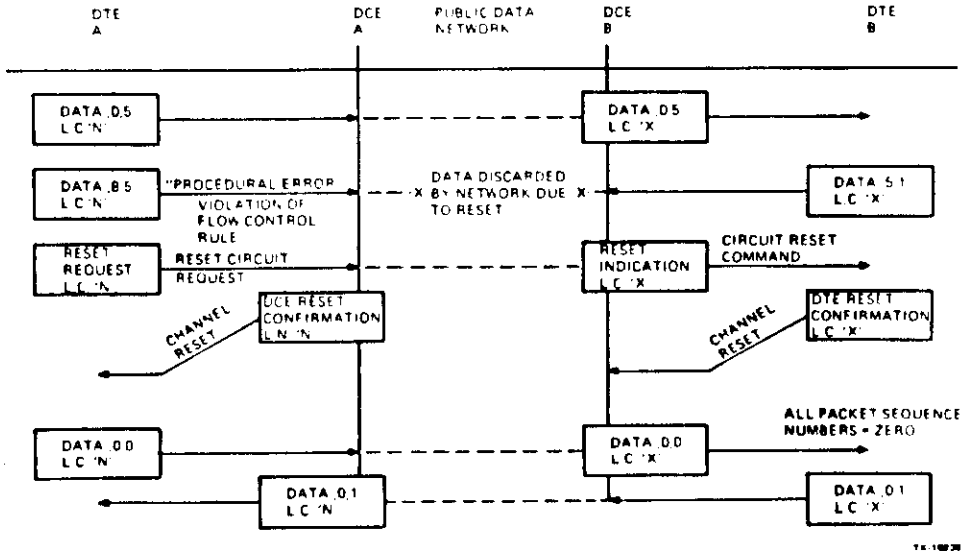


Figure 2-28 Resetting a Virtual Circuit

If a catastrophic failure (such as a power failure) occurs, either the DTE or DCE can initiate a Restart operation. The Restart operation allows all switched virtual circuits to be cleared and deassigned, and all permanent virtual circuits to be reset. The Restart sequence is also initiated by the local DCE every time the local DTE Frame level connects or reconnects to the DCE Frame level module. (This is when the level 2 modules change from the "SYNC" state to the "RUN" state on the physical link. Refer to Sections 4 and 5 in the DNA X.25 Frame Level Functional Specification).

A DCE can initiate the restart sequence by transmitting a Restart Indication packet to its local DTE. The reception of the restart packet forces the DTE to clear or reset (as necessary) all of its virtual circuits. When all of the virtual circuits are cleared or reset, the DTE transmits a Restart Confirmation packet back to the DCE to indicate that the clear/reset operation/s are completed. Figure 2-29 depicts the restart sequence when initiated at DTE A on Logical Channel Zero. The remote DTEs are forced to clear all switched virtual circuits and reset all permanent virtual circuits to DTE A.

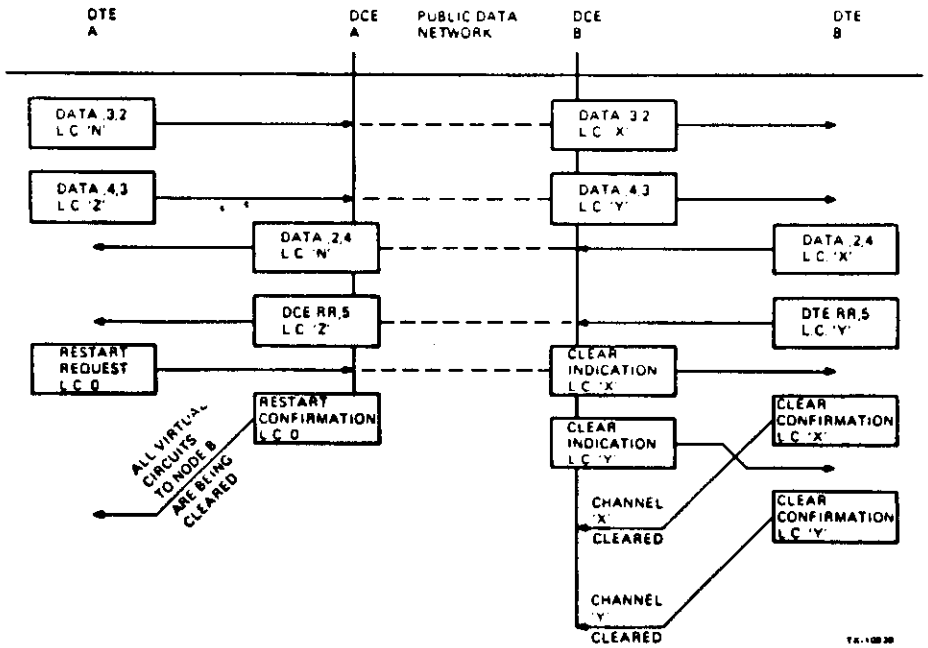


Figure 2-29 Restart of All Virtual Circuits To and From DTE A

Another possible error condition between DTEs and DCEs is collision. There are three types of collision possible. The X.25 packet level protocol is responsible for their recovery. The three types of collision are:

1. Reset Collision - Both DTE and DCE transmit on the same logical channel at the same time as the reset request (DTE to DCE) and reset indication (DCE to DTE) packets.

2. Clear Collision - Both the DTE and DCE transmit on the same logical channel at the same time as the clear request (DTE to DCE) and clear indication (DCE to DTE) packets.

3. Call Collision - Both the DTE and DCE allocate at the same time as the same logical channel number for assignment. The DTE transmits a call request and the DCE simultaneously transmits an incoming call packet with the same logical channel number assigned.

Reset collision and Clear collision are both handled in the same manner. The packet level protocol treats both as if a reset or clear confirmation packet had been received by both the DTE and DCE alike. Figure 2-30 shows the reset collision recovery while Figure 2-31 shows the clear collision recovery sequences.

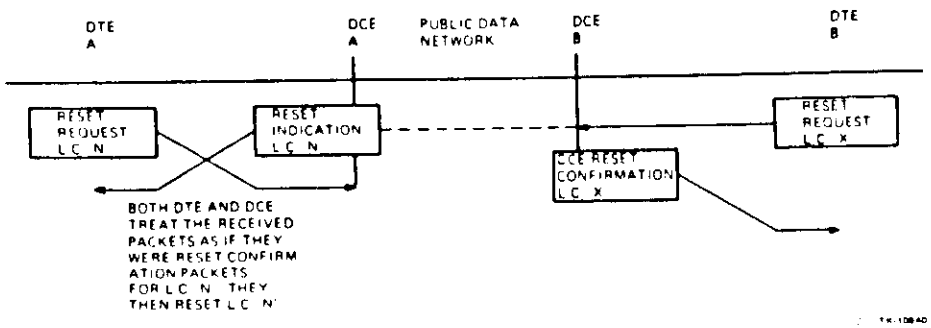


Figure 2-30 Reset Collision and Recovery

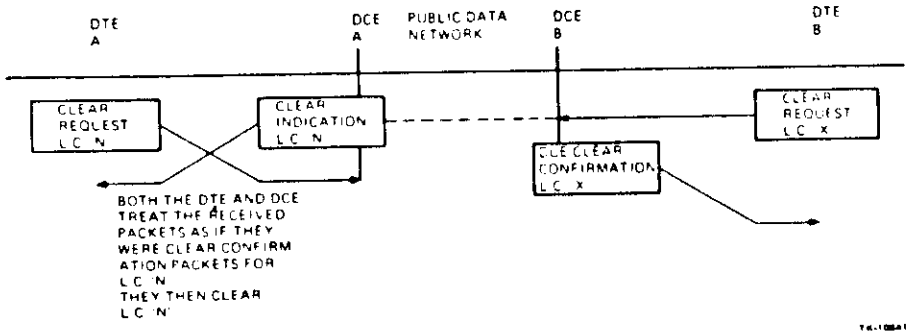


Figure 2-31 Clear Collision and Recovery

Since both DTEs and DCEs can independently assign logical channel numbers for virtual circuits, there is a chance of outgoing and incoming call setup packets colliding. Collision happens if both the DTE and DCE assign the same channel numbers to the call setup packets.

To minimize the chance of call collision, logical channel numbers are assigned in sets depending upon their intended use. The logical channel sets for channel number assignment are:

- Logical channel number zero: Reserved for restart and diagnostic packets.
- Permanent virtual circuit set: Reserved for permanent virtual circuit use only.
- Outgoing calls only set: Only the DTE can assign channels numbers in this set.
- Incoming calls only set: Only the DCE can assign channel numbers in this set.
- Common set: Both the DTE and DCE can assign channel numbers in this set.

To further reduce the risk of call collision, the DTE and DCE allocate logical channel numbers from opposite ends of the range in each set. Normally, the DTE allocates the lowest number available; the DCE allocates the highest number available. The actual order used is defined by network management.

If a call collision does happen, the packet level protocol:

- Processes the Call Request packet
- Discards the Incoming Call packet
- Generates a Clear Indication packet to be transmitted to the remote DTE for the incoming call packet.

Figure 2-32 shows the operations performed by a local DCE when a call collision condition is detected. Remember, if two DTEs simultaneously attempt to establish a virtual circuit with each other, normally two different virtual circuits will be established, unless a call collision occurs at one of the DTE/DCE pairs.

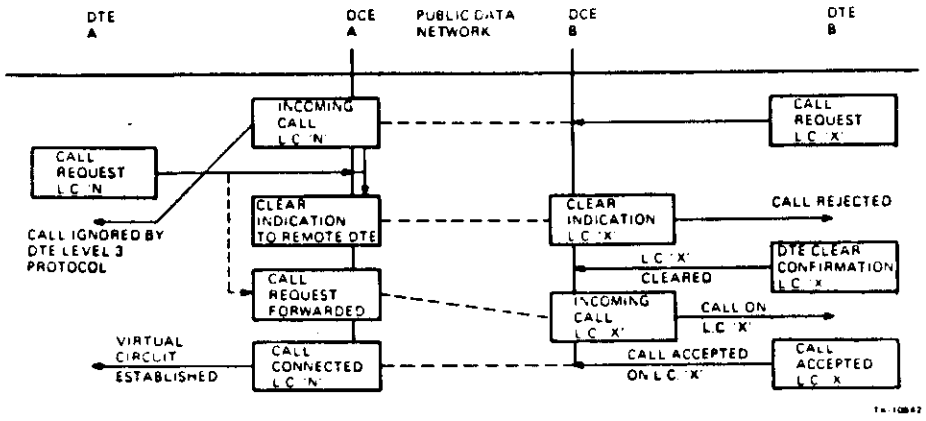


Figure 2-32 Call Collision Recovery

Data Transfer - Data can be transferred over a virtual circuit only after the circuit is established. Data packets transferred prior to the establishment cause an error condition and initiate a clear sequence of the virtual circuit. Permanent virtual circuits are established after the restart sequence, and therefore, always established unless another restart sequence is initiated. Permanent virtual circuits do not require the call setup sequence, as do switched virtual circuits. Data transfer over a permanent virtual circuit does not require as much circuit control overhead from the packet level protocol, because it is established after the completion of the restart sequence.

Each data packet transferred on either permanent or switched virtual circuits is sequentially numbered by a send sequence number. A send sequence number is carried by each data packet and is used to specify the position of the packet within a sequential data stream. Send sequence numbers are modulo 8, or modulo 128; they range from 0 to 7 or from 0 to 127. The default on all public data networks is modulo 8. For this reason DNA uses a default of modulo 8. The modulo numbering scheme can be changed; it is a network management decision and can be written as a network parameter by the network manager. The specific numbering scheme used is determined by the PPSN DCE. The local DTE and DCE must agree on the packet sequence numbering used. Usually, the numbering scheme used is decided upon at the time the user subscribed to the PPSN for communications service.

Another modulo 8 or 128 sequence number carried by each data packet is the Receive Sequence number. Receive sequence numbers are used to authorize the transmission of additional packets by acknowledging correctly received packets. The receive sequence number acknowledges all received packets whose send sequence number was one count less than the present receive sequence number value. The receive sequence number functions are similar to those of the response number field contained within the DDCMP message header. They both perform the "piggy-backed" acknowledgement function. The X.25 receive sequence number can be carried by the following packet types:

- Data packets (DCE or DTE Data packets)
- Receive Ready packets (DCE or DTE `RR' packets)
- Receive Not Ready packets (DCE or DTE `RNR' packets)

The X.25 packet level protocol uses the send and receive sequence numbers to perform packet flow control. Flow control takes place separately in each direction of data transfer, on an individual logical channel basis. Flow control is based on a window concept. A window is a range of packets that are authorized for transmission or reception across the DTE/DCE interface. The window size is selected independently for each direction of data transfer, and for each channel between the DTE and DCE pair. There are two windows created in each packet level protocol for each logical channel established between the DTE and DCE pair: one for transmitter control and one for receiver control. The window ranges delimit specific packet send and receive sequence numbers. The window ranges are flexible; the upper and lower limits change as data is transmitted, received, and acknowledged over the logical channel. The actual size of the window, the area of allowable packet numbers between the upper and lower limits of the window, is set by default or by network management on a per-channel and direction-of-flow basis.

Packets can only be transmitted if their send sequence number is greater than or equal to the lower window edge, and less than the upper window edge. Packets are only processed by the packet level if their send sequence number is next and is within the receive window limits. If the received packet does not conform to this flow control rule, it is considered a procedural error by the packet level protocol, and a reset of the logical channel is performed.

When the packet level receives a receive sequence number, it uses it as the new lower edge of the transmit window. Receiving receive sequence numbers constitutes an acknowledgement of all outstanding transmitted packets up to and including the receive sequence number value minus one. Thus, the transmit window is updated, and now allows the transmission of additional data packets, except where the receive sequence number is carried by an RNR packet.

Figure 2-33 shows the operation of a window at a DTE station. The window size (W) is 3; the size is indicated as the numerical value between the lower and upper limits of the window. The lower limit of the window is 2, as set by the last received receive sequence number. Therefore, the last packet acknowledged had a send sequence number of 1.

The next packet transmitted by this DTE must have a send sequence number between the values of 2 and 4. If no further receive sequence numbers are received by this DTE, the window limits cannot be changed and no additional packets can be transmitted after packet number 4. However, if the DCE transmits anything other than an RNR packet back to the DTE, with a new receive sequence number between 2 and 4, the DTE's transmit window can be changed. It will rotate according to the receive sequence number received from the DCE, so that 'W' remains equal to 3 and the new lower limit of the window is equal to the value of the just received receive sequence number. At this time, additional packets can be transmitted to the DCE.

If the DCE had sent the receive sequence number via an RNR packet, the window would have updated accordingly, but no further data packets could be transmitted from the DTE. The RNR packet is an indication that the DCE is not ready to receive any additional packets from the DTE. When the DCE is ready, it will transmit an RR packet to the DTE. The reception of the RR at the DTE indicates that the DCE is now ready to resume the transfer of packets between them. The DTE, which may now transmit, will resume where it left off. The packet transmitted next is determined by the lower limit value of its transmit window.

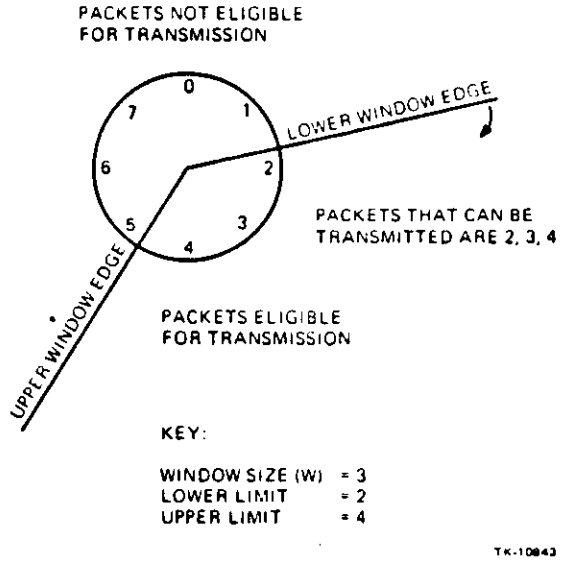


Figure 2-33 Packet Flow Control Window Operation

A DTE can construct a logical message from a number of received consecutive packets. It can also construct a number of consecutive packets from a user's logical message (using a bit contained in the packet level header field, called the "more data data bit"). If set, it indicates that this packet is part of a larger logical message. If not set, it indicates that this packet is either the last packet of a larger logical message, or a single logical message that was small enough to fit into a single packet for transfer. Figure 2-34 uses consecutive packets to transfer a large user message across the public data network.

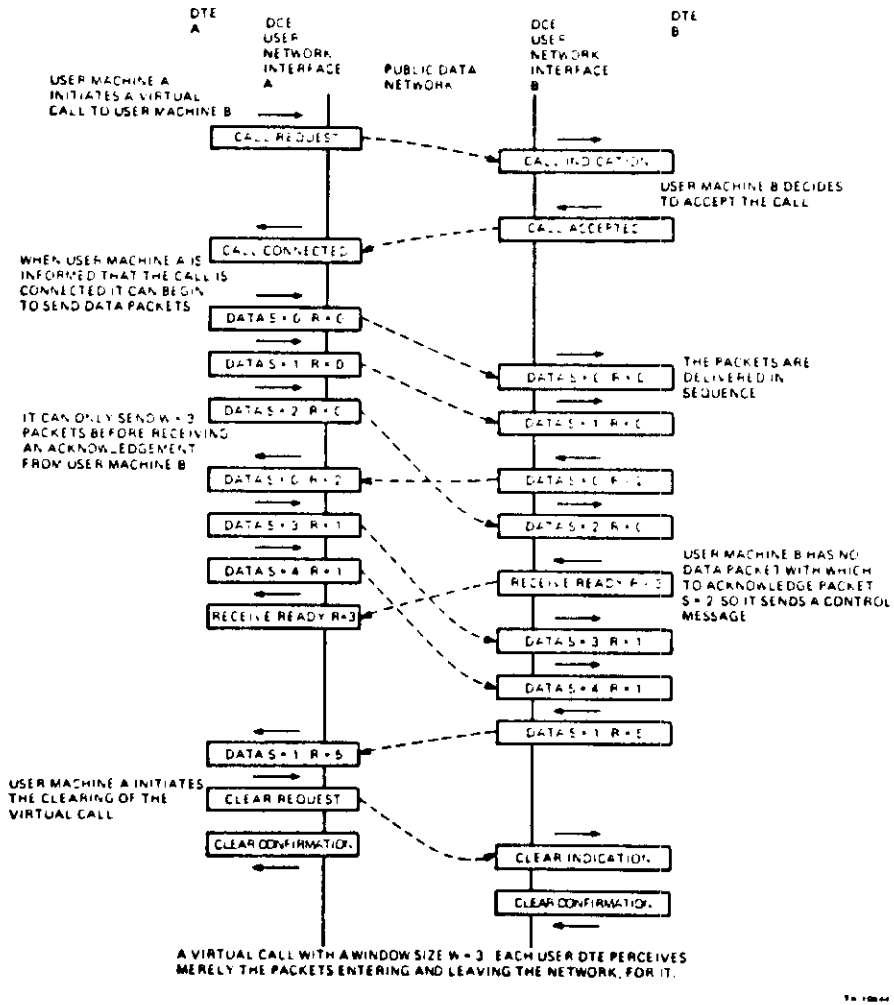


Figure 2-34 Packet Data Flow

Another type of user data packet possible under control of the X.25 packet level protocol is an Interrupt packet. Interrupt packets are used to transfer one byte of interrupt data to a remote DTE. Interrupt packets are not subject to the flow control rules normal data packets are; they are also processed as quickly as possible. They jump the normal data packet queues for processing and are delivered to the user DTE even if it is not currently accepting data packets. The interrupt packet can contain user data, but normally contains control type information. For example, a user of a typewriter-like terminal presses the break key to stop the flow of data from a distant computer. This action would be transmitted over the network by the interrupt packet to the distant computer.

Interrupt packets are acknowledged by Interrupt Confirmation packets. Once a DTE has issued an interrupt packet over a specific channel, it cannot issue another interrupt packet over that channel until the first one has been acknowledged. Only one outstanding interrupt packet at a time is allowed for each direction of data flow. Figure 2-35 shows the transmission and acknowledgement of an interrupt packet.

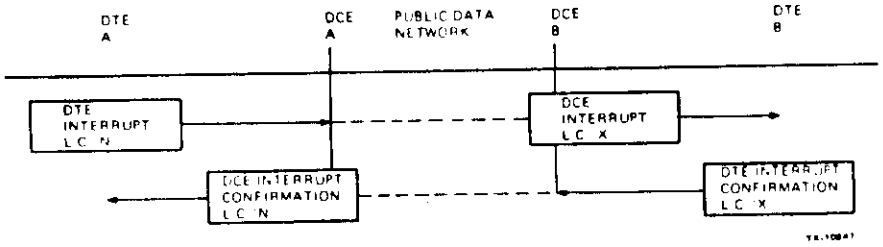


Figure 2-35 Interrupt Packet Flow

2.5.3 X.25 Message Formats

X.25 messages can be broken down into Frame and Packet level messages. Table 2-1 listed the different types of Frame level messages, and Tables 2-2 and 2-3 listed the different types of Packet level messages. Figure 2-36 represents the X.25 Frame and Packet level message format supported by DNA. It shows the Frame level control information fields appended by the frame level protocol, and the packet level control fields appended by the packet level protocol. Figure 2-36 also shows the relationship between the client/user's data and the control information fields appended by the X.25 protocols.

Notice the similarity between the DDCMP message enveloping and the CCITT X.25 (HDLC) type of message enveloping. The Frame level message envelopes the Packet level message, which in turn envelopes the user or client data. Each message portion is responsible for performing different functions and operations once decoded at the receiving DTE or DCE. Remember too, that the I Frame message is the only Frame level message that contains a data field. The Packet level message is the I Frame message data field.

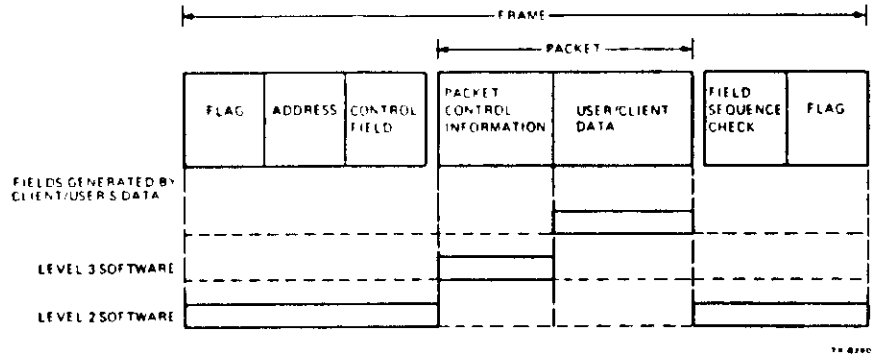


Figure 2-36 CCITT X.25 Message Format

2.6 ETHERNET MODULE

For more information on the Ethernet Module and its interactions with DNA, read:

1. Chapter 2, Section 2.2, in the DECnet DIGITAL Network Architecture (Phase IV) General Description
2. Chapters 1-4 in the DNA NI Node Product Architectural Functional Specification
3. Chapters 1-7, and Appendix A in the DNA NI Data Link Architectural Functional Specification

2.6.1 Ethernet Functional Description

The Ethernet protocols supported by DNA are resident in the Data Link and Physical Link layers of the DNA structure. This section discusses both DNA layers as they pertain to the Ethernet module.

The Ethernet local area network provides a communications facility for high-speed data exchange among computers and other digital devices located in a moderate size geographic area. Together the Data Link and Physical Link layers define the Ethernet structure, and ensure compatibility to every other implementation of the Ethernet network. The primary characteristics of the Ethernet Physical Link layer are:

- Data exchange rate of 10 million bits per second
- Maximum station separation of 2.8 Kilometers
- Maximum of 1024 stations
- Shielded coaxial cable medium using base-band signaling (offers a virtually error-free channel for data transfer)
- Supports a branching, non-rooted tree topology (an advanced branching, bus type topology, very similar to the multipoint topology discussed in the Network Concepts SPI). Figure 2-37 shows the concept of a non-rooted tree topology.

- Ability to send and receive data, non-simultaneously, between any two or more data link entities on the same network.
- Ability to detect the presence of another station's transmission while not transmitting itself. (Ethernet uses a signal called Carrier Sense for this detection.)
- Ability to detect the presence of another station's transmission while actively transmitting (Ethernet uses a signal called Collision Detect for this detection.)
- A total worst-case round trip signal propagation delay of 450 bit times, or approximately 45 microseconds at the 10 Million bits per second rate.

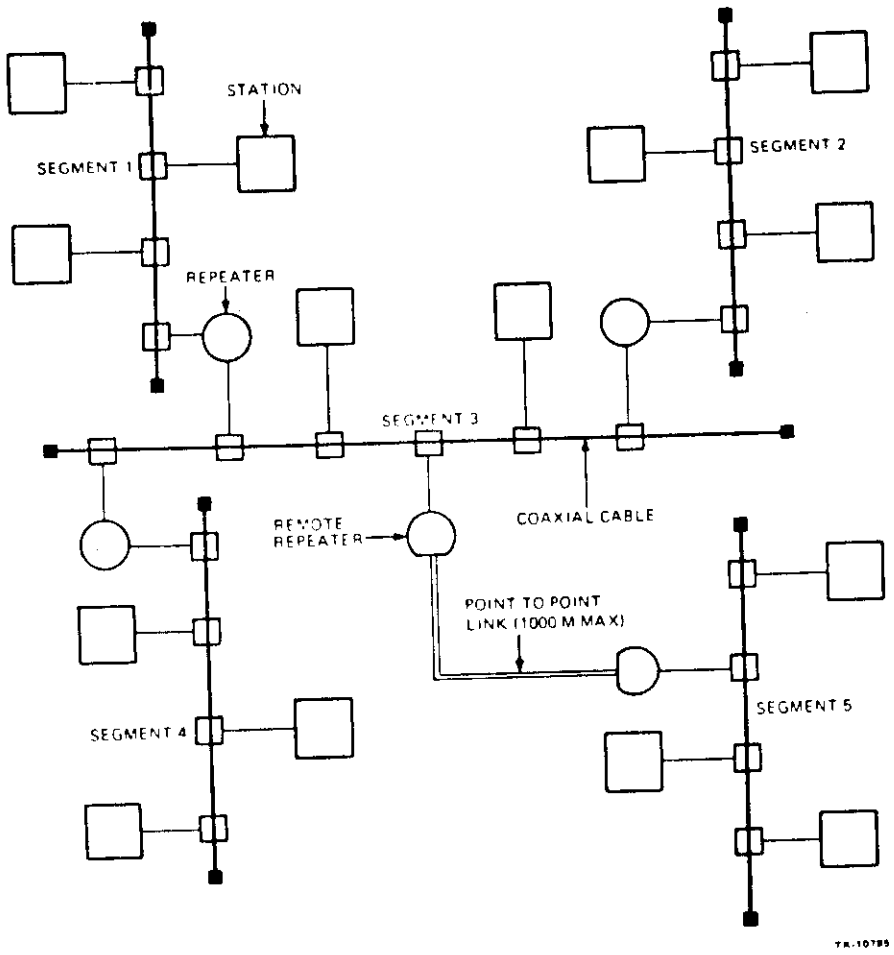


Figure 2-37 Ethernet Non-Rooted Tree Topology

The Ethernet Physical Link layer:

- Transmits and receives serial bit streams between the Data Link layer and the communication medium
- Generates clock signals for link synchronization and timing
- Detects a non-idle channel, using the Carrier Sense Signal
- Detects a data collision on the channel, using the Collision Detect Signal
- Generates and removes coding-specific preamble information. (The preamble is a 64-bit pattern of alternating 1s and 0s that ends in two consecutive 1s. The preamble is inserted before the first bit of the frame to be transmitted, then removed once it is received. It is used for message synchronization and timing.)
- Codes and decodes the data bit stream between the Data Link layer and the coaxial cable medium. (The data link data bits are in binary coded form, and the coaxial cable bits are in Manchester phase coded form.)

The Physical Link layer functions are grouped into two major areas:

1. Channel Access Control
2. Data Encoding and Decoding

The Data Link layer functions are also grouped into two major areas:

1. Data Encapsulation and Decapsulation
2. Link Management

The primary characteristics of the Data Link layer are:

- A link control procedure using a fully distributed peer protocol, with statistical contention resolution, called Carrier-Sense Multiple Access With Collision Detection Protocol (CSMA/CD)
- A message protocol that supports variable length message frames, between 64 and 1518 bytes total frame length
- Offers Best-Effort delivery service for data frame transfers.

The functions performed by the Data Link layer are:

- Data Encapsulation

  Framing - Defining the format of message packets using control information fields appended before and after the data supplied by higher layers.

  Constructing message frames - Building the different fields to be appended onto the user data. The information for the fields is supplied by the higher DNA levels, except for the Frame Check Sequence Value (FCS). This function includes assembling and disassembling Frame messages transmitted or received over the network.

  Addressing - Handles the construction and disassembly of source and destination addresses for the frame message.

  Error detection - Limited to the detection of physical channel bit errors, and the detection and recovery from transmission collisions.

- Link Management

  Channel allocation - Avoids transmission collisions by controlling the use of the channel.

  Contention Resolution - Access to the channel is controlled by CSMA/CD, which handles channel collisions by "backing-off" to retransmit at a later time.

Figure 2-38 shows the Ethernet Data and Physical Link layers, including the split between the Ehternet module layers and sublayers. The Ethernet modules treat all higher DNA layers as Client layers.

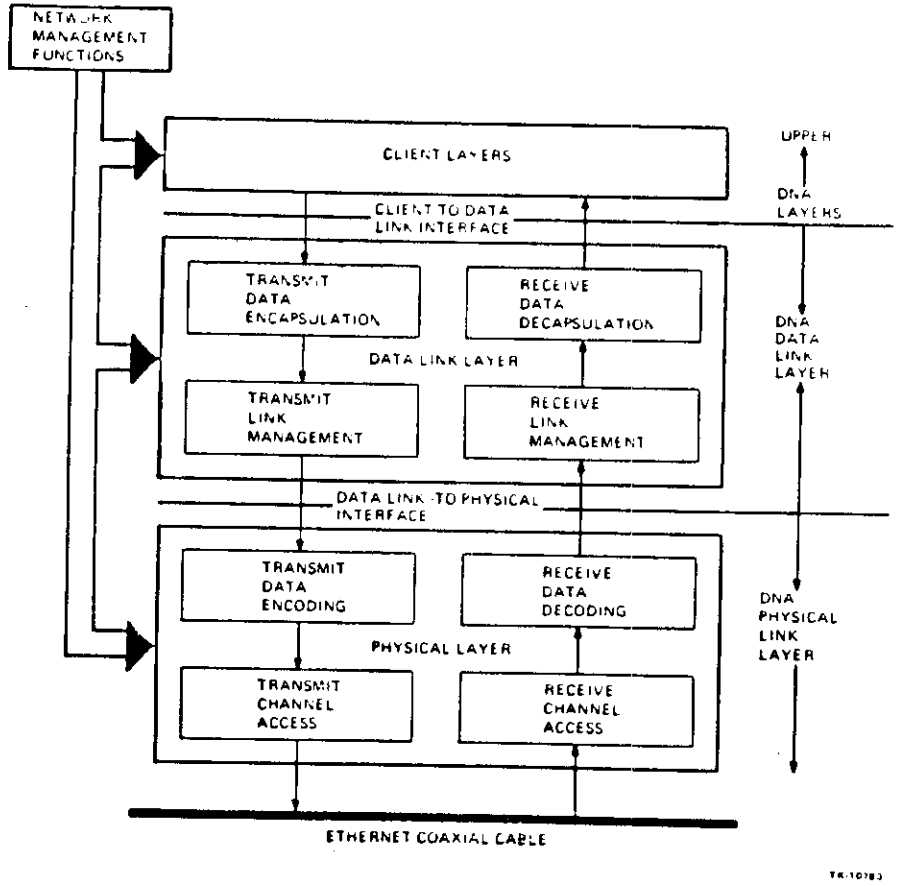


Figure 2-38 Ethernet Data and Physical Link Layer Sublayers

Figure 2-39 shows the DNA Ethernet Architecture, the functions performed by each, and the typical hardware where the actual protocol implementation occurs.

The following is a description of the transmit and receive functions performed by each layer and sublayer shown in Figure 2-39.

Transmit Flow

1. The Ethernet modules, specifically the Data Link layer module, accepts data and frame control information for transmission from the Client layer. The data to be transmitted first arrives at the Data Encapsulation sublayer.

2. The Data Encapsulation sublayer accepts the data and control field information from the client layer. It then appends the control information to the client data, generates a Frame Check Sequence (FCS) value for the frame, and appends it to the back or end of the newly built frame. The frame is now passed down to the Link Management sublayer.

3. The Link Management sublayer then decides whether to transmit or wait to transmit based upon information obtained from the Physical Link layer. Once the Physical Link layer detects that the channel is free (or idle), the link management sublayer initiates the transmission process, and passes the message frame down to the Physical Link layer for action.

4. The Physical Link layer, specifically the Encode and Decode sublayer, encodes the message frame for transmission. This sublayer changes the physical bits in the message frame from Binary coded to Manchester phase coded form. It also generates the frame preamble pattern, and once generated, passes the preamble and message frame as one steady stream of bits to the Transmit Access sublayer for transmission.

5. The Transmit Access sublayer accepts the bit stream from the Encoding sublayer and places the preamble and message frame onto the network channel (coaxial cable).

Receive Flow

1. At the receiving station, the frame is detected by the Receive Access sublayer in the Physical Link layer. The Receive Access sublayer responds to the preamble pattern by synchronizing with the bit stream. Once synchronized, it passes the preamble pattern and the message frame up to the decoding sublayer for processing.

2. The Decoding sublayer decodes the Manchester phase coded preamble and message frame into binary coded form. It strips the preamble pattern from the message frame, then passes the frame up to the Data Link layer as it is received and decoded.

3. The Physical Link layer sends a signal to the Data Link layer, called Carrier Sense, to flag the Data Link layer that reception has started. The Data Link layer then accepts the bit stream from the Decode subsection.

4. The Data Link layer accepts the message frame (the preamble has been striped by the Physical Link layer) and stores the received bits until the signal carrier sense from the receive access sublayer stops. When this signal stops, it indicates that the channel has gone idle, signalling the end of the transmitted frame. During the receiving process, the Link Management sublayer checks for transmission collisions. (Transmission collision detection and recovery is discussed later.) If none are detected, the frame is passed to the Data Decapsulation sublayer.

5. The Data Decapsulation sublayer strips the frame control fields and checks the FCS value to see if transmission errors have occurred. Then the frame address field is interrogated to ensure that this message frame is for this station. If it is the client data, frame control information and a status code are passed up to the client layer for further processing.

The status code is used to indicate whether any damage was suffered by the frame during transmission across the channel. The client layer within the context of DNA is the higher layer of the DNA structure.

Unlike DDCMP and X.25, Ethernet protocols do not automatically cause the retransmission of the data containing errors. Ethernet does not perform any acknowledgements or negative acknowledgements for received data. These functions are the responsibility of the client layer. This is a result of the Ethernet's "Best-Effort" delivery service. It delivers the frame and handles the detection and recovery of transmission collisions. In a DECnet station, the End Communication layer that implements the NSP protocol performs all of the other error recovery procedures for the Ethernet channel.

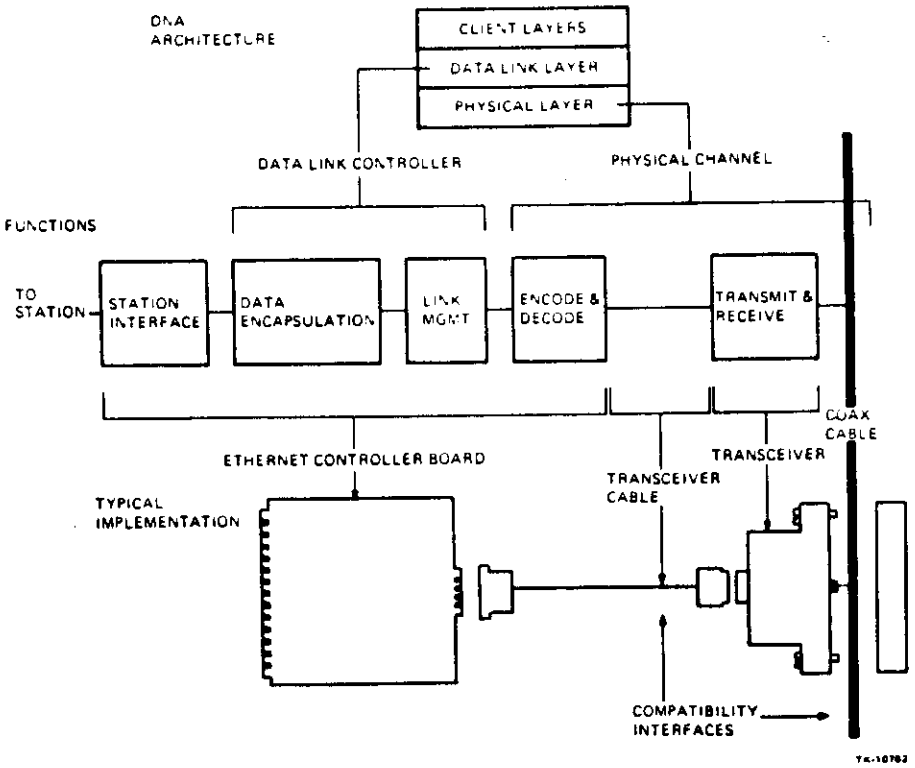


Figure 2-39 Ethernet Architecture and Implementation

2.6.2 Ethernet Functional Operations

The Ethernet modules provide a mechanism for exchanging client data over the Local Area network. This mechanism detects transmission collisions and recovers from those collisions. It addresses client data to:

- A specific physical station on the network
- A specific logical group of stations on the network
- All stations on the network

This mechanism also detects any bit errors within a message frame caused by the physical channel during transmission and reception.

These operations are divided between the two Data Link layer sublayers: Data Encapsulation/Decapsulation and Link Management. The specific operations performed by each are listed below.

- Data Encapsulation/Decapsulation
  - Message Framing
  - Message Addressing
  - Error Detection
- Link Management
  - Channel Allocation (collision avoidance)
  - Contention Resolution (collision handling)

2.6.2.1 Data Encapsulation/Decapsulation - The data encapsulation and decapsulation functions of the Data Link layer comprise the construction, destruction, and processing of frames. The subfunctions of framing, addressing, and error detection are done for both transmission and reception.

The following is a generic description of each subfunction; specific transmission and reception actions are discussed separately later in this section.

- Framing - No explicit framing information is needed. The necessary queues, carrier sense and transmitting, are present in the interface to the Physical Link layer.

- Addressing - Two address fields are provided to identify the source and destination stations for the frame.

- Error Detection - A frame check sequence field, a 32-bit value used to detect transmission errors caused by the Physical Link layer operations.

Framing - There are four control information fields appended to the clients' data (see Figure 2-40).

1. Destination Address Field - The physical or multicast address of the destination station or stations.

2. Source Address Field - The physical address of the station that transmitted the frame.

3. Type Field - A two-byte field uses the client layer to identify the specific protocol associated with the frame.

4. Frame Check Sequence Field - Contains a 32-bit cyclic redundancy check value used to detect transmission errors.

All message fields are a fixed size except the data field. The data field may contain any integral number of bytes between the minimum of 46 and the maximum of 1,500. The total frame can be from a minimum of 64 bytes to the maximum of 1,518 bytes, counting the frame control information fields appended to the data field.

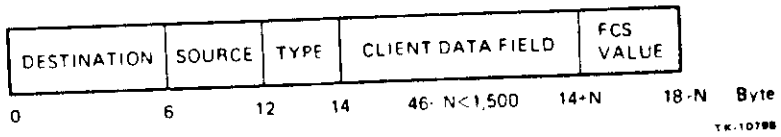


Figure 2-40 Data Link Layer Message Frame Format

Addressing - The Ethernet frame is addressed by two 6-byte address fields. The fields are used to define the source and destination stations of the frame.

The Source field specifies the station that sent the frame. The source address is not interpreted at the data link level; it is passed up to the higher DNA layers for processing.

The destination field specifies the station or stations for which the frame is intended. The first bit of the frame distinguishes physical and multicast addresses. If the bit equals a logical zero, it defines the destination address as a physical station address. It may be either a:

- Physical Address - The unique address associated with a particular station on the network, or

- Multicast Address - A multidestination address associated with one or more stations on the network. There are two kinds of multicast addresses:

  Multicast group address - An address associated with a group of logically related stations

  Broadcast Address - A predefined multicast address that denotes all stations on the network.

During transmission, the destination, source, and type field information is passed down to the data encapsulation sublayer from the Client layer. The sublayer then appends this information to the front of the client data.

During reception, the sublayer strips this information and passes it along with the received client data up to the Client layer, the higher DNA layers.

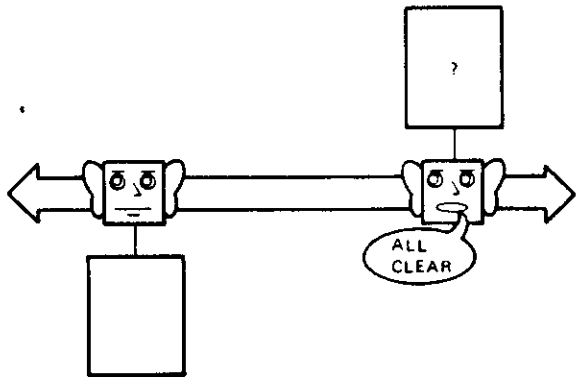
Error Detection - The data encapsulation sublayer generates a frame check sequence value for each frame to be transmitted, and places this value at the end of the frame in the FCS field. The FCS value is used to detect transmission errors at the time of reception. Unlike DDCMP or X.25 protocols, the Ethernet protocols do not use the FCS value to determine whether or not a retransmission is necessary to correct the error. The Ethernet protocols use this field solely to detect the presence of errors. If an error is detected, the frame with the error and a status code that identifies the error as a frame check error is passed up to the client layer for further processing and recovery procedures.

The DNA End Communication layer is the Client layer that performs the error recovery procedures in an Ethernet network. Ethernet protocols do not perform any message sequencing or error recovery procedures. It only transmits and receives clients' frames, and performs link management for collision detection and recovery.

Link Management - Ethernet Link Management can be thought of as two separate operations: transmission control and reception control.

Transmission Control - Link Management for transmitting consists of the following operations:

Carrier Deference - Causes the Data Link layer to monitor the communication channel via the Physical Link layer's signal Carrier Sense. When Carrier Sense is present, it indicates that some other station is currently transmitting on the channel. If the channel is busy, carrier sense is present, and the Data Link layer defers all transmissions until 9.6 microseconds after carrier sense has stopped. Once the channel is free (or idle) the Data Link layer can begin to transmit. Figure 2-41 illustrates the process of carrier deference.



CARRIER SENSE MULTIPLE ACCESS
LISTEN-BEFORE-TALKING TO AVOID
COLLISIONS.

TK-10802

Figure 2-41 Carrier Sense Multiple Access Channel Clear Detection

Collision Handling - Collision handling occurs once the Data Link layer has finished deferring to an already transmitting station. It initiates the steps described earlier to begin transmitting on the now idle channel. It is still possible for this station to experience contention for the channel. Contentions are a result of transmission collisions. Collisions can occur until total acquisition of the channel, or network, has been accomplished. Acquisition of the network is accomplished through the deference of all other stations' Data Link layers. Contention resolution or collision handling is performed through two separate operations:

1. Collision Detection with Enforcement
2. Collision Backoff with Retransmission

Collision detection and enforcement operations detect the presence of transmission collisions and then ensure that the collision is detected by all other stations on the channel.

Collision detection is provided by the Physical Link layer using a signal called Collision Detect. The Physical Link layer forwards this signal to the Data Link layer for action. If the Data Link layer receives the collision detect signal while transmitting, it will not stop the transmission immediately. It keeps transmitting for at least another 32 (but not more than 48) bit times. This forced transmission is called Collision Enforcement. It ensures that the other transmitting station or stations detect the collision. The 32 to 48 bits transmitted are called the "Jam" signal. The Jam signal guarantees that the duration of the transmission collision is enough to ensure its detection by all other stations on the network. Figure 2-42 illustrates the process of "jamming" when collision is detected.

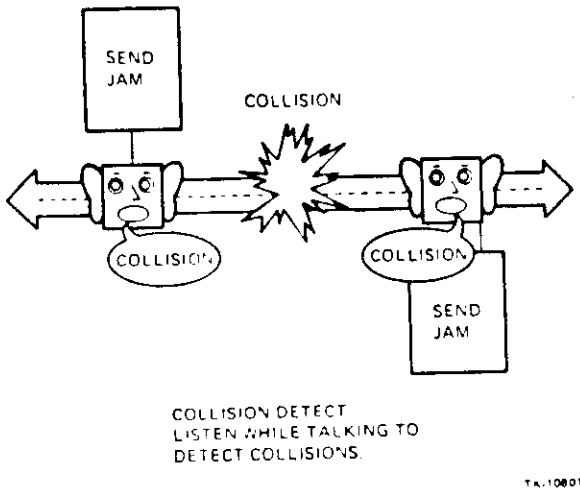


Figure 2-42 Collision Detection and Channel Jamming

Once the collision is detected, an operation called backoff with retransmission defers transmission to the first station that wanted to transmit, and allows it to retransmit the message frame that experienced the collision. If for some reason another station starts to transmit and the deferring stations don't get to transmit when they are finished deferring, they will defer again. Deferral can occur up to 16 times (the first time plus 15 retries) before the attempt to transmit that particular frame is aborted. If aborted, the Data Link layer flags the error to the Client layer and, unless intervened by the Client layer, begins the transmission procedures for the next frame to be transmitted. It flags the Client layer with a status code indicating that the first 16 attempts to transmit the frame had failed. Remember, all 16 attempts must be unsuccessful, and must result in collision before trying to transmit the next frame.

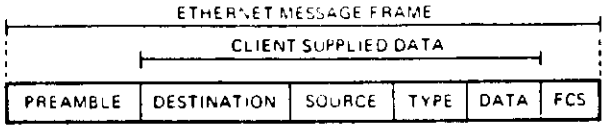
Reception Control - Ethernet Receive link management checks all incoming frames for FCS errors and frame length errors.

FCS errors are not recoverable at the Ethernet layer level. All FCS errors must be handled by the Client layer. In DNA, these errors are handled by the End Communication layer. The Data Link layer only detects these errors; it does not handle their recovery. If an FCS error is detected, the Data Link layer passes the frame containing the error, and a status code indicating the error, to the Client layer. As far as Ethernet is concerned, it has fulfilled its responsibility and has completed the processing for that frame.

Frame length errors are detected and recovered at the Data Link level. They are the result of transmission collisions. Since occasional collisions are a normal part of link management, the frame length errors are not reported to the Client layer. Frames received with less than 64 bytes are classified as frames with length errors, and are discarded without the knowledge of the Client layer. The frame that resulted in the error is discarded because a correct copy of the discarded frame is retransmitted by the source station as a function of transmission control.

2.6.3 Ethernet Message Formats

The Ethernet Data Link layer has only one type of message, called a frame. The data encapsulation sublayer builds the frame for transmission; the data decapsulation sublayer breaks down the frame upon reception. Figure 2-43 illustrates the standard Ethernet frame format. The numbers below each frame field indicate its length in bits. The Physical Link layer's 64-bit preamble pattern field is also shown. The preamble is appended then stripped upon reception by the Physical Link layer. The preamble is appended to all frames that are sent over the network channel (coaxial cable).



PREAMBLE — Used for physical link synchronization.

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.
- **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 $46<n<1500$. Full transparency is provided, in the same 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.

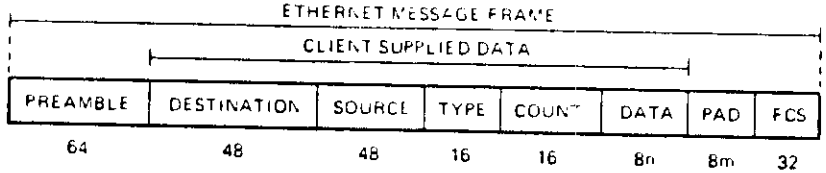
FCS — The frame check sequence. This field contains the CRC-32 polynomial check on the rest of the frame.

TX-10769

Figure 2-43 Ethernet Frame Format

DNA also defines a standard convention for transmitting higher-level protocol data within the Ethernet frame format. The higher level protocol data is not usually long enough to fill the minimum message size requirement. The Ethernet minimum frame length is 64 bytes. This convention allows the transmission and reception of these very small higher-level protocol messages. It pads the protocol data so that it appears to be large enough to qualify as a valid message frame.

Figure 2-44 shows the Ethernet message frame used to transfer higher-level protocol data. Notice the fields added to pad the protocol data, specifically the count field. The count field keeps track of how much data in the data field is really data and how much is padded data. Padded data is any amount of bytes necessary to fulfill the minimum message size rule; these bytes contain all zeros for data.



PREAMBLE — Used for physical link synchronization.

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 $0<n<1498$. Full transparency is provided, in the sense that any sequence of 8 bit bytes may appear in the data field.

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

TX-10797

Figure 2-44 Padded Ethernet Frame Format

Data Link Control

MODULE TEST

Answer the following questions by circling the letter next to the best possible solution. After you have finished the test, check your answers against the Answer Sheet provided in your Tests and Answers booklet. Do not proceed to the next module until you have correctly answered all of the following questions.

1. DDCMP Data message numbers range from __ to ___ .

   a. 0, 256

   b. 0, 255

   c. 1, 256

   d. 1, 255

2. What is the hexadecimal value of the special 8-bit character used by DDCMP to accomplish link synchronization?

   a. 26

   b. 90

   c. 96

   d. 7F

3. In nonbroadcast circuits, how many physical lines are needed to support full-duplex, point-to-point data communication?

   a. 1

   b. 2

   c. 4

   d. 6

4. When it has determined that it has correctly received a data message from an adjacent node, DDCMP will perform which of the following actions?

   a. Return a NAK to the transmitting station.

   b. Pass the data up to the Routing layer for further processing.

   c. Reject the message due to CRC errors.

   d. Acknowledge receipt of the data message by returning a STRT to the transmitting station.

5. DDCMP Control messages are identified by an identity field containing the value ENQ. ENQ is equal to a value of ____ in hexadecimal notation.

   a. 05

   b. 08

   c. 81

   d. 90

6. The X.25 recommendation is divided into how many different levels?

   a. 1

   b. 3

   c. 5

   d. 7

7. In the X.25 Frame level, how are Command messages that are transmitted from the DTE to the DCE addressed?

   a. 1

   b. 3

   c. a

   d. b

8. X.25 Frame level messages that contain data are called _____ frames.

   a. Data or (D)

   b. Receive Ready or (RR)

   c. Information or (I)

   d. Unnumbered or (U)

9. Which of the following Public Data Network Services does not support DNA Phase IV?

   a. Permanent Virtual Circuit Service

   b. Switched Virtual Circuit Service

   c. Datagram Delivery Service

   d. DCE to DTE message acknowledgements

10. In the case of a Call Collision, the X.25 Packet level will do which of the following to recover from the error?

    a. Discard the incoming call packet

    b. Process the incoming call packet normally

    c. Transmit an RNR frame to the Remote DTE

    d. Flag the local user via an interrupt flag

11. An Ethernet circuit can exchange data at a rate of ____ million bits per second.

    a. 1

    b. 10

    c. 15

    d. 20

12. All nodes or stations on a single Ethernet Cable are _____ nodes.

    a. Adjacent

    b. Remote

    c. Nonbroadcast

    d. Router

13. The Ethernet Data Link layer offers what type of delivery service to its clients?

    a. Positive acknowledgement with automatic retransmission

    b. Worst-Case Control with automatic retransmission

    c. Best-Effort with indefinite automatic retransmission if collisions are detected

    d. Best-Effort with up to but no more than 16 automatic retransmissions if collisions are detected

14. Ethernet Data messages can communicate no less than ____ and no more than ____ bytes of client data.

    a. 25, 16,383

    b. 46, 1,900

    c. 46, 1,500

    d. 64, 1,518

15. What is the value used to pad the Ethernet message so that higher DNA layers can exchange protocol information?

    a. 1 byte or more of 0s

    b. 1 byte or more of 1s

    c. 1 byte or more of alternating 1s and 0s

    d. 1 byte or more of a randomly generated 1s and 0s pattern

ROUTING

INTRODUCTION

This module introduces, describes, and illustrates the operations performed, and message formats used, by the Routing layer of the DNA.

OBJECTIVES

To use DECnet in technical support of applications environments, Software Services and Customer Personnel must be able to:

1. Define and illustrate the terms associated with the DNA's Routing layer.
2. Identify the Message Formats used by the DNA's Routing layer.
3. Describe the functional operations performed by the DNA's Routing layer.

LEARNING ACTIVITIES

1. Study the information in this module.
2. Read Chapter 3, The Routing Layer, in the DECnet DIGITAL Network Architecture (Phase IV) General Description.
3. Take the module test at the end of this module.
4. Correct the test using the answer sheet provided in the Test and Answers booklet. Review the material on any questions you may have missed before going on to the next module.

RESOURCES

1. DECnet DIGITAL Network Architecture (Phase IV) General Description
2. DNA Routing Layer Functional Specification, Phase IV, Version 2.0

ROUTING LAYER

3.1 LAYER PURPOSE

The Routing layer has two major purposes:

1. To route message packets from a source to a destination node, within the same physical node or across an entire network, via intermediate network nodes.

2. To manage the message packet flow across the network by preventing old or undeliverable packets from saturating the network.

3.2 LAYER INTERFACES

There are three layer interfaces defined between the Routing layer and the other DNA layers. Two of these interfaces are to its adjacent layers; the Data Link and End Communication layers. The third interface is to the Network Management layer. Figure 3-1 shows the relationships between the Routing layer and the other layers of DNA.

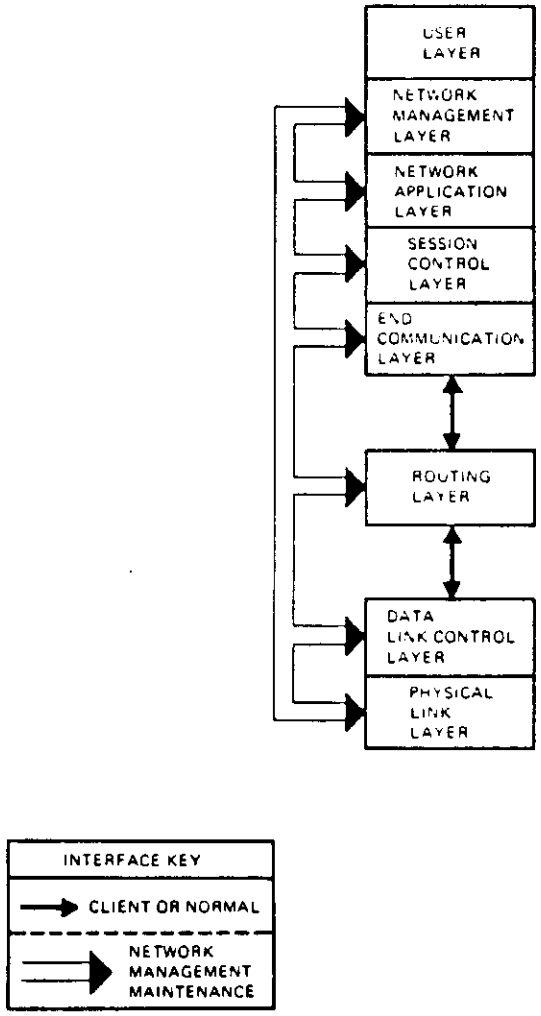


Figure 3-1 Routing Layer Interfaces

3.3 ROUTING FUNCTIONAL DESCRIPTION

The Routing layer routes and manages message packet flow through the network. The Routing layer consists of two sublayers:

1. Initialization
2. Control

Figure 3-2 shows the logical positioning of these two sublayers with respect to the other DNA layers.

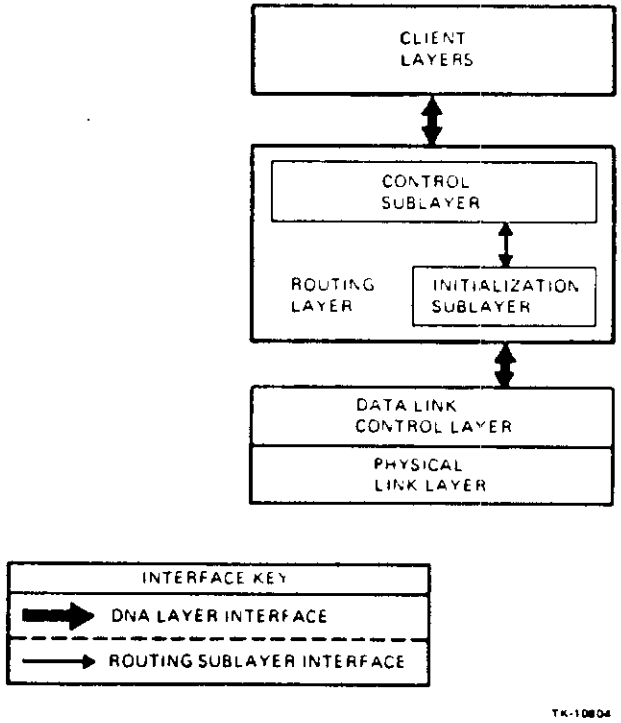


Figure 3-2 Logical Positioning of the Routing Sublayers

3.3.1 Initialization

The Initialization sublayer performs the following functions:

1. Masking - Masks the physical and logical circuit/channel characteristics of the Data Link layer modules from the Routing Control sublayer. The Routing Control sublayer knows of only two generic link types: Nonbroadcast (DDCMP and X.25) and Broadcast (Ethernet). The Initialization sublayer interfaces the Control sublayer with the specific link type or types being used by the network.

2. Initialization - Identifies the adjacent node and the adjacent node's Routing layer, and performs node verification, if required.

3. Monitoring of the physical circuit - Monitors errors detected by the Data Link layer module or modules.

3.3.2 Control

The Control sublayer performs the following functions:

- Masking - Masks the physical and topological circuit/channel characteristics of the network from the higher DNA layers. The combined masking functions of the Initialization and Control sublayers ensure that the End Communication and higher DNA layers do not have to act or function differently in supporting the different DECnet network types available.

- Routing - Determines the path (physical circuit/channel) of a message packet to its intended destination. The destination could be the adjacent node or a node on the far side of the network. With few exceptions, which are covered later, the packet is routed through the network until it reaches its intended destination via this Routing layer function.

- Congestion control - Manages the transmit and receive buffers for nodes that support route-through. When several transmit buffers are available, both local and route-through traffic are handled on a first come, first-served, basis. If, however the supply of transmit buffers available is below a certain limit, local message traffic is not allowed, until more buffers are available to the Routing layer. Route-through traffic is allowed to continue. This prevents a node from tying up network resources and becoming a bottleneck for traffic between other network nodes that require this node for route-through functions.

- Packet lifetime control - Prevents old or undeliverable message packets from wasting the network resources. Packets that have been looping through the network (have visited too many nodes for route-through) are discarded.

3.3.3 Phase IV Routing

The Routing layer defined by DNA Phase IV:

- Permits two types of Routing implementations at a network node:

  Routing - Sometimes referred to as a Full-Routing Node:

  1. Routes message packets from other nodes, referred to as route-through or packet-switching. The route-through operations are done using routing algorithms. Full-Routers (route-through nodes) are the only nodes that contain or execute the routing algorithms.

  2. Receives message packets from any other Phase III or IV node.

  3. Sends message packets from itself to any other Phase III or IV node.

  Nonrouting - Also called an End Node:

  1. Receives message packets addressed to itself from other nodes.

  2. Sends message packets from itself to any other Phase IV node.

  3. Has only one active circuit connecting it to the network. End nodes cannot perform route-through operations; they do not contain or execute any of the route-through algorithms.

- Nodes support two generic types of network circuits:

  Broadcast - Ethernet circuits

  Nonbroadcast - DDCMP or X.25 circuits

- Has introduced two new concepts for networking:

  Adjacency - Prior to Phase IV, adjacency was described as being physically next to (adjacent) to another node. Under DNA Phase IV, adjacency is described as being logically one hop away from another node. For example, all nodes in a Multipoint or Ethernet topology are adjacent nodes even if they are not physically next to each other.

  Designated Router - A specific node on an Ethernet network that is chosen to route message packets on behalf of the end nodes on the local Ethernet circuit.

- Is compatible with DNA Phase III routing. However, DNA Phase IV nodes differ from Phase III nodes in the following ways:

  Phase IV nodes can be attached to an Ethernet; Phase III nodes cannot.

  Phase IV nodes can handle larger networks than Phase III nodes. A Phase III node can communicate with up to as many as 255 network nodes, while a Phase IV node can communicate with up to 1023 network nodes. A Phase III node cannot communicate with a Phase IV node whose addresses are out of the Phase III node's addressing range. A Phase III node also cannot be on the path to the out-of-range Phase IV node.

  Phase IV message packets differ in format from Phase III message packets. Phase IV nodes translate message packets into Phase III format when they communicate with a Phase III node.

3.3.3.1 Topologies - Figure 3-3 is an example of a supported Phase IV network. It shows the implementation of DDCMP, Ethernet, and X.25 circuits, as well as DNA Phase III and Phase IV compatibility between routing and non-routing nodes.

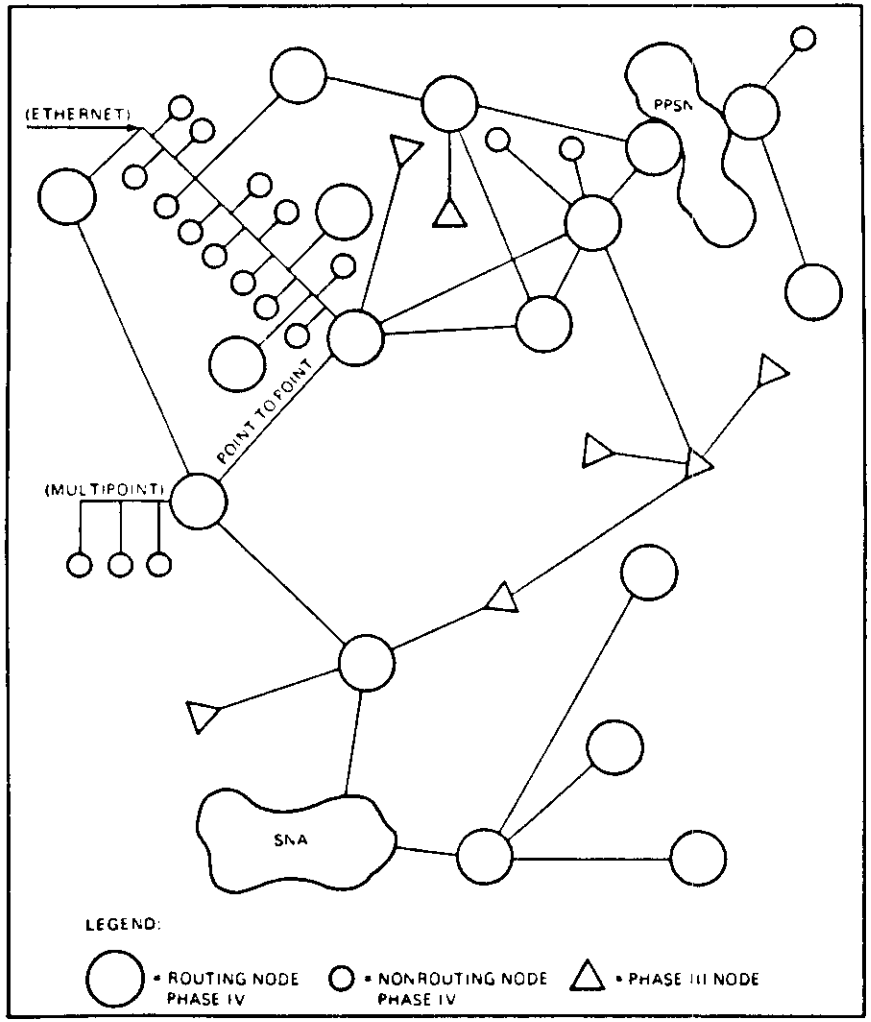


Figure 3-3 A Possible DECnet Phase IV Routing Topology

3.3.3.2 Concepts - The DNA Routing layer is responsible for exchanging message packets between nodes in the network. The Routing layer sends message packets over the most cost-effective circuit available, selected by the routing algorithm.

Algorithm - The routing algorithm provides the Routing layer with Adaptive Routing, which is the ability to constantly update, and compensate for changes made to the network topology without operator intervention. Adaptive routing does not compensate for circuit loading (message traffic saturation). Circuit loading is determined by the amount of message packets transmitted on any given circuit at any given time. When this amount is greater than the circuit's packet-handling capacity, the circuit is loaded (or saturated).

The routing algorithm determines the best path for the packet by building a Routing Data Base Table in each routing node in the network. Each node's Routing Data Base Table is different. The tables are built during network initialization, then updated by a change to the network topology or characteristics, or the expiration of a timer at a network node.

Data Base Table - The Routing Data Base Table stores information that is built by sending and receiving routing messages to/from all adjacent nodes. The adjacent nodes in turn send and receive their own routing messages to/from their adjacent nodes, who in turn repeat the process to their adjacent nodes (a snowball effect). The adjacent node that receives the routing message, updates its own Routing Data Base Table, then forwards the new information to its adjacent nodes. Eventually every initialized route-through node has information in its own table that can be used to determine the best path to route message packets. The best path is always the most cost-effective path, not necessarily the shortest path.

Hops and Visits - A hop is the logical distance from one node to another node in the network. Path distance (called path length), is calculated by adding up the total number of hops from the source node to the destination node. Adjacent nodes are always one hop away from each other; the local node is always zero hops from itself.

To prevent old or undeliverable packets from wasting network resources, each node in the network has a value called maximum hops stored within its Routing layer. If a node is farther in hops than the maximum limit at the source node, the message cannot be sent; that destination is unreachable. Maximum hops can be set as high as 31 in the Routing layer by the network management layer.

The Routing layer also stores the maximum visits value. The maximum visits value determines whether or not a packet received for route-through can be forwarded. All data packets have Route Headers appended to them by the source node the header identifies:

- The node that originated the message
- The destination node
- The number of nodes the packet has been forwarded through

A field (called the Forwarding Field), in the Packet Route Header portion of the message packet, is used to count the number of route-through nodes that have forwarded the packet to its destination. As packets are forwarded through the network by route-through nodes, the Forwarding Field is incremented by one. If the count in the Forwarding Field is greater than the maximum visits value at the node currently doing the route-through for the packet, the routing algorithm of that node discards the packet. Once a packet is discarded, it cannot be recovered. The Routing layer architecture limits maximum visits to a value of 63. However, it can be set to any specific value equal to or less than 63 by the Network Management layer.

Both maximum hops and maximum visits are set by the system manager of each node in the network. These limits prevent old, undeliverable, or excessively looped messages from saturating the network circuits. The maximum hops limit imposed by DNA is 31. Network Management layer software will not allow the maximum hops value to exceed the maximum visits limit.

It is suggested that the maximum hops limit be set first, then the maximum visits limit value set to: Maximum hops + X, where X is equal to ($1 < X <$ maximum hops limit). In other words, maximum visits must be larger than the maximum hops by at least one but less than twice the value of maximum hops.

The maximum visits limit of other nodes in the path to the destination are not considered by either the source or route-through nodes when transmitting a packet over the network to its destination.

Circuit Cost - Circuit cost is an arbitrary positive integer that is set for each circuit connected to a node. The cost of a circuit is determined by each node's system manager. The cost of a circuit does not have to be the same value on both ends of the circuit. For example, the circuit connecting nodes A and B in Figure 3-4 could have a cost of 2 at node A and a cost of 4 at node B.

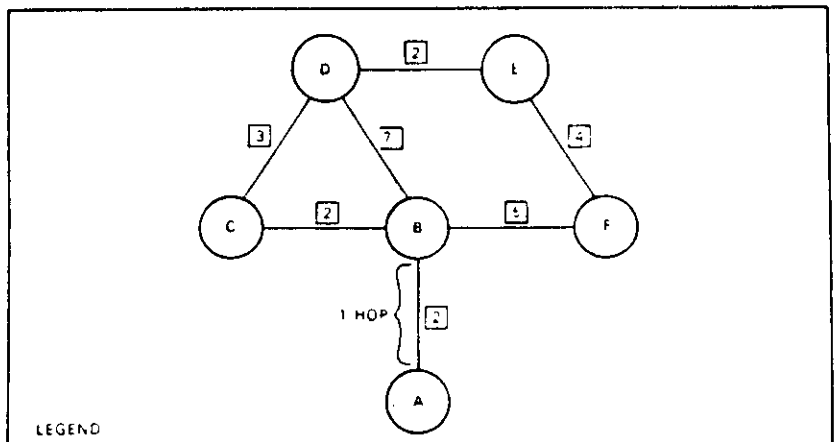
When initializing the network (when routing messages are first being transmitted and received) each network node knows only about itself and its adjacent nodes. The routing messages eventually update the entire network, so that every Routing node knows the best or most cost-effective local circuit over which to send a packet. The routing messages contain both cost and hop information for all adjacent nodes within the network. A source node need only select what it believes to be the cheapest path to the destination node and send the message packet over the local circuit that corresponds to the selected path. The first node in the path, the adjacent node, must be a route-through node so that the packet can be forwarded along its way. The route-through node performs the same calculations as the source node and forwards the packet over what it currently believes to be the cheapest path to the intended destination via its own local circuits.

This process of routing and rerouting continues until the packet:

- reaches its intended destination
- is received by a route-through node whose maximum visits limit has been exceeded.

Remember, routing does not automatically adjust to traffic flow or circuit loading; it simply sends message packets over what it believes to be the cheapest path to a particular destination node at the time the packet was queued for transmission. Also, if a packet is received by a route-through node and the packet header Forwarding Field value exceeds the maximum visits allowed by that route-through node, the packet will be discarded. Routing does not guarantee delivery of all offered packets; this task is left to the End Communication layer's NSP protocol.

Figure 3-4 shows how a node chooses the cheapest path to send a message packet to another nonadjacent node.



NODE A WANTS TO SEND A PACKET TO NODE D. THERE ARE THREE POSSIBLE PATHS

	PATH COST	PATH LENGTH
A TO B, B TO C, C TO D	2 + 2 + 3 = 7*	3 HOPS
A TO B, B, 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

TA-10848

Figure 3-4 Selecting the Most Cost-Effective Path Between Nodes

3.4 ROUTING FUNCTIONAL OPERATIONS

The following is a summary of the functional components contained within each Routing sublayer, and the processes that make up each of the different components. Figure 3-5 shows the logical positioning of the two Routing sublayers with respect to the Routing, End Communication, and Data Link layers of DNA. It also shows the different sublayer components, the processes that operate within each component, and the operations performed by each of the different processes. The following list can be used to identify the different layers, sublayers, components, and processes shown in Figure 3-5. Details are provided in the sections that follow the figure.

Data Link Layer

Routing Layer

I. Routing Initialization Sublayer

A. Initialization and Circuit Monitoring Components

1. Ethernet Circuit
2. DDCMP Circuit
3. X.25 Circuit
4. X.25 Circuit Mapping Data Base

II. Routing Control Sublayer

B. Routing Component

5. Decision Process
6. Update Process
7. Forwarding Process
8. Receive Process
9. Routing Data Base
10. Forwarding Data Base

C. Congestion Control Component

D. Packet Lifetime Control Component

11. Loop detector process

End Communication Layer

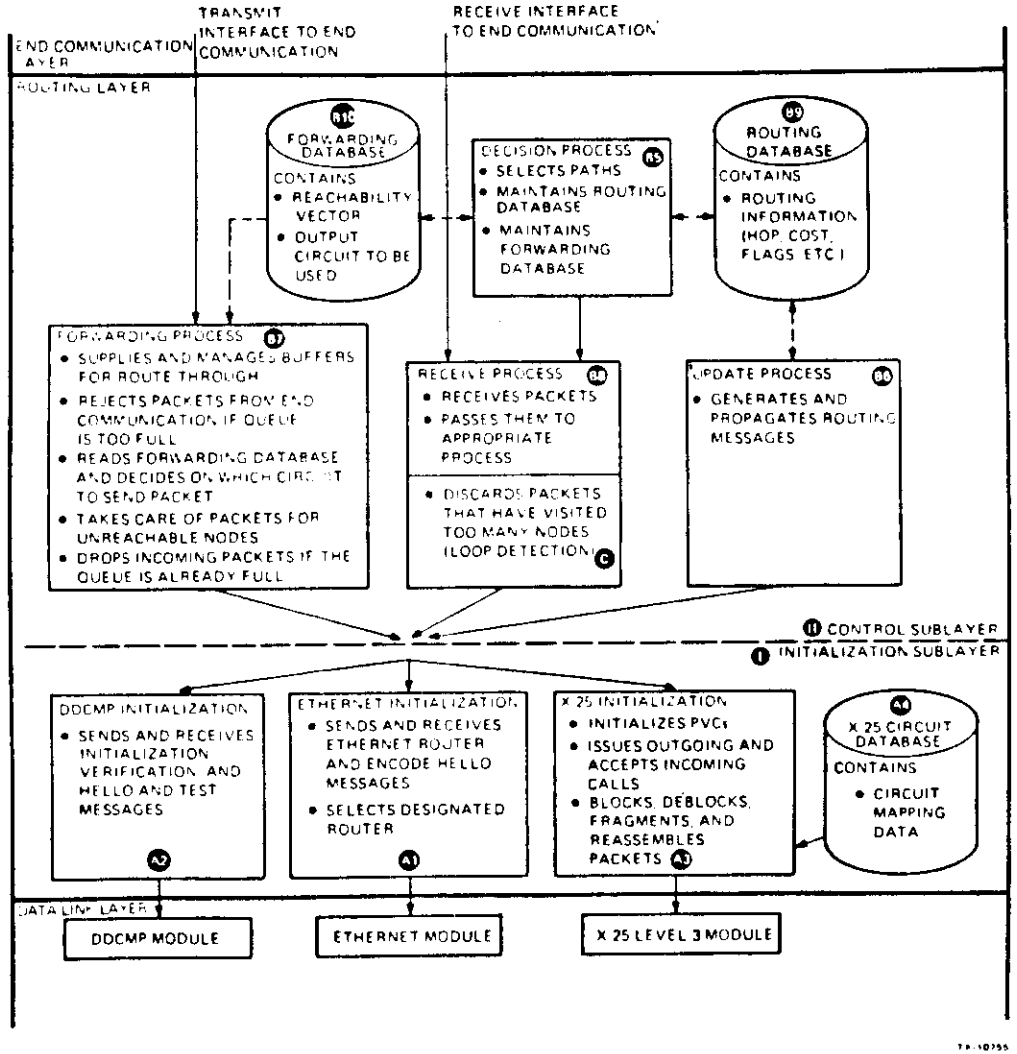


Figure 3-5 Routing Layer Sublayers, Components, and Processes

3.4.1 Routing Initialization Sublayer

The Initialization sublayer masks the characteristics of the Data Link layer from the Routing Control sublayer. It performs the following functions:

- Link initialization (start up) for adjacent node's Routing layers
- Identification of an adjacent node's Routing layer type

  Phase IV routing node

  Phase IV end node

  Phase III routing node

  Phase III end node
- Adaptation to the circuit characteristics of the line that connects to the adjacent node
- Circuit monitoring (detects the loss of communication with an adjacent node)

Routing initialization is a start-up procedure for adjacent nodes that identifies the Routing layer type. The start-up procedure adapts the local node's Routing layer to the characteristics of the circuit that provides the link between the adjacent nodes.

Initialization and circuit monitoring operations depend on the circuit type that links the adjacent nodes. There are two circuit types supported by DNA Phase IV:

1. Nonbroadcast (DDCMP and X.25)
2. Broadcast (Ethernet)

3.4.1.1 Nonbroadcast Circuit Initialization - The initialization and identification of adjacent node's Routing layers on a Nonbroadcast circuit is accomplished by exchanging Routing initialization and verification messages. The operations performed are functionally similar for both DDCMP and X.25 circuits. DDCMP initialization adapts the Routing layer's operation to the block size of the physical line managed by the Data Link layer; X.25 initialization adapts the Routing layer's operation to the X.25 Data Packet size selected when the virtual circuit is established.

There are some differences between the initialization of DDCMP and X.25 circuits. These differences however, affect only the contents of the routing message envelope and the size of the data packet passed down to the Data Link layer (amount of data bytes to be transmitted). The Routing layer initialization handshaking and node type identification sequences are the same for both nonbroadcast circuit types. Figures 3-6 through 3-11 show the initialization handshaking sequences required to initialize and identify the different Routing layer types that may reside in the adjacent node.

The major differences between DDCMP and X.25 initialization operations are:

- DDCMP supports the Hello and Test message; X.25 circuits do not.
- X.25 initialization performs some additional operations not required by either DDMCP or Ethernet circuits:

  Blocks and Deblocks Routing Layer Messages - Communication over X.25 circuits is in the form of small packets; Routing initialization blocks and deblocks DNA datagrams into X.25 packets 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 fragments and reassembles Routing layer messages so that they fit in the available X.25 packet size.

  Checksums X.25 Packets - The Routing initialization sublayer appends a checksum value to every Data Packet transmitted to ensure the integrity of the data transmitted over the virtual circuit.

Nonbroadcast Routing layers exchange the following message types to initialize, identify, and monitor the circuit between them:

Initialization Message - The Routing layer sends this message when initializing a nonbroadcast circuit. The message contains information about the node's Routing layer type and version (Phase IV or Phase III), maximum Data Link layer receive block size, and whether or not verification is required.

Verification Message - This message is sent and used only for verification purposes on nonbroadcast circuits, and only if the initialization message indicates that verification is necessary. It ensures that only authorized users be allowed access to your system and network.

Hello and Test Message - The Hello and Test message is used to test an adjacency to determine if it is still operational. Routing sends this message periodically on nonbroadcast circuits in the absence of other normal data 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 adjacent node, routing considers that adjacency as being nonoperational, or down.

Phase IV End Node Initialization - Figure 3-6 shows the following sequence of events that initialize and identify the routing layers of two adjacent nonbroadcast Phase IV end nodes.

1. Node A starts the sequence by transmitting the Routing initialization message to node B.
2. Node B responds by transmitting its Routing initialization message back to node A.
3. Both Routing layers then exchange verification messages if required by the Initialization messages.

The exchange of the initialization messages initializes and identifies the two adjacent nodes as being DECnet Phase IV end nodes.

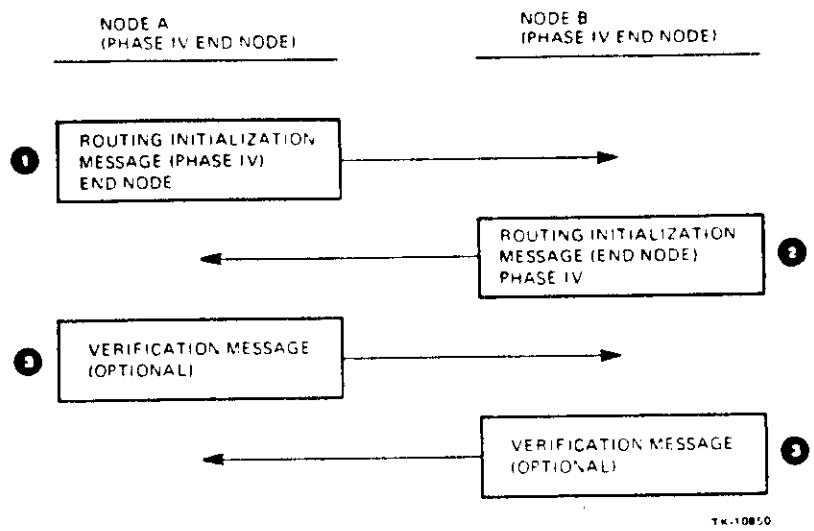


Figure 3-6 Phase IV to Phase IV Nonbroadcast Circuit End Node Initialization and Identification

Phase IV to Phase III End Node Initialization - Figure 3-7 shows the following sequence of events that initialize and identify the Routing layers of two adjacent nonbroadcast end nodes.

1. Node A starts the sequence by transmitting the Routing initialization message (Phase IV end node) to node B.
2. Node B responds by transmitting its Routing initialization message (Phase III end node).
3. Node A receives node B's Phase III Routing initialization message and determines that node B is in fact a Phase III end node.
4. Node A then reformats its Routing initialization message into DECnet Phase III format and retransmits it to node B.
5. Both routing layers then exchange verification messages if required by the initialization messages.

The exchange of the initialization messages initializes the Routing layers of the two adjacent nodes and informs node A that it must reformat all messages transmitted to node B so that they will be compatible to node B's Routing layer.

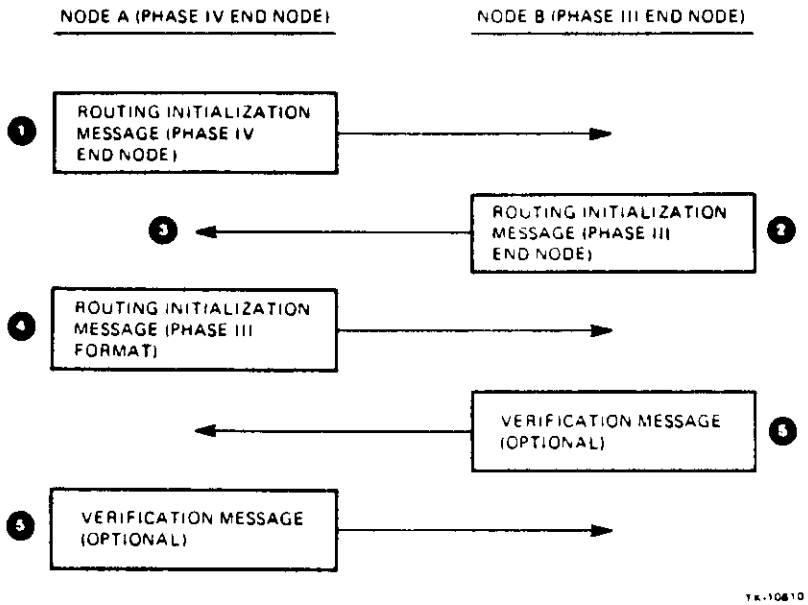


Figure 3-7 Phase IV to Phase III Nonbroadcast Circuit End Node Initialization and Identification

Phase IV Routing Node Initialization - Figure 3-8 shows the following sequence of events that initialize and identify the Routing layers of two adjacent nonbroadcast Phase IV full-routing nodes.

1. Node A starts the sequence by transmitting the Routing initialization message to node B.
2. Node B responds by transmitting its Routing initialization message back to node A.
3. Both Routing layers would then exchange verification messages if required by the initialization messages.
4. The exchange of initialization messages initializes and identifies the two adjacent nodes as DECnet Phase IV Routers. Since both are identified as routing nodes, they begin the process of updating each other's Routing Data Base Structures. (This process is covered later in the Routing Control Sublayer section using the Routing message.)

NOTE

In DECnet Phase IV, the Routing message is segmented. Phase IV routing nodes only exchange new or changed information. This protects CPU and Network resources from waste caused by transmitting redundant information.

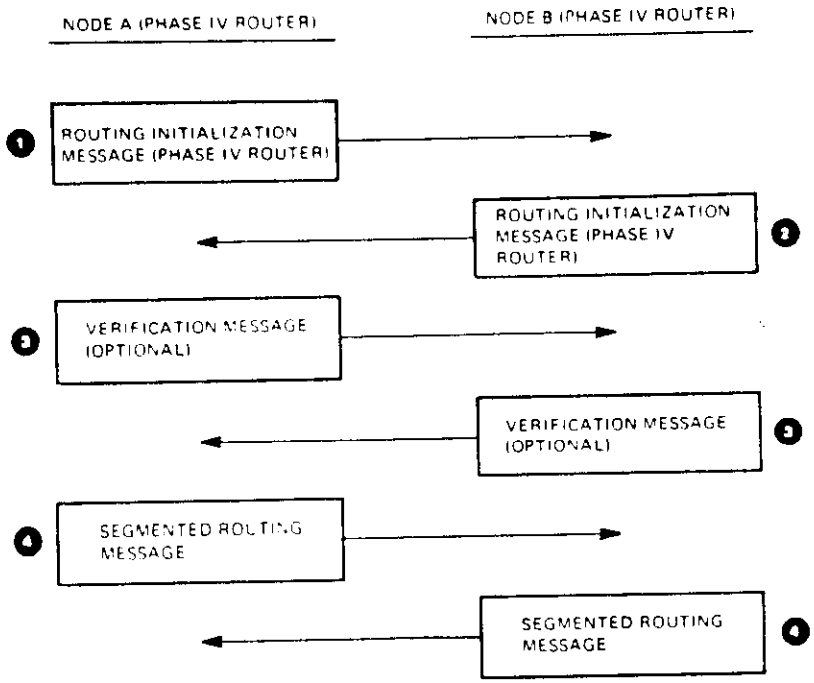


Figure 3-8 Phase IV to Phase IV Nonbroadcast Circuit Routing Node Initialization and Identification

Phase IV to Phase III Routing Node Initialization - Figure 3-9 shows the following sequence of events that initialize and identify the Routing layers of two adjacent nonbroadcast routing nodes.

1. Node A starts the sequence by transmitting the Routing initialization message (Phase IV Router) to node B.
2. Node B responds by transmitting its Routing initialization message (Phase III Router) back to node A.
3. Node A receives node B's Phase III Routing initialization message and determines that node B is in fact a Phase III routing node.
4. Node A then reformats its Routing initialization message into the Phase III format and retransmits it to node B.
5. Both Routing layers would then exchange verification messages if required by the initialization messages.
6. The exchange of initialization messages initializes and identifies the two adjacent nodes as routers but the Phase IV node (node A) must reformat all messages sent to node B. Since both are identified as routing nodes, they begin the process of updating each other's Routing Data Base Structures. The Routing Data Base update is accomplished using the Routing message.

NOTE

In this example, the two adjacent routers cannot exchange Phase IV segmented Routing messages. Node B does not know what a segmented Routing message is, and cannot interpret it when it is received. Therefore, node A must transmit all messages in the Phase III format when communicating with node B.

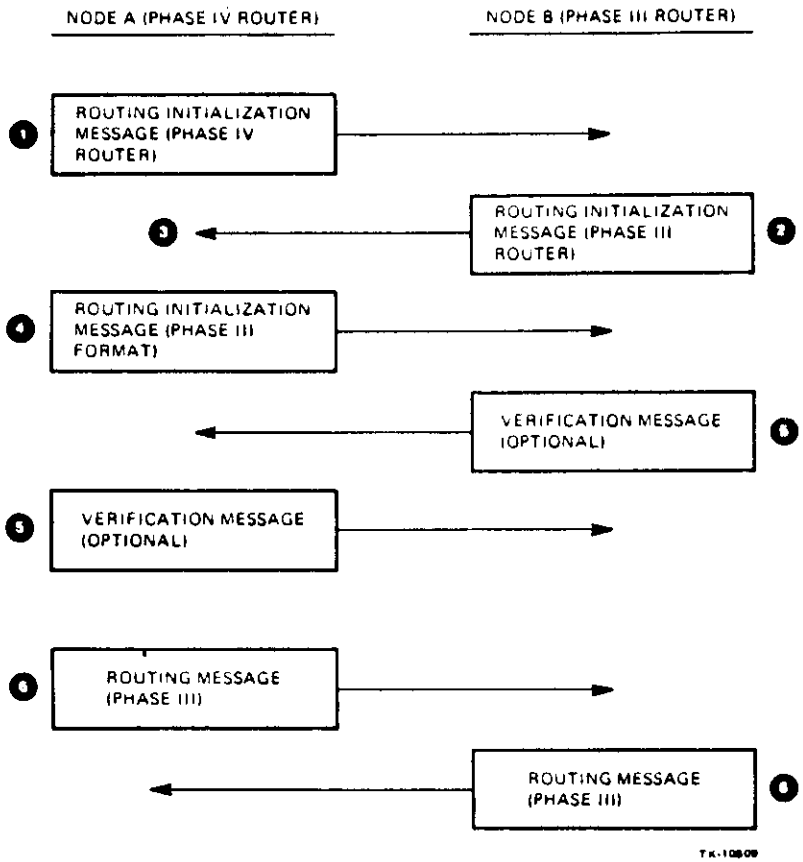


Figure 3-9 Phase IV to Phase III Nonbroadcast Circuit Routing Node Initialization and Identification

Phase IV Routing to Phase IV End Node Initialization - Figure 3-10 shows the following sequence of events that initialize and identify the Routing layers of two adjacent nonbroadcast nodes.

1. Node A starts the sequence by transmitting the Routing initialization message (Phase IV Router) to node B.
2. Node B responds by transmitting its Routing initialization message (Phase IV end node) back to node A.
3. Both Routing layers would then exchange verification messages if so required by the initialization messages.

The exchange of the initialization messages initializes and identifies node B to node A as being a Phase IV end node, and node A to node B as being a Phase IV routing node. Because Node B is an end node, they will not exchange Routing update messages. Node B does not have the capability of performing route-through for network message traffic.

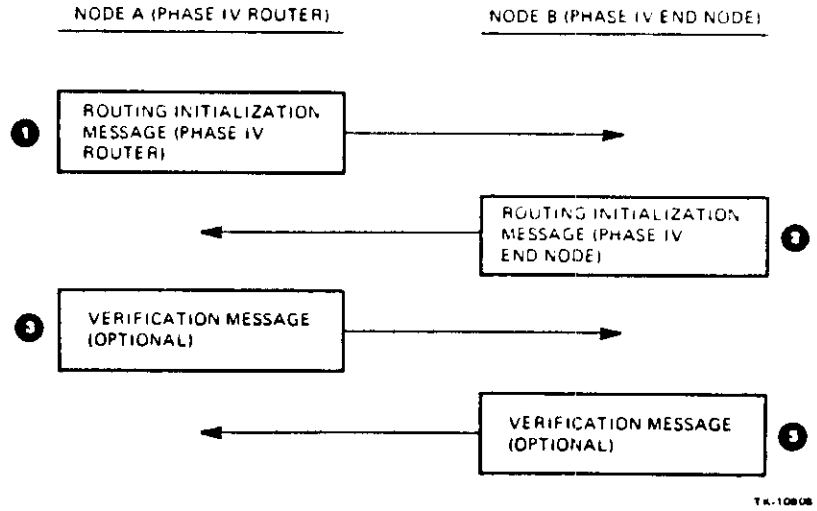


Figure 3-10 Phase IV Routing Node to Phase IV End Node Nonbroadcast Circuit Initialization and Identification

Phase IV Routing to Phase III End Node Initialization - Figure 3-11 shows the sequence of events that initialize and identify the Routing layers of two adjacent nonbroadcast nodes.

1. Node A starts the sequence by transmitting the Routing initialization message (Phase IV Router) to node B.
2. Node B responds by transmitting its Routing initialization message (Phase III end node) back to node A.
3. Node A receives node B's Phase III Routing initialization message and determines that node B is in fact a Phase III end node.
4. Node A must then reformat its Routing initialization message into DECnet Phase III format and retransmit it to node B.
5. Both Routing layers would then exchange verification messages if required by the initialization messages.

The exchange of the initialization messages initializes the Routing layers of the two adjacent nodes and informs node A that it must reformat all messages transmitted to node B so they are compatible to node B's Routing layer. It also informs node A that node B is an end node and cannot perform route-through for network traffic. Node A and node B will not exchange any routing update messages other than Hello and Test messages.

ROUTING LAYER

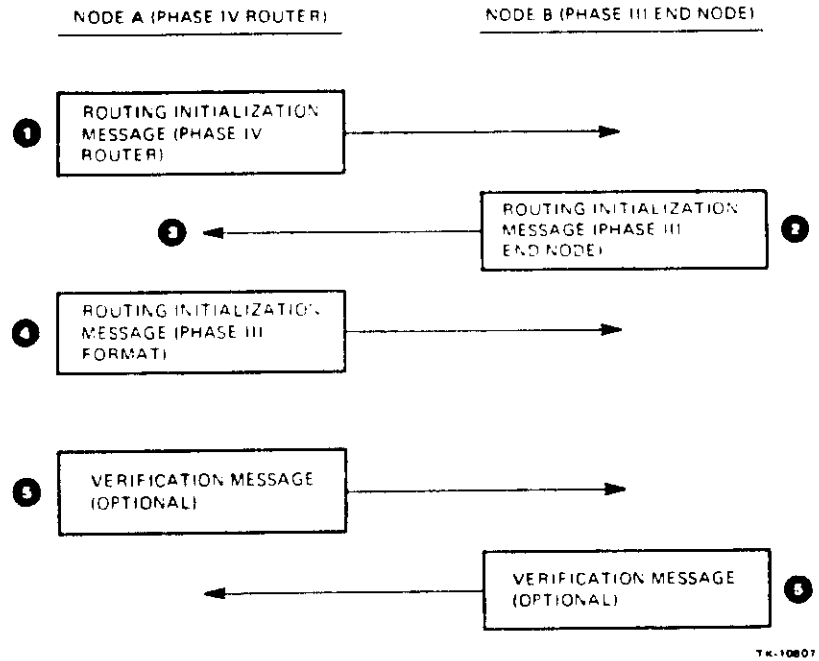


Figure 3-11 Phase IV Routing Node to Phase III End Node Nonbroadcast Circuit Initialization and Identification

3.4.1.2 Broadcast Circuit Initialization - The initialization, identification, and circuit monitoring of adjacent Routing layers on a broadcast circuit is accomplished by exchanging the following two Ethernet Routing initialization messages:

1. **Ethernet Router Hello Message** - The Ethernet Router message is used for both circuit initialization and monitoring of route-through nodes on an Ethernet circuit. Each Ethernet Router periodically broadcasts an Ethernet Hello message to all other routers on the same Ethernet circuit using multicast operation. (Multicast operation was discussed in the Ethernet section of the Data Link Layer module.)

   The Router Hello message contains a list of all known routers on the Ethernet circuit from which the sending router has recently received Ethernet Router Hello messages. By exchanging these 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 that selects a single router on the Ethernet circuit to be the Designated Router for that Ethernet circuit.

   The Designated Router sends the Ethernet Router Hello message to both routers and end nodes alike. It also helps any two end nodes discover that both are on the same Ethernet circuit by forwarding a packet from one to the other as an Intra-Ethernet packet. The Intra-Ethernet packet informs the receiving node that the source node is on the same physical Ethernet circuit and that direct communication can be accomplished simply by addressing that node within the Ethernet message packet header field "Destination Address". Then subsequent communication between the two end nodes can be made directly, without the aid of the Designated Router.

2. **Ethernet Endnode Hello Message** - The Ethernet Endnode Hello message is used for initialization and periodic monitoring of end nodes on an Ethernet circuit. Each Ethernet End Node periodically broadcasts an Ethernet Endnode Hello message to all routers on the Ethernet circuit by using the multicast operation. The message is used by the Routers to maintain the status (up or down) of the end nodes on the Ethernet circuit.

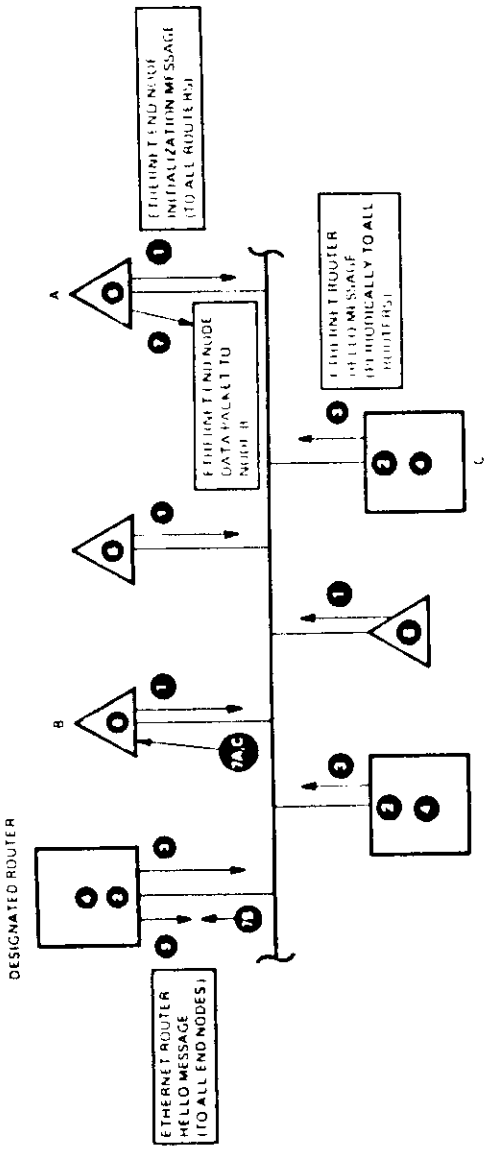
Ethernet End Node Initialization - Figure 3-12 shows the following sequence of events that initialize and identify the Routing layers of adjacent broadcast nodes. The following steps are provided to help guide you through the broadcast circuit initialization sequences with respect to an Ethernet End Node.

1. Node A, an Ethernet End Node, multicasts the Ethernet Endnode Hello message" to all routers on the local Ethernet circuit. This message is repeated periodically to keep all routers updated as to the status of node A (up or down).
2. The routers update their data bases to include the non-routing end node A.
3. The routers multicast the Ethernet Router Hello message periodically to all other routers on the Ethernet circuit.
4. The routers decide on one router to be the Designated Router for the circuit.
5. The Designated Router (the router with the highest router priority) additionally multicasts the Ethernet Router Hello message to all end nodes.
6. The non-routing node A stores the information about the designated router in a subarea of its Routing layer called cache. Cache is a temporary storage area for known current adjacencies to the Ethernet End Node. The cache is updated by the Ethernet End Node's Routing layer each time an Intra-Ethernet packet is received.
7. When the end node, A, wants to communicate with another node, B, it:

   a. Checks cache for information about node B. If there is such information, node A addresses node B directly.

   b. If there is no information in the cache about node B, node A addresses the message to the Designated Router for route-through to node B.

   c. If there is no information in the cache about node B, and there is not yet a router assigned as the Designated Router, node A addresses node B directly.



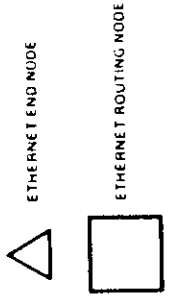


Figure 3-12 Phase IV Ethernet End Node Initialization and Identification

Ethernet Routing Node Initialization - Figure 3-13 shows the sequence of events that initialize and identify the Routing layers of adjacent broadcast nodes. The following steps are provided to help guide you through the broadcast circuit initialization sequences with respect to an Ethernet Routing Node.

- Node C, an Ethernet Routing Node, multicasts the Ethernet Router Hello message to all routers on the Ethernet circuit.
- The other routers on the Ethernet circuit (since all nodes on the Ethernet circuit are adjacent nodes) update their data bases to include the information for node C.
- Each router then decides on the Designated Router for the Ethernet circuit. The decision is based upon the router priority contained in the Ethernet Router Hello message. The router with the highest priority becomes the Designated Router; in case of a tie, the router with the highest node address becomes the new Designated Router for the circuit.
- The Designated Router periodically multicasts the Ethernet Router Hello message to all other routers and end nodes alike. Node C may or may not have become the new Designated Router for the circuit.
- Node C learns about the end nodes from the Ethernet Endnode Hello messages and about the other routers on the circuit from the Ethernet Router Hello messages.

NOTE

Due to resource restraints, Ethernet Networks should be limited to a maximum of 10 routers per Ethernet Circuit.

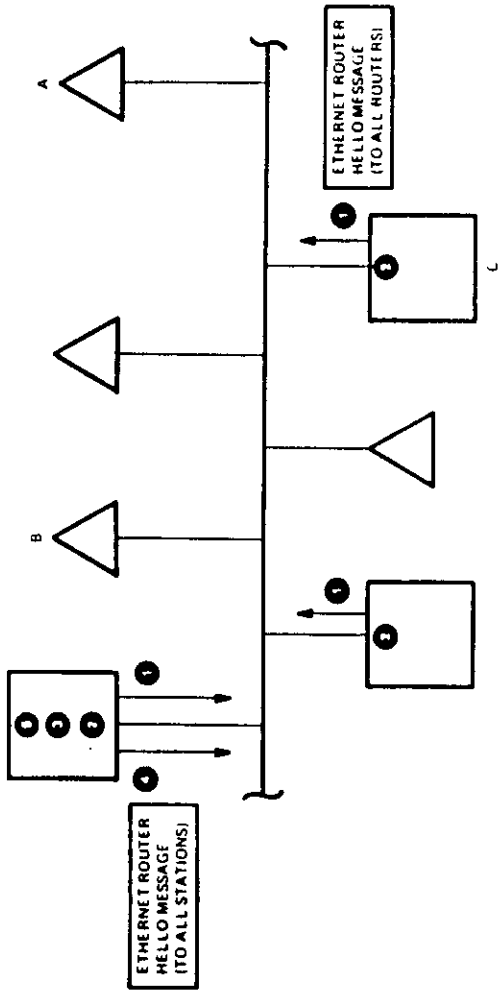


Figure 3-13 Phase IV Ethernet Routing Node Initialization and Identification

3.4.2 Routing Control Sublayer

The Routing Control sublayer supplies full-duplex packet transmission between any pair of nodes within the network. It is independent of the specific Data Link layer below it. All it knows about the Data Link layer is whether it contains either or both broadcast (Ethernet) or nonbroadcast (X.25 or DDCMP) circuits. The Routing Control sublayer also masks the physical and topological characteristics of the network from the higher DNA layers. Figure 3-4 showed an overall functional view of the Routing Control sublayer operations. It also summarized sublayer functions that are performed by the different sublayer components and processes.

3.4.2.1 Routing Component - Figure 3-14 shows the different processes that make up the Routing component. This component consists of:

- Decision Process
- Update Process
- Forwarding Process
- Receive Process
- Routing Data Base
- Forwarding Data Base

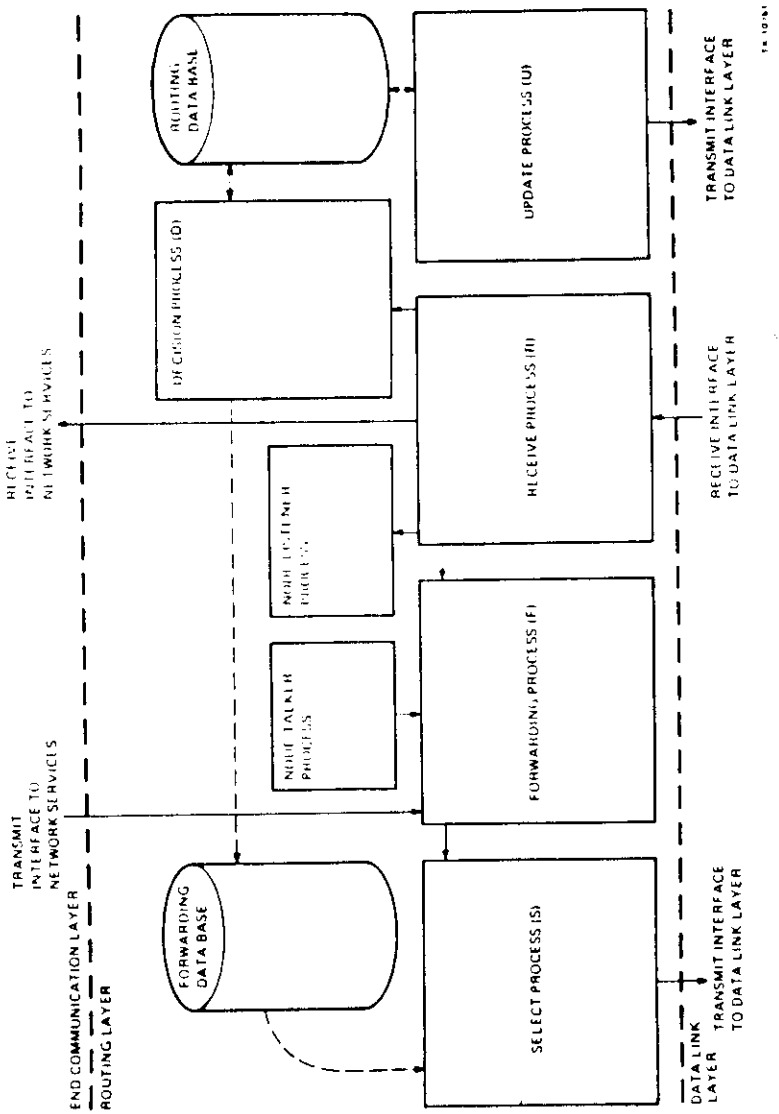


Figure 3-14 Routing Control Sublayer's Routing Components

Decision Process - The Decision process maintains and updates the data bases used to select the most cost-effective path to a particular destination node at any given instant in time.

The decision process selects the best path to each network node that could be a messages destination. This process uses two algorithms to determine the minimum path cost and distance or hops to each known destination in the network. Path selection is based upon the minimum path cost to a destination node. The hops information to the destination node is not used to determine the best path. The minimum cost and hops for a path to a destination node is determined by storing all of the possible path's cost and hops in a matrix, called the Routing Data Base Hop/Cost Matrix. This matrix is updated when the topology is changed by the Update process. (The Hop/Cost Matrix and Update process are detailed later in this section.) Figure 3-15 shows the decision process functions.

The decision process algorithms use the information stored in the Matrix to perform the following operations:

- Form a data base called the Routing Update Message Vector. This vector is used by the Update process to create a message called the Routing Update message. The Routing Update message can be a partial or complete copy of the Routing Update Message Vector built from the Routing Data Base Matrix. The Routing Update message is transmitted to all adjacent route-through nodes when queued for transmission by the Update Process.

- Form a data base called the Forwarding Data Base Vector. This vector stores the circuit's identification used to send message packets to any known destination node.

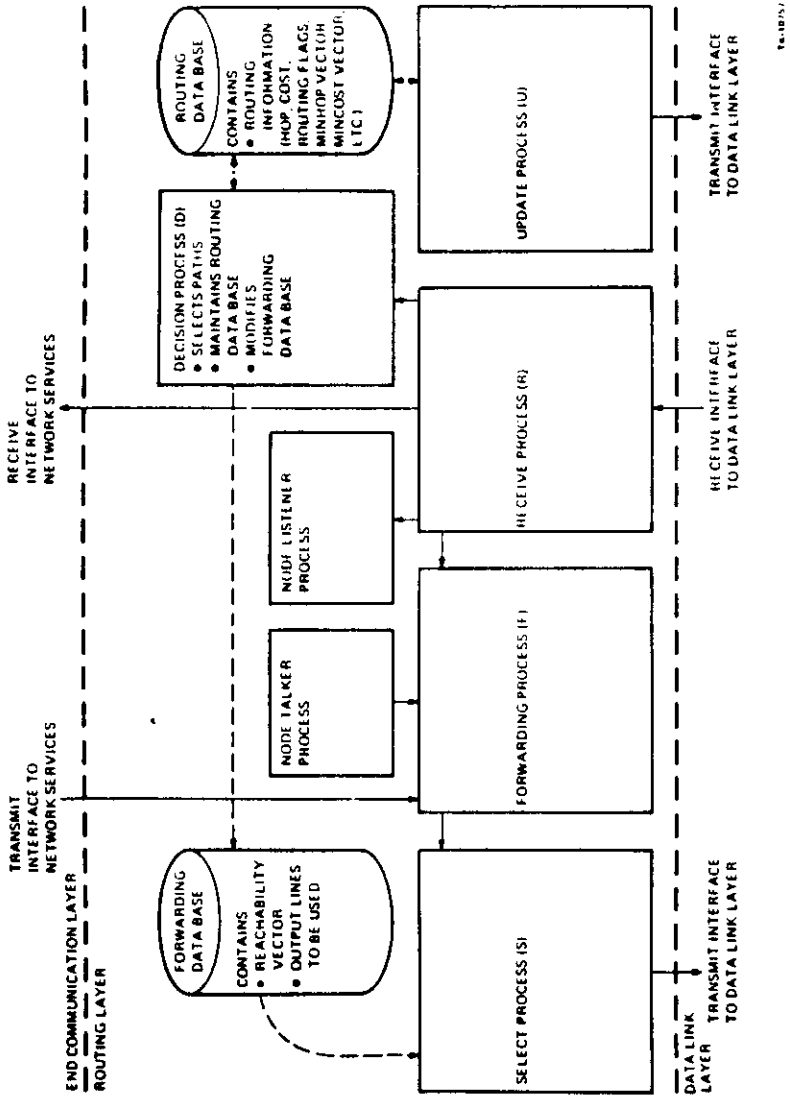


Figure 3-15 The Decision Process

The decision process also builds a buffer that contains a flag called the Send Routing Message Flag. This buffer contains a flag for all known adjacent nodes. The decision to set the flag for any particular circuit to an adjacent node is based upon information obtained by the Initialization sublayer.

Setting the flag allows the Update process to transmit the Routing Update message to the adjacent node connected by that circuit. The decision process initially sets the flag to identify an adjacent node as a route-through node. Once set, it remains set until a network change affects that circuit or the adjacent route-through node connected by that circuit. Circuits without this flag are connected to end nodes, nodes that are not route-through nodes. Circuits to end nodes must not be identified by the Forwarding Data Base Vector as a circuit to be used for communication with nonadjacent destination nodes.

The decision process obtains the information for its decisions from the Routing Update and Initialization messages. Routing Update messages inform network nodes of topological changes. Changes are reported through hop and cost information updates for known node adjacency pairs. (Initialization messages were covered earlier in this module.)

The two algorithms executed by the decision process are called the Connectivity and Assignment algorithms. Executing these algorithms results in updates to the data bases used by the decision process. The connectivity algorithm updates the path length portion of the Routing Data Base Matrix (path length is the cumulative hops for the entire path). The assignment algorithm updates the total path cost portion of the Routing Data Base Matrix. (Total path cost is the cumulative circuit costs for the entire path.)

Update Process - The Update process is responsible for building and sending the Routing Update message once the Routing Update Message Vector has been created or updated by the decision process.

The Routing Update message is transmitted to all adjacent route-through nodes; the Send Routing Message Flag is set for the circuit that connects that node. The receiving adjacent route-through node uses the Routing Update message to build or update its own Routing Data Base Matrix and Routing Update Message Vector.

A Routing Update message contains path cost and length information for groups of destination nodes. The information reported or updated can cover the entire known network or a small part of the network, depending upon:

- The size of the buffer available to the Update process for transmitting the Routing Update message.
- The amount of information or number of adjacency pairs that require updating.

DNA Phase IV uses a segmented Routing message that allows Routing layers to communicate either a total Routing Data Base Matrix Update (the complete Routing Update Message Vector for all known destination paths as in DNA Phase III) or only the portions of the Routing Update Message Vector that have changed since the last Routing Data Base Matrix update resulted.

Routing Update message transmission under the control of the update process is event-driven with periodic backup. The update process transmits a Routing Update message to an adjacent node only under the following conditions:

Network changes are detected - A buffer is made available to the update process, a Rate Control Frequency Timer (T2) expires (T2 is a 60 second timer), a change in the network topology is detected since the timer's expiration, and the Send Routing Message Flag is set for the circuit to that adjacent node.

No network changes are detected - A buffer is made available to the update process, no new network topological changes have been detected, the Send Routing Message Flag is set for the circuit to that adjacent node, and a Background Frequency Timer (T1) or (BCT1) has expired (T1 or BCT1's durations are system manager selectable. It is suggested that they both be set for 10 second intervals. T1 is used for Nonbroadcast circuits; BCT1 is used for Broadcast circuits).

The operation performed by the Update process is shown in Figure 3-16.

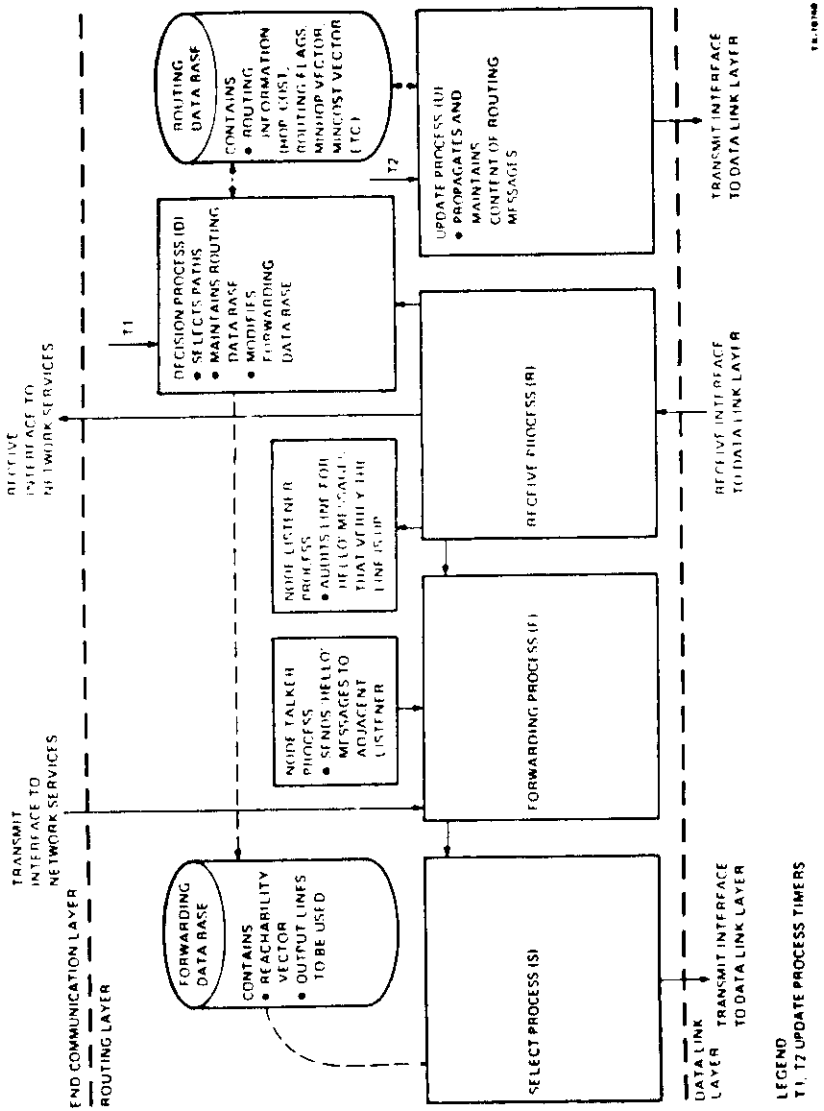


Figure 3-16 The Update Process

Forwarding Process - The Forwarding process supplies and manages the buffers necessary to support both route-through packet traffic for all known, operational adjacencies and local packet traffic from the local End Communication layer to all known operational adjacencies. Figure 3-17 shows the operations performed by the Forwarding process.

The Forwarding process performs the actual Forwarding Data Base Vector lookup to select the proper transmission circuit for sending packets to a given destination. If a destination is unreachable, this process either returns the packet to its sender or discards it, depending on the packet's Routing Packet Header request.

The Forwarding process is also responsible for packet formatting or reformatting to facilitate communications between all DECnet routing and non-routing, broadcast and nonbroadcast nodes.

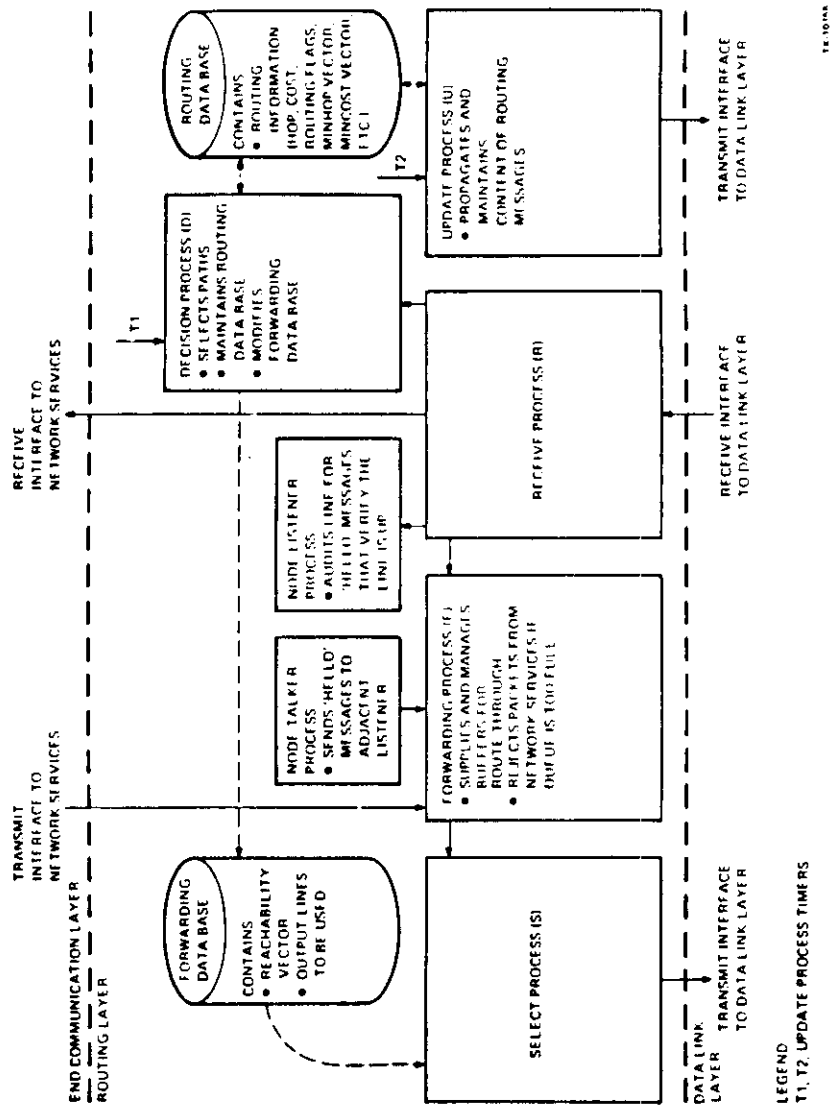


Figure 3-17 The Forwarding Process

The Forwarding and Decision processes interact when a message packet is queued for transmission.

- The Decision process - Builds and maintains the Routing Data Base Matrix. From this matrix, it maintains an updated Forwarding Data Base Vector. The vector contains the circuit's identification path which correlates the most cost-effective path to any given known destination.

- The Forwarding process - Reads the circuit's identification from the Forwarding Data Base Vector for the message's destination. It selects this circuit for transmission, then formats the message packet to conform to that circuit's transmission requirements (to ensure that the packet's byte size does not exceed the transmission circuit's maximum data field byte size, and if necessary, make changes to Routing header information so that the message packet is compatible with the adjacent node's Routing layer characteristics). Once the packet is correctly formatted, the Forwarding process passes it down to the Data Link layer module that manages the selected circuit.

Receive Process - The Receive process receives packets from the different Data Link layer modules, inspects the packet's route header information, then directs the packet to the appropriate component process or End Communication layer for futher processing.

Table 3-1 shows the possible Routing packet types and the component, process, or DNA layer to which they will be directed by the Receive process.

Table 3-1 Routing Packet Types and Their Destinations

Routing Packet Type	Destination
Routing Update Messages	Decision Process
Hello Message	Node Listener Process
Packet for Self (local node)	End Communication layer
Packet for Another Destination (requires forwarding)	Forwarding Process

3.4.2.2 Congestion Control Component - The Congestion Control Component consists of a single process called Transmit Management that:

- Manages buffers by limiting the maximum number of packets on a queue for a given circuit.

- Regulates the ratio of packets received directly from the End Communication layer to the packets received for route-through service. Packets that require forwarding (route-through service) have higher priority.

- Prevents circuit congestion by rejecting locally generated packets to keep route-through service from being degraded.

- Discards packets that are queued for an adjacent node that has gone down (become unreachable).

- Checks packet size for each packet to be transmitted. This prevents data from being lost due to an insufficient number of bytes available in the data field of the Data Link layer protocol used for transmission. The amount of data that can be transmitted by the Data Link layer depends upon the Data Link layer circuit type used for transmission. The Forwarding process performs the packet reformatting if reformatting is necessary.

3.4.2.3 Packet Lifetime Control Component - This component prevents excessive looping of route-through packets by discarding all packets that have visited too many nodes. The major functions of the Packet Lifetime Control component are performed by the Loop Detector process. This process prevents excessive packet looping. It counts the number of nodes that a packet has visited (been routed through) and discards the packet if it has exceeded the local node's visit limit.

Figure 3-18 shows the operations performed by the Receive process and the Congestion and Lifetime Control components. It shows: 1) the Receive process distributing received packets 2) the Congestion Control component managing the number of local packets allowed to be transmitted considering the number of packets that require forwarding, and 3) the Lifetime Control component discarding old undeliverable route-through packets.

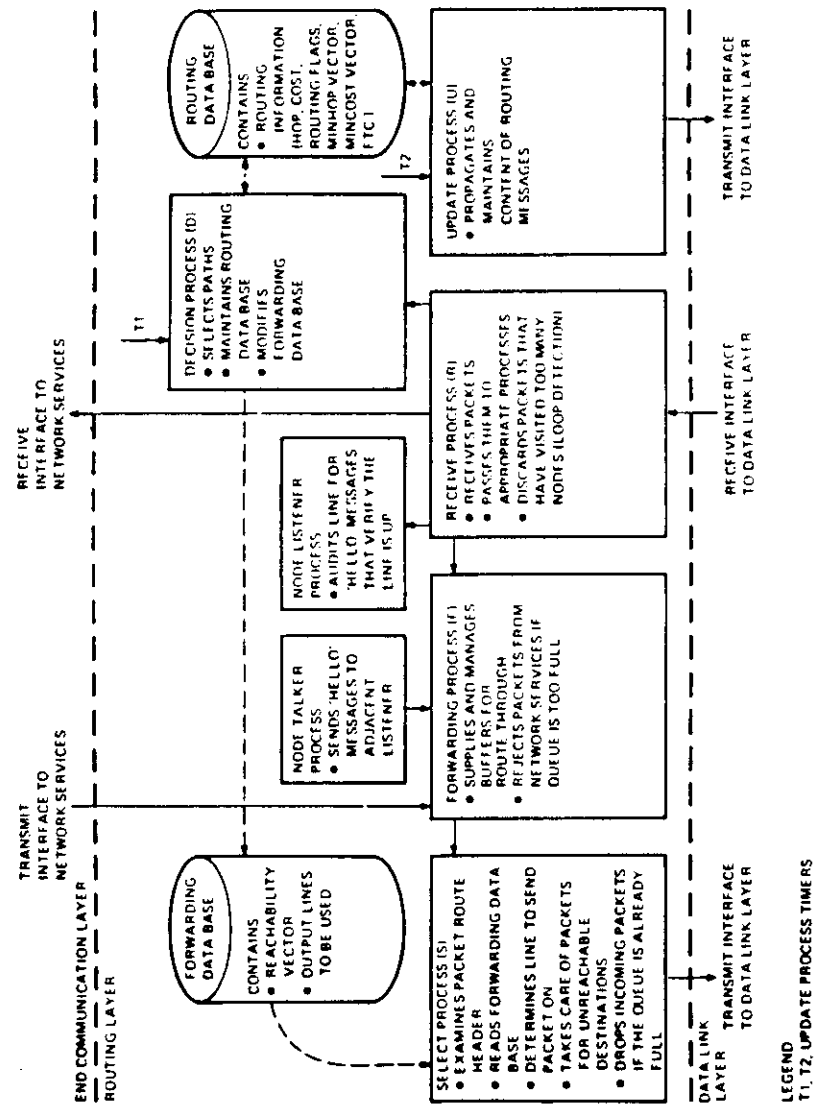


Figure 3-18 Operations Performed by the Receive Process, Congestion Control, and Lifetime Control Components

3.5 ROUTING LAYER MESSAGE FORMATS

There are two types of Routing Layer Messages used by DNA Phase IV:

1. Data Packets - Carry data to and from the End Communication layer. The Routing layer processes append a packet route header to all messages queued for transmission by the End Communication layer.

2. Control Messages - Exchange routing level information between Routing layer software modules in adjacent network nodes. This routing level information is used to initialize, identify, update, and monitor the operations performed by the adjacent Routing layer modules.

3.5.1 Data Packets

There are two Route Header formats used for the data packets; the format used depends on the circuit type connecting the adjacent nodes and the adjacent node's Routing layer type. The circuit could be either Broadcast (Ethernet) or Nonbroadcast (DDCMP or X.25); the Routing layer type could be either route-through or end node. The two packet route headers used are:

- Phase IV Data Packet Routing Header (called the Short Route Header)

- Ethernet Endnode Data Packet Routing Header (called the Long Route Header)

The Short Route Header is appended to all data packets transmitted through the network except when communication is with an Ethernet end node. If communication is with an Ethernet end node, the Ethernet Endnode Data Packet Routing Header (the Long Route Header) is appended to the data packet. Figure 3-19 shows the two types of data packet routing headers. The numbers under the different fields in this and the following Routing message format illustrations indicate the length of the fields in bits.

Phase IV Data Packet Routing Header Format

RTFLG	DSTNODE	SRCNODE	FORWARD
8	16	16	8

Bits

Ethernet Endnode Data Packet Routing Header Format

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

Bits

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

TK 10/86

Figure 3-19 Data Packet Route Headers

3.5.2 Control Messages

There are six types of Routing Control messages used by the Routing layer's Initialization sublayer; the message(s) used are dependent upon the circuit used to connect to the adjacent node. Figure 3-20 shows the message format for all six control messages. The following description identifies each control message and summarizes their use by the Routing Initialization sublayer.

- Common to all circuit and Routing layer types:

  Routing Message - Provides information on path cost and path length for a set of destinations to update the Routing Data Base of an adjacent node.

- Used only on Nonbroadcast DDCMP circuits:

  Hello and Test Message - Test an adjacency to determine if it is still operational. Routing sends this message periodically on Nonbroadcast circuits in the absence of other normal traffic. Upon receipt of this or any other message, routing starts a timer (or restarts the timer). If the timer expires before another message is received from that node, routing considers that adjacency down. Hello and Test messages are transmitted as a function of the Node Talker process in the Initialization sublayer. The timer that indicates "Adjacency Down" is controlled by the Node Listener process in the Initialization sublayer. Normally the Node Talker process sends Hello and Test messages twice as often as needed to restart the Node Listener timer. Essentially, the Listener timer is twice as long as the Talker timer.

- Used only on Nonbroadcast circuits (both DDCMP and X.25 links):

  Initialization Message - Contains information about the node type, required verification, maximum Data Link layer receive block size, and Routing version (DNA Phase III or IV). This message is used when initializing a Nonbroadcast circuit.

  Verification Message - Used for verification purposes on Nonbroadcast circuits if the initialization message indicates that verification is required.

- Used only on Broadcast circuits (Ethernet links):

  Ethernet Router Hello Message - Initializes and monitors routers on an Ethernet circuit. It is multicast from routers to all other routers and from the Designated router to all end nodes. The Router Hello message is used to determine which router on the circuit is to be the Designated router, and to update the Routing Data Bases of all routers on the circuit. It also is used to update the cache memories in end nodes so that they know who is the Designated router for the circuit.

  The Hello message is transmitted periodically to determine if there are any new nodes to add or any nonoperational nodes to drop from the list of known operational adjacencies. The Node Talker and Node Listener processes control the use of the Hello message. These processes operate on Ethernet circuits as they do on Nonbroadcast circuits. The major difference is the length of the Talker and Listener timers; for Ethernet support, the Listener timer is eight times longer than the Talker timer.

  Ethernet Endnode Hello Message - Initializes and monitors end nodes on an Ethernet circuit. Each end node periodically multicasts the Endnode Hello message to all routers on the circuit so that the routers can maintain current status on that end node. This message is also controlled by the Node Talker and Listener processes. These processes operate the same as the Talker and Listener processes discussed earlier. The lengths of the timers are the same as the Ethernet Router Talker and Listener timers. The Listener timer is eight times longer than the Talker timer.

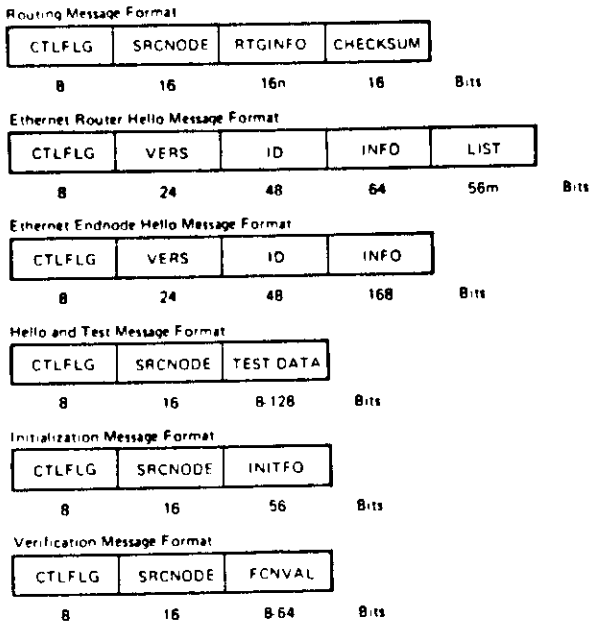


Figure 3-20 Control Message Formats

3.6 MODULE EXERCISE

Table 3-2 lists the network and node parameters for Figure 3-21. Carefully study both the figure and the table. After you have given yourself ample time to become familiar with the network and its parameters, perform the steps and answer the questions on the following pages. When you have finished, compare your answers to those provided.

Table 3-2 Network and Node Parameters

Node	Circuit	Circuit Cost	Node Limits	
			Max. Hops	Max. Visits
A	A-B	2	10	19
B	B-A	2	10	15
	B-C	8		
	B-D	10		
	B-F	2		
C	C-B	8	1	2
	C-D	6		
D	D-B	10	6	11
	D-C	3		
	D-E	2		
E	E-D	2	1	2
	E-F	4		
F	F-B	2	3	4
	F-E	4		

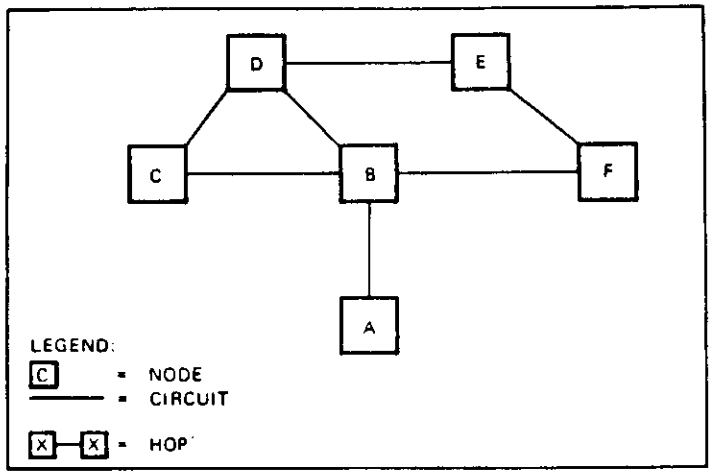


Figure 3-21 Nonbroadcast Network Topology

1. Using Figure 3-4 as a reference, fill in the circuit cost between each node in the network shown in Figure 3-21 using the values and parameters listed in Table 3-2.

2. Beside each node in Figure 3-21, write the node's value for maximum hops and maximum visits.

3. A user at node A wishes to send a message to a user at node D. With a pencil, trace the path that you believe the data must follow to get from the Source node (node A) to the Destination node (node D).

   Check your responses thus far with the solutions provided. If everything is correct, proceed. If you have made any errors, go back to the section of the module that covers the area in which you had problems, then correct your errors and proceed.

4. Will the user message transmitted from node A ever arrive at node D?

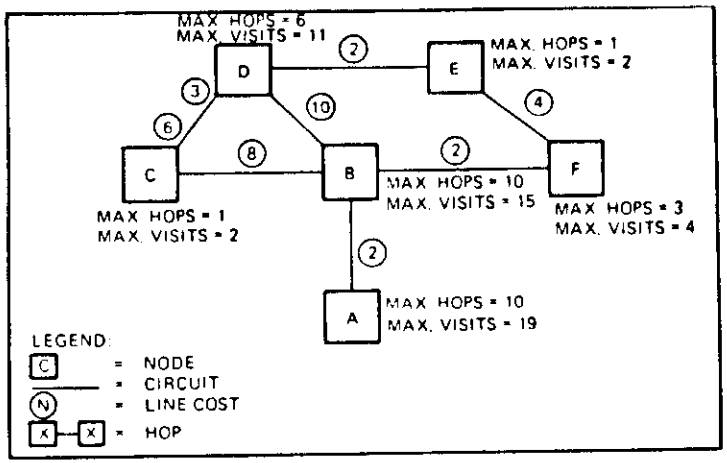
5. If YES; How did it get there and what operations were necessary at the different route-through nodes?

6. If NO; Why not, and what do you think might be a possible solution to the problem?

   Compare your answers to the solutions provided on the next page. If your responses agree with the solutions, take the module test. If they do not agree, go back and study this module. The concepts covered in this module are extremely important and act as building blocks for the rest of the course.

SOLUTIONS TO EXERCISE

1. - 3.



PATH	PATH COST	PATH LENGTH
A TO B, B TO F, F TO E, E TO D	(2) + (2) + (4) + (2) = 10*	4
A TO B, B TO D	(2) + (10) = 12	2
A TO B, B TO C, C TO D	(2) + (8) + (6) = 16	3

TK-10858

Nonbroadcast Network Topology
Showing the Desired Responses

The best path from node A to node D is via nodes B, F, and E. The message packet destined to node D must travel the path through nodes B, F, and E.

4. NO

5. An answer to this question indicates that an error was made when you were performing any one or all of the following steps:

   - Drawing the packet's path from node A to node D
   - Calculating the number of hops to node D from node A
   - Considering node E's maximum visits limit value

6. Node E's maximum visits value is too small to allow it to forward any packets from node A to node D. The number of hops the packet has already made is 3. Node E will only forward packets whose Forwarding Field is less than or equal to 2. Therefore, node E's routing algorithm will discard any packets received from node F that were originally transmitted from nodes A, C, or D.

   The Data Link and Routing protocols of nodes A, B, F, and E have been satisfied. There were no physical link errors encountered, but the message never reached node D via the cheapest path. In this case, since the routing protocols do not guarantee delivery, the problem is left up to the End Communication layer to correct.

   The End Communication layer will eventually time out and retransmit the message, and node E must again discard the message packet. After node A has tried to retransmit the message enough times to exceed a parameter value in the NSP protocol, the operator will be flaged with an error message. The error will indicate that the logical link between nodes A and D could not be connected.

   To correct this problem, the user at node A must inform Node A's System Manager of the fault. The System Manager must contact the Network Manager, who can do two things to correct this type of error:

   1. Request a change of the circuit costs at node B so that packets will not be passed to node E via node F for route-through service

   2. Request a change in the maximum visits limit for node E.

   Until one of these changes is made, the routing algorithm will always try to send message packets to node D via nodes B, F, and E, because it is the most cost-effective path between the source and destination nodes. The packets will also continue to be discarded by node E and produce error messages at node A indicating that the logical connection could not be established to node D.

MODULE TEST

Answer the following questions by circling the letter next to the best possible solution. After you have finished the test, check your answers against the Answer Sheet provided in your Tests and Answers booklet. Do not proceed to the next module until you have answered all of the following questions.

1. There are __ layer interfaces defined between the Routing layer and the other DNA layers.

   a. 2

   b. 3

   c. 4

   d. 5

2. The two generic types of network circuits supported by DNA Phase IV are:

   a. Broadcast, Multipoint

   b. Broadcast, Ethernet

   c. Nonbroadcast, Point-to-Point

   d. Nonbroadcast, Broadcast

3. The value set for the maximum visits limit should be _____ the maximum hops limit.

   a. Greater than or equal to

   b. Less than or equal to

   c. Greater than maximum hops limit by 1, but less than twice

   d. Less than maximum hops limit by 1, but no less than half

4. To transmit a message from node A to node D, the message must pass through nodes B and C. Using the following network parameters, what is the total circuit cost and distance between nodes A and D.

Path	Cost	Hops
A to B	2	1
B to C	2	1
C to D	5	1

a. Cost = 6, Length = 9

b. Cost = 9, Length = 3

c. Cost = 5, Length = 9

d. Cost = 10, Length = 3

5. The Routing layer's Loop Process Detector is part of the ____.

a. Routing Initialization sublayer

b. Routing component

c. Congestion Control component

d. Packet Lifetime Control component

6. What is a major difference between X.25 and DDCMP initialization operation?

a. DDCMP supports Hello and Test messages; X.25 does not.

b. X.25 supports Hello and Test messages; DDCMP does not.

c. DDCMP "Blocks" and "Deblocks" Routing layer messages; X.25 does not.

d. There are no differences between DDCMP and X.25 circuits.

7. What information is contained in the Routing Update message?

a. Circuit cost and hops

b. Node type

c. Routing layer version number

d. DNA Phase III or IV identification

8. What is the Routing layer destination process for Hello messages?

a. Decision process

b. Node Listener process

c. Forwarding process

d. Initialization process

9. Which of the following Routing messages is not used by Broadcast circuits?

a. Routing message

b. Ethernet Router Hello message

c. Ethernet Endnode Hello message

d. Hello and Test message

10. How many bits are used to form the Routing message Control Flag field?

a. 4

b. 8

c. 10

d. 16

END COMMUNICATION

INTRODUCTION

This module introduces, describes, and illustrates the operations performed, and message formats used, by the End Communication layer of the DNA.

OBJECTIVES

To use DECnet in technical support of applications environments, Software Services and Customer Personnel must be able to:

1. Define and illustrate the terms associated with the DNA's End Communication layer.
2. Identify the Message Formats used by the DNA's End Communication layer.
3. Describe the functional operations performed by the DNA's End Communication layer.

LEARNING ACTIVITIES

1. Study the information in this module.
2. Read Chapter 4, The End Communication Layer, in the DECnet DIGITAL Network Architecture (Phase IV) General Description.
3. Take the module test at the end of this module.
4. Correct the test using the answer sheet provided in the Test and Answers booklet. Review the material on any questions you may have missed before going on to the next module.

RESOURCES

1. DECnet DIGITAL Network Architecture (Phase IV) General Description
2. DNA NSP Functional Specification, Phase IV, Version 4.0

4.1 LAYER PURPOSE

The End Communication layer is responsible for the reliability and sequentiality of data exchanged between two processes regardless of their location within the network. The End Communication layer provides system-independent, process-to-process communication service to the DNA. It provides DNA with error-free data exchange and also guarantees the proper delivery of data. Data is transferred through the network via a logical connection between the two communicating processes. This logical connection is called a logical link. The End Communication layer creates, destroys, and manages the logical link between two communicating processes. A logical link permits two-way simultaneous transmission of normal data messages and independent two-way simultaneous transmission of interrupt messages. Figure 4-1 illustrates a logical link created by the End Communication layer over different physical links between two nonadjacent network nodes.

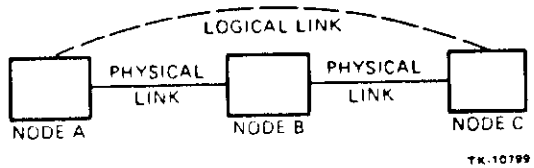


Figure 4-1 A Logical Link vs Two Physical Links

The End Communication layer uses a protocol called the Network Services Protocol (NSP). The NSP (detailed later in this module) performs all of the End Communication layer functions.

4.2 LAYER INTERFACES

There are three layer interfaces between the End Communication layer and the other DNA layers. Two of these interfaces are to its adjacent layers, the Routing and Session Control layers; the third interface is to the Network Management layer. Figure 4-2 shows the relationship between the End Communication layer and the other layers of DNA.

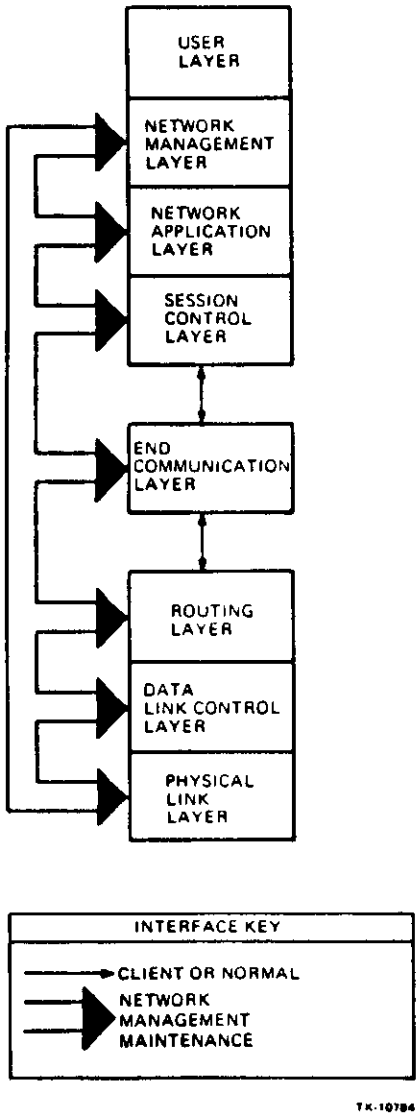


Figure 4-2 End Communication Layer Interfaces

4.3 END COMMUNICATION (NSP) FUNCTIONAL DESCRIPTION

Network Services Protocol (NSP) allows node-to-node and process-to-process communications in the same or different nodes within the DECnet network. It performs message handling and information flow control functions for network message traffic.

While DDCMP, X.25, and Ethernet are called the physical link protocols, NSP is called the logical link protocol. NSP allows the various operating systems on different network nodes to communicate with one another.

NSP performs the following functions:

- Creates and destroys logical links.
- Guarantees the sequential delivery of data and control messages to a specified destination using 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 the proper order upon reception.

NSP acts as a multiplexer; it receives messages from multiple processes located in higher DNA layers and passes them over a single path to the DNA Routing layer. The Routing layer in turn routes these messages to the appropriate Data link layer protocol module for transmission on the physical line to the messages' destination. Figure 4-3 shows the message multiplexing action performed by the NSP protocol. The NSP protocol is also the user's interface into the network as in Figure 4-4. Figure 4-5 shows that the NSP protocol is also the network's interface to the user.

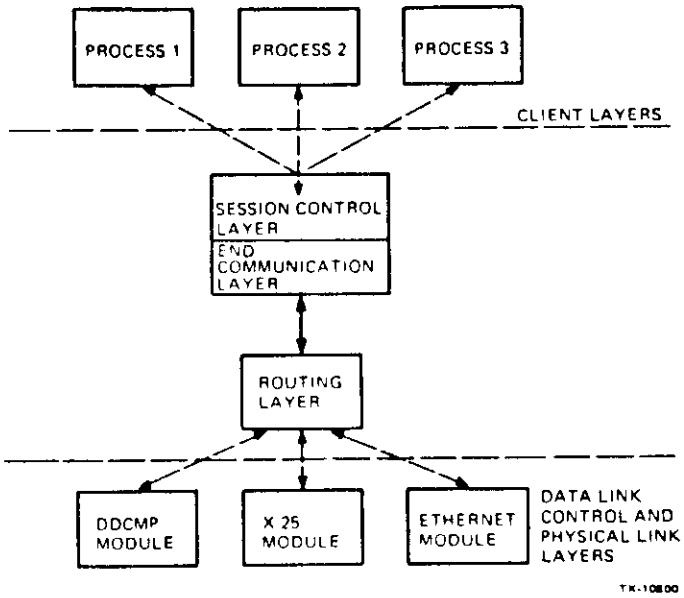


Figure 4-3 NSP Message Multiplexing

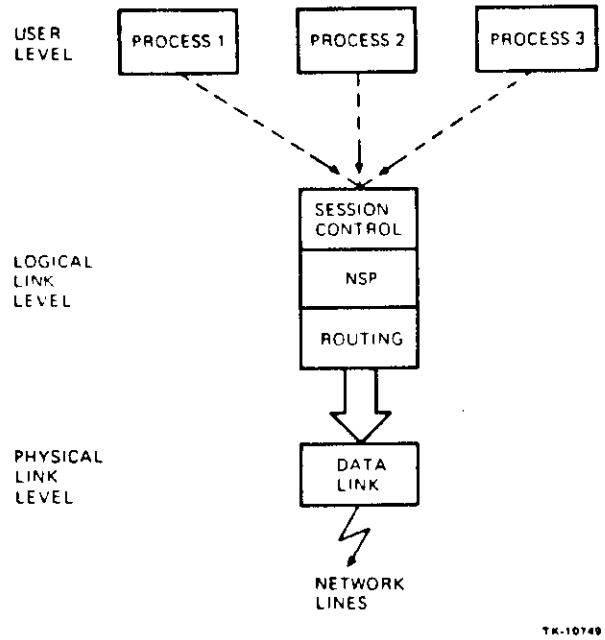


Figure 4-4 NSP - The User's Interface into the Network

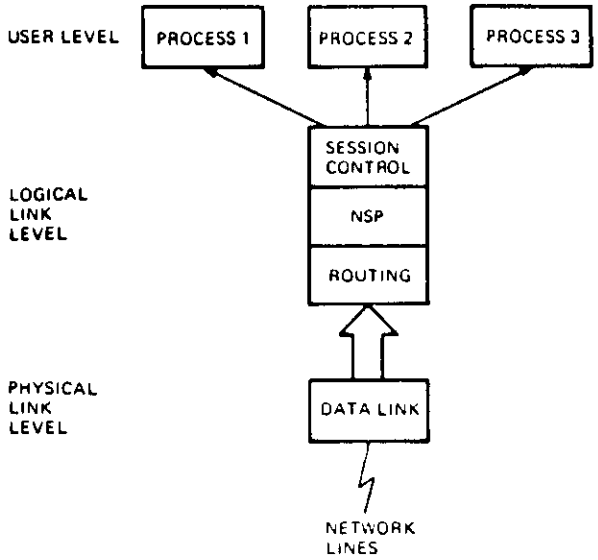


Figure 4-5 NSP - The Network's Interface to the User

Due to the interaction of the Session Control and End Communication layers, the user need not be concerned with the physical location of other nodes in the network or transmission of messages to those nodes. The user need only know the name or identity of the process and the name or identity of the node on which it resides. The Session Control and End Communication layers handle all message segmentation and sequencing; the Routing and Data Link layers handle all physical link control and error detection functions.

NSP depends on the lower DNA layers (Routing and Data Link layers) to provide error-free physical links and pass any message to the proper destination. The destination's Routing layer passes the received message up to its End Communication layer where its NSP protocol handles the logical link, end-to-end error checking and message resegmentation. Figure 4-6 shows the end-to-end, logical link functions performed by the NSP protocol.

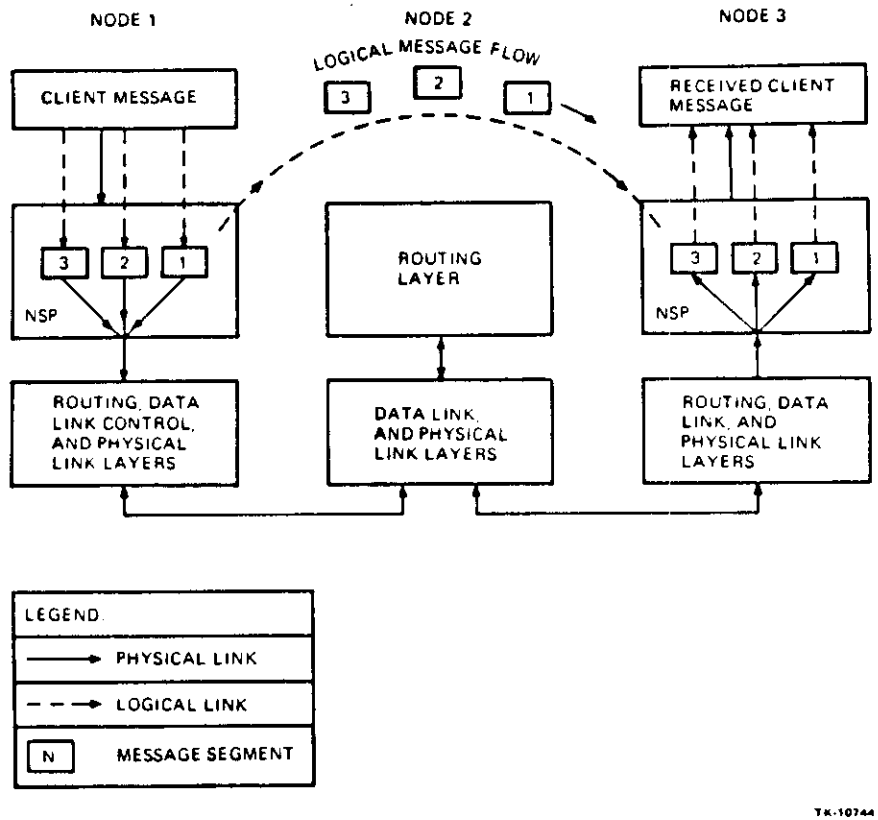


Figure 4-6 NSP End-To-End Logical Link Functions

NSP attempts to establish a communications path (logical link) between two user level processes. Once the logical link is established, NSP allows the processes to exchange messages (Refer to Figure 4-7).

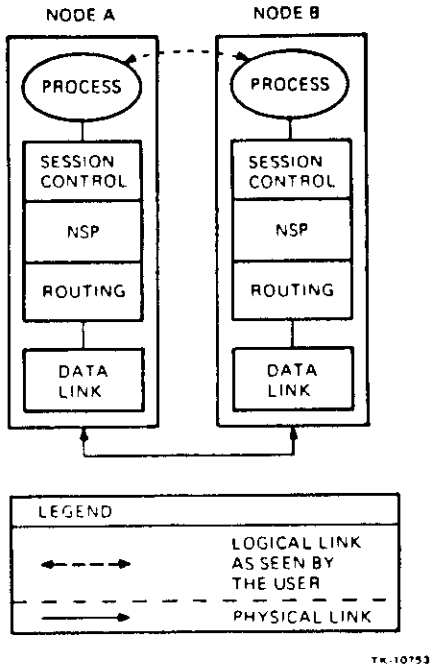


Figure 4-7 The Logical Link from the User's Perspective

The creation of a logical link is a cooperative arrangement between two processes, each indicating its desire to establish a mutual link. Once a logical link is established, data can be exchanged between the two processes. One process cannot send a message without the receiving process first requesting the transfer, and allocating buffer space to hold the resulting message. This ensures that the links do not become congested with unwanted traffic. It also helps to minimize system buffer occupancy.

There is an exception to this rule that allows a process to send a short, unsolicited interrupt message to another process to notify it of an error, or to wake it up if it is dormant. To do this, however, a logical link must already exist between the two processes.

Either process can direct its NSP to destroy the logical link. The NSP module at one end of the logical link (Channel) notifies the NSP module at the other end of the link (Channel) that it is disconnecting the logical link. Logical links created for other processes will still exist. Processes are not limited to only one logical link; they may have many logical links at one time.

The NSP modules at each node in the network can identify the logical links for which they are responsible. This enables the NSP to pass messages over the correct logical link and to destroy only the specified logical link whose destruction is requested. NSP modules do this by identifying each logical link with a unique number. This number is known as a logical link address. Figure 4-8 shows how an NSP module assigns a number to each logical link it has created.

NSP provides a single facility for two processes (called dialogue processes) to communicate with each other. This facility is called logical link service. The logical link service allows a dialogue process to establish a connection to another dialogue process. Once the logical link is established, the dialogue processes can exchange data via the logical link. When the data transfer is complete (no more data to be exchanged), either dialogue process may request that the logical link be disconnected. A logical link provides both a guaranteed delivery (a guarantee that the exchanged information went to a storage area that is accessible by the destination dialogue process) and a guaranteed sequentiality. (When the receiving user process reads the data from the receive data buffer, the data is guaranteed to be in the same sequence as it was when the sending user process queued it for transmission.)

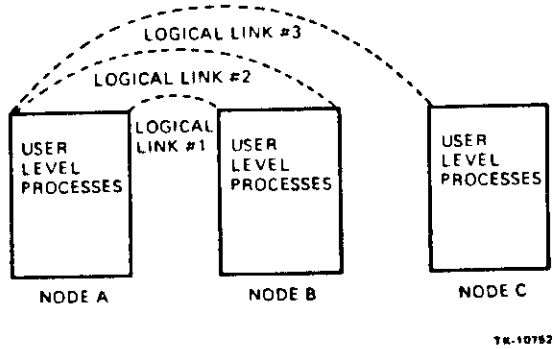


Figure 4-8 Logical Link Number Assignment

4.4 NSP OPERATIONS

The operations performed by the NSP Protocol consist of four functions:

1. Creation, maintenance, and destruction of logical links
2. Segmentation and reassembly of data
3. Error Control
4. Flow Control

The NSP functions are accomplished through the use of NSP messages. There are three basic types of NSP messages:

1. Data
2. Acknowledgement
3. Control

Table 4-1 summarizes the different NSP message types and gives a brief description of each. Each message type is detailed later in the NSP Message Formatting section of this module.

Table 4-1 NSP Messages and Descriptions

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 acknowledgement.
	Data Request	Carries data flow control information and optionally a Data Segment acknowledgement (also called Link Service message).
	Interrupt Request	Carries interrupt flow control information and optionally a Data Segment acknowledgement (Link Service message).
Acknowledgement	Data Acknowledgement	Acknowledges receipt of either a Connect Confirm message or one or more Data Segment messages, and optionally an Other Data message.
	Other Data Acknowledgement	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.

Table 4-1 NSP Messages and Descriptions (Cont)

pe	Message	Description
ntrol	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.

Due to network constraints, a dialogue message may not be sent in one piece. In that case, either NSP or the user (depending on the user's operating system) breaks the message into smaller units (dialogue segments), transmits them through the network, and reassembles the segments to form the original message for final delivery to the dialogue process. To efficiently break up dialogue (data) messages into segments, a single parameter is required by the transmitter for each logical link. This parameter is called Transmit Segment Size. It is important to note that the NSP segmentation is done only for data, not for Interrupt or Link Service (Control) messages. (These types are explained later in this module.)

When a dialogue process requests a connection to establish a datapath to another process, NSP attempts to create a logical link. To provide such services, the NSP in one node must be able to exchange messages with the NSP in another node.

For the logical link service, NSP guarantees that messages sent over a logical link are delivered to the destination process in the same order in which they were sent. If NSP is unable to do this (for example, if there is a broken physical link), the sending user is notified.

Data messages usually are not transmitted from the sending process until the receiving process issues a receive request that is transmitted to the sending node. A data message is not sent until NSP knows that there is buffer space available at the destination to receive the message. If a message arrives and there is no receive request outstanding, a fatal error occurs and NSP disconnects the logical link.

4.4.1 Creation, Maintenance, And Destruction Of Logical Links

The primary functions of NSP are to create, operate (maintain), and destroy logical links at the request of the Session Control layer (the next higher DNA 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 in which it was sent. An equivalent term for logical link is Virtual Circuit. The NSP mechanisms that set up a link, check for data errors, and manage the data flow, are all transparent to the user processes.

There can be several logical links at any given time, even to 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 4-9 shows some typical logical link connections. Notice the differences between the network physical links and the NSP created logical links. Also, notice the logical link from node N to node N: this logical link is connecting two user processes on the same node (node N).

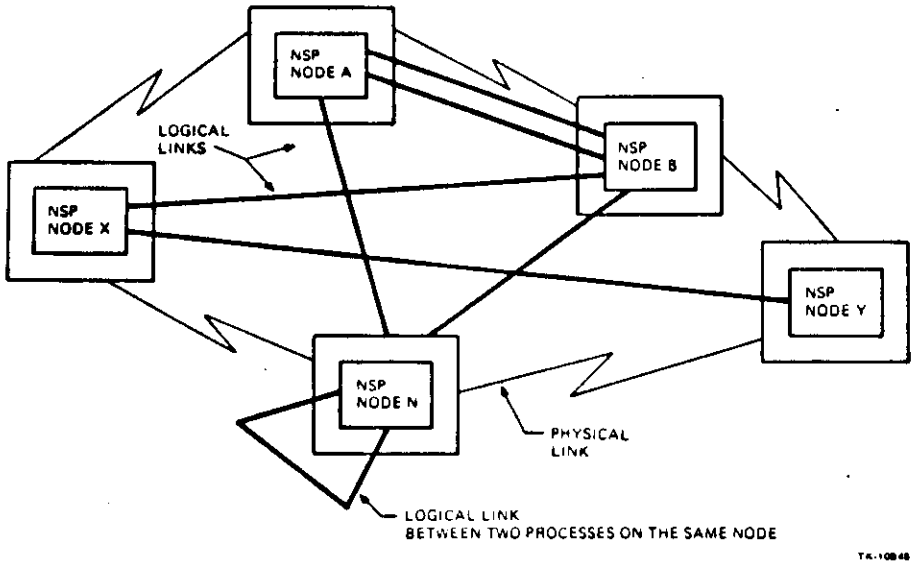


Figure 4-9 Typical Logical Link Connections

NSP establishes, maintains, and destroys logical links by exchanging control messages with other NSP modules on different nodes, or by exchanging the same control messages with itself to create, manage, and destroy logical links between two local user processes. (Refer to the logical link connections for node N in Figure 4-9.) Figure 4-10 shows a typical control message exchange used to create, manage, and destroy a logical link. In this figure, an NSP module first initiates a connection, then sends data, and finally, disconnects the link, based on commands received from the Session Control layer. The messages used to control the data flow (Flow Control messages) are not shown; they are detailed later in this module.

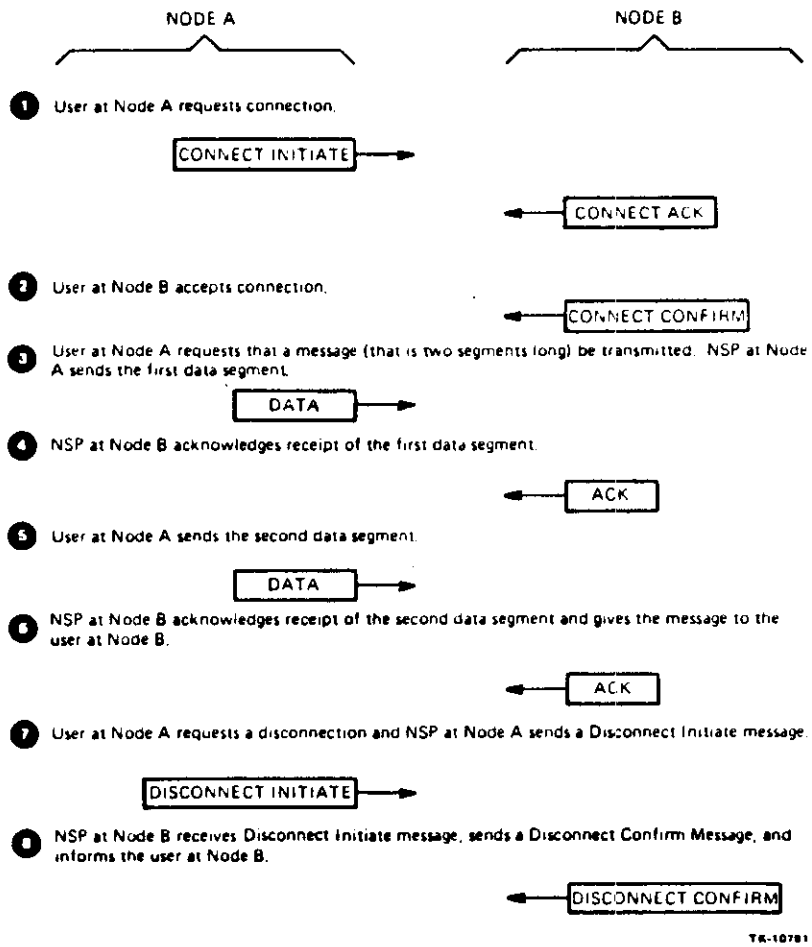


Figure 4-10 Typical Message Exchange Between Two NSP Modules

NSP operates in full-duplex on a logical link. The user process at either end of the link can disconnect at any time.

Logical links are made up of two subchannels, each carrying messages in both directions:

- Normal Data Subchannel - Carries Data Segments (data messages)
- Other Data Subchannel - Carries:

  Interrupt messages

  Data Request messages

  Interrupt Request messages

4.4.1.1 Segmentation and Reassembly of Data - Since the Routing layer limits the amount of data that can be sent in any single datagram, the NSP protocol must break up user messages passed down from the Session Control layer for transmission. NSP breaks up normal user data buffers into smaller message buffers called segments. These segments are numbered and appended using control information by the NSP. Once numbered and formatted correctly, they are passed down to the Routing layer for transmission. The receiving NSP module uses the numbers (called sequence numbers) and the control information for each data segment appended by the transmitting NSP, to reassemble the received segments into a correctly sequenced user message. The user message is placed into a receiving Session Control buffer according to each segment's sequence number. Thus, the received message is passed to the destination user process in the exact sequence in which it was generated by the source user process.

NSP only segments normal data messages; it does not segment any interrupt messages. Interrupt messages are limited in size and, therefore, always fit into a single Routing layer datagram. Figure 4-11 shows the operation of data message segmentation and reassembly by two logically linked NSP modules.

❶ Node A's Session Control queues a user data message for transmission.

❷ NSP first segments the message into smaller units (called datagram segments) and issues a sequence number to each as they are segmented.

❸ It then appends control information to each data segment prior to passing it down to the Routing layer for transmission.

❹ The Receiving NSP module strips off the control information and checks the received datagram's sequence number.

❺ The datagram, now called a data segment, is placed into the receive buffer allocated by the Session Control layer. The NSP module handles the placement of the data segments into the receive buffer.

❻ Once the entire data message is received, NSP turns over control of the buffer to the Session Control layer for further processing.

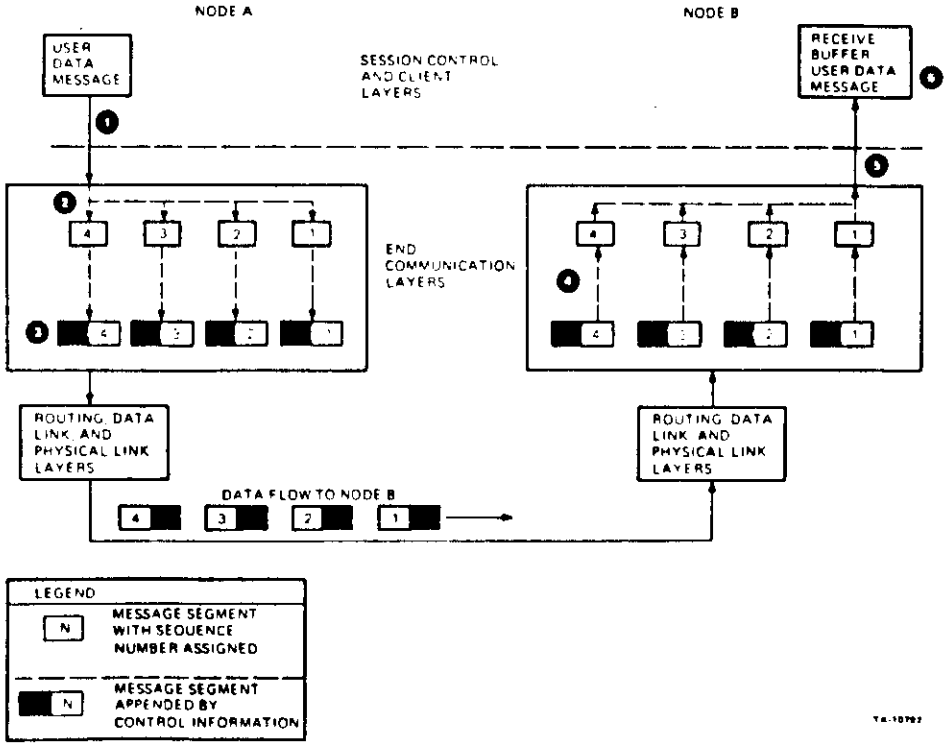


Figure 4-11 Data Message Segmentation and Reassembly

4.4.2 Error Control

The NSP modules at each end of a logical link must positively, or as an optional feature, negatively acknowledge received data. The decision to acknowledge on a segment or message basis is made at the time the logical link is connected. Messages received in error or out of sequence are discarded, then negatively acknowledged.

Acknowledgement or negative acknowledgement takes place in much the same way as did message acknowledgement under the control of DDCMP. Acknowledgments or negative acknowledgements can be direct or implied. Direct acknowledgement/negative acknowledgement is when each message received is either acknowledged or negatively acknowledged as it is processed. Implied acknowledgement/-negative acknowledgement is when the last message/segment received without error is acknowledged and all other previously received messages/segments from the source NSP are assumed to be without error.

If the transmitting NSP receives a negative acknowledgement or fails to receive a positive acknowledgement during a timeout interval, it retransmits the data.

Figure 4-12 shows data segmentation, reassembly, and acknowledgement operations performed by the NSP modules for two communicating user processes.

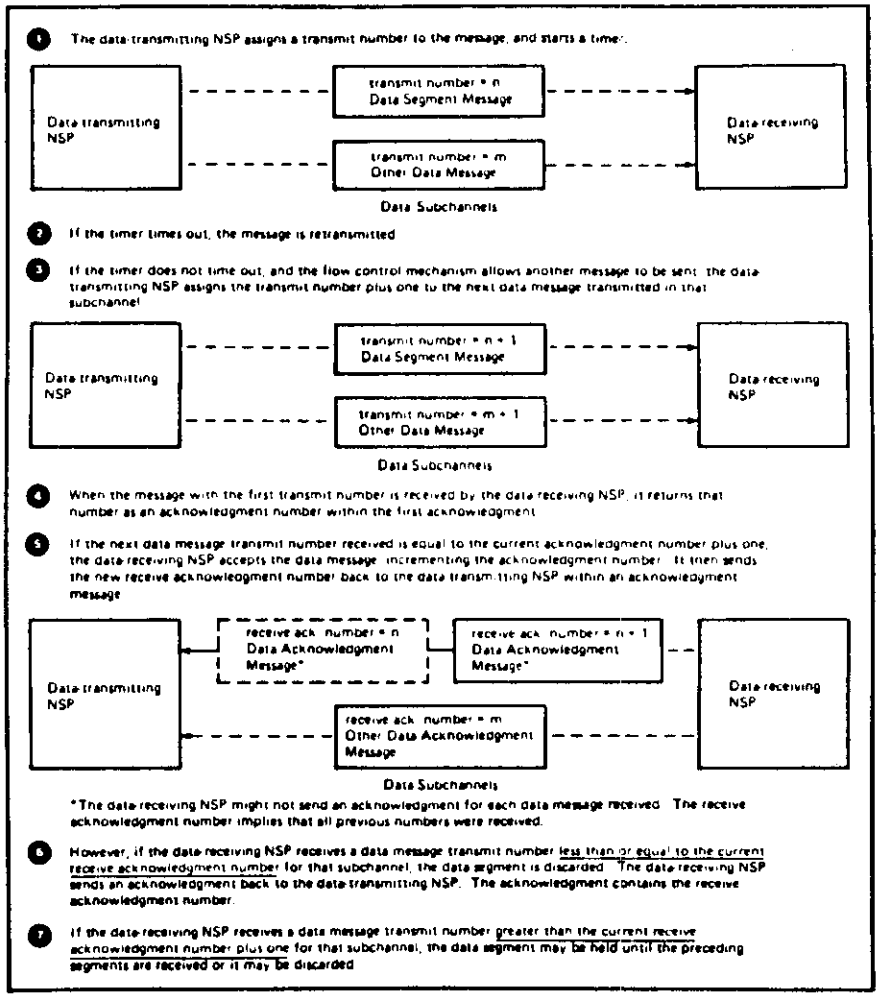


Figure 4-12 NSP Error Control via Data Acknowledgement

4.4.3 Flow Control

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

Deadlock occurs when a node's transmit and receive buffers are all full and cannot be returned. For example, normally, once a message is transmitted, it must be acknowledged by the receiving station before the buffer can be returned as complete. The acknowledgement of the transmitted buffer must be received by the source node which must have an empty receive buffer to receive the acknowledgement. If all of the transmit buffers are full, no more messages can be transmitted until some of those allocated transmit buffers are returned as complete. If all allocated receive buffers are full, the acknowledgement cannot be received.

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

1. None
2. Segment - The receiver sends a request count of the number of segments it can accept.
3. Message - The receiver sends a request count of the number of Session Control messages it can accept. (Note that message flow control is obsolete.)

Flow control uses a request count message to carry a request count from the receiver to the transmitter. The transmitter uses the request count to determine when data may be transmitted to the receiver. In addition, the receiver can always tell the transmitter via a request count message to either stop or start sending data under the normal request count conditions previously set up. The receiving NSP also controls interrupt data flow using an interrupt request count message. The interrupt request count is the same as the normal data request count, except it has only one mode of operation: interrupt flow control at the message level.

Flow control incorporates the implied and piggy-backed acknowledgement techniques used by the DDCMP Data Link layer to perform NSP data acknowledgements. The following message types may contain an acknowledgement for a Data Segment message:

- Data Request message
- Interrupt Request message
- Interrupt message
- Other Data Acknowledgement message

In addition, the Data Segment and Data-Acknowledgement messages may contain an acknowledgement for an Other Data message. Combining acknowledgements with either data or control messages reduces the number of NSP messages required per user message and, therefore, increases the DNA's effectiveness and performance, giving the user a more cost-effective network. Figure 4-13 shows the NSP's flow control operation.

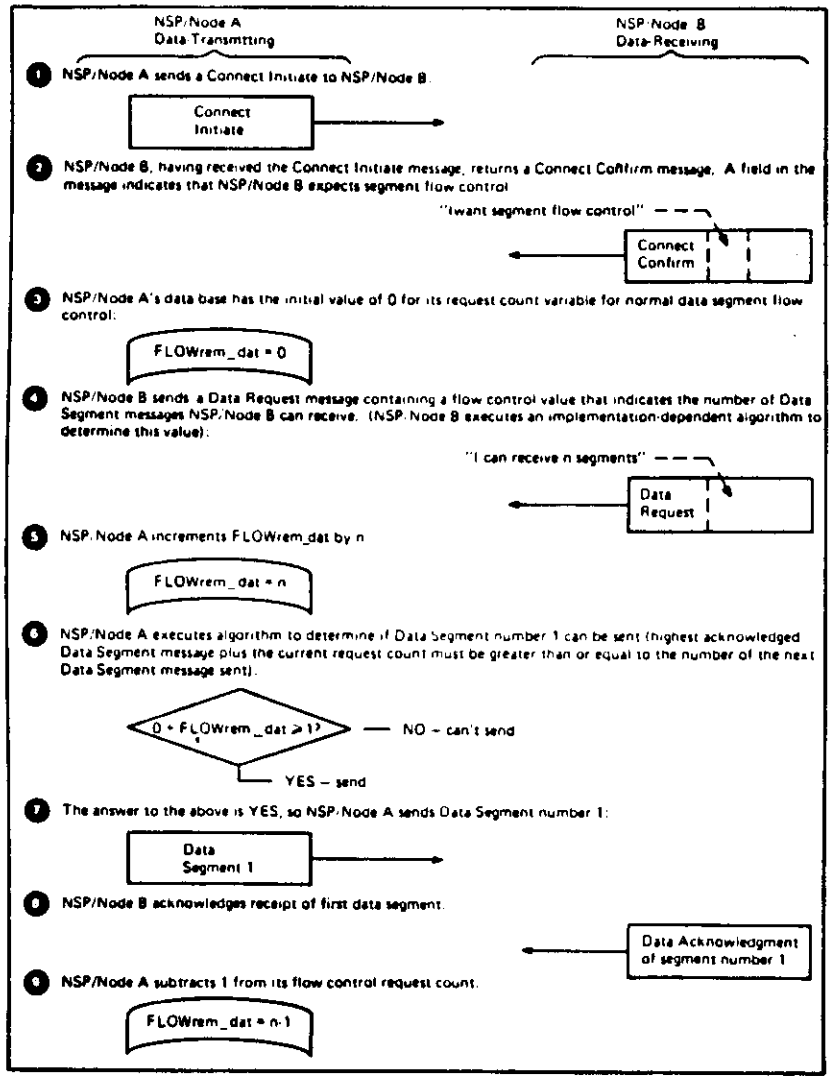


Figure 4-13 Segment Flow Control

4.5 NSP MESSAGE FORMATTING

Logical link service, flow control, and error control for DNA are provided by NSP messages and the functions they perform. There are three types of NSP messages:

1. Data Messages
2. Acknowledgement Messages
3. Control Messages

Table 4-2 summarizes the functions performed by each NSP message type.

Table 4-2 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.)
Data (also called Other Data)	Interrupt	Carries urgent data, originating from higher DNA layers.
	Data Request	Carries data flow control information (also called Link Service message).
	Interrupt Request	Carries interrupt flow control information (also called Link Service message).
Acknowledgement	Data Acknowledgement	Acknowledges receipt of either a Connect Confirm message or one or more Data Segment messages, and optionally an Other Data message.

Table 4-2 NSP Messages (Cont)

Type	Message	Description
	Other Data Acknowledgement	Acknowledges receipt of one or more Interrupt, Data Request or Interrupt Request messages.
	Connect Acknowledgement	Acknowledges receipt of a Connect Initiate message or Retransmitted Connect Initiate message.
Control	Connect Initiate	Carries a logical link connect request fromm a Session Control module.
	Connect Confirm	Carries a 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 (or Retransmitted Connect Initiate message) is received and there are no resources to establish a new port (also called Disconnect Confirm message).
	Disconnect Complete	Acknowledges receipt of a Disconnect Initiate message (also called Disconnect Confirm message).
	No Link	Sent when a message is received for a nonexisting link (also called Disconnect Confirm message).
	No Operation	Does nothing (included for compatibility with NSP V3.1).

All NSP messages have the same basic format; only their contents differ. Figure 4-14 shows the basic NSP message format.

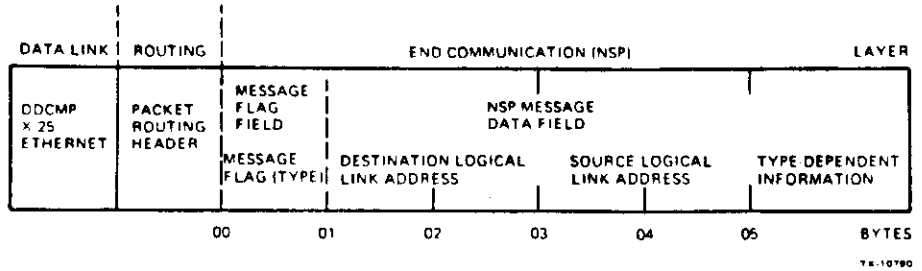


Figure 4-14 Basic NSP Message Format

The basic NSP message is divided into two fields:

1. Message Flag Field - Indicates the type of NSP message (Data, Acknowledgement, or Control). It is used to indicate the function the NSP message is to perform (e.g., create a logical link, transmit data, disconnect a logical link, acknowledge the receipt of data over the logical link, etc.).

2. Message Data Field - Contains either data or control information that is to be exchanged - Contains control information (Source and Destination Logical Link Addresses) and either client data or NSP control information. The exact contents of this field varies depending upon the type of NSP message being discussed.

For details on the specific subfields contained in each major field of NSP messages, refer to Section 8.0 in the NSP Functional Specification, Phase IV, Version 4.0.

MODULE TEST

Answer the following questions by circling the letter next to the best possible solution. After you have finished the test, check your answers against the Answer Sheet provided in your Tests and Answers booklet. Do not proceed to the next module until you have answered all of the following questions.

1. What protocol is used by the End Communication layer?

   a. DDCMP

   b. Routing Algorithm

   c. DAP

   d. NSP

2. Which of the following is not an NSP message type?

   a. Acknowledgement

   b. Control

   c. Data

   d. Interrupt

3. What DNA layer requests the End Communication layer to establish logical links?

   a. Network Management

   b. Session Control

   c. Network Application

   d. User

4. What message type is used by the End Communication layer to establish a logical link?

   a. Acknowledgement

   b. Control

   c. Data

   d. Interrupt

5. How many generic fields are in an End Communication layer message header?

   a. 2

   b. 4

   c. 6

   d. 8

6. The End Communication layer identifies a message's source and destination by _____.

   a. Circuit identification numbers

   b. Node names

   c. User Process names

   d. Logical Link addresses

7. Two communicating processes are called _____ processes.

   a. Logical

   b. Talking

   c. Dialogue

   d. User

8. What is a major difference between a Physical and a Logical Link?

   a. Logical Links exist only between adjacent nodes.

   b. Physical Links exsit only between adjacent nodes under DDCMP control.

   c. Logical Links are made over many Physical Links.

   d. Logical Links are made over only one Physical Link.

9. What End Communication layer message is used to destroy a Logical Link?

   a. Connect Initiate

   b. Data

   c. Disconnect Initiate

   d. Other Data

10. Which of the following is not a method of Flow Control used by the End Communication layer?

    a. Packet

    b. Message

    c. None

    d. Segment

SESSION CONTROL

INTRODUCTION

This module introduces, describes, and illustrates the operations performed, and message formats used, by the Session Control layer of the DNA.

OBJECTIVES

To use DECnet in technical support of applications environments, Software Services and Customer Personnel must be able to:

1. Define and illustrate the terms associated with the DNA's Session Control layer.
2. Identify the Message Formats used by the DNA's Session Control layer.
3. Describe the functional operations performed by the DNA's Session Control layer.

LEARNING ACTIVITIES

1. Study the information in this module.
2. Read Chapter 5, The Session Control Layer, in the DECnet DIGITAL Network Architecture (Phase IV) General Description.
3. Take the module test at the end of this module.
4. Correct the test using the answer sheet provided in the Test and Answers booklet. Review the material on any questions you may have missed before going on to the next module.

RESOURCES

1. DECnet DIGITAL Network Architecture (Phase IV) General Description
2. DNA Session Control Functional Specification, Phase IV, Version 1.0

5.1 LAYER PURPOSE

The Session Control layer resides immediately above the End Communication and directly below the Network Application layers of DNA. It provides system-dependent, process-to-process communication functions. The functions performed by the Session Control layer bridge the gap between the End Communication layer and the logical link functions required by the high-level user processes running under an operating system.

5.2 LAYER INTERFACES

There are four interfaces between the Session Control layer and its environment. Three of these interfaces are defined by DNA for interfacing the other layers of DNA. The fourth interface is to the User layer (the Operating System and User Processes). This later interface is completely system-dependent.

Figure 5-1 shows the interfaces between Session Control and its environment:

1. End Communication layer
2. Network Application layer
3. Network Management layer (Normal and Maintenance interfaces)
4. User layer (Operating System and User Process interfaces)

Unlike other DNA layers or modules covered thus far, the Session Control layer is not self-contained. It cannot be easily isolated from other non-DECnet modules. It is the point (or one of the points) at which DECnet is integrated with an operating system.

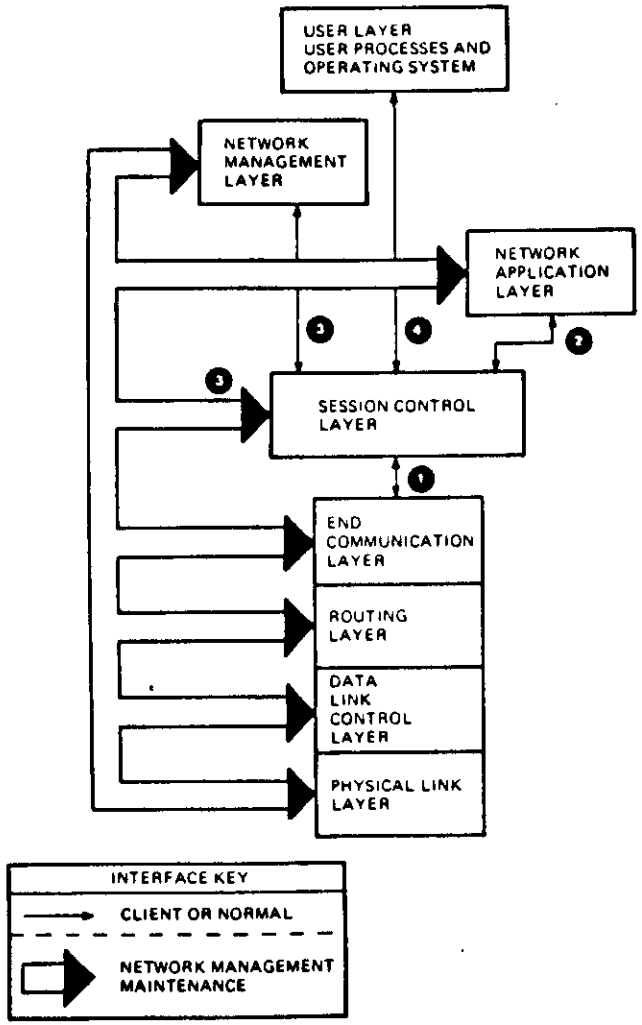


Figure 5-1 Session Control Layer Interfaces

5.3 SESSION CONTROL FUNCTIONAL DESCRIPTION

The Session Control and End Communication layers work together to make logical links available to the end users within a network. End users are modules that reside in the User, Network Application, or Network Management layers (Figure 5-1). A logical link is a virtual communication channel that temporarily connects two end users so that they can exchange data. From the perspective of the End Communication layer, each logical link connects two Session Control modules in the same or different nodes. The Session Control layer bridges the gap between the end users requiring logical link service and the End Communication layer which actually creates, maintains, and destroys the logical links.

End users communicate directly with Session Control to request logical link service. This communication, which is the end user interface, varies from one operating system to another. However, the functions this interface provides, regardless of the local operating system, include:

- Requesting a logical link to an end user
- Receiving a logical link request from an end user
- Accepting or rejecting a logical link request
- Sending and receiving data
- Terminating a logical link

These functions are similar to those that Session Control requests from the End Communication layer. In response to requests from end users, Session Control makes parallel requests to the End Communication layer. Unlike the end user interface, the End Communication interface is not system-dependent. It is defined and specified within the architecture of DNA.

An important concept to understand is that the End Communication layer provides the Session Control layer with logical link service. The Session Control layer, in turn, provides this service to the end user processes.

The Session Control layer, using the services of the End Communication layer, can create one or more logical links to other Session Control modules in the same or different nodes within the network. Session Control and End Communication communicate using individual data bases called Ports. A port is a space in memory (generally in a designated or shared pool) that contains control variables for managing the link. Each node within a network has a number of available ports, allocated and controlled by the End Communication layer. Even though the ports are maintained and controlled by the End Communication layer, they must be opened or closed by requests from the Session Control layer. Since each end of a logical link has its own port, the creation of a logical link can be thought of as the temporary association between two ports.

Session Control and End Communication use the Ports to manage the logical links they create on behalf of the end user. For this reason, both End Communication and Session Control refer to logical links in terms of their associated ports.

Session Control requests that the End Communication layer allocate or "open" a port when it receives an end user request for a logical link, or when it needs a port open to receive an incoming connect request. If sufficient resources are available, End Communication opens a port as requested. When Session Control closes a port, End Communication deallocates the port's resources.

At any given time, each end or port of a logical link is in a port state, which is determined by Session Control requests and End Communication messages pertaining to the link. The state of a port is represented by a variable in the port data base. Session Control maintains two data bases: the Port State data base and the Node Name data base (covered later in this section). Table 5-1 defines all possible port states. A port can only be in one state at any given time.

Table 5-1 Port States

State	Explanation
OPEN (O)	The local Session Control has issued an OPEN call which allocated the port.
CONNECT-RECEIVED (CR)	NSP has received a Connect Initiate message.
DISCONNECT-REJECT (DR)	The local Session Control has issued a REJECT call while the port was in the CONNECT-RECEIVED state.
DISCONNECT-REJECT-COMPLETE (DRC)	NSP has received a Disconnect Complete message while in the DISCONNECT-REJECT state. (The remote port is or has been in the REJECTED state.)
CONNECT-CONFIRM (CC)	The local Session Control has issued an ACCEPT call, while the port was in the CONNECT-RECEIVED state.
NO-RESOURCES (NR)	NSP has received a No Resources message while in the CONNECT-INITIATE state. (The remote NSP does not have an available port in the OPEN state.)
NO-COMMUNICATION (NC)	NSP has received its own Connect Initiate message while in the CONNECT-INITIATE state because Transport was unable to deliver the message.
CONNECT-DELIVERED (CD)	NSP has received a Connect Acknowledgement message while in the CONNECT-INITIATE state. (A destination port is or has been in the CONNECT-RECEIVED state.)
REJECTED (RJ)	NSP has received a Disconnect Initiate message while in the CONNECT-INITIATE or CONNECT-DELIVERED state. (The remote port is or has been in the DISCONNECT-REJECT state.)

Table 5-1 Port States (Cont)

State	Explanation
RUNNING (RUN)	NSP has either received a Connect Confirm message while in the CONNECT-INITIATE or CONNECT-DELIVERED state or received a Data, Data Request, Interrupt Request, Data Acknowledgement, or Other Data Acknowledgement message while in the CONNECT-CONFIRM state. The logical link may be used for sending and receiving data.
DISCONNECT-INITIATE (DI)	The local Session Control has issued a DISCONNECT-XMT or an ABORT-XMT call while in the RUNNING state.
DISCONNECT-COMPLETE (DIC)	NSP has received either a Disconnect Complete message or a Disconnect Initiate message while in the DISCONNECT-INITIATE state. (The remote port is, or has been in either the DISCONNECT-NOTIFICATION state or the DISCONNECT-INITIATE state.)
DISCONNECT-NOTIFICATION (DN)	NSP has received a Disconnect Initiate message while in the RUNNING state. (The remote port is or has been in the DISCONNECT-INITIATE state.)
CLOSED (CL)	The local Session Control has issued a CLOSE call while the local port was in the DRC, DN, DIC, NC, NR, or CI state. This is not really a state of the port, but is used for descriptive purposes to indicate that the port is not there.
CLOSED NOTIFICATION (CN)	NSP has received a No Link message while in the DISCONNECT-INITIATE or DISCONNECT-REJECT state. (The remote NSP closed the remote port.)

Since logical links are an association between two ports, and ports have specific states, logical links also have states. The state of a logical link is determined by the combination of possible port states at each end of the link. The product of the port states is called the logical link state.

When one Session Control module attempts to connect with a second Session Control module, End Communication places the requesting port in the Connect-Initiate (CI) state. End Communication then attempts to associate the source local port with a destination port that is in the Open (O) state. If the association is successful, a logical link is formed and its initial state is CI/O, the product of the two communicating ports' state. Figures 5-2 and 5-3 show the normal logical link, port states and state transitions. Ports and logical links can be in only one state at any given time.

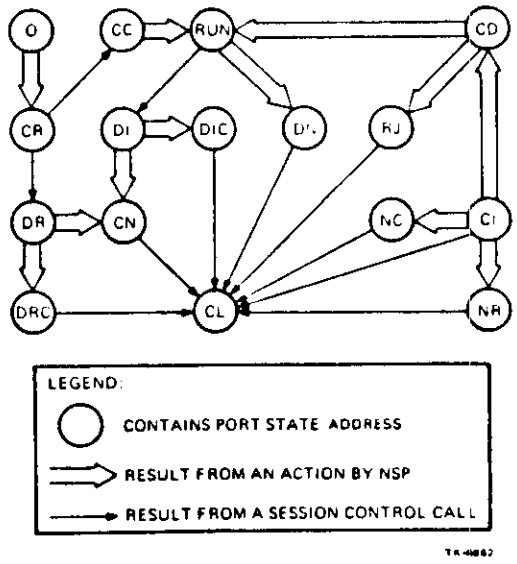


Figure 5-2 Port States

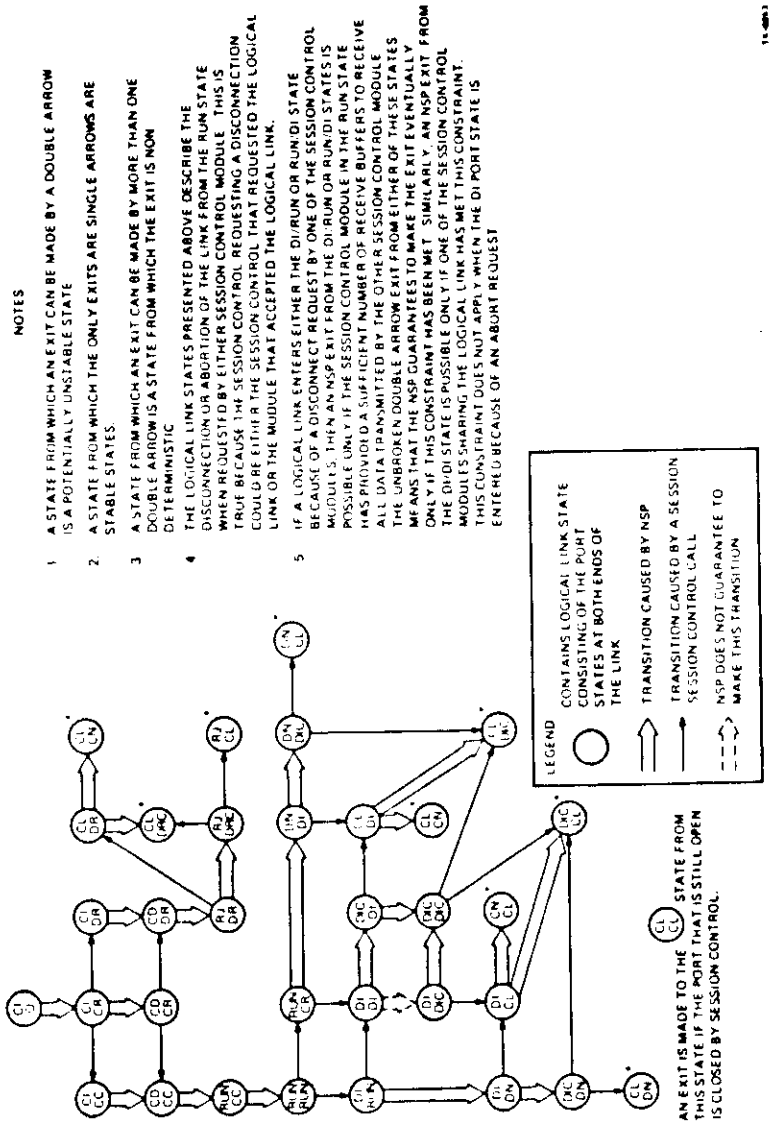


Figure 5-3 Logical Link States

Session Control performs the logical functions associated with the management of logical link communications; the End Communication layer performs the mechanical functions associated with the logical link communications. The logical functions performed by the Session Control layer to facilitate logical link communication 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 a channel number. (The channel number is used only for loopback testing, a Network management fault isolation tool provided to the end user by the Network Management layer of DNA.) The node name mapping table enables the Session Control layer 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 layer 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.

- 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.

Figure 5-4 shows a model of Session Control operating within a network node. The Session Control layer requires two data bases:

(1) Node Name Mapping data base

(2) State data base

Other data bases or components required by a Session Control module are operating system-dependent. These specific data bases are not within the scope of this course and, therefore, not covered. They are however, covered by other higher-level system-specific Data Communications courses.

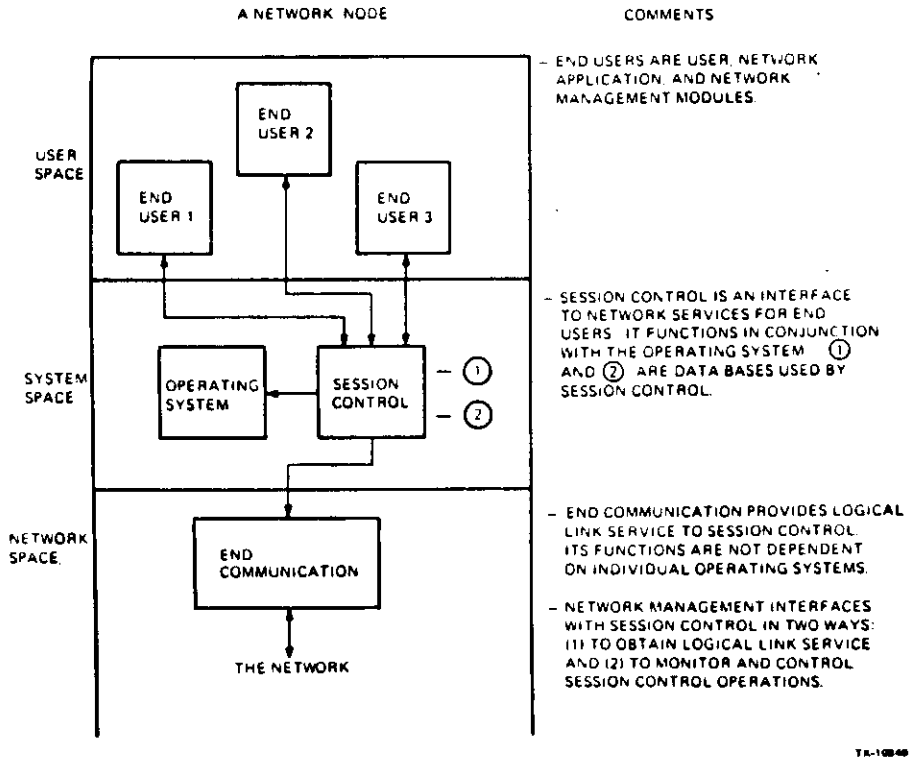


Figure 5-4 The Session Control Model

5.4 SESSION CONTROL OPERATIONS

Session Control extends the functions offered by the End Communication layer (NSP) into the end user layers. These functions are:

- Requesting a connection - Requests logical links on behalf of end user processes. Connections are requests to establish a logical link between end users (the source and destination).
- Receiving a connection - Handles the connection requests passed up from the End Communication layer to an end user process.
- Sending and receiving data - Handles data passed between the End User, Session Control, and End Communication layers.
- Disconnecting and aborting logical links - Passes disconnect and abort requests directly between the end user processes and the End Communication layer.
- Optionally monitoring logical links - Session Control can optionally monitor logical link functions in a system-dependent manner.

5.4.1 Requesting A Connection

When Session Control receives a connection request from an end user process (the source) it performs the following:

1. Identifies the destination node address or channel number Session Control may use either a node name mapping table or an optional "alias" node name mapping table to identify the destination node address or channel number. The "alias" node name mapping table is basically an extension of the mandatory node name mapping table. The existence, maintenance, and use of either a node name mapping table or the "alias" node name mapping table is system-dependent.

2. Formats connect data to be passed to the End Communication layer -- The operations involved with the formatting of the connect data message is defined by the Session Control Functional Specification. However, how Session Control obtains the destination and source end user names, access control information, and end user connect data that make up the message, is system-dependent.

3. Issues a connect request to the End Communication layer -- Session Control issues the connect request to the End Communication layer. If sufficient resources are available, it opens a port for the connection requested by the Session Control layer.

   The request is successful if the End Communication layer places the port in the Running state. When the port is placed in the Running state, the Session Control layer informs the end user (the source) that the connection request has succeeded. Session Control must now make the accept data, if any, available to the source end user.

   The request is unsuccessful, if the End Communication layer places the port in the No-Resources, No-Communication, or Rejected states. If unsuccessful the Session Control layer returns information to the end user (the source) indicating that the connection request has failed. The extent of the information concerning the failure is system-dependent.

4. Optionally starts an outgoing connection timer -- The outgoing connection timer is optional; timer processing is system-dependent. If the timer is used and the End Communication layer does not place the port associated with the request in the No-Resources, No-Communication, Rejected, or Running state before the timer expires, Session Control returns a timeout rejection to the source end user and closes the port. The timer value used can be either a Network Management default value or an "error tolerance" argument value included in the connect request. If used, an argument value will override the default value set by the Network Management layer; this value is system-dependent.

5.4.2 Receiving A Connect Request

Session Control maintains one or more ports in the Open state to detect incoming connect requests. The End Communication layer notifies the Session Control layer of an incoming request by changing the state of the port from Open to Connect-Received. When the Session Control layer detects the port's new state, it performs the following:

1. Obtains data for the incoming connect request - Session Control obtains the destination end user name, source end user name, access control information, and end user connect data for the incoming connect request. This data is obtained by parsing the connect data received from the End Communication layer.

2. Validates access control information - Access control information is a system-dependent function and may be processed in any number of ways. For example, one Session Control module may log the logical link onto the local system. Another Session Control module may perform no validation of its own and pass the information directly to the end user process for validation.

3. Identifies, creates, or activates the destination end user Session Control either maps the received destination end user name to an existing end user, or creates or activates a destination end user to receive the connect request. If Session Control cannot identify, create, or activate a destination end user, it issues a Reject to the End Communication layer which causes the logical link to be disconnected.

4. Maps the source node's address or channel number to a node name, if there is one - After identifying, creating, or activating a destination end user, Session Control uses a node name mapping table to map the source node's address or channel number to a node name. If session Control cannot find an entry that corresponds to the source node's address or channel number, it identifies the source node as "unknown".

5. Delivers the incoming connect request to the end user process - If Session Control has identified, created, or activated a destination end user, it delivers the incoming connect request to the end user in a system-dependent way. Session Control also delivers information that identifies the source node who sent the connect request. The form that this information takes is system-dependent, but it may be one of the following:

   - Source Node's Name
   - The Source Node Identification, "UNKNOWN"
   - Source Node's Channel Number

   If the destination end user accepts the connection, Session Control issues an Accept to the End Communication layer, which then passes the accept in the form of a connect confirm message to the source node's end user. If the destination end user rejects the connection, Session Control issues a Reject to the End Communication layer, which then passes the reject in the form of a disconnect initiate message to the source node's end user.

6. Optionally starts an incoming connection timer -- Session Control can optionally start an incoming connection timer. The timer starts when Session Control makes the connection request available to the destination end user. If the end user does not accept or reject the connection request before the timer expires, Session Control rejects the connection request to the End Communication layer, and causes it to send a disconnect initiate message back to the source node's end user. The incoming connection timer processing is a system-dependent function.

Figure 5-5 shows the message exchange between two Session Control and End Communication layer pairs while attempting to establish a logical link.

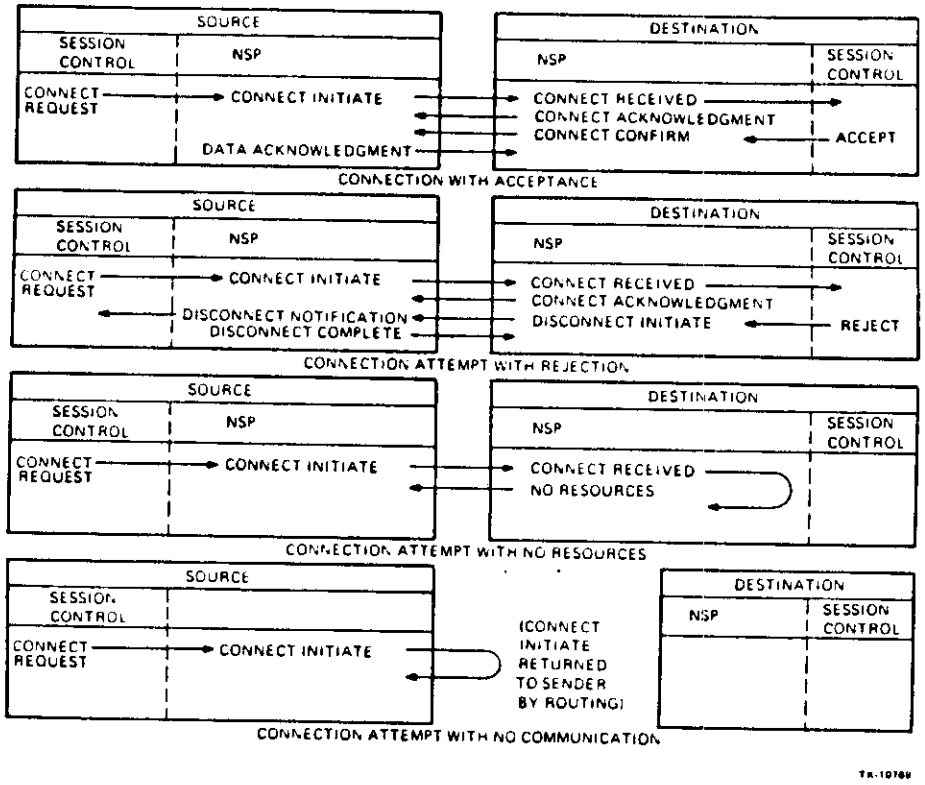


Figure 5-5 Establishing a Logical Link

5.4.3 Sending And Receiving Data

Sending and receiving data is a system-dependent function. Session Control passes end user requests for sending and receiving data directly to the End Communication layer. The Session Control layer handles all initial logical functions necessary for data exchange, such as: Data buffering scheme, Data integrity guarantees, and Data transfer interface type. The End Communication layer handles all mechanical functions, such as: Segmentation and reassembly of data, Error control, and Flow control for transmitting and receiving data.

Since the interface between end users and the Session Control layer is system-dependent, Session Control can handle user requests to transmit and receive data in several ways. For more information concerning the data transfer interface and data buffering schemes, refer to Section 5.3 in the DNA Session Control Functional Specification.

5.4.4 Disconnecting And Aborting A Logical Link

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 destination end user process. Figure 5-6 shows how the Session Control layer passes the disconnect and abort requests directly to the End Communication layer for transmission.

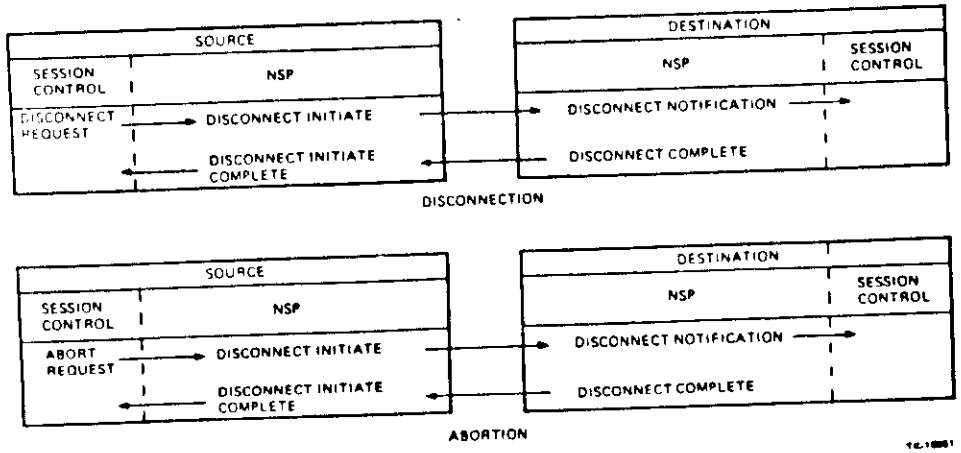


Figure 5-6 Disconnecting and Aborting a Logical Link

5.4.5 Monitoring A Logical Link

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

- Detecting probable network disconnections 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.

Since these functions are optional and system-dependent, they are beyond the scope of this course. However, they are covered in other higher level system-specific Data Communications courses. You may also refer to Sections 5.5 and 6.0 in the DNA Session Control Functional Specification for more information concerning the methods and possibilities available for monitoring a logical link.

5.5 SESSION CONTROL MESSAGE FORMATTING

Session Control defines only two message types used to create a logical link:

- Connect Data Message - Used to establish or connect the logical link.
- Reject/Disconnect Data Message - Used to destroy the logical link.

Figure 5-7 shows the format of both message types and lists the names of each field contained within each message type. 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 Bytes

Reject/Disconnect Data Message Format

REASON	DATACTL
2	n 2+n Bytes

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)

TK-10783

Figure 5-7 Session Control Message Formats

MODULE TEST

Answer the following questions by circling the letter next to the best possible solution. After you have finished the test, check your answers against the Answer Sheet provided in your Tests and Answers booklet. Do not proceed to the next module until you have answered all of the following questions.

1. A Logical Link is an association between two _____.

   a. Operating systems

   b. End Communication layers

   c. Ports

   d. Nonadjacent Routing layers

2. What information is used to validate an incoming connect request at the Session Control layer level?

   a. Node Address information

   b. Access Control information

   c. User Process identification information

   d. Connect Initiate message information

3. What message is issued to the End Communication layer from the Session Control layer when an end user accepts a requested connection?

   a. Accept

   b. Reject

   c. Go-Ahead

   d. Initiate Confirm

4. What is the Connect Data Message used for?

   a. To connect the Logical Link

   b. To disconnect the Logical Link

   c. To reject the Logical Link

   d. To identify the Logical Link address

5. Which of the following IS NOT a function performed by the Session Control layer?

   a. Requesting a connection

   b. Sending and receiving data

   c. Monitoring the Logical Link

   d. Establishing the Logical Link

6. How many interfaces are defined by DNA for the Session Control layer?

   a. 2

   b. 4

   c. 6

   d. 8

7. The major function performed by the Session Control layer is:

   a. To initialize the Physical link.

   b. To bridge the gap between the physical and logical functions required for data communications.

   c. To buffer End Communication layer functions from Routing layer functions.

   d. To perform network testing when commanded to do so by the Network Management layer.

8. What does CI/O mean in terms of ports and port states?

   a. Session Control has cleared the Logical Link to perform input/output functions.

   b. An association between two network nodes has been formed and they are ready to print a user data buffer on an output device.

   c. An association bewteen two ports has been established and they are currently both in the run state ready to perform input/output functions.

   d. A Logical Link has been formed and it is currently in the Connect Initiate/Open state.

9. How many information fields are contained in the Session Control Connect data message?

   a. 3

   b. 5

   c. 7

   d. 9

10. How many bytes are used in the Reject/Disconnect message Reason field?

    a. 1

    b. 2

    c. 8

    d. 16

NETWORK APPLICATION

INTRODUCTION

This module introduces, describes, and illustrates the operations and message formats used by the Network Application layer of the DNA.

OBJECTIVES

To use DECnet in technical support of applications environments, Software Services and Customer Personnel must be able to:

1. Define and illustrate the terms associated with the DNA's Network Application layer.
2. Identify the message formats used by the DNA's Network Application layer.
3. Describe the functional operations performed by the DNA's Network Application layer.

LEARNING ACTIVITIES

1. Study the information in this module.
2. Read Chapter 6, The Network Application Layer, in the DECnet DIGITAL Network Architecture (Phase IV) General Description.
3. Take the module test at the end of this module.
4. Correct the test using the answer sheet provided in the Tests and Answers booklet. Review the material on any questions you may have missed before going on to the next module.

RESOURCES

1. DECnet DIGITAL Network Architecture (Phase IV) General Description
2. DNA Data Access Protocol (DAP) Functional Specification, Phase IV, Version 7.0
3. DNA X.25 Gateway Access Functional Specification, Phase IV
4. DNA SNA Gateway Access Functional Specification, Phase IV
5. DNA Network Virtual Terminal Functional Specification, Phase IV

6.1 LAYER PURPOSE

The Network Application layer provides generic network applications to end users in the DNA User Layer. Any DECnet user is free to design, implement, and install protocols that meet the functional requirements of this layer.

6.2 LAYER INTERFACES

There are four interfaces defined for the Network Application layer:

1. Network Management Interfaces - One for Normal and one for Maintenance functions
2. Session Control Interface - Normal or Client interface between layers
3. Data Link Layer Interface - Used on Gateway Systems to IBM SNA and/or PPSN X.25 networks
4. User Layer Interface - Task- or application-dependent. There can be many user tasks or processes interfacing the Network Application layer at any given time.

Figure 6-1 shows the interfaces between the Network Application layer and the other layers defined by DNA.

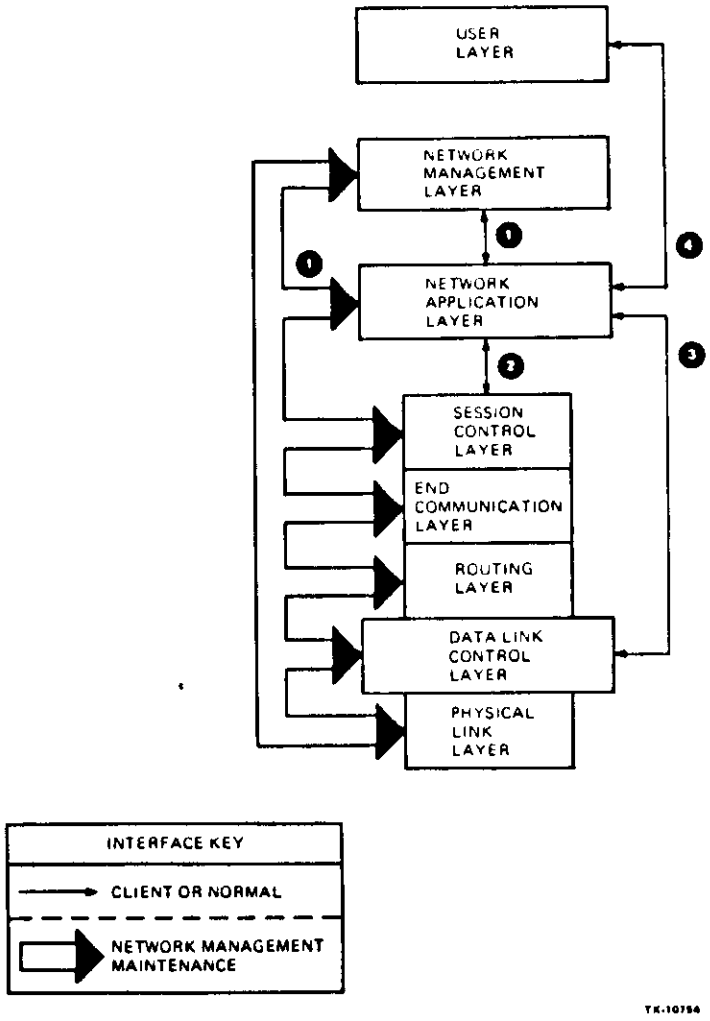


Figure 6-1 Network Application Layer Interfaces

6.3 NETWORK APPLICATION LAYER FUNCTIONS AND OPERATIONS

The Network Application layer contains a number of independent protocol modules. These modules access data or provide communication services to end users. Currently, there are five DNA Phase IV DIGITAL-supplied protocol modules in the Network Application layer, but their number is not restricted. Users are not restricted to DIGITAL-supplied protocols. The Network Application layer can contain and use as many protocol modules as the user wishes. The only requirement is that the user-written protocol modules meet the functional requirements of the layer.

The five DIGITAL-supplied protocol modules are:

1. Data Access Protocol (DAP) - DAP permits remote file access and file transfer independent of the operating system's I/O structure.

2. Network Virtual Terminal Protocols - These protocols permit terminals connected locally to a host DECnet system or to a Terminal Concentrator system to access remote host DECnet systems. This remote access allows the user to issue interactive commands and run application programs on the remote host DECnet system as a local system user.

3. 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 (PPSN). The user's process need not reside in the DECnet system directly connected to the X.25 PPSN (the DTE Node for the X.25 network).

4. SNA Gateway Access Protocol - This protocol permits user-written modules and a number of DIGITAL-supplied modules to communicate with IBM host application programs and application subsystems in a Systems Network Architecture (SNA) network. The user's process need not reside in the same DECnet system that is directly connected to the SNA network.

5. Loopback Mirror - This protocol consists of three messages and is used to interface between the Network Management Loopback Access Routines and the Network Application layer's loopback mirror. These protocol modules are used to test logical links and are covered in the Network Management Layer module of this course.

Each of the following sections covers the functions and operations of the different Network Application layer modules.

6.3.1 Data Access Protocol (DAP)

The Data Access Protocol is an application level protocol. Its primary purpose is to permit the user to access remote files within the DECnet environment. This file access is accomplished regardless of the remote operating system's I/O structure. DAP makes the DECnet network's resources available to all valid, authorized end users.

6.3.1.1 DAP Functional Description - DAP is a set of messages and rules governing the exchange of messages between two cooperating processes that provide DECnet remote file access. Table 6-1 lists and briefly describes the most commonly used DIGITAL-supplied DECnet remote file access facilities. These facilities reside in the User layer and interface directly between the DAP protocol and the end user.

DAP is designed to minimize the amount of protocol overhead required for remote file access within a computer network. For example, DAP uses default values for specified fields wherever possible. In addition, a file transfer eliminates the need for DAP control messages once the file data flow begins. Finally, relatively small file records can be blocked together and sent as one large message under the control of the DAP module.

Within DECnet, DIGITAL Operating Systems can use DAP to provide the following functions and features:

- Support heterogeneous file systems
- Retrieve a file from an input device (a disk or tape file, card reader, terminal, etc.)
- Store a file on an output device (a magnetic disk or tape, line printer, terminal, etc.)
- Transfer files between DECnet nodes
- Support deletion and renaming of remote files
- List directories of remote files
- Recover from transient errors and reporting fatal errors to the user
- Allow multiple data streams to be sent over a logical link
- Allow remote command file submission and execution

- Permit sequential, random, and indexed (ISAM) access of records
- Support sequential, relative, and indexed file organizations
- Support using wildcards as file specifications for sequential file retrieval, file deletion, file renaming, and command file execution
- Permit the use of an optional file checksum value to ensure file integrity

Table 6-1 DECnet Remote File Access Facilities

Facility Name	Description
File Access Listener (FAL)	FAL receives user I/O requests at the remote node and acts on the user's behalf. This is a remote DAP-speaking server process.
Network File Transfer (NFT)	NFT operates at the user level. It interfaces to a DAP-speaking accessing process to provide DAP functions, and 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. RMS can transmit and receive DAP messages over logical links. These DAP messages are sent to a remote FAL to complete the request. To the user, remote file access is handled as local file access, except that a remote node name, and possibly access control information, is necessary for remote file access.

Table 6-1 DECnet Remote File Access Facilities (Cont)

Facility Name	Description
Network File 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. VAX/VMS does not require NFT to access remote files.
Network Management Modules	Network Management modules use DAP services to obtain remote files to down-line load other remote nodes and transfer up-line dumps for storage.

Figure 6-2 shows a file transfer between two DECnet nodes using the DAP protocol and DECnet remote file access facilities.

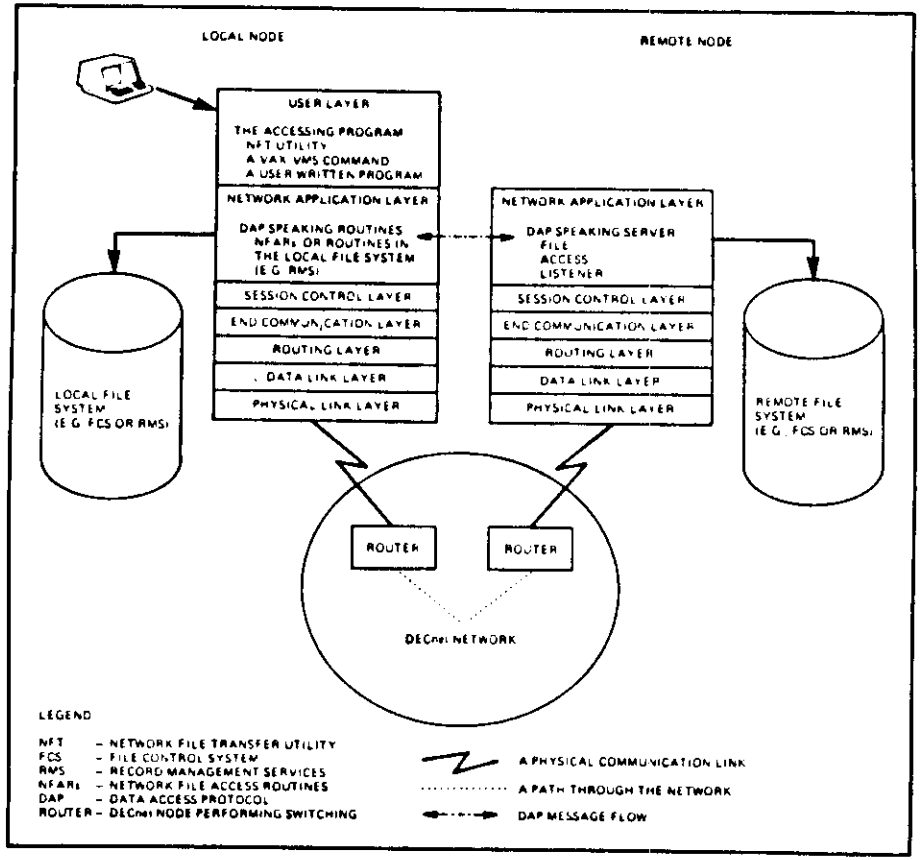


Figure 6-2 Node-To-Node File Transfer Using DAP

6.3.1.2 DAP Operations - DAP allows two cooperating processes to exchange data files: the local user's process, and the server process that acts on the user's behalf at the remote node.

The user's I/O commands for accessing the remote file are mapped to equivalent DAP messages and transmitted via a logical link to a server at the remote node. The server interprets the DAP commands and actually performs the file I/O functions for the user.

Figure 6-3 shows the DAP message exchange between a user and a server process when a remote file is accessed by the user. The following is a list of messages in the order they are exchanged during a remote file access. The user is accessing or requesting to transfer a sequential file from a remote DECnet node.

1. Configuration messages - Provide information about the operating and file systems resident at the two communicating nodes.
2. Attributes messages - Supply information about the file itself.
3. Access Request messages - Used to open the requested file at the remote node.
4. Data Stream Set Up and Get messages - A series of control and acknowledgement messages to set up the data flow stream for both sequential and random access file transfers. A control message is used to initiate the data flow and another control message is used to actually get the data.
5. Access Complete messages - Used to terminate the data stream after the completion of the file transfer.

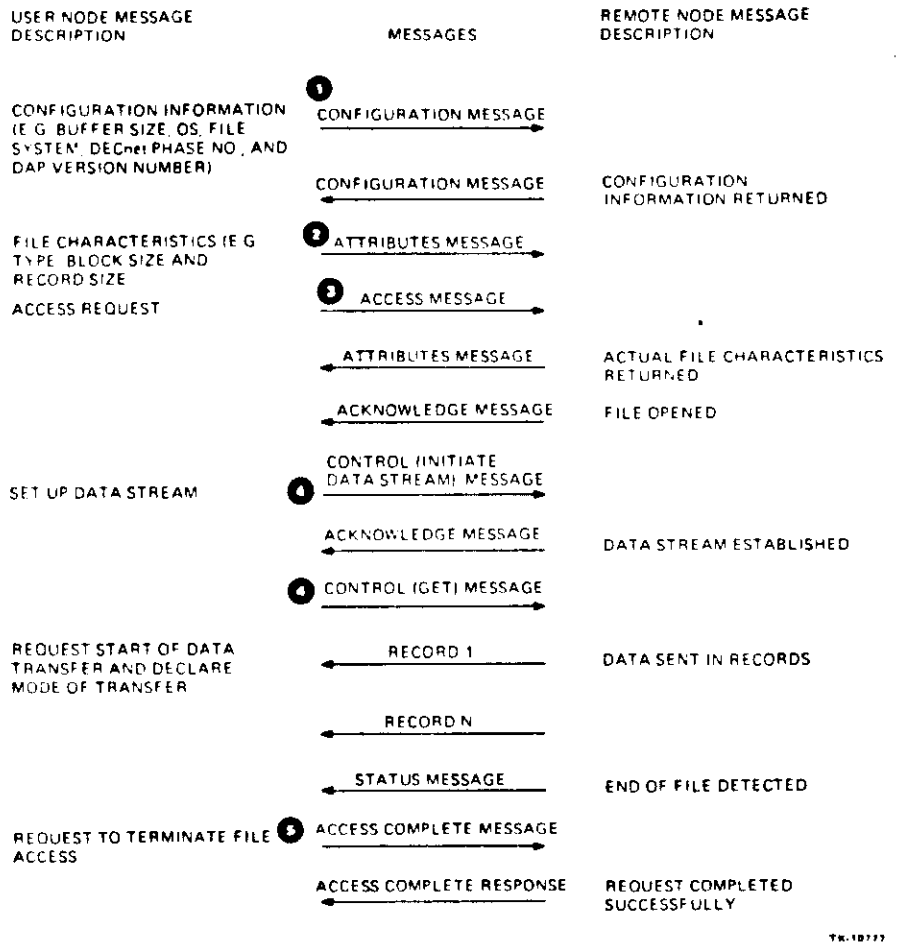


Figure 6-3 DAP Message Exchanges for Sequential File Retrieval

Table 6-2 lists the different types of messages exchanged by DAP-speaking processes to access remote files.

Table 6-2 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 stream.
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.

Table 6-2 DAP Messages (Cont)

Message	Function
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.

6.3.2 Network Virtual Terminal Protocols (NVT)

The Network Virtual Terminal Protocols are application level protocols. Their primary purpose is to permit the user to access remote nodes within the DECnet environment as a local command terminal. The remote or Host Node access is accomplished regardless of the node's operating system. NVT makes all of the DECnet nodes in a network available to end users who are valid, authorized users.

6.3.2.1 NVT Functional Description - NVT service provides the following functions and features:

- Distributes terminal handling functions between two operating systems.
- Supports heterogeneous host systems. Allows different operating systems to cooperate via common protocols, and a host operating system to manage a terminal independently of the server node.
- Allows a server node to connect to a specified host node, or a host node to connect to a specified server node remote terminal.
- Provides terminal input/output characteristics and management functions at the operating system services level.
- Offers standard terminal services, featuring good performance and device independence. Optionally, it offers methods for controlling the terminal's behavior in considerable detail.
- Supports high-availability implementations. The NVT protocols contain functions that restart interrupted communication.

The NVT protocols are divided into two sublayers in the Network Application to accommodate future enhancement. The two NVT sublayers are:

- **Command Terminal Protocol** - Allows terminal communication with Command Language Processors and application level programs. Its functions are oriented mainly toward command line input and output, but are general enough to support a broad range of video and hardcopy terminal applications.
- **Terminal Communication Protocol** - Controls the logical connections between applications and terminals. It extends the services offered by the DNA's Session Control layer by establishing logical links between two endpoints that are specific to terminal services. The endpoint at the host system is called a Portal; the endpoint at the server is called a Logical Terminal. The NVT connection created between the portal and the logical terminal is called a Binding.

Figure 6-4 shows the organization of the Network Virtual Terminal service protocols within the Network Application layer.

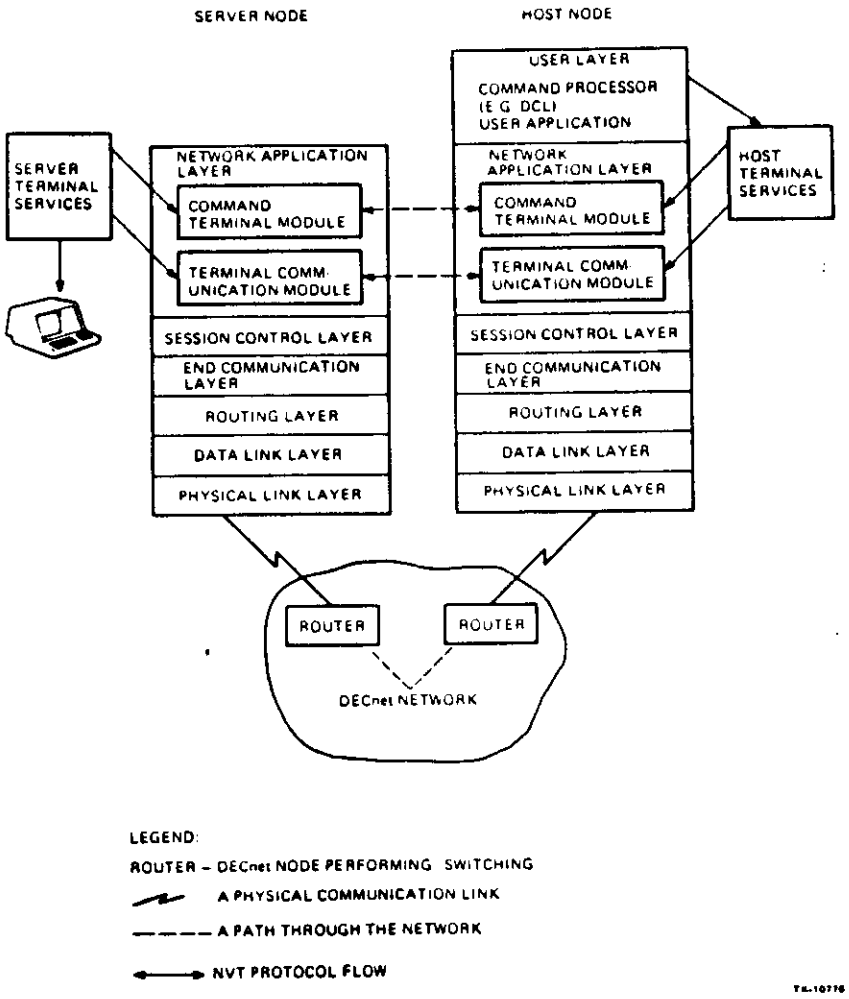


Figure 6-4 Network Virtual Terminal Service Protocol Organization

6.3.2.2 NVT Operations - In a typical network command terminal session, a terminal management module at a server node system with an active terminal requests a binding to some host node system. The binding is done by invoking a Terminal Communication Services function. (This is done typically in response to a user DCL command such as "Set Host" from the terminal operator.) Figure 6-5 shows the typical NVT protocol messages exchanged during a remote network command terminal session.

The Terminal Communication Services protocol module initiates a logical link via the Session Control layer's logical link functions to its counterpart in the host system. On discovering the incoming logical link, the host's Terminal Communication protocol module allocates a portal, thus beginning the formation of the binding. The host module then accepts the logical link. Once the logical link has been formed:

1. The server module sends a Bind Request message to the host module.
2. A Terminal Management module in the host recognizes the binding request, and causes the Terminal Communication Services protocol module to send a Bind Accept message. (The Terminal Management module may be acting on behalf of the host system's login process.)
3. Once the binding has been formed, the host readies the binding for the Command Terminal protocol by sending the Enter Command Mode message. (This action takes place in connection with the first terminal input/output request from the Login process at the host system.)
4. The respective protocol modules can now begin speaking the Command Terminal protocol language by each sending an Initiate message to the other.
5. After initialization, the dialogue of the remote terminal service session can begin.

The remote terminal service dialogue is a series of requests and responses. The terminal service requests normally originate with an application program in the host system. The application program issues requests to the host system's terminal services process which in turn issues the requests to the host system's NVT protocol module. The server system's NVT protocol module receives and reproduces those requests remotely and reissues them to the server system's terminal services process.

Termination of the remote terminal session can come about in a number of ways. A typical shutdown begins with a Logout command by the application process in the host system. As a result, the host's Terminal Communication Services module sends an Unbind Request message to the server. The server responds by releasing its resources.

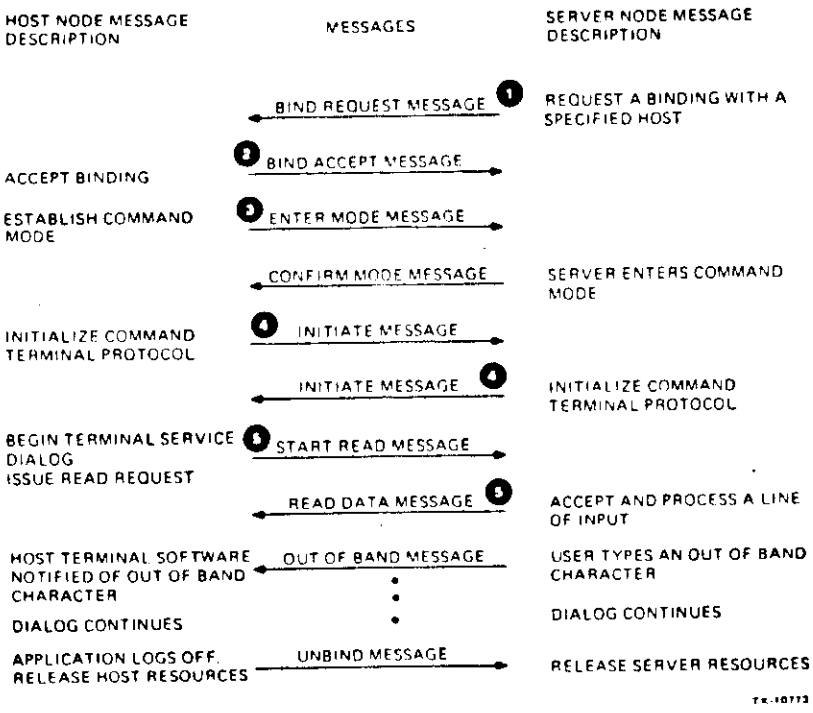


Figure 6-5 NVT Protocol Message Exchange

The messages used by the NVT protocol modules are specific to each sublayer module. Table 6-3 lists the different messages used by the Command Terminal Protocol module. Table 6-4 lists the different messages used by the Terminal Communication Protocol module.

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 the terminal on the 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	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 the Check Input message.
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.

Table 6-4 Terminal Communication Protocol Messages

Message	Function
Bind Request	Requests a binding and identifies the version and type of sending system.
Rebind Request	Requests a rebinding (used to reestablish a broken communication).
Unbind	Requests that a binding be released.
Bind Accept	Accepts a Bind Request message.
Enter Mode	Requests entry of a new mode. (The only mode currently defined is command mode, which 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).

6.3.3 X.25 Gateway Access Protocol

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 a DECnet node that is a DTE on the X.25 network over a DECnet logical link. The user program's local node need not be directly connected to the X.25 network. The user program's local node Network Application layer communicates from its X.25 Gateway Access module to the remote (DTE) node's Network Application layer X.25 Gateway Server module. The remote node's Gateway Server module in turn communicates via the Datalink Control layer's X.25 level 2 and 3 modules to the X.25 network DCE. The node that is a DTE to the X.25 Network (the X.25 Gateway Server node) is often called the Gateway System. The Gateway System makes X.25 Network facilities available to all remote nodes in the DECnet network. The X.25 Access and Gateway modules communicate by exchanging X.25 Gateway Access Protocol messages.

Figure 6-6 shows the relationship between the DNA X.25 modules and the X.25 network foreign DTE. It also shows the interactions between the user, X.25 Gateway Access, X.25 Gateway Server, and the Foreign DTE processes.

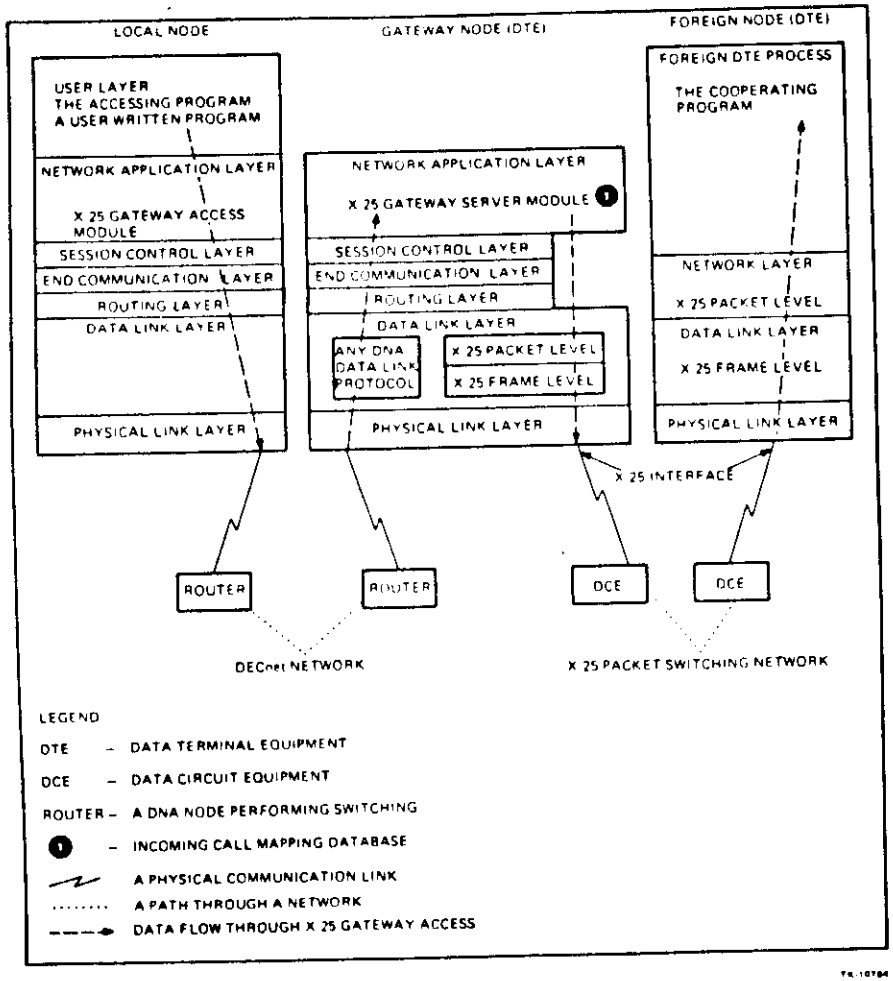


Figure 6-6 X.25 Gateway Access Module Relationships

6.3.3.1 X.25 Gateway Access Functional Description - The X.25 Gateway Access Protocol provides the following functions and features to the DECnet network user:

- 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 using this gateway function. Refer back to the Datalink Control layer, X.25 modules.)

- Supports all facilities and modes of operation defined in CCITT Recommendation X.25, except Datagram Service and the Delivery Confirmation bit.

- Supports user access to both Permanent Virtual Circuits (PVCs) and Switched Virtual Circuits (SVCs). Switched Virtual Circuits may be 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 need not reside in the Gateway System 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 or destination.

- Allows 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.

DECnet uses the following protocol modules to provide the X.25 Gateway Access service:

- **X.25 Gateway Access Module** - Interfaces with user-written programs and communicates via the DECnet network with the DTE directly connected to the X.25 PPSN (the Gateway system).

- **X.25 Gateway Server Module** - Resides in the DTE on the X.25 PPSN. This is the module that gives the Gateway System its name. The Gateway Server module communicates with the X.25 Gateway Access module over the DECnet network's logical links and acts on requests made by the X.25 Gateway Access module and the X.25 network. It uses the facilities of the Datalink Control layer X.25 Packet level modules (levels 2 and 3) to gain access to the X.25 PPSN.

- **X.25 Packet Level Module** (Level 3) - Resides in the Datalink layer of the Gateway system and acts as the DTE on the X.25 PPSN. It interfaces with the X.25 Frame level module for its connection to the X.25 DCE. This module is described in detail in the Datalink Control Layer module of this course.

- **X.25 Frame Level Module** (Level 2) - This module also resides in the Datalink Control layer of the Gateway system. It provides the X.25 Frame level protocol communicates with the X.25 PPSN's DCE. This module is also described in detail in the Datalink Control Layer module of this course.

6.3.3.2 **X.25 Gateway Access Operations** - The X.25 Gateway access operations involve three processes:

1. The user's process
2. The X.25 Server process
3. The foreign X.25 DTE process

The user process accesses the X.25 packet level protocol functions by issuing calls to the local X.25 Gateway Access module. The user's calls to the Gateway Access module are converted to X.25 Gateway Access Protocol messages, then issued as calls to the local Session Control layer for transmission over the DECnet logical links to the X.25 Network Gateway system.

The Gateway system's lower DNA layers receive the X.25 Gateway Access Protocol messages and pass them up to the Network Application layer X.25 Gateway Server module for processing. The Gateway system's X.25 Gateway Server module accesses the Datalink layer X.25 modules directly. The Datalink layer X.25 modules then establish the connection or channel to the X.25 DCE node for access to the X.25 Public Packet-Switching Network. The user is now logically connected to a foreign (non-DECnet) DTE process through the X.25 network.

In a typical X.25 Gateway Access Protocol dialogue, the first messages exchanged between the User's node and the Gateway system open a Permanent Virtual Circuit or place a virtual call (Switched Virtual Circuit) to the X.25 Gateway system's DCE to the X.25 network. The PVC or SVC are opened when the user's process issues calls to the X.25 Gateway Access module. The X.25 Gateway Access and Server modules then exchange the following protocol messages:

1. The X.25 Gateway Access module transmits either an Open message (for PVCs) or Outgoing Call message (for SVCs) to the Gateway system's X.25 Gateway Server module.
2. The user process is informed by an Incoming Accept message that the Foreign DTE has accepted the call.
3. Once the PVC has been opened, or the SVC has been accepted, the user process can exchange data with the Foreign DTE process.
4. When the user process wishes to terminate the virtual call, a Clear Request message is sent to the X.25 Gateway Server.
5. The Server then clears the call by issuing a Clear Request Packet to the X.25 Network.
6. The user process is informed of the termination by receipt of a Clear Confirm message from the Gateway system's X.25 Gateway Server module.

Figure 6-7 shows X.25 Gateway Access Protocol message exchange between a user process and the Gateway system (PPSN DTE node) via the X.25 Gateway Access and Server modules.

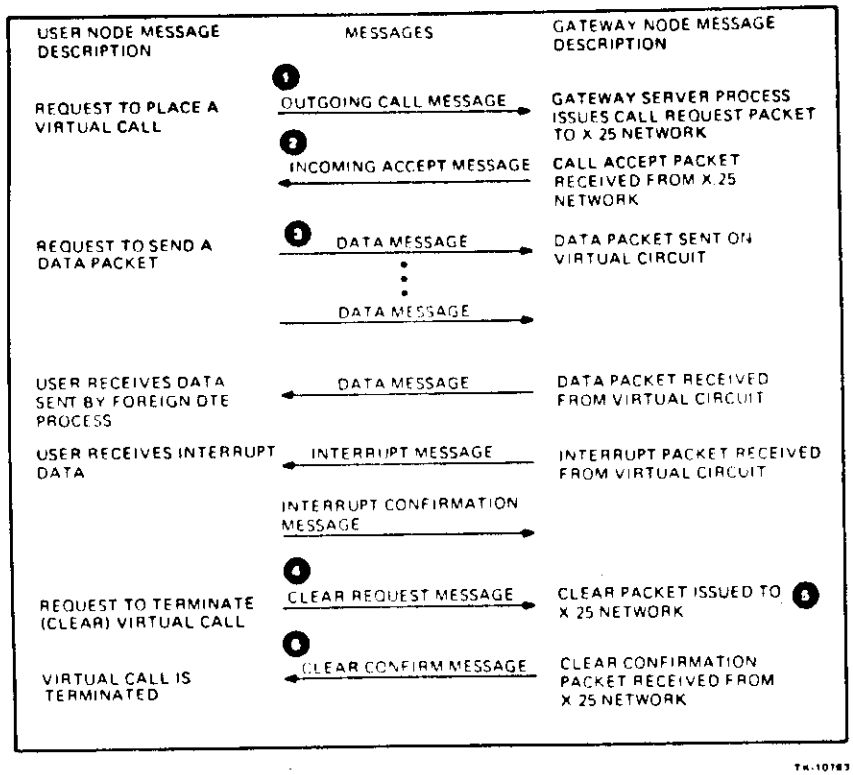


Figure 6-7 X.25 Gateway Access Protocol Message Exchange

Table 6-5 lists the X.25 Gateway Access Protocol messages used to communicate between the X.25 Gateway Access and X.25 Gateway Server modules.

Table 6-5 X.25 Gateway Access Messages

Message	Function
Open	Requests the use of a Permanent Virtual Circuit (PVC).
Open Accept	Accepts a PVC Open request and allocates the PVC to the Gateway Access user.
Open Reject	Rejects a PVC Open request, refusing allocation of the PVC to the Gateway 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 acceptance of an outgoing or incoming virtual call for lack of resources.
Incoming Accept	Indicates acceptance of an outgoing virtual call.
Incoming Call	Indicates that 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 that a clear request packet has been received from the X.25 network.
Clear Confirm	Indicates that a clear confirmation packet has been received from the X.25 network.
Reset Request	Requests that a virtual circuit be reset.
Reset Confirmation	Indicates a reset confirmation by either user of the X.25 network.

Table 6-5 X.25 Gateway Access Messages (Cont)

Message	Function
Reset Marker	Marks the place in a stream of Data messages where a reset occurred.
Data	Contains Data Packets outbound from, or inbound to, a user process.
Interrupt	Contains Interrupt Packet data outbound or inbound to a user process.
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).

6.3.4 SNA Gateway Access Protocol

The 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 (at times called SNA Secondary Logical Units or SLUs, in IBM terminology) residing anywhere in a DECnet network. To accomplish this, SNA Gateway Access communicates with a DECnet node that acts as a Physical Unit Type 2 (PU2) on the SNA network over a DECnet logical link. The user program's local node need not be directly connected to the SNA network. The user local node's Network Application layer communicates from its SNA Gateway Access module to the remote (PU2) node's Network Application layer SNA Gateway Server module. The remote node's Gateway Server module in turn communicates via an SNA Protocol Emulator to gain access to a process (called a Primary Logical Unit or PLU) in the SNA Network Host (called a PU5 node).

The SNA Access and Gateway modules in the DECnet nodes communicate by exchanging SNA Gateway Access Protocol messages.

Figure 6-8 shows the user program's local node, the DECnet Gateway system, and the SNA network.

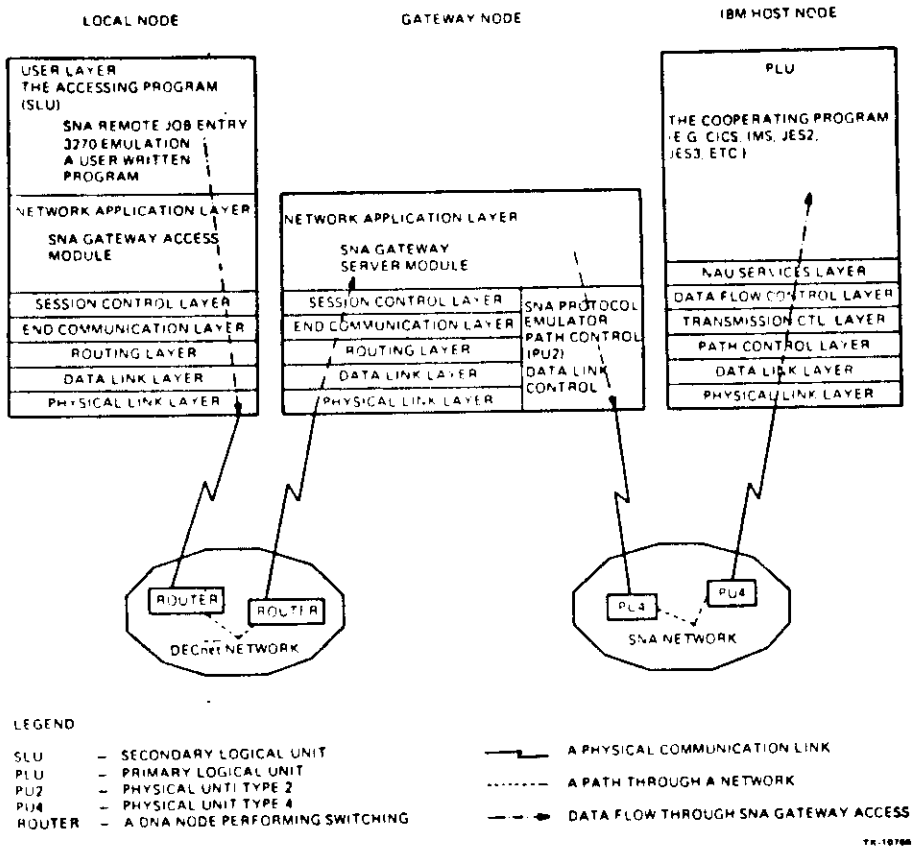


Figure 6-8 SNA Gateway Access Module Relationships

6.3.4.1 SNA Gateway Access Functional Description - The SNA Gateway Access Protocol provides the following functions and features to the DECnet network user:

- Supports communication between user-written (and a number of DIGITAL-written) programs in a host DECnet system and modules in a IBM host (PU5) system over an SNA network.
- Supports programs in DECnet systems that 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 exchange data in SNA sessions from any DECnet node whose DNA contains an SNA Gateway Access module. The user program's local node does not have to be the SNA network's PU2 node, nor does it have to be directly connected to the SNA network.
- Automatically manages the SNA session pacing for the user.
- Recovers from transient errors and reports fatal errors to the user.

DECnet implementations require the following protocol modules to provide SNA Gateway Access facilities and services:

- SNA Gateway Access Module - Must reside in the user program's local node. It receives user SNA session requests and communicates with the DECnet system that is directly connected to the SNA network (the PU2 node).

- SNA Gateway Server Module - Must reside in the DECnet system directly connected to the SNA network. This DECnet system acts as a PU2 node to the SNA network. The PU2 to the SNA network can also be referred to as the Gateway system, just as the DECnet X.25 DTE node was referred to as the X.25 Gateway system. This Network Application layer module communicates with the SNA Gateway Access module over a DECnet logical link and acts on requests made by the SNA Gateway Access module in the remote DECnet node and the PLU in the IBM PU5 node. The SNA Gateway Server uses the facilities offered by the SNA Protocol Emulator to gain access to the SNA network.

- SNA Protocol Emulator - Is a set of protocol modules in the Gateway system that perform the functions of SNA Path Control and Data Link Control. It connects the Gateway system to the SNA network as a Physical Unit Type 2 (PU2) node. The SNA protocol emulation modules also perform the functions necessary to communicate with the SNA network's System Service Control Point (SSCP) that controls the SNA network functions, such as network startup, control, and shutdown.

6.3.4.2 SNA Gateway Access Operations - There are three processes involved in the SNA Gateway Access operation:

1. The user's process
2. The SNA Gateway Server process
3. The IBM Host application process (PLU)

The user's process gains access to the SNA SLU functions by issuing calls on the SNA Gateway Access Routines in its local DECnet node. The routines translate these calls into SNA Gateway Access Protocol messages and transmit them over a DECnet logical link to the SNA Gateway Server process. The process transmits and receives SNA Basic Information Units (BIUs) by accessing the functions of the SNA Protocol Emulator. The SNA Protocol Emulator connects directly to the SNA network as a PU2 node. The SNA Protocol Emulator also contains the functions of the SNA Path Control and Data Link Control (SDLC) layers.

The messages are exchanged during a normal SNA Gateway Access session:

❶ The first message exchange establishes a communication session with a PLU in the PU5 node (an IBM Host node process). A typical SNA Gateway Access protocol dialogue is started by the Gateway Access transmitting a Connect Data message.

❷ The Gateway Server acting as the Gateway system exchanges Init-Self and Bind messages with the PLU after receipt of a Connect message from the Gateway Access process.

❸ Once the Gateway Server receives the Bind message from the PLU it will send a Bind Data message to Gateway Access process.

❹ The Gateway Access process transmits a Bind Accept Message to acknowledge receipt of the Bind Data message.

❺ Once Binding is complete the Gateway Access and Gateway Server modules exchange data with the PLU by sending and receiving Normal Data and Interrupt Data messages.

❻ The user process terminates the session by informing the PLU using the appropriate Data Flow Control Request Unit (RU).

❼ The PLU can then terminate the session by sending an Unbind message to the Server process.

❽ When the Unbind message is received by the Server process, the session terminates with an appropriate reason code, and the Gateway Server sends an Abort message to the user process.

❾ The user process senses the session termination when it receives the Abort message from the Gateway Server process.

Figure 6-9 shows the SNA Gateway Access protocol messages exchanged between the Gateway Access and Server processes when a session with a PLU is requested by a user process.

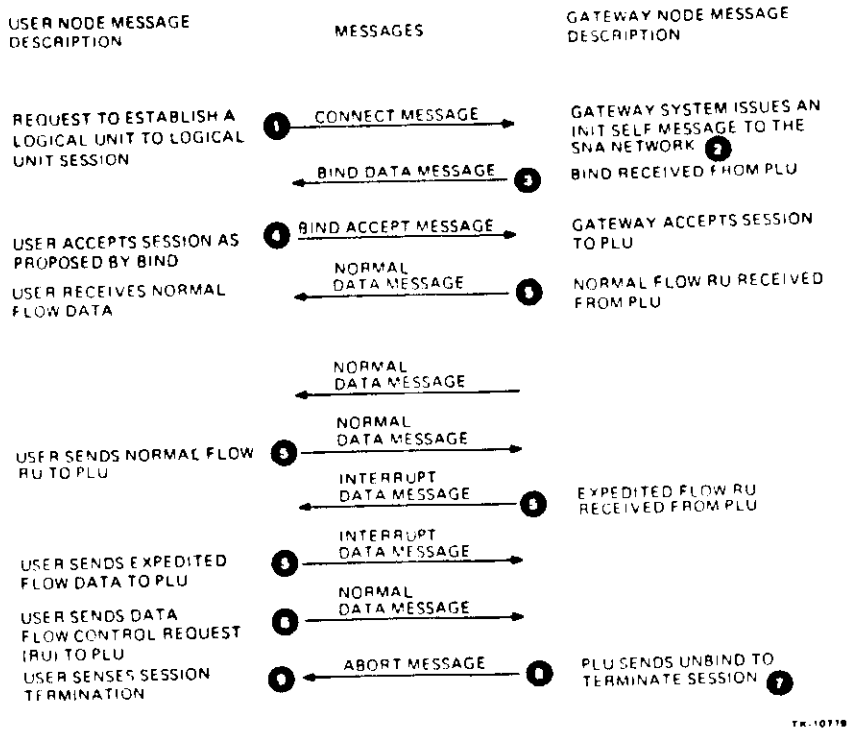


Figure 6-9 SNA Gateway Access Protocol Message Exchange

To accomplish SNA network remote access, the SNA Gateway Access and Server modules exchange SNA Gateway Protocol messages. These messages are listed in Table 6-6.

Table 6-6 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 successfully completed.
Abort	Requests abnormal termination of a Session, or indicates that an Unbind has been received from the PLU.

6.4 NETWORK APPLICATION LAYER MESSAGE FORMATTING

Network Application layer protocol messages are similar in format. However, each message's exact field format and length depends on the Network Application layer protocol module and the message type specified.

For a detailed description of the different message types, refer to the following references:

- DNA Data Access Protocol Functional Specification, Section 3.0
- DNA Network Virtual Terminal Functional Specification, Section 4.16
- DNA X.25 Gateway Access Functional Specification, Section 4.2
- DNA SNA Gateway Access Functional Specification, Section 3.2

Figure 6-10 shows the general format of messages used by the DAP protocol module.

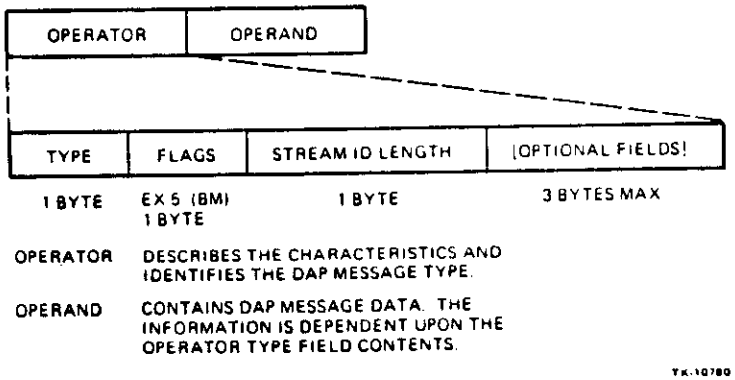


Figure 6-10 General DAP Protocol Message Formatting

Figure 6-11 shows the general format of messages used by the NVT, X.25, and SNA protocol modules.

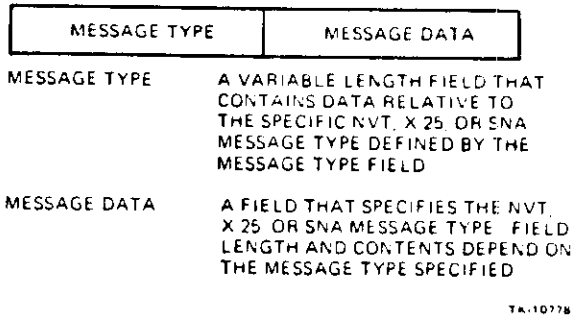


Figure 6-11 General NVT, X.25, and SNA Protocol Message Formatting

Network Application Layer

MODULE TEST

Answer the following questions by circling the letter next to the best possible solution. After you have finished the test, check your answers against the Answer Sheet provided in your Tests and Answers booklet. Do not proceed to the next module until you have answered all of the following questions.

1. Which of the following interfaces IS NOT defined by DNA for the Network Application layer?

   a. Network Management Interface

   b. Session Control Interface

   c. Data Link Interface

   d. Routing Interface

2. How many DIGITAL-supplied protocol modules are in the Network Application layer?

   a. 3

   b. 5

   c. 7

   d. 9

3. What is the major function performed by the DAP protocol?

   a. Provides Remote File Access

   b. Controls the Mail Utility

   c. Allows User-to-User Communication

   d. Provides SNA Gateway Services

4. What is FAL?

   a. File Access Looker

   b. File Access Listener

   c. Fault Analysis Language

   d. Fault Analysis Logger

5. What is NFARs?

   a. A standard file access system used under VAX/VMS operating systems

   b. A standard user access system used under RSTS operating systems

   c. A set of FORTRAN-callable subroutines used under RSX operating systems

   d. A set of FORTRAN-callable subroutines that interface IBM SNA nodes

6. What DAP message is exchanged after the exchange of Configuration messages?

   a. Access message

   b. Attributes message

   c. Acknowledge message

   d. Control (Get) message

7. What function is provided by NVT?

   a. Remote File Access

   b. User-to-User Communication

   c. Remote Terminal Access

   d. Public Packet-Switching Network (PPSN) Access

8. Which node transmits the NVT Bind Request Message?

   a. Server Node

   b. Host Node

   c. Remote Node

   d. Distant Node

9. What is the function performed by the X.25 Gateway Server module?

   a. Allows remote DECnet nodes access to a DECnet Gateway Node.

   b. Allows the DECnet Gateway node access to a PPSN network DCE node.

   c. Communicates directly to any Data Link Control layer protocol module.

   d. Allows the DECnet Gateway node to function as a DCE for the PPSN.

10. IBM SNA network nodes see the DECnet SNA Gateway node as a _____ node.

   a. PU2

   b. PU4

   c. PU5

   d. PLU

NETWORK MANAGEMENT

INTRODUCTION

This module introduces, describes, and illustrates the operations performed, and message formats used, by the Network Management layer of the DNA.

OBJECTIVES

To use DECnet in technical support of applications environments, Software Services and Customer Personnel must be able to:

1. Define and illustrate the terms associated with the DNA's Network Management layer.
2. Identify the Message Formats used by the DNA's Network Management layer.
3. Describe the functional operations performed by the DNA's Network Management layer.

LEARNING ACTIVITIES

1. Study the information in this module.
2. Read Chapter 7, The Network Management Layer, in the DECnet DIGITAL Network Architecture (Phase IV) General Description.
3. Take the module test at the end of this module.
4. Correct the test using the answer sheet provided in the Tests and Answers booklet. Review the material on any questions you may have missed before going on to the next module.

RESOURCES

1. DECnet DIGITAL Network Architecture (Phase IV) General Description

2. DNA Network Management layer Functional Specification, Phase IV, Version 4.0

3. DNA Low-Level Maintenance Operations Architectural Specification, Phase IV, Version 3.0

7.1 LAYER PURPOSE

The Network Management layer provides the network manager, network operators, and user-written programs with the ability to plan, control, monitor, and maintain the operation of centralized or distributed DECnet networks.

7.2 LAYER INTERFACES

There are five interfaces defined for use by the Network Management layer:

1. User Layer Interface - Normal adjacent layer interface that allows user access to Network Management functions.

2. Network Application Layer Interface - Provides certain Network Testing functions (described later) to users via Network Management layer functions.

3. Session Control Layer Interface - Provides the Network Manager with direct access to the Session Control's logical link functions.

4. Data Link Control Layer Interface - Used for maintenance testing, line servicing and remote system loading and dumping functions. It allows the Network Management layer direct access to the Protocol Modules in the Data Link layer.

5. Network Management Maintenance Interface - Connects the Network Management layer to all the other DNA layers. This interface is used for Network Management functions such as setting parameter values for the lower DNA layers, or monitoring the values of the various counters resident in lower DNA layers.

Figure 7-1 shows the different interfaces between the Network Management layer and the other layers defined by DNA.

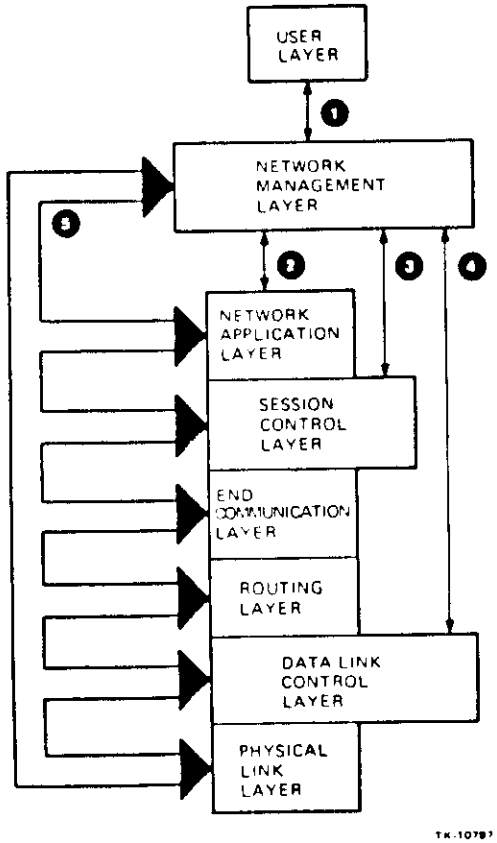


Figure 7-1 Network Management Layer Interfaces

7.3 NETWORK MANAGEMENT LAYER FUNCTIONAL DESCRIPTION

The Network Management layer allows the system manager, network operators, and user-written programs to control and monitor network operation. It also provides information for planning and troubleshooting a DECnet network.

The Network Management layer performs the following functions:

- Loads and dumps remote operating systems - A system manager can down-line load an operating system into an unattended, remote DECnet node; a remote node can up-line dump its operating system to a host node for examination by the system manager or network operator.

- Changes and examines network parameters - Network Operators and system managers can set and change local node and network parameters such as:

  Circuit parameters

  Line parameters

  Module parameters

  Node parameters

  Logging parameters

  The Network operator and system manager can also examine local and network parameters such as:

  Timer settings

  Line types

  Node names

- Examines network counters and events that indicate network performance - The Network Management layer contains an Event Logger routine that records significant events that happen in the lower DNA layers.

- Tests links at both the data link and logical link levels A system manager or operator can queue a test message for transmission to be looped back to its origin from either a hardware or logical (software) point in the network.

- Sets and displays the states of lines and nodes - The system manager or operator can reconfigure the network by turning nodes and links on and off.

The Network Management layer is actually a group of software modules called components. The Network Management components, for the most part, reside in the Network Management layer, however, there are some Network Management components in other DNA layers, including the Network Application and User layers. Another component, the Event Logger, resides in the Network Management layer, but has a queue in each of the lower DNA layers.

The Network Management layer has four outstanding characteristics that make it a very versatile part of the overall DNA structure:

1. Both programs and terminals can access and control a DECnet network via a set of functionally discrete calls and commands.

2. Control over a DECnet network can be either distributed or centralized. Distribution of control can be either partial or complete.

3. Network Management is Modular; a DECnet system is not required to implement the architecture in its entirety.

4. 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 requirements. This allows the managers of a network to construct their own network management philosophy.

7.4 NETWORK MANAGEMENT OPERATIONS

Table 7-1 lists the Network Management components, gives a brief summary of their functions, and the location of their DNA layer. Figure 7-2 shows the relationship between the Network Management components listed in Table 7-1.

Table 7-1 Network Management Components, Location and Function

Component	Location	Function
Network Control Program (NCP)	User Layer	A user-level utility that interfaces with the lower level modules. NCP can be used to control, monitor, and test the DECnet system and network. Refer to the DNA Network Management Functional Specification, Section 4.0, for specific information concerning the use and commands provided by the NCP Utility.
Network Management Access Routines	Network Management Layer	Provide generic Network Management functions. The routines communicate across logical links with the Network Management Listener by exchanging Network Information and Control Exchange (NICE) Protocol messages.
Network Management Listener	Network Management Layer	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	Network Management Layer	Take function requests from the Access Routines, translating the requests into system-dependent calls.
Link Watcher	Network Management Layer	Senses requests for service on a link from an adjacent node. The Link Watcher handles state changing for automatic remote load, dump, and trigger functions.

Table 7-1 Network Management Components, Location and Function (Cont)

Component	Location	Function
Maintenance Functions	Network Management Layer	Provide the actual protocol operation to support system maintenance functions such as down-line load and data link loopback testing. The protocol used is the Maintenance Operation Protocol (MOP).
Link Service Functions	Network Management Layer	The higher-level Network Management modules interface to the Link Service Functions for services that require a direct connection to the Data Link layer. This function bypasses the Session Control, End Communication, and Routing layers of DNA.
Event Logger	Network Management Layer	Records significant events occurring in the lower layers. DNA specifies the event types in the Network Management interface to each lower 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 receivers at other nodes of event occurrences. Events travel to a specified Sink Node for console, file, or monitor output.
Loopback Access Routines and Loopback Mirror	Network Application Layer	Used for logical link loopback tests, the Local Network Management functions can interface to the Loopback Access Routines, which use the Loopback Loopback Mirror protocol to loop test messages to or from a remote Loopback Mirror.

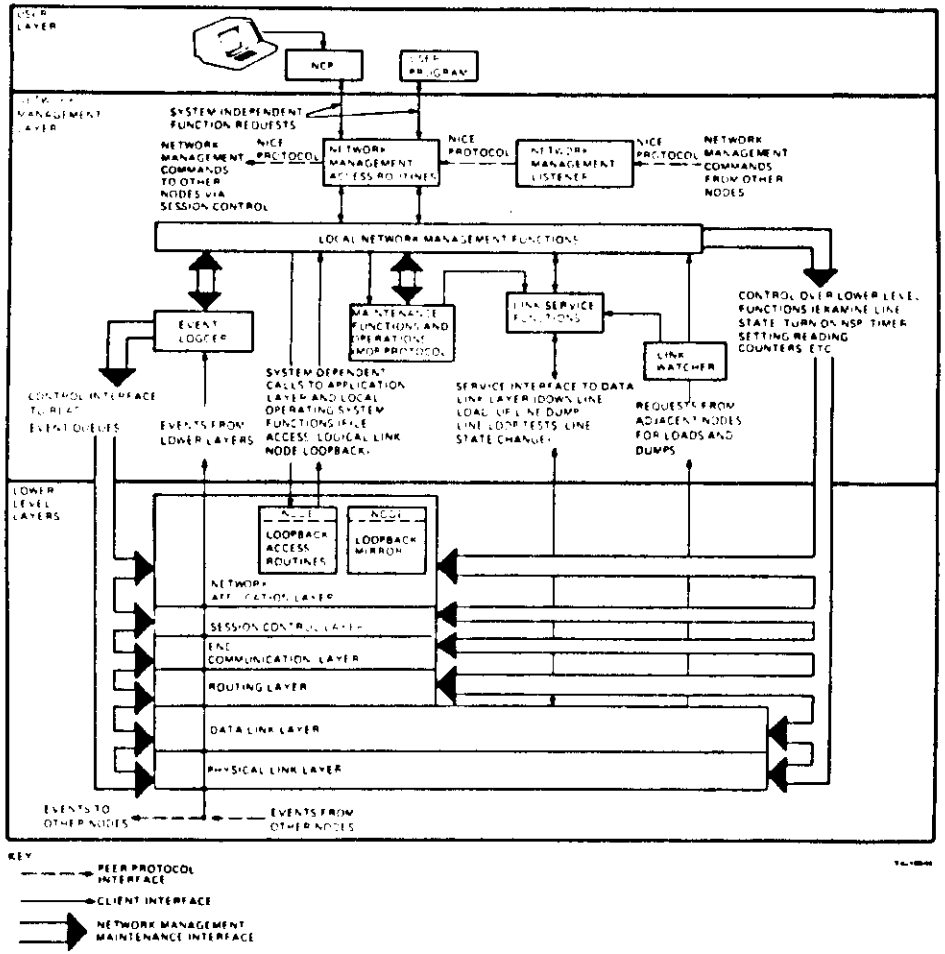


Figure 7-2 Network Management Components Relationships

There are three major functions performed by the Network Management Layer:

1. Network Control
2. Monitoring
3. Testing

Control and monitoring use the following Network Management protocols:

1. Network Information and Control Exchange (NICE) Protocol

2. Event logger Protocol

The network testing functions are divided into two categories:

1. **Node Loopback Testing** - Uses another Network Management protocol called the Loopback Mirror protocol with the NICE protocol. This method of testing uses logical links to loopback test messages from other network nodes to the message originator.

2. **Circuit Loopback Testing** - Is performed in a special mode of operation called the Maintenance Mode using the Maintenance Operation Protocol (MOP). The MOP protocol resides in the Network Management layer and acts on requests from the Network Management layer, but is not a Network Management protocol. The specific MOP functions are described later in this module.

7.4.1 Network Control And Monitoring

Network managers or network operators (under certain conditions) can control and monitor both the local node and network functions. Access to the Network Management layer functions of control and monitoring can be from either user-issued NCP commands or user-written program commands. Regardless of the access method, the user's management commands are translated by the Network Management Access Routines into NICE protocol messages for network functions, or function requests for local system-dependent calls.

Network Management controls and monitors both the local node and the DECnet network using:

- **Parameters** -- DNA specifies line, circuit, node, module, and logging parameters that are accessible through the Network Management layer. Parameters are internal network values that are important to the system manager. The system manager can examine, and, in some cases change, these values to control and manage the network. These values are classified as:

  Characteristics - Generally remain constant until changed by the system manager.

  Status - Reflect the condition of lines, circuits, nodes, modules, or loggers.

- Counters - DNA specifies line, circuit, node, and module counters that are internal network variables. These keep track of network and system errors and activity at each DNA layer in the local node. Counter values are returned to the users in response to user level requests.

- Events - Events are significant changes in the network. For example, changes in certain error counters may trigger several events that are automatically stored for the user by the Event Logger.

Each DNA layer's Functional Specification contains its Network Management interface, including Parameters, Counters, and Events. The DNA Network Management Functional Specification (Section 3.0) contains tables that define all of the Network Management Parameters, Counters, and Events.

7.4.2 Network Node Level Loopback Testing

Node level loopback testing consists of sending test messages over a logical link to be looped back at a specific point. The message sender is called the Executor; the message's destination is called the Target. The network operator may set a hardware loopback device on a line, line device, or modem between the executor and its adjacent target. Alternatively, software components can loopback test data. For example, FAL (a user program) or the Loopback mirror can loopback to their associated access routines. Figures 7-3 and 7-4 show some types of node level loopback testing available to network operators and managers.

There are two basic types of node level loopback testing. One type uses the Network Management layer Loopback Access routines and Loopback Mirror in the Network Application layer. These tests use logical links and the Loopback Mirror Protocol to send, receive, and loopback test data from the executor to the target node. The other type of node level loopback test does not use the Network Management software. This type of loopback testing allows a user task to loop back test data for another user task. For example, a file transfer can be used to test the logical link. Figures 7-3 and 7-4 give some examples of node level loopback testing using Network Management or User tasks to test logical links.

Figure 7-3 shows the loopback test data path after the user issued an NCP command called SET NODE node-id CIRCUIT circuit-id. This command causes the test data to be transmitted over a particular logical circuit and insists that the data be looped back either from a hardware loopback device that acts as an adjacent node, or the actual adjacent network node's Routing layer software. In either case the data is queued for transmission over a specified logical circuit path.

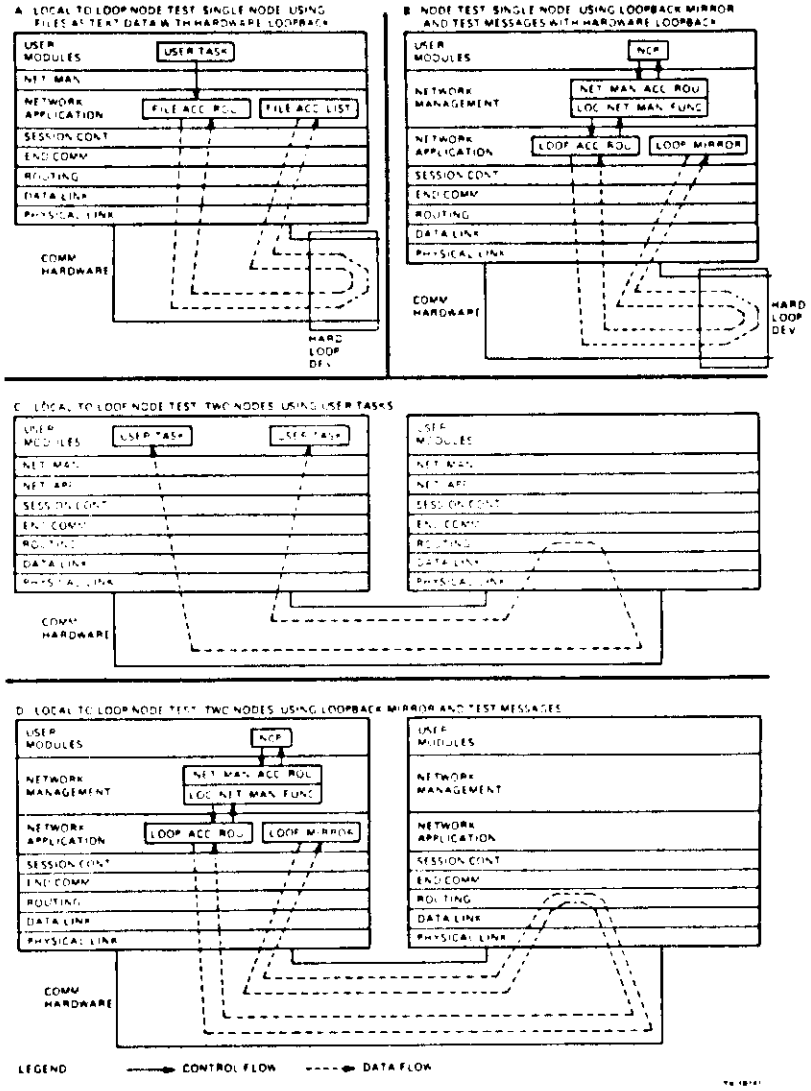


Figure 7-3 Node-Level Testing Using an Adjacent Loopback Node

Figure 7-4 shows node level loopback testing without the SET NODE node-id CIRCUIT circuit-id command. In these examples, the test data is transmitted over the cheapest logical path to the target node. The target node in these tests does not have to be adjacent to the test executor and may be executor, itself. This method of testing only verifies the ability of any two network nodes to communicate; it does not verify that the least-cost circuit path is operational.

In these examples, the test data is routed through the network to the target node over what each route-through node (if applicable) believes to be the least-cost path at the time the message was queued for transmission. Therefore, if the normal path to a node is down, and there is an alternative path for the communication, the routing algorithm at the various route-through nodes automatically reroutes the data over the now cheapest path to the destination. (Remember that with the Routing layer, if a circuit is down, its cost is raised to an infinite value and, therefore, not chosen by the routing algorithm as the circuit to use for the message exchange.)

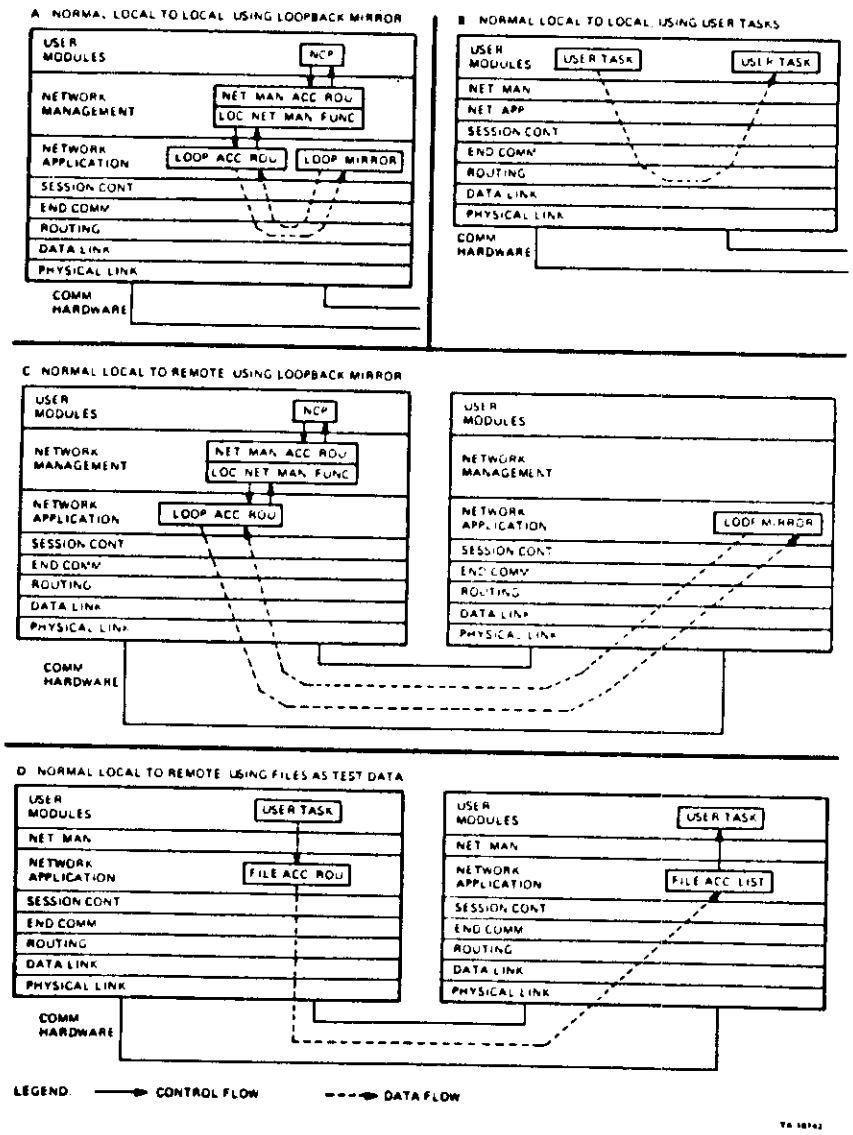


Figure 7-4 Node-Level Testing Using Logical Link Loopback Tests

7.4.3 Network Circuit Level Loopback Testing

Network Management uses the Maintenance Operation Protocol (MOP) to perform remote loading, dumping, controlling, and circuit level loopback testing. The functions of remote loading, dumping, and controlling are presented later in the MOP section of this course module. Circuit level loopback testing is a procedure for isolating faults in a physical connection between two adjacent nodes.

The Node level loopback tests described earlier are used to isolate the adjacent node pairs (where the fault is most likely to be found) from the rest of the network. Circuit level tests should be used to isolate the fault to either a hardware or software area. Software faults are best isolated using the node level loopback testing methods shown in Figure 7-4. Hardware faults are best isolated using the circuit level loopback testing methods shown in Figure 7-5.

There are three basic ways to perform circuit loopback tests:

1. Using NCP commands, the operator can set certain line devices to loopback mode, then perform the loopback tests. This method places the communications line device into an internal loopback mode, and causes all test messages to be looped back at the communications controller itself. (The test data is never actually placed on the physical line; it is turned around locally by the communications line device.)

2. Using NCP commands without any hardware loopback devices, an operator can cause test data to be looped back from the adjacent node's Maintenance Functions component in its Network Management layer.

3. Using hardware loopback devices, such as loopback test connectors or setting a loopback switch on a modem, the operator can manually set the point at which the test data is to be looped back. Once the loopback point is set up, the operator can execute the loopback test using NCP commands.

Figure 7-5 shows the different types of circuit loopback tests available. Notice that the use of the circuit loopback tests bypass the following DNA layers:

- Network Application
- Session Control
- End Communication
- Routing

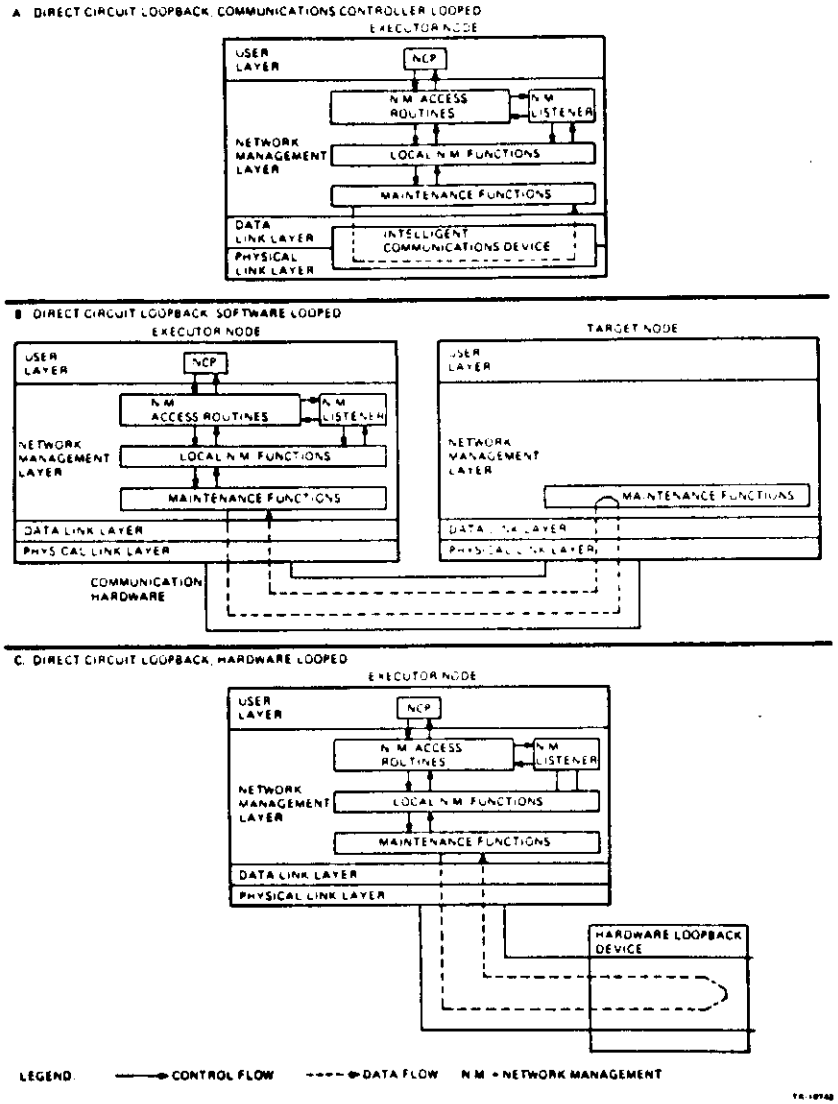


Figure 7-5 Circuit Level Loopback Testing

7.5 MAINTENANCE OPERATION PROTOCOL (MOP) FUNCTIONAL DESCRIPTION

Maintenance Operations are special, primitive functions that must be available without the services of any DNA layers between the Network Management and Data Link Control layers of DNA. The Maintenance mode is used for the following functions:

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

Maintenance operations must be available without the services of any DNA layers between Network Management and Data Link Control. This is because one of the nodes involved in the operation may be in a state where it cannot support more than a minimal Logical Data Link; 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 Control layer protocols. DDCMP supports all functions in its maintenance mode. The X.25 Frame Level supports circuit loopback in a special loopback mode. Ethernet supports all functions as maintenance protocol types.

With the exception of the Ethernet, maintenance operations are accomplished using the MOP protocol. Ethernet, loopback testing is done by including a standard protocol in the Ethernet Specification.

Figure 7-6 shows the relationship between MOP and the rest of the DNA layers and protocols.

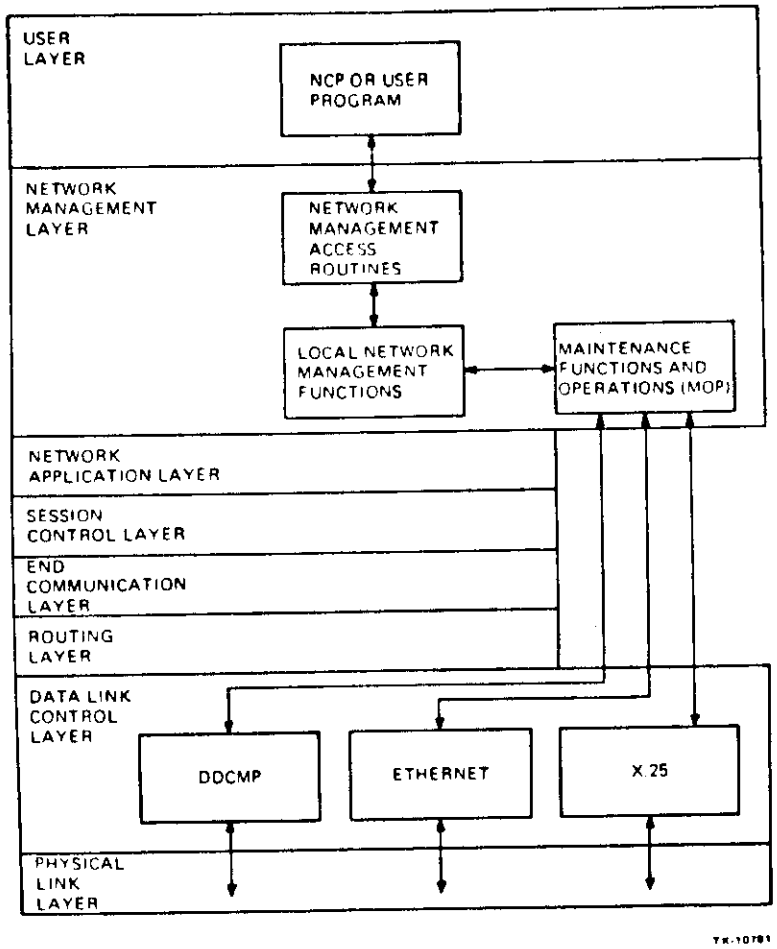


Figure 7-6 Relationship Between MOP and DNA

7.6 MAINTENANCE OPERATION PROTOCOL (MOP) OPERATIONS

In response to either higher level commands (user level commands) or MOP messages received from remote systems, MOP sends appropriate messages within the Data Link message envelope. MOP messages and functions handle all message acknowledgement, time-out, and retransmission functions that are normally handled by the higher DNA layers such as the Routing and End Communication layers.

The node being serviced by the MOP operation (down-line loaded, up-line dumped, controlled, or tested) is called the Target node. The node providing the services is called the Executor node. This naming convention is compatible with the normal Network Management naming convention discussed earlier. MOP messages pass alternately (this is logically a half-duplex message exchange) between the executor and the target node. The Executor and Target nodes for MOP functions must also be adjacent nodes.

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

In all cases of MOP message interchange, it is the responsibility of the Request For Service message sender to time-out and retransmit if a response is not received.

7.6.1 Down-Line Loading

Either the target or the executor node can initiate a down-line load. In either case:

1. The target sends a Program Request message to tell the executor what is needed and to indicate its willingness to cooperate.
2. The executor node's Link Watcher process receives the Program Request message and passes it to the Local Network Management Function's process for handling.
3. The Local Network Management Function's process responds by causing the Maintenance Functions and Operations process to send a Memory Load message (with or without a transfer address).

The major message exchange consists of Request Memory Load messages from the target and Memory Load responses from the executor.

The sequence is completed when the executor or host system sends a Memory Load with a Transfer Address or a Parameter Load with Transfer Address message. The Transfer Address causes the target node to begin the execution of the program code loaded into its main memory by the Memory Load message or messages.

On Ethernet Data Links, a target node can multicast its program request and select an executor from nodes that respond with an Assistance Volunteer message.

Figure 7-7 shows the operation and message exchange for a down-line load request when initiated from:

1. The target node
2. The executor node

NOTE

Remember, the executor and target nodes must, in all cases, be adjacent nodes.

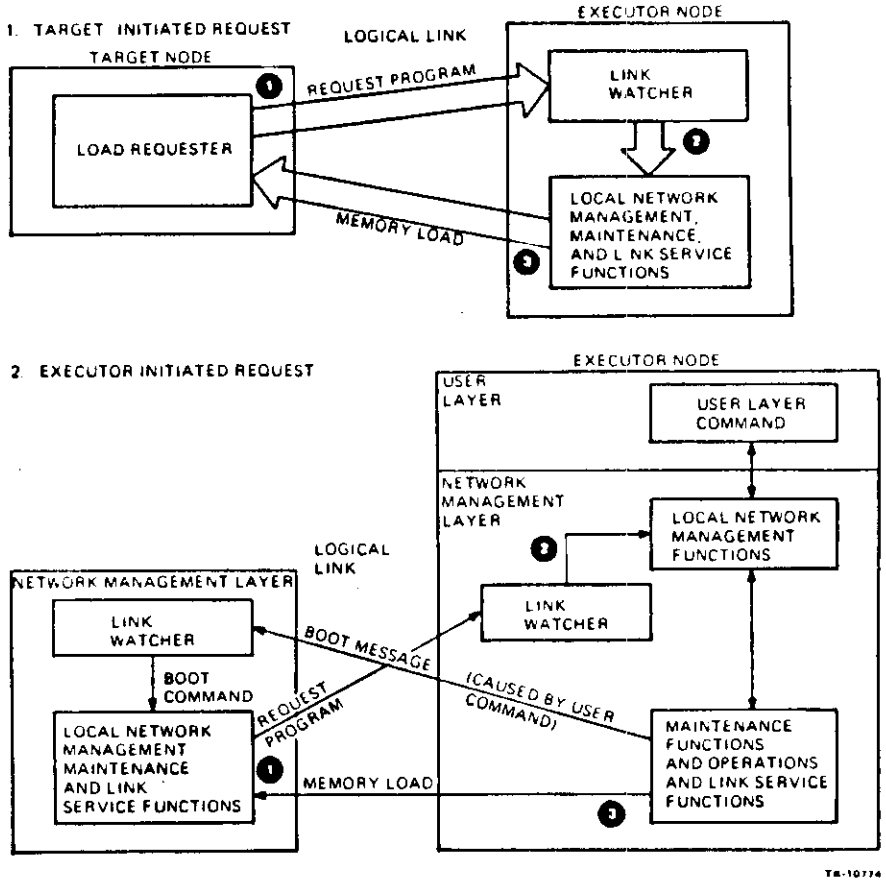


Figure 7-7 Down-Line Load Request Operation and Message Exchange

7.6.2 Up-Line Dumping

Up-line dumping is similar to down-line loading. The target node starts the memory dump. During the dump sequence it sends:

Request Dump Service messages ❶

Memory Dump Data messages ❸

The executor controls the dump process by sending:

Request Memory Dump messages ❷

Dump Complete messages ❹

Figure 7-8 shows the operation and message exchange for an up-line dump sequence.

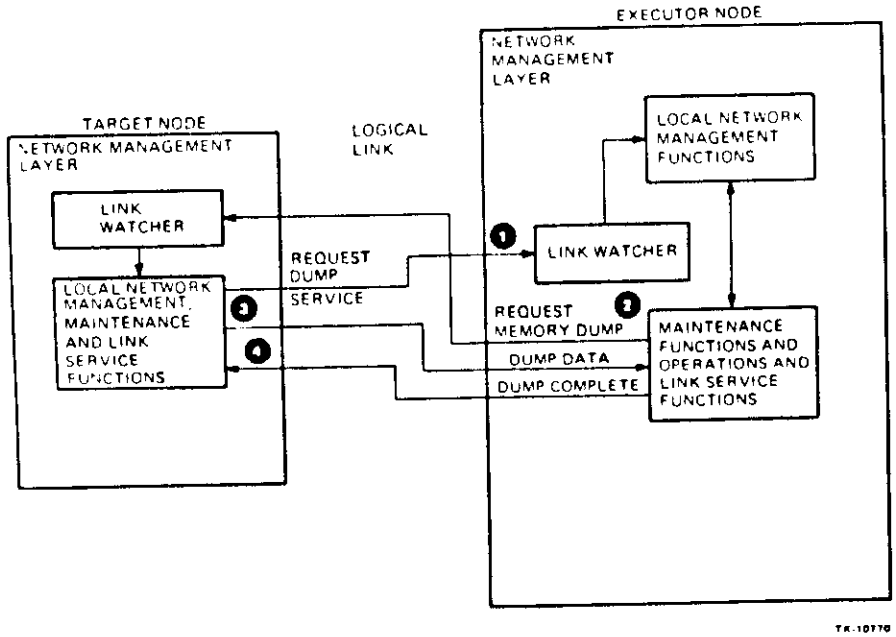


Figure 7-8 Up-Line Dump Operation and Message Exchange

7.6.3 Circuit Testing

MOP tests the data link by looping a test message from a point in the physical connection. By manually moving the loopback point and isolating components, the user can diagnose circuit faults. There are two methods for using MOP tests:

A. Loopback performed by an adjacent node - The executor of the tests (the active side) sends a Loop Data message ❶ on the circuit and waits for a response. The passive or target node returns a Looped Data message ❷ if the loopback is performed.

B. Loopback performed by other than an adjacent node - Loopback can be performed by loopback connectors, modems, or hard-wired drivers. The active side transmits a Loop Data message ❶ ; the passive side then returns the same Loop Data message ❷ .

MOP is not used for circuit testing on an Ethernet Data Link. Instead, the Ethernet Standard Loop Protocol is used.

Figure 7-9 shows the message exchange sequences during a MOP circuit test operation for both types of circuit loopback testing.

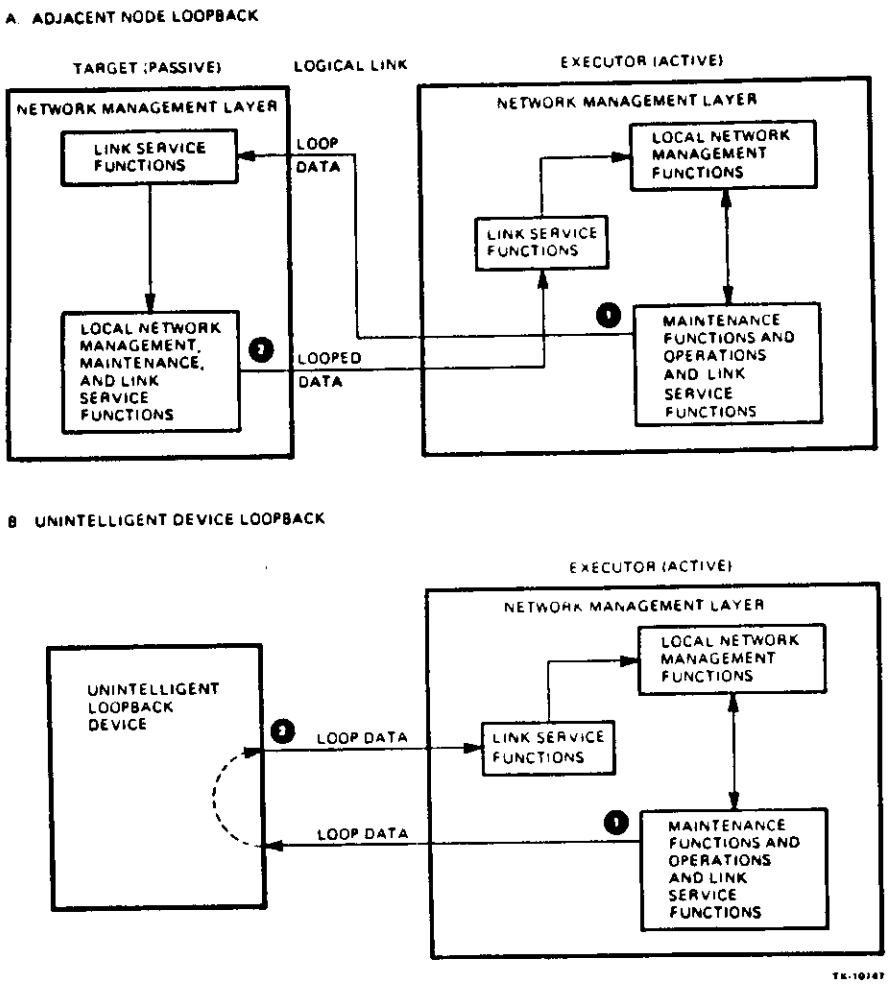


Figure 7-9 Circuit Testing Operation and Message Exchange

7.6.4 System Control

MOP can be used to control remote, possibly unattended, systems through what can be thought of as a virtual console. System Control functions can:

Using the Boot Message ❶, restart a remote system.

Using the Reserve Console Message ❷, an executor can take control of a remote system console and proceed through a complete command dialogue. The user's commands are carried in the Console Command and Poll Messages ❸ sent by the executor. The target node responds with Console Response and Acknowledge Messages ❹.

Using the Release Console Message ❺, the executor node can end the session. (A lack of executor messages within a time-out period at the target node also causes the session to end.)

Figure 7-10 shows the operation and message exchange sequence for controlling a remote target system.

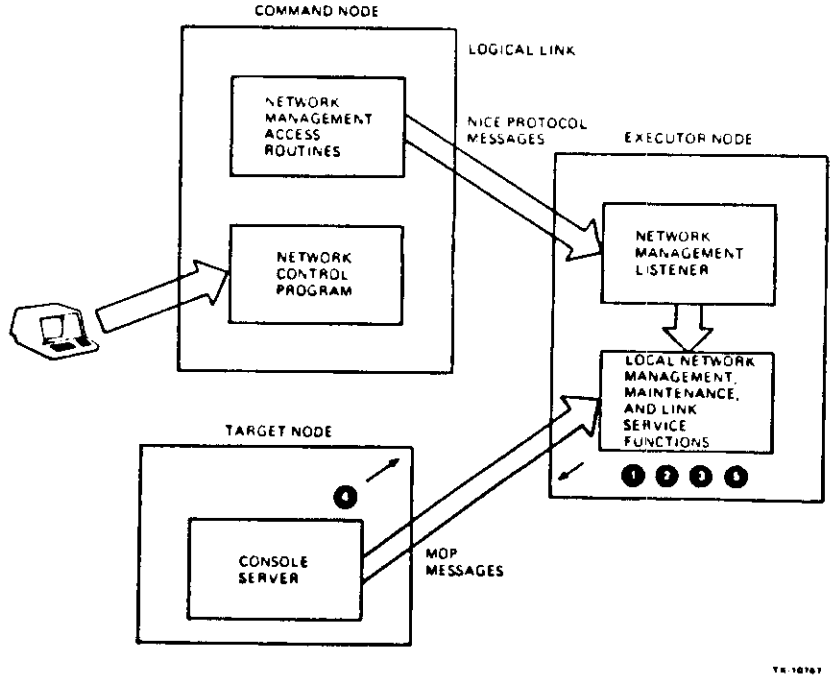


Figure 7-10 Remote System Control and Message Exchange

7.7 NETWORK MANAGEMENT LAYER MESSAGE FORMATS

There are three Network Management Protocols and another special Maintenance Mode protocol that may be accessed by the Network Management components. The three normal protocols are:

1. **Network Information and Control Exchange (NICE) Protocol** - NICE handles most Network Management functions.

2. **Event Logger Protocol** - Event logger logs significant events from remote network nodes.

3. **Loopback Mirror Protocol** - Physically resident in the Network Application layer, it handles the Network Management node loopback functions for system and network testing.

The Maintenance Mode protocol (MOP Protocol), a special protocol resident in the Network Management layer, is only used when testing, controlling, down-line loading, or up-line dumping the local or adjacent network system(s).

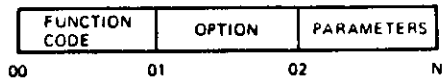
7.7.1 NICE Protocol Messages

Table 7-2 lists the NICE protocol messages and gives a brief description of each message function. Figure 7-11 shows the basic NICE protocol message format.

Table 7-2 NICE Protocol 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 Response	Requests a system-specific Network Management function. Provides request status and requested information in response to a NICE request.

REQUEST MESSAGES:



FUNCTION CODE	IDENTIFIES THE MESSAGE TYPE.
OPTION	IDENTIFIES THE TARGET NODE BY NODE ID, OR CIRCUIT ID.
PARAMETERS	NICE MESSAGE DATA TYPE AND DATA

RESPONSE MESSAGES:

RETURN CODE	IDENTIFIES THE RETURN MESSAGE TYPE
ERROR DETAIL	ERROR INFORMATION THAT PERTAINS TO THE SPECIFIC RETURN CODE TYPE
ERROR MESSAGE	SYSTEM-DEPENDENT DATA THAT FURTHER DESCRIBES THE STANDARD ERROR DETAIL MESSAGE
ENTITY ID	IDETIFIES THE ENTITY TYPE REPORTING THE ERROR
DATA	INFORMATION RESULTING FROM A TEST OPERATION OR INFORMATION RETURNED FROM A READ OR READ AND ZERO OPERATION

TA-10778

Figure 7-11 NICE Protocol Message Formats

7.7.2 Event Logger Protocol Message

There is only one Event Logger Protocol message: the Event message. It provides the following information when an event occurs:

- The Event Sink (Where the event is to be logged: Console, File, and/or Monitor)
- The event type and classification
- 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

Figure 7-12 shows the format of the Event Logger Message.

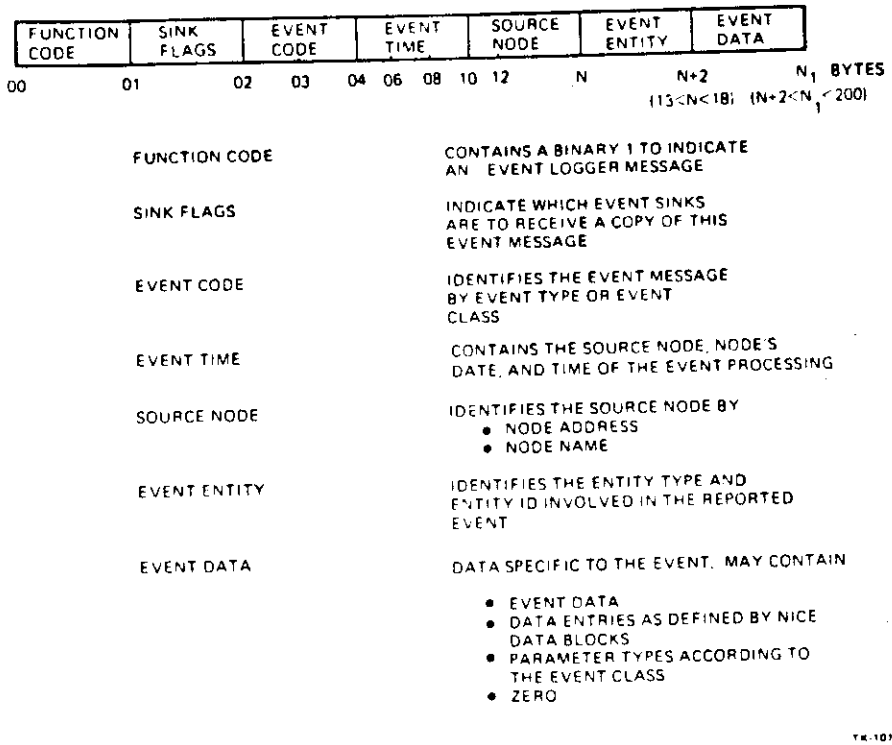


Figure 7-12 Event Logger Message Format

7.7.3 Loopback Mirror Protocol Messages

Table 7-3 lists the three Loopback Mirror protocol messages and gives a brief description of each message's function. Figure 7-13 shows the basic message format used by the Loopback Mirror protocol messages.

Table 7-3 Loopback Mirror Protocol Messages

Message	Description
Command	Requests a loop test and sends the data to be looped.
Connect Accept Data	Is sent in response to the Session Control Connect Request message. Informs the loop requestor of the maximum length message (in bytes) that can be looped.
Response	Returns status information and the looped back data.

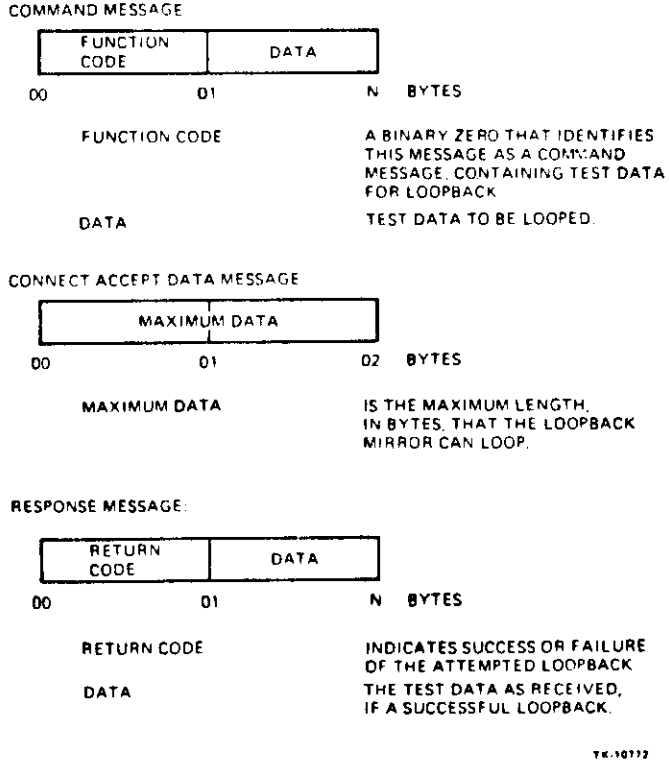


Figure 7-13 Loopback Mirror Protocol Message Formats

7.7.4 MOP Protocol Messages

Table 7-4 lists the MOP protocol messages and gives a brief description of each message function. Figure 7-14 shows the general message format used by the MOP messages.

Table 7-4 MOP Protocol Messages

Message	Description
Memory Load with Transfer Address	Loads the image data into memory at the load address, and the system to be started at the transfer address.
Memory Load Without Transfer Address	Loads the image data 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.
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 the willingness to perform dump or load service in response to a Request Program or Request Dump Service message. (Only applies to Broadcast circuits).
Loop Data	Contains data to be sent back in a loopback test.
Looped Data	Returns the data from a Loop Data message.

Table 7-4 MOP Protocol Messages (Cont)

Message	Description
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 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 dialogue 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.

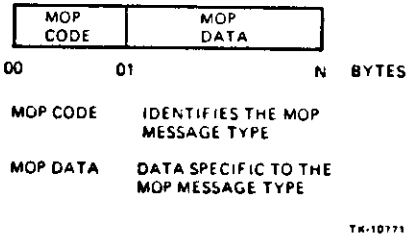


Figure 7-14 MOP Protocol Message Format

MODULE TEST

Answer the following questions by circling the letter next to the best possible solution. After you have finished the test, check your answers against the Answer Sheet provided in your Tests and Answers booklet. Do not proceed to the next module until you have answered all of the following questions.

1. How many interfaces are defined by DNA for the Network Management Layer?

   a. 1

   b. 3

   c. 5

   d. 7

2. What is NCP?

   a. A User-layer utility used to perform network control

   b. A User-layer tool used to perform system shutdown

   c. A Network management tool that provides Remote system loading abilities

   d. A Network Application layer utility that performs error checking

3. What Network Management layer protocol performs circuit loopback testing?

   a. NICE

   b. MOP

   c. NVT

   d. Loopback Mirror

4. Which Network Management layer protocol provides Network Control and Monitoring features?

   a. NICE

   b. NVT

   c. MOP

   d. Event Logger

5. What DNA layer contains the Loopback Mirror protocol?

   a. Network Management

   b. Network Application

   c. Session Control

   d. User

6. What method of network testing is best suited for identifying hardware or software faults?

   a. Node-level loopback testing

   b. Circuit-level loopback testing

   c. Component-level loopback testing

   d. Communications Controller diagnostics

7. What is MOP?

   a. Management Operation Protocol

   b. Maintenance Option Products

   c. Network Management Operations

   d. Maintenance Operation Protocol

8. What Network Management protocol performs down-line loading functions?

   a. NICE

   b. MOP

   c. Event Logger

   d. NCP

9. What Network Management protocol provides the Trigger Bootstrap message?

   a. NICE

   b. MOP

   c. Event Logger

   d. NCP

10. Which of the following is NOT a Loopback Mirror Protocol message?

    a. Command Message

    b. Connect Accept Message

    c. Control Message

    d. Response Message

MESSAGE EXCHANGE

INTRODUCTION

This module discusses the different message formats used by the various DNA layers. It gives examples that show sessions between DECnet network nodes. The protocol messages exchanged between these DECnet nodes is shown at two levels of detail. The first level is an overall basic representation of the messages that are exchanged between the DECnet nodes to facilitate user-to-user, process-to-process, or user-to-process communication. The second level details the Data Link Control layer DDCMP protocol message exchange between the two nodes. It also provides a representation of the Routing and End Communication layer protocol messages exchanged between the nodes. (Ethernet and X.25 message exchanges at the Data Link Level are not presented. Most Customers and Software Specialists do not have a way to capture these message types.)

OBJECTIVES

To use DECnet in technical support of applications environments, Software Services and Customer Personnel must be able to identify and interpret the messages used by the various DNA layers.

LEARNING ACTIVITIES

1. Study the information in this module.
2. Complete the exercise and module test located at the end of this module.
3. Correct the test using the answer sheet provided in the Tests and Answers booklet. Review the material on any questions you may have missed.
4. Take the DNA SPI Final Examination found at the end of the Tests and Answers booklet.
5. Correct your final examination using the answer sheet provided. Review the material on any questions you may have missed.

8

RESOURCES

1. DECnet DIGITAL Network Architecture (Phase IV) General Description
2. DNA DIGITAL Data Communications Message Protocol (DDCMP) Functional Specification, Version 4.1
3. DNA NI Data Link Architectural Functional Specification, Version 1.0
4. DNA NI Node Product Architectural Functional Specification, Version 1.0
5. DNA X.25 Frame Level Functional Specification
6. DNA X.25 Packet Level Functional Specification
7. DNA Routing Layer Functional Specification, Phase IV, Version 2.0
8. DNA End Communications (NSP) Functional Specification, Phase IV, Version 4.0
9. DNA Session Control Functional Specification, Version 1.0
10. DNA X.25 Gateway Access Functional Specification
11. DNA SNA Gateway Access Functional Specification
12. DNA Network Virtual Terminal Functional Specification
13. DNA Data Access Protocol (DAP) Functional Specification, Version 7.0
14. DNA Network Management Functional Specification, Version 3.0
15. DNA Low-Level Maintenance Operations Architectural Functional Specification, Version 3.0

8.1 BASIC MESSAGE EXCHANGE

Figure 8-1 shows the sequence of events and the protocol messages exchanged during a communication session between a local user and a remote DECnet node process (FAL). In this example, the local user has requested to read a file from a disk found on a remote DECnet node.

The following list is keyed to Figure 8-1.

I. Initialization of the Data Link Control Layer

- ❶ The local user enters an operating system specific command requesting to read a file residing on the remote system disk.
- ❷ The local operating system passes the request to the Network Application layer RMS utility for processing. The request is either a user-written program, Network utility, or VAX/VMS DCL command request.
- ❸ RMS, a DAP speaking Network Application layer module, passes a Logical Link Connect Request to the Session Control layer for processing.
- ❹ Session Control, after formatting the connect data, issues a Connect Request to the End Communication layer.
- ❺ End Communication passes a Connect Initiate message to the Routing layer, which begins the process of establishing a logical link on behalf of the user.
- ❻ The Routing layer determines the least-cost circuit to an adjacent node that is on the path to the destination node (the node that controls the remote disk). If the circuit to the adjacent node has not yet been initialized, it passes a Routing Initialization message to the proper Data Link layer module. If the circuit to the adjacent node has previously been initialized and the adjacent node is currently reachable, the Routing layer passes the NSP header and data, as data preceded by the Phase IV Data Packet Routing Header, to the Data Link Control layer module for transmission.

❼ The Data Link layer receives either the Routing Initialization or Phase IV Data Packet Routing message. Either message causes it to either initialize the Physical link to the adjacent node and transmit the Routing Initialization message as the first data message (if the Physical link is not yet initialized) or transmit the NSP Connect Initiate message enveloped by the Routing layer's Data Packet Routing message (if the Physical link has already been initialized).

II. Creation of the Logical Link

❽ The adjacent node's Communications controller receives the messages from the source node. It responds with some type of message acknowledgement (if applicable) depending upon its circuit type.

❾ The Communications controller (or Data Link layer Protocol Module, if applicable) passes the received Initialization or Connection Request up to the Routing layer for processing.

❿ The Routing layer processes the received data and initializes the link at the Routing layer level by exchanging Routing messages with the source node via Data Link layer data messages. Once initialized, the routing layer allows the NSP protocol to exchange messages over the physical link to establish the desired logical link for the user. NSP messages are exchanged as data by the Routing layers.

⓫ The NSP layers may now create and transfer messages over the logical link.

⓬ Session Control must either accept or reject the local node's user request. If accepted, it exchanges DAP-formatted data messages (DAP formatted messages contain the user's requested file data) to source and destination nodes.

⓭ RMS at the local system and FAL at the remote system exchange DAP data messages to transfer the user's requested file data across the network.

III. Disconnecting the Logical Link

⓮ The DAP speaking processes exchange Access Complete messages to disconnect the link.

⓯ NSP layers exchange Disconnect Initiate and Disconnect Confirm messages to destroy the logical link.

⓰ Routing and Data Link Control layers still maintain the physical circuit connection by exchanging:

Data Link Level:

Control Messages
Acknowledgment Messages
Data Messages

Routing Level:

Ethernet Router/Endnode Hello Messages
Nonbroadcast Hello and Test Messages

This message exchange keeps the Data Link and Routing layers initialized until halted by a Network Error or a Network Management command.

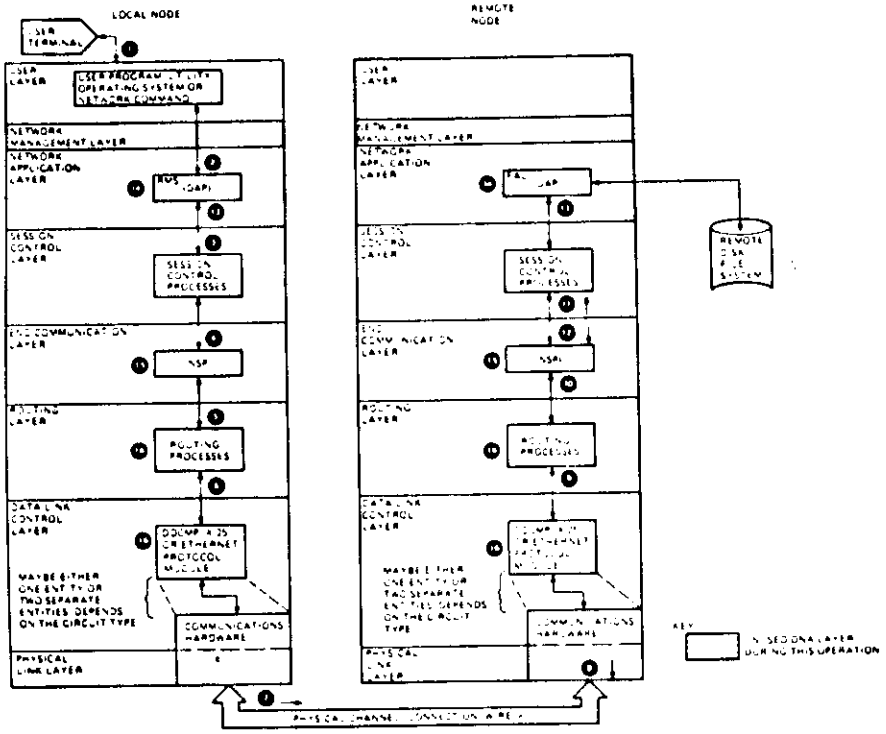


Figure 8-1 Basic DECnet Message Exchange

8.2 DETAILED MESSAGE EXCHANGE

Figures 8-2 through 8-4 and Examples 8-1 through 8-3 show message exchanges over the physical channel between the local source and remote destination node shown in Figure 8-1. The message exchange is shown in detail at the following three DNA layer levels:

1. Data Link Control Layer
2. Routing Layer
3. End Communication Layer

8.2.1 Data Link Layer

The Data Link layer can be one of two circuit types: Broadcast and Nonbroadcast.

1. Broadcast (Ethernet circuits) types do not exchange any control or acknowledgement messages; they simply exchange client data contained within a Data Link Data Frame message.

2. Nonbroadcast (X.25 and DDCMP circuits) types do exchange Data Link Control, Acknowledgement, and Data messages to initialize and maintain open communications circuits between two adjacent nodes.

   - DDCMP circuits remain on once started until they are commanded to halt or encounter some network error.

   - X.25 circuits are initiated or established for each user request received. Depending on the link type (Switched or Permanent), X.25 protocols at the DTE and DCE exchange messages to open the channel for user data exchange.

Figure 8-2 shows the exchange of Control, Acknowledgement, and Data messages used to provide communication between the local and remote nodes when they are connected by a DDCMP circuit.

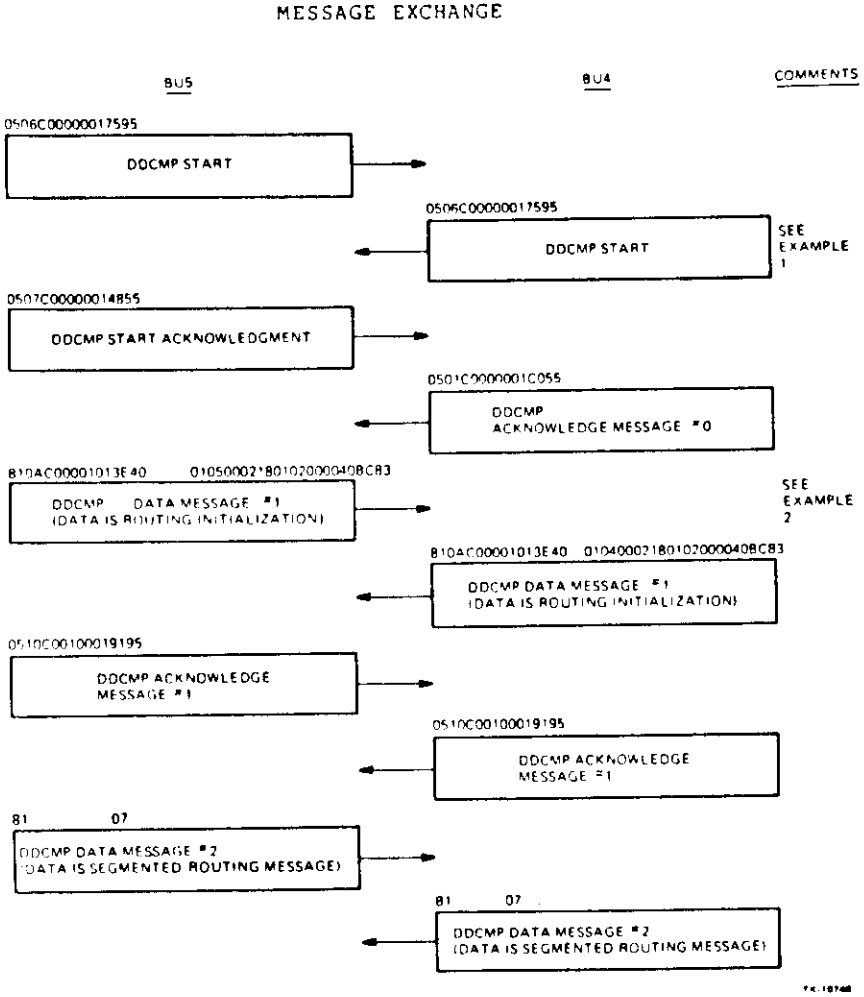


Figure 8-2 DDCMP Protocol Message Exchange

8.2.2 Routing Layer Message Exchange

The Routing layer messages exchanged depend upon the circuit type. The Routing layer distinguishes the difference between Broadcast and Nonbroadcast circuits.

Figure 8-3 shows the Routing protocol messages exchanged on Nonbroadcast DDCMP circuits.

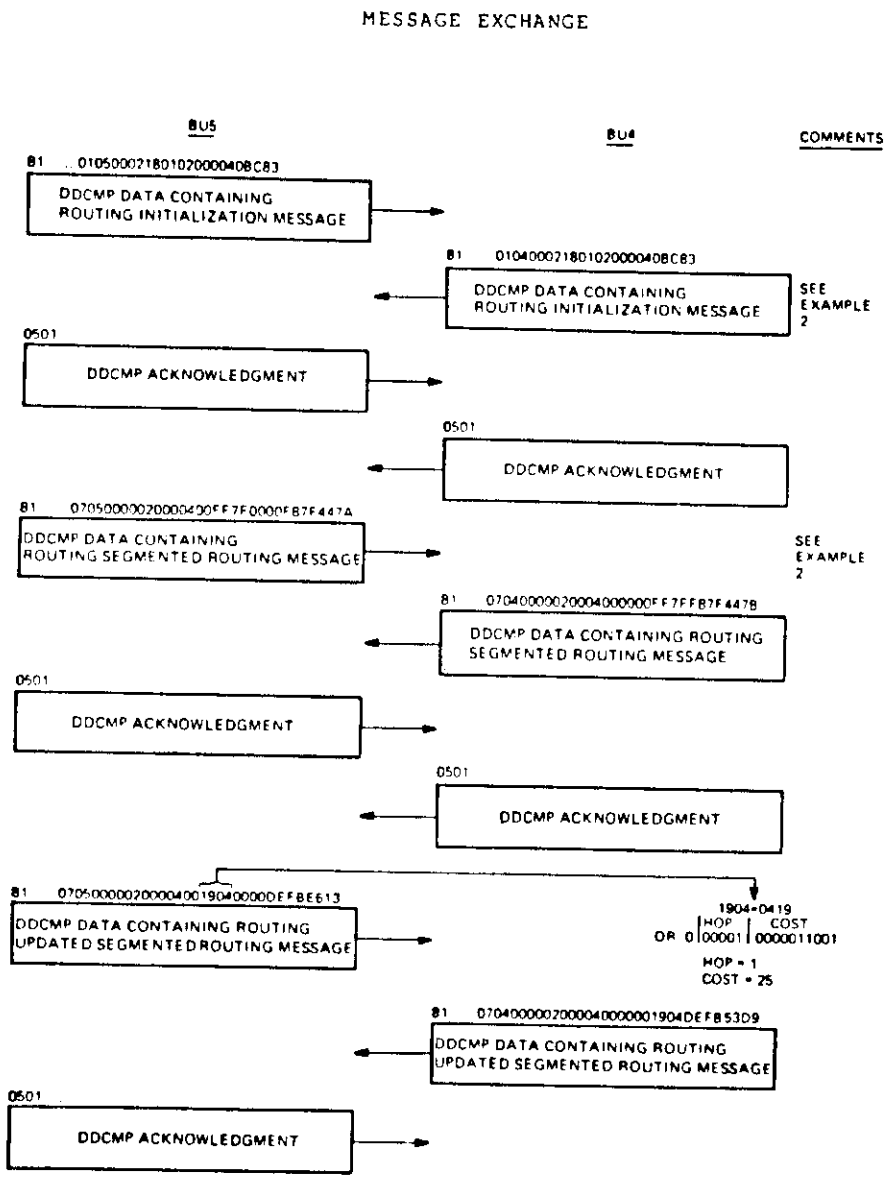


Figure 8-3 Routing Protocol Message Exchange

8.2.2.1 End Communication Layer Message Exchange - The End Communication and higher DNA layers function independently of the physical circuit type. For this reason Figure 8-4 shows the NSP message exchange on a DDCMP circuit. (There are no differences, except for message size, between DDCMP, X.25, or Ethernet circuits at the NSP or higher levels.)

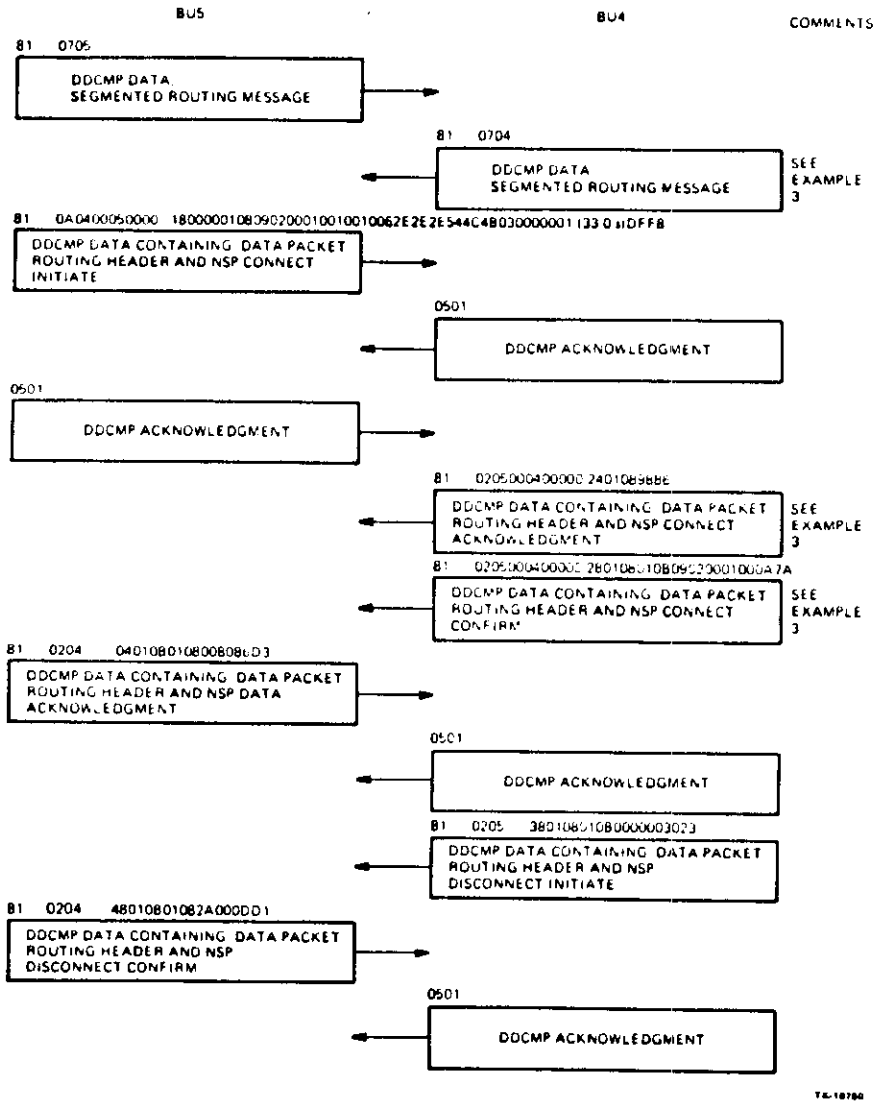


Figure 8-4 NSP Protocol Message Exchange

8.2.2.2 Protocol Message Decoding - The last section of each DNA layer's Functional Specification details the format and meaning of each field contained in the different messages used by that specific protocol. Examples 8-1 through 8-3 will help familiarize you with decoding DNA protocol messages. These examples are decoded from, and correlated to, the protocol messages shown in Figures 8-2 through 8-4.

FIELD NAME	HEX CONTENT	INTERPRETATION
ENQ	05	DDCMP CONTROL MESSAGE
TYPE	06	DDCMP START MESSAGE (1 - ACK 2 - NAK 3 - REP 6 - STRT 7 - STACK)
FLAG	C0	SELECT = 1 QSYNC = 1
RESPONSE NO.	00	NOT USED IN CONTROL MESSAGES
TRANSMIT NO.	00	NOT USED IN CONTROL MESSAGES
STATION NO.	01	ALWAYS 01 ON POINT TO POINT LINKS
BLOCK CHECK	7595	CYCLIC REDUNDANCY CHECK

Example 8-1 DDCMP Message Decoding

Routing Initialization:

Field Name	Hex Content	Interpretation
DDCMP HEADER	810AC0000101 3E40	
CTLFLG	01	Routing Control FLAG whose format is: 0000 xxx1 Hex where xxx = 0 = Initialization Message 1 = Verification Message 2 = Hello and Test Message 3 = Level 1 Routing Message 4 = Level 2 Routing Message 5 = Ethernet Router Hello Message 6 = Ethernet Endnode Hello Message
SRCNODE	0500	Interpreted as Source Node Number = 0005, which is the Node Address for Node BU5.
TLINFO	02	Routing Layer type where: 0 = Reserved 1 = Level 2 Routing Node 2 = Level 1 Routing Node 3 = Non-Routing (End) Node
BLKSIZE	1801	Interpreted as 0118 which is 280
TLVER	020000	Routing Layer Version
TIMER	40	Hello Timer Value in Seconds
BLOCK CHECK	BC83	DDCMP Block Check, CRC2

Example 8-2 Routing Nonbroadcast Message Decoding
(Sheet 1 of 2)

Segmented Routing Message:

Field Name	Hex Content	Interpretation
DDCMP HEADER	81	
CTLFKG	07	Identifies the message as a routing information message
SRCNODE	0500	The source node is 0005
RES	00	Reserved
SEGMENT:		
COUNT	0200	Number of Node IDs the routing information field (RTGINFO) is reporting for
STRTID	0400	First node ID reported on in the RTGINFO field 0400 = Node Address 0004 or BU4
RTGINFO	FF7F	Each RTGINFO field is formatted 0, Hop (5 bits), Cost (10 bits) If a node is not up, or is unreachable, the value for Hops and Cost is FF7F or maximum: 0 11111 1111111111 Hops Cost FF7F is the Hops and Cost from Node 5 to Node 4.
	0000	0000 is the Hops and Cost from Node 5 to itself. Notice that the segmented routing message only reports on pertinent nodes, and does not try to report on the entire network at one time.
CHECKSUM	F57F	Checksum for the routing segments
BLOCK CHECK	447A	DDCMP Block Check, CRC2

Example 8-2 Routing Nonbroadcast Message Decoding
(Sheet 2 of 2)

NSP Connect Initiate Message:

Field Name	Hex Content	Interpretation
DDCMP HEADER	812FC00204019517	
ROUTE HEADER	0A0400050000	0A = 00001010 which according to the routing specification means return to sender if destination is unreachable. 0400 = Destination Node 0004 0500 = Source Node 0005 00 = Visit Count is 0
NSP MESSAGE FLAG	18	NSP Connect Initiate Message
DSTADDR	0000	Destination Logical Link Address
SCRADDR	0108	Source Logical Link Address is 08F1
SERVICES	09	Request Message
INFO	02	NSP V4.0
SEGSIZE	0001	Maximum Data Segment is 0100 (256.) Bytes
DATA-CTL	0010010062E2E2E544C4B030000001 (33. 0s) . . . T L K	The Connect Initiate Data Field with the Task Name and Access Control Information
BLOCK CHECK	DFF8	DDCMP Block Check, CRC2

NSP Connect Acknowledge:

Field Name	Hex Content	Interpretation
DDCMP HEADER	8109800404016911	
ROUTE HEADER	020500040000	See Example 8-2 for explanation
MSGFLG	24	Connect Acknowledge Message
DSTADDR	0108	Logical Link Destination Address =0801
BLOCK CHECK	9886	

Example 8-3 NSP Message Decoding
(Sheet 1 of 3)

NSP Connect Confirm:

Field Name	Hex Content	Interpretation
DDCMP HEADER	8110C00405014083	
ROUTE HEADER	020500040000	
MSGFLG	28	Connect Confirm Message
DSTADDR	0108	Logical Link Destination Address is 0001
SRCADDR	010B	Logical Link Source Address is 0B01
SERVICES	09	Request Message
INFO	02	NSP V4.0
SEGSIZE	0001	Segment Size = 0100 (256.) in Bytes
DATA-CTL	00	No User Supplied Data
BLOCK CHECK	0A7A	DDCMP Block Check, CRC2

NSP Data Acknowledgement Message:

Field Name	Hex Content	Interpretation
DDCMP HEADER	810DC0050601DDB1	
ROUTE HEADER	020400050000	See Example 8-2 for details
MSGFLG	04	NSP Data Acknowledgment
DSTADDR	010B	Destination Logical Link Address
SRCADDR	0108	Source Logical Link Address
ACKNUM	0080	The format for this would be: 15 14 12 11 0 1 QUAL NUMBER Thus 8000 means acknowledge NSP message 0.
BLOCK CHECK	86D3	DDCMP Block Check, CRC2

Example 8-3 NSP Message Decoding
(Sheet 2 of 3)

NSP Disconnect Initiate Message:

Field Name	Hex Content	Interpretation
DDCMP HEADER	810EC00A0D01AE82	
ROUTE HEADER	0205	
MSGFLG	30	NSP Disconnect Initiate Message
DSTADDR	0108	Destination Logical Link Address
SCRADDR	010B	Source Logical Link Address
REASON	0000	First Two Bytes of Session Control Disconnect Data
DATA-CTL	00	Session Control Disconnect Data = 0
BLOCK CHECK	3023	DDCMP Block Check, CRC2

NSP Disconnect Confirm Message:

Field Name	Hex Content	Interpretation
DDCMP HEADER	810DC00D0B0158E3	
ROUTE HEADER	0204	
MSGFLG	48	NSP Disconnect Confirm Message
DSTADDR	010B	Destination Logical Link Address
SRCADDR	0108	Source Logical Link Address
REASON	2400	Interpreted as 002A or 42., which is Disconnect Complete
BLOCK CHECK	0DD1	

Example 8-3 NSP Message Decoding
(Sheet 3 of 3)

8.3 MODULE EXERCISE

Use the following blank pages to decode the protocol messages shown in Figures 8-2 through 8-4.

a. Reference your DNA Functional Specifications for each of the indicated protocol layers as you decode each protocol message.

b. Check your decoded answers against Examples 8-1 through 8-3 in this module. If any of your message decodes do not match, decode the message again. If your message decodes still do not match, consult your Course Administrator.

c. Once you have successfully decoded all of the messages shown in Figures 8-2 through 8-4, take the module test.

MODULE TEST

Answer the following questions by circling the letter next to the best possible solution. After you have finished the test, check your answers against the Answer Sheet provided in your Tests and Answers booklet. Once you have answered all of the following questions correctly, take the course Final Examination in your Tests and Answers booklet.

1. What message type is the DDMCP message 0506C00000017595 ?

   a. DDCMP STACK

   b. DDCMP STRT

   c. DDCMP DATA

   d. DDCMP MAINTENANCE

2. The DDCMP message 0501C0000001C055 acknowledges what received message number?

   a. 00

   b. 01

   c. C0

   d. C055

3. What DDCMP message header code is used to carry Routing layer information?

   a. 05

   b. 81

   c. 90

   d. 96

4. The Routing message CTLFLG containing 07 (HEX) identifies this message as a (an) _____ Routing message.

   a. Initialization

   b. Routing Information

   c. Hello and Test

   d. Ethernet Endnode Hello and Test

5. What NSP message type contains a 24 (HEX) in the MSGFLG field?

   a. Connect Initiate

   b. Request

   c. Data Acknowledgement

   d. Connect Acknowledgement

6. What NSP message type contains a 48 (HEX) in the MSGFLG field?

   a. Connect Initiate

   b. Disconnect Confirm

   c. Data Acknowledgment

   d. Connect Confirm

7. What node is the message 81.....
   07050000020004 00FF7F0000F57F447A transmitted from?

   a. 0000

   b. 0004

   c. 0005

   d. FF7F

8. What is the destination Logical link address defined by the following message:
   81..... 0A0400050000 18000001080902000 1.....DFF8

   a. 0000

   b. 0108

   c. 0A04

   d. DFF8

9. What is the source Logical Link Address defined by this message: 81..... 020400050000 04010B010800080 86D3

   a. 0204

   b. 0401

   c. 0108

   d. 86D3

10. What hexadecimal value identifies an NSP Disconnect Initiate message when it is contained in the NSP MSGFLG field?

    a. 01

    b. 04

    c. 38

    d. 48

APPENDIX: Glossary

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

ACOUSTIC COUPLER -- A device that converts electrical signals into audio signals, enabling data to be transmitted over the public switched telephone network via a conventional telephone handset.

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 the 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 the local node by a single physical line. In Multipoint Networks, all nodes in the network are Adjacent Nodes. For more details, see circuit and line definitions.

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.

ANCILLARY CONTROL PROCESSOR -- A program that acts as an interface between user software and an I/O driver.

AMPLITUDE MODULATION (AM) -- A method of transmission whereby the amplitude of the carrier wave signal is modified in accordance with the amplitude of the signal wave.

ASYNCHRONOUS TRANSMISSION -- Transmission in which time intervals between transmitted characters can be of unequal length. Transmission is controlled by start and stop elements at the beginning and end of each character. Also called Start-Stop transmission.

AUTOMATIC CALLING UNIT (ACU) -- A dialing device supplied by the communications common carrier. This device permits business machine to automatically dial calls over the communications network.

BACK-OFF -- The Ethernet Data Link procedure that ensures stable behavior on overloaded Ethernets by delaying retransmission 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 greater the data throughput.

BATCH PROCESSING -- A technique of data processing in which jobs are collected and grouped before processing. Data is thus normally processed in a deferred mode.

BAUD -- A unit of signaling speed equal to the number of discrete conditions or signal events per second. In asynchronous transmission, the unit of signaling speed corresponding to one interval is 20 milliseconds; that is, if the duration of the unit interval is 20 milliseconds, the signaling speed is 50 baud. Baud is the same as bit-per-second (bps) only if each signal event represents exactly one bit.

BINARY DIGIT (Bit) -- In binary notation, either of the characters 0 or 1. "Bit" is the commonly used abbreviation for Binary Digit.

BINARY SYNCHRONOUS PROTOCOL (BISYNC) -- 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.

BIT -- Abbreviation for BINARY DIGIT.

BLOCK -- Data transmitted as a unit, over which a coding procedure is usually applied for synchronization or error control purposes.

BPS (Bits Per Second) -- The commonly used measure for data transfer rate. (Other notations are bit(s), b.p.s., bit/sec., etc.)

BYTE -- Assumed to be 8 bits throughout unless stated otherwise. Commonly equal to a character.

CACHE -- A temporary storage subarea within an Ethernet End Node's Routing layer which maintains a list of the currently known adjacencies to that end node.

CARRIER -- A continuous frequency capable of being modulated or impressed with a signal.

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.

CENTRALIZED (COMPUTER) NETWORK -- A computer network configuration in which a central node provides computing power, control, or other services. Compare: DECENTRALIZED NETWORK.

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.

CIRCUIT -- A logical point-to-point connection between two nodes. On a Multipoint line, where there can be more than two nodes on a single line, several circuits can be established over the single line - one circuit between each slave and the master. You can turn on as many circuits as necessary, up to the maximum number of circuits allowed by the routing data base for the executing node. The routing data base has an upper limit of 16 circuits, depending upon the type of operating system used.

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.

CIRCUIT NUMBER -- A circuit number is a decimal number that is part of the identification of a circuit in a Multipoint line. The number is used internally to identify the circuit. For example, DMP-1.3 is a circuit-id that refers to circuit number 3 on DMP-1. The node to which the executor is connected by this circuit can be named BOSTON and have a node address of 27. However, the tributary address can be 123, which is the number recognized by the hardware that controls the connection. Note that the circuit number, node address, and tributary address are unrelated and can all be different.

CLIENT LAYER -- A module or protocol that requests the services of another module or protocol.

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

COMMAND NODE -- The node where a Network Control Program (NCP) command originates.

COMMON CARRIER -- In data communications, a public utility company that is recognized by an appropriate regulatory agency as having a vested interest and responsibility in furnishing communication services to the general public, e.g., Western Union, The Bell System, General Telephone, etc.

COMPONENT -- An element in the network that can be controlled and monitored. Components include lines and nodes.

COMPUTER NETWORK -- An interconnection of assemblies of computer systems, terminals, and communications facilities.

CONDITIONING -- The addition of equipment to leased voice-grade lines to provide specified minimum values of line characteristics required for data transmission, e.g., equalization and echo suppression.

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

CONGESTION CONTROL -- The routing layer component that manages buffers by limiting the maximum number of packets on a queue for a line. Also called transmit management.

CONTROLLER -- A hardware device that controls activity on a physical line. It resides on the system bus and its operations are directed by a device driver. A single DMP11 or DMV11 is a hardware controller.

CONTROL NODE STATION -- The station on a network that supervises the network control procedures such as polling, selecting, and recovery. It is also responsible for establishing order on the line in the event of contention, or any other abnormal situation arising between any stations on the network. (Compare with **TRIBUTARY STATION.**)

COST -- See circuit cost.

CSMA/CD -- See carrier-sense multiple access with collision detection.

DATA BASE -- The collection of information available to a computer system; a structured collection of information as an entity or collection of related files treated as an entity.

CONCENTRATION -- Collection of data at an intermediate point from several low and medium-speed lines for retransmission across high-speed lines.

DATA COMMUNICATION -- The interchange of data messages from one point to another over communication channels. (See also DATA TRANSMISSION.)

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 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.

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

DATA LINK -- A logical connection between two stations on the same circuit. On a multipoint link there can be multiple circuits.

DATA TERMINAL EQUIPMENT (DTE) -- The equipment, typically a computer system or terminal, comprising a data source and sink connected to common carrier communication facilities.

DATA TRANSMISSION -- Sending data from one place for reception elsewhere. Compare: DATA COMMUNICATION.

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

DATA TYPE -- ASCII data is subject to formatting conversion by the DECnet software, depending on the data's record attributes. Image data is a stream of bits to which the software applies no interpretation.

DECENTRALIZED (COMPUTER) NETWORK -- A computer network where some of the network control functions are distributed over several network nodes. Compare: CENTRALIZED NETWORK.

DEMODULATION -- The process of retrieving an original signal from a modulated carrier wave. This technique is used in data sets to make communication signals compatible with computer signals.

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 identity of the Ethernet Routers. The Router chosen is the one with the highest station address breaking ties.

DIAL-UP LINE -- A communications circuit that is established by a switched circuit connection.

DISTRIBUTED NETWORK -- A network configuration in which all node pairs are connected either directly, or through redundant paths through intermediate nodes.

DOWN-LINE LOAD -- The process by which one node in a computer network transfers an entire system image or a program (task) image to another node and causes it to be executed.

DTE -- See data terminal equipment.

ECHO SUPPRESSOR -- A device used to suppress the effects of an echo.

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

END NODE -- A topological description of a non-routing node. Since a non-routing node cannot perform route-through and supports only a single 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 the "user space" of a network node and communicates with session control to obtain logical link service.

EQUALIZATION -- Compensation for the increase in signal attenuation that occurs as the signal's frequency increases. Its purpose is to produce a flat frequency response.

ERROR CONTROL -- The NSP function that ensures the reliable, sequential delivery of NSP data messages. It consists of sequencing, acknowledgement, 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 million bit per second baseband coaxial cable.

EVENTS -- Occurrences that are logged for recording by Network Management.

EXECUTOR NODE -- An active network node connected to one end of a line used for a load, dump, or line loop test; it is the node that executes the request.

FLOW CONTROL -- The NSP function that coordinates the flow of data on a logical link in both directions, from transmit buffers to receive buffers, to ensure that data is not lost, to prevent buffer deadlock, and to minimize communications overhead.

The protocol mechanism that ensures that the sending station does not overrun the receiving station with more data than it can accept.

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

FRAME LEVEL -- Level II of the CCITT X.25 recommendation that defines the link access procedures for data exchange over the link between the DTE and DCE.

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

FREQUENCY DIVISION MULTIPLEXING (FDM) -- Dividing the available transmission frequency range into narrower bands, each of which is used for a separate channel.

FREQUENCY MODULATION (FM) -- A transmission method that changes the frequency of the carrier wave to correspond to changes in the signal wave.

FRONT-END PROCESSOR -- A communications computer associated with a host computer. It can perform line control, message handling, code conversion, error control, and application functions such as control and operation of special-purpose terminals.

FULL-DUPLEX LINE/CHANNEL -- The line can transmit data in both directions simultaneously. A full-duplex line allows a node to send and receive data at the same time. The channel line services concurrent communications in both directions (to and from the station).

FULLY-CONNECTED NETWORK -- A network in which each node is directly connected with every other node.

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 that contains the full complement of Routing components. A full routing node performs route-through functions.

GATEWAY -- A module or set of modules that transforms the conventions of one network into the conventions of another.

HALF-DUPLEX LINE/CHANNEL -- Transmits data in either direction, but only in one direction at any given time. The line cannot be used to send and receive data simultaneously. The channel permits two-way communications, but in only one direction at any time.

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

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

HOST NODE -- Provides services for another node (for example, during a down-line task load).

INTERACTIVE COMMUNICATION -- A protocol that allows one system to interact with a connected system at the transaction level rather than at the file level.

INTERFACE -- A shared boundary defined by common physical interconnection characteristics and meanings of interchanger signals. A device or equipment that allows communication between two systems, e.g., a hardware component or a common storage register. A shared logical boundary between two software components.

INTRA-ETHERNET PACKET -- A packet forwarded by a Router over an Ethernet to a destination end node; 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 that defines a 7-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 line reserved for the exclusive use of a leasing customer without inter-exchange switching arrangments. Also called a PRIVATE LINE.

LINE -- A physical path connecting adjacent nodes, over which circuits can be established. In Multipoint Networks there is only one line that is shared by all nodes in the network.

LINE COST -- An arbitrary integer value assigned to a line between two adjacent nodes. Each line has a separate cost. Packets are routed on paths with the least cost. Nodes on either end of a line can assign different costs to the same line.

LINE OR LINK LEVEL LOOPBACK -- Testing a specific data link by sending messages directly to the Data Link layer and over a wire to a device that returns the message to the source.

LINE NUMBER -- A line number is a decimal number or alphanumeric name that consists of up to 16 characters and/or numerics to uniquely identify a specific line on a node.

LINK -- Any specific relationship between two nodes in a network. A communications path between two nodes. A data link (refer to LINE).

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 responsible for channel allocation (collision avoidance) and contention resolution (collision handling).

LOCAL NODE -- A frame of reference; the node at which the user is physically located (Compare: REMOTE NODE).

LOGGING -- Recording information from an occurrence that has potential significance in the operation and/or maintenance of a network 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 an its DCE for a given virtual channel.

LOGICAL LINK -- A carrier of a single stream of full-duplex traffic between two user-level processes. A virtual channel between two end users in the same node or in separate nodes. Session control acts as an interface between an end user requiring logical link service, and NSP, which actually creates, maintains, and destroys logical links. Many logical links may be sent (multiplexed) within a single physical link

LOOP NODE -- A node associated with an adjacency for loop testing purposes. The NCP SET NODE CIRCUIT command sets the loopback node name.

MASTER STATION -- A station that controls a channel at a given instant for the purpose of sending data messages to a slave station (whether or not it actually does). A master station controls the polling of all tributary stations.

MAXIMUM ADDRESS -- The maximum number of nodes the local node can handle in its routing data base.

MAXIMUM COST -- An operator-controlled transport 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-controlled transport 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 network nodes having the greatest routing cost, where routing cost is the cost of the least-cost path between a given pair of nodes.

MAXIMUM PATH LENGTH -- The routine distance between the two nodes of the network having the greatest routing distance, where routine distance is the length of the least-cost path between a given pair of nodes.

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

MESSAGE -- The unit of communication as seen by the user; it can be segmented into several packets to traverse the network, or in some cases several messages can be carried in one packet.

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

MODEM -- Modulator-demodulator. A device that modulates and demodulates signals transmitted and received over communications circuits. Often referred to as a DATASET.

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 two or more data streams on a single channel. In DNA, NSP is the only protocol that multiplexes.

MULTIPOINT CONNECTION OR LINE -- A network configuration in which more than two computers are attached to the same line. Use of this type of line normally requires some kind of polling mechanism, addressing each tributary station with a unique ID. Also called multidrop. (Compare POINT-TO-POINT CONNECTION.)

NETWORK -- A configuration of two or more computers linked to share information and resources.

NETWORK DIAMETER -- The distance between the two nodes of the network having the greatest reachability distance, which is the length of the shortest path between a given pair of nodes.

NODE -- A network management component consisting of a DIGITAL system that supports DECnet software.

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, network services, and transport 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 line.

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 alpha character.

NODE NAME MAPPING TABLE -- Defines the correspondence between node names and node addresses or channel numbers. Session control uses the table to identify destination nodes for outgoing connect requests and source nodes for incoming connect requests.

NOISE -- Undesirable disturbances in a communication system. Noise can generate errors in transmission.

NON-ROUTING (END) NODE -- Can send and receive packets to other nodes in the network, but packets cannot be forwarded or routed through it. A Phase III or Phase IV DECnet node connected to the network by a single active circuit.

NULL MODEM -- A device that interfaces between a local peripheral that normally requires a modem, and the computer near it that expects to drive a modem to interface to that device. An imitation modem in both directions.

OBJECT TYPE -- Numeric value that may be used for process or task addressing by DECnet processes instead of a process name.

OPERATING SYSTEM -- An integrated collection of service routines for supervising the sequencing and processing of programs by a computer. An operating system provides access to the features of a central processor, and also organizes and optimizes a central processor and peripheral equipment for a certain range of applications.

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.

PACKETS -- A group of bits including data and control elements switched and transmitted as a composite whole. The data and control elements and possibly error control information, are arranged in a specified format. When stripped of its route header and passed up to the End Communication layer, it becomes a datagram.

PACKET LEVEL -- Level III of the CCITT X.25 recommendation that defines the packet format and control procedures for exchanging packets.

PACKET LIFETIME CONTROL -- The routing component that monitors lines to detect if a line has gone down, and prevents excessive packet looping by discarding packets that exceed 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 test, the node that loops back the test message.

PATH -- The route a packet takes from source node to destination node. This can be a sequence of connected nodes between two nodes.

PATH COST -- The sum of the line costs along a path between two nodes. Path cost is direction-dependent; cost from A to B is not necessarily equal to cost from B to A.

PATH LENGTH -- The sum of the hops along a path between two nodes. Path length is the number of lines a packet must go through to reach its destination.

PEER PROTOCOL -- A protocol for communication between modules in the same layer in different nodes.

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

PHASE III NODE -- Runs a Phase III implementation of DECnet and supports routing as either a full-routing or non-routing (end) node. (Refer to ROUTING NODE and NON-ROUTING NODE.)

PHASE IV NODE -- Runs a Phase IV implementation of DECnet and supports routing as either a full-routing or non-routing (end) node. (Refer to ROUTING NODE and NON-ROUTING NODE.)

PHASE MODULATION (PM) -- Transmission method that varies the phase angle of the carrier wave according to the signal.

PHYSICAL LINK -- An individual hardware-addressable communications path. In terms of hardware, a physical link is a combination of either a channel and its controllers, or a channel, controllers, and units.

PIGGYBACKING -- Sending an acknowledgement within a returned data message.

PIPELINING -- Sending messages without waiting for individual acknowledgement of each successive message.

POINT-TO-POINT (LINK/CHANNEL/CIRCUIT) -- The direct connection of two nodes by a single link, channel, or circuit. A network configuration in which a connection is established between two (and only two) computers (compare MULTIPOINT CONNECTION).

POLLING -- The process of inviting another station or node to transmit data (refer to CONTROL STATION and TRIBUTARY STATION).

PORT -- A collection of control variables and parameters for managing logical links. Each logical link has a port at each end. Each End Communication layer at each node in the network has a number of available ports. When Session Control requests a logical link, or that a port be opened to receive an incoming connect request, NSP allocates a port if sufficient resources are available.

PROTOCOL -- A basic procedure or set of rules that govern and control the flow of messages between computers. Also, a set of conventions between communicating processes on the format and content of messages to be exchanged. DIGITAL Network Architecture (DNA) uses three basic protocols in a layered structure as the framework for DECnet.

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

REACHABLE NODE -- A destination node to which the DECnet routing module has determined there is a usable path for packet exchange.

REAL-TIME SYSTEM -- A system performing computation during the actual time the related physical process transpires, so that the results of the computation can be used in guiding the process.

REASSEMBLY -- Placing multiple, received data segments by NSP into a single session control receive buffer.

REMOTE JOB ENTRY (RJE) -- Submission of jobs through an input device that has access to a computer through a communications link. The mode of operation that allows input of a batch job by a card reader at a remote site and receipt of the output via a line printer or card punch at a remote site.

REMOTE NODE -- A frame of reference; any node other than the one at which the user is located in the network (Compare: LOCAL NODE).

REQUEST COUNT
1) Variables that NSP uses to determine when to send data.
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 to determine when to send data.

RETRANSMISSION -- Resending data messages not acknowledged within a certain period of time. This is usually part of a protocol's error control mechanisms.

RECORD MANAGEMENT SERVICES (RMS) -- This file system is 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 -- Directing packets from source nodes to destination nodes by one or more intervening nodes. Routing nodes permit route-through. Also called PACKET SWITCHING.

ROUTING -- A network function that directs data message packets from a source node to a destination node.

ROUTING NODE -- A full routing node can forward packets to other nodes in the network and can be adjacent to all other types of nodes. A Phase III or Phase IV DECnet node that contains the complete set of transport or routing modules, and can deliver, receive, and route packets through.

SATELLITE (NODE) -- A node that depends on another node (host) for software loading and control. A satellite node has little or no peripheral equipment of its own.

SEGMENT -- The data carried in a data segment message. NSP divides the data from session control transmit buffers into numbered segments for transmission by routing.

SEGMENTATION -- Dividing normal data from session control transmit buffers into numbered segments for transmission over logical links.

SERIAL TRANSMISSION -- Transmission method in which all information is sent sequentially on a single channel, rather than simultaneously, as in parallel transmission.

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 that contains one or more servers. A server system is often delegated server functions.

SINK NODE -- A node that receives and records Network Management events.

SLAVE NODE (STATION) -- A tributary station that can send data only when polled or requested to by a master control station. In some multiplex situations, a tributary can act as both a slave and a master.

SOLICITED MESSAGES -- Normal data messages that network tasks explicitly send and receive.

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 Data Link layer protocol, this is 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 communications 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 travel in two different subchannels.

SWITCHED LINE -- A communications link for which the physical path can vary with each usage, e.g., the dial-up telephone network.

SWITCHED VIRTUAL CIRCUIT -- A temporary association between two X.25 DTEs.

SYNCHRONOUS, SERIAL DATA TRANSMISSION -- Transmission in which the data characters and bits are transmitted at a fixed rate with the transmitter and receiver synchronized. This eliminates the need for start-stop elements, thus providing greater efficiency. (Compare: ASYNCHRONOUS TRANSMISSION.)

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.

TELECOMMUNICATIONS -- Data transmission between a computing system and remote devices on another computing system.

TIME DIVISION MULTIPLEXING -- A system of multiplexing in which channels are established by connecting terminals one at a time at regular intervals by means of an automatic distribution.

TOPOLOGY -- The physical or logical placement of nodes in a computer network.

TRANSPARENT DATA -- Binary data transmitted with the recognition of most control characters suppressed. DDCMP provides data transparency because it can receive data containing bit patterns that resemble DDCMP control characters.

TRIBUTARY ADDRESS -- A master control station uses a numeric address to poll the tributary node. You can set and change these addresses during network operation. This address is interpreted by the peripheral equipment on the multipoint line to determine which hardware interface should respond. Note that the tributary address may be different from the node address and circuit number.

TRIBUTARY NODE (STATION) -- A station other than the control station on a centralized multipoint data communications system that can communicate only with the control station when polled or selected by the control station.

UNATTENDED OPERATION -- The automatic features of a node's operation that permit the transmission and reception of messages on an unattended basis.

UNIT -- The hardware controlling one channel on a multiple line controller. A unit, a controller, and associated data link modules form a station.

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 -- Used 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 that requests service from a DNA 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. For example, FILE,*;* specifies all known types and versions of all files named FILE.

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