CONTENTS

COURSE DESCRIPTION

This lecture/lab course prepares system programmers, application programmers, and system and network managers to use, install, monitor, and troubleshoot DECnet on RSX-11M/M-PLUS/S systems. This course prepares specialists to support the DECnet-RSX software. Emphasis is placed on the following topics: defining the user interface, using the network utilities, and generating a DECnet system. These topics are also emphasized: performance considerations, internal components, structures, data flow, and presales of DECnet-RSX (for SWS only). This course includes a section on writing programs that perform intertask communication using RSX-11M/M-PLUS/S supported languages.

COURSE GOALS

Upon completion of this course the specialist should be able to:

- Describe the DECnet-RSX product features.
- Use the Network Management interface to perform monitoring, managing, and testing functions.
- Perform DECnet-RSX generation and installation.
- Write programs that perform intertask communications using any of the supported RSX-11M/M-PLUS/S languages.
- Trace the DECnet-RSX internal data structures and data flows.

NON-GOALS

This course is not intended to:

- Provide support training for Ethernet or any of the products on the Ethernet Server Base. These topics are handled separately in the Ethernet Server Support course (SWS only). Customers receive Ethernet Server related information from the Local Area Network Seminar.
- Be a comprehensive network troubleshooting course.
- Teach heterogeneous network support.

PREREQUISITES

Before enrolling in this course, you should be able to:

- Discuss on a technical level, the functions and layers of the DIGITAL Network Architecture (DNA).
- Describe the characteristics and functions of the network components.
- Program in any of the supported languages available on RSX-11M/M-PLUS/S (BASIC-PLUS II, FORTRAN-77, and MACRO-11).
- Generate and support a RSX-11M/M-PLUS/S system.

In addition to the skills listed, the SWS specialist should have a minimum of six months support of RSX-11M/M-PLUS/S.

These courses may help provide the skills necessary to meet the above prerequisites:

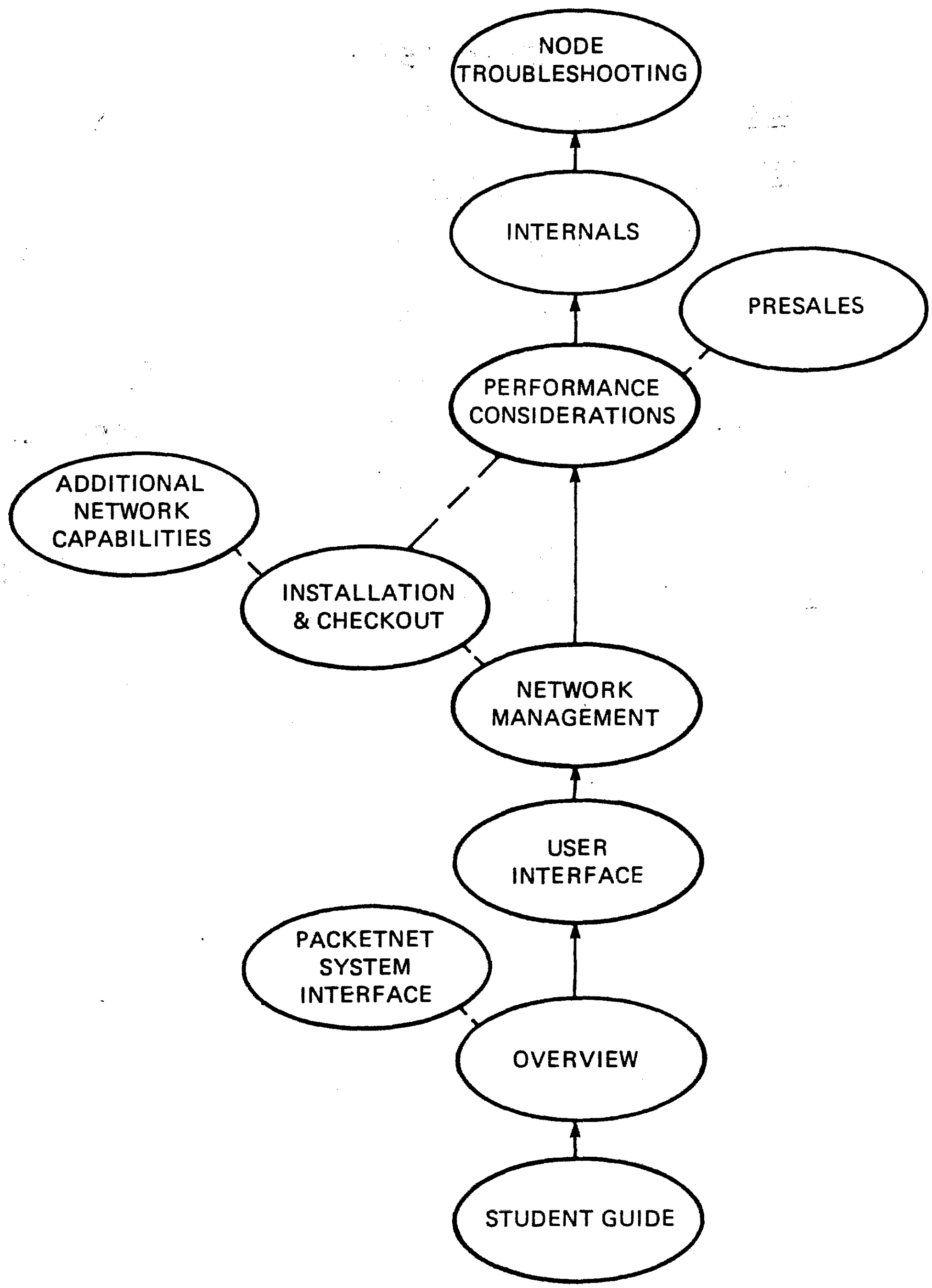
- RSX-11M/M-PLUS/S Support (J1Ø37)
- Network Concepts (SPI) (EY-1379E-SG-ØØØ2)
- DIGITAL Network Architecture (SPI) (EY-Ø149E-SG-ØØØ1)
- Communications Interfaces and Modems Workshop (J1Ø29)
- Packet Switching Concepts (SPI)
- Programming in MACRO-11 (J2Ø5Ø-A)

COURSE TOPICS

During the study of this course, you will be asked to:

- Use DECnet-RSX utilities
- Perform a DECnet-RSX generation and installation including:
  - Selection of specific components
  - Partition layout
  - SYSGEN specific items
  - Performing a PREGEN
  - Performing a NETGEN
  - Installation and Checkout of DECnet
- Use the Network Management Utilities to:
  - Configure a node
  - Configure a network
  - Test nodes and network
  - Control node and network operation
  - Inspect and monitor nodes in the network
- Use available tools to troubleshoot node problems
- Write and test intertask communication programs

COURSE MAP



MKV84-0579

COURSE MODULE BOOKLETS

You should have the following module booklets that are included for this course.

1. DECnet-RSX Overview (EY-0155E-SG-0202)
2. DECnet-RSX User Interface (EY-0155E-SG-0302)
3. DECnet-RSX Network Management (EY-0155E-SG-0402)
4. DECnet-RSX Packetnet System Interface (EY-0155E-SG-0802)
5. DECnet-RSX Installation and Checkout (EY-0155E-SG-0502)
6. DECnet-RSX Additional Network Capabilities (EY-0155E-SG-0602)
7. DECnet-RSX Performance Considerations (EY-0155E-SG-0902)
8. DECnet-RSX Internals (EY-0155E-SG-0702)
9. DECnet-RSX Node Troubleshooting (EY-0155E-SG-1002)
10. DECnet-RSX Presales (EY-0155E-SG-1102)

CONTENTS

FIGURES

TABLES

INTRODUCTION

This module presents the major features of DECnet-RSX.

DECnet is a set of software products that implement the DIGITAL Network Architecture (DNA). For RSX-11 operating systems, DECnet is implemented as a layered product. DECnet-RSX represents the following products:

- DECnet-11M, Version 4.Ø (runs on RSX-11M, Version 4.1 or later)
- DECnet-11M-PLUS, Version 2.Ø (runs on RSX-11M-PLUS, Version 2.1 or later)
- DECnet-11S, Version 4.Ø (runs on RSX-11S, Version 4.1 or later)

DECnet-RSX also represents the following products in part:

- RSX-11M PSI, Version 2.Ø (runs on RSX-11M, Version 4.1 or later)
- RSX-11M-PLUS PSI, Version 2.Ø (runs on RSX-11M-PLUS, Version 2.1 or later)

DECnet allows computers to be connected into a network either by using facilities provided by a common carrier (AT&T) or by using local available hardware interfaces (DEUNA). Either way, DECnet is transparent to a network user, because DNA layers isolate the user from the physical characteristics of the network. The physical network itself may be shared among users.

The Packetnet System Interface (PSI) allows computers to be connected by using the Packet Switching Network (PSN) which supports the CCITT X.25 set of recommendations. The PSI product can be integrated into DECnet during a NETGEN procedure and, in such a case, DECnet tasks are able to communicate with one another by using the X.25 network. For a DECnet user, the facility is transparent. As in DECnet, PSI lines may be shared among many users.

The Direct Line Access Interface (DLX) allows two tasks located on adjacent nodes to communicate. This interface is simple; a user has full control over the physical line that connects the nodes. The line cannot be shared among many users.

OBJECTIVES

To be successful in presales activity, the specialist must be able to:

- List the important features of DECnet-RSX products.
- Identify minimum hardware and software requirements for:
  - DECnet and PSI
  - DECnet
  - PSI
  - DLX
- Explain the general data flow between tasks in DECnet and/or PSI.

To be successful with DECnet-RSX, a network user must be able to:

- Specify the network interfaces available to a user.
- Identify the major features of a logical link and a virtual circuit.
- List available programming facilities.
- Identify the network layers involved in a task-to-task communication for:
  - DECnet
  - PSI
  - DLX

A network manager must be able to:

- Specify the basic features of the Network Management Layer.
- Identify the differences between:
  - Down-line system loading and task loading
  - Routing and nonrouting nodes
  - Path length and path cost
  - Circuit and line

RESOURCES

1. DECnet-RSX Programmer's Reference Manual
2. DECnet-RSX Guide to User Utilities
3. DECnet-RSX Post-Installation and Checkout Procedures
4. DECnet-RSX Release Notes
5. DECnet-RSX System Manager's Guide
6. DECnet-RSX Network Generation and Installation Guide
7. Introduction to DECnet
8. Overview of DECnet-RSX
9. DECnet DNA (Phase IV) General Description
10. RSX-11M/M-PLUS Executive Reference Manual

DECnet IMPLEMENTATIONS

DECnet is:

- An integral part or a layered product on top of the operating system (DECnet/M/M-PLUS/S is a layered product).
- A facility that connects various operating systems into a network.
- An interface between an end user and a network.

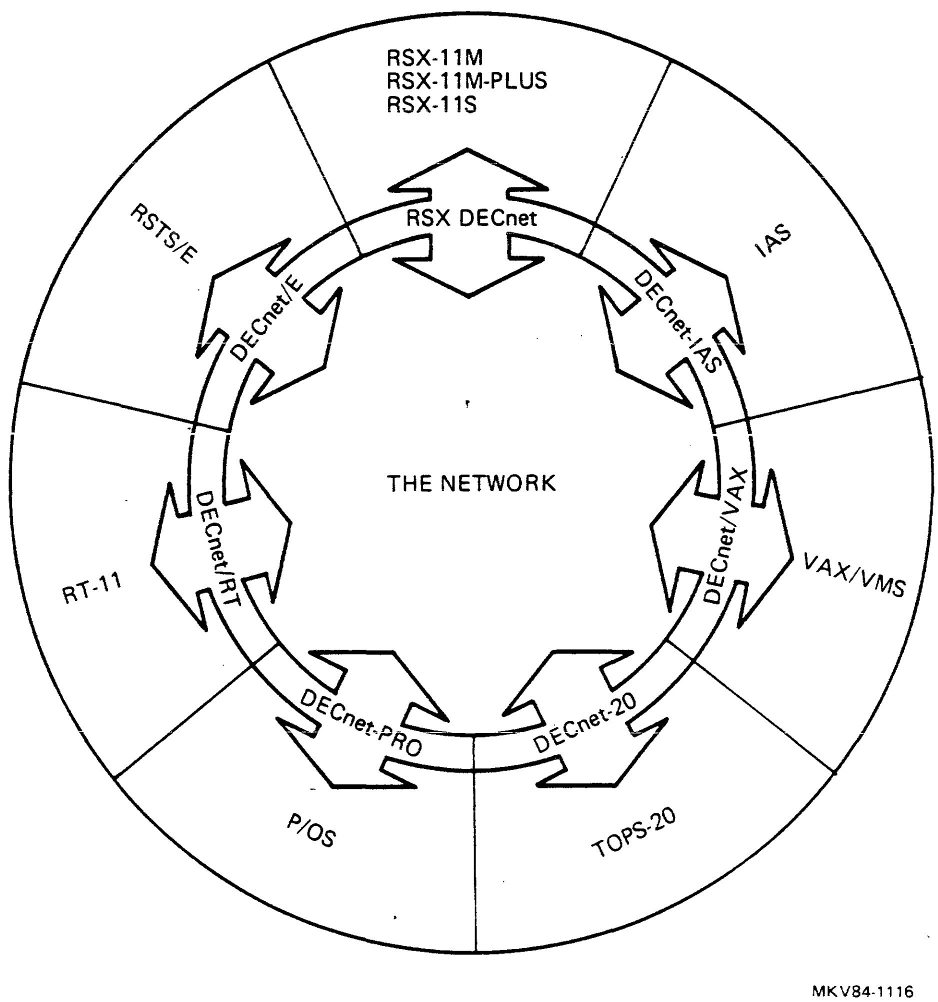


Figure 1 DECnet Implementations - Interface Between Operating Systems and the Network

DECnet FUNCTIONS

- Task-to-task communication
- Network resource access
  - File transfer
  - File access
- Network command terminal
- Down-line system loading/up-line system crash dump
- Down-line task loading/up-line task checkpointing/down-line task overlaying
- Network management
- Terminal communication
- Direct line access
- Routing
- Local area networks
- Multipoint
- Packetnet System Interface

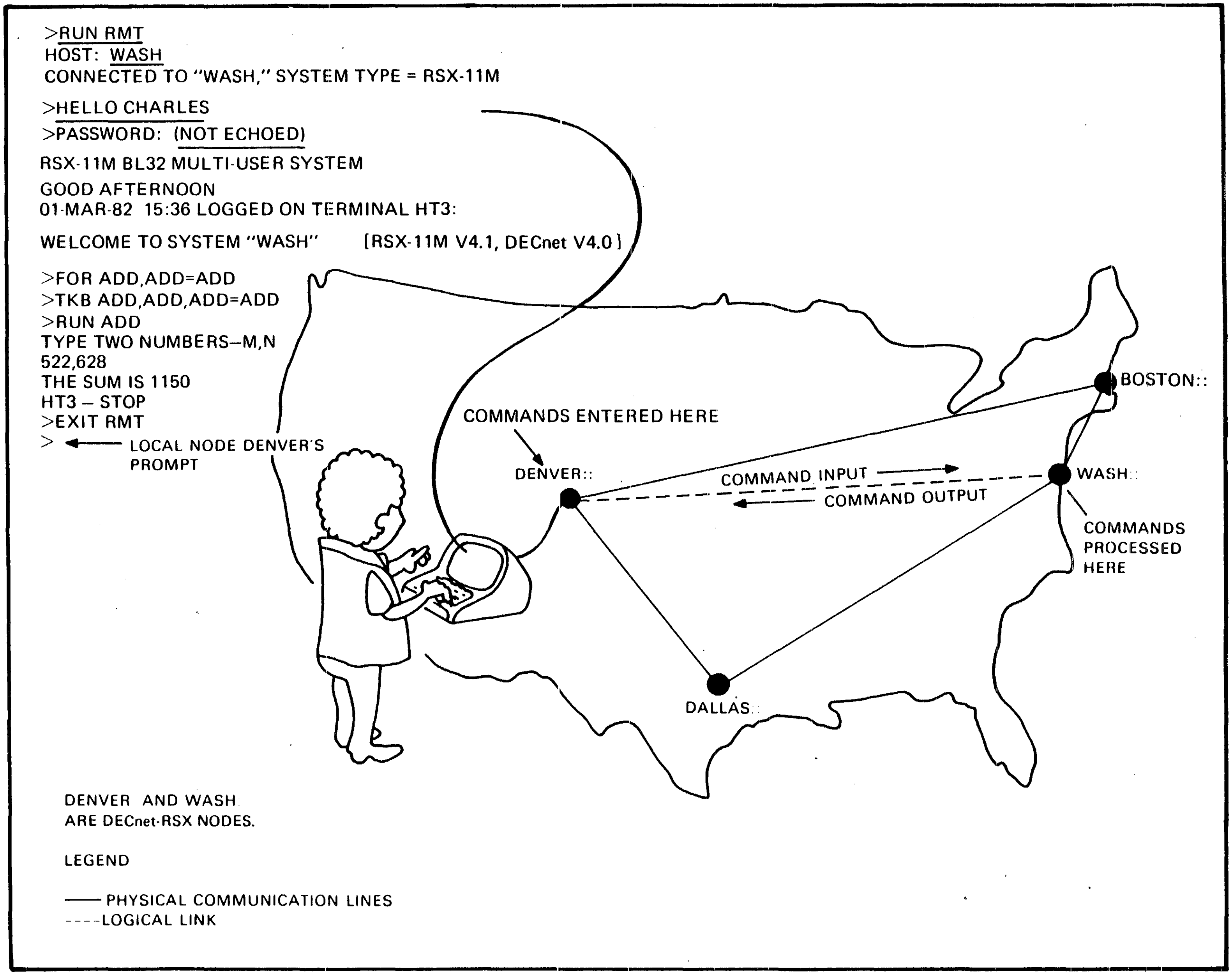


Figure 2 Network Command Terminal Usage

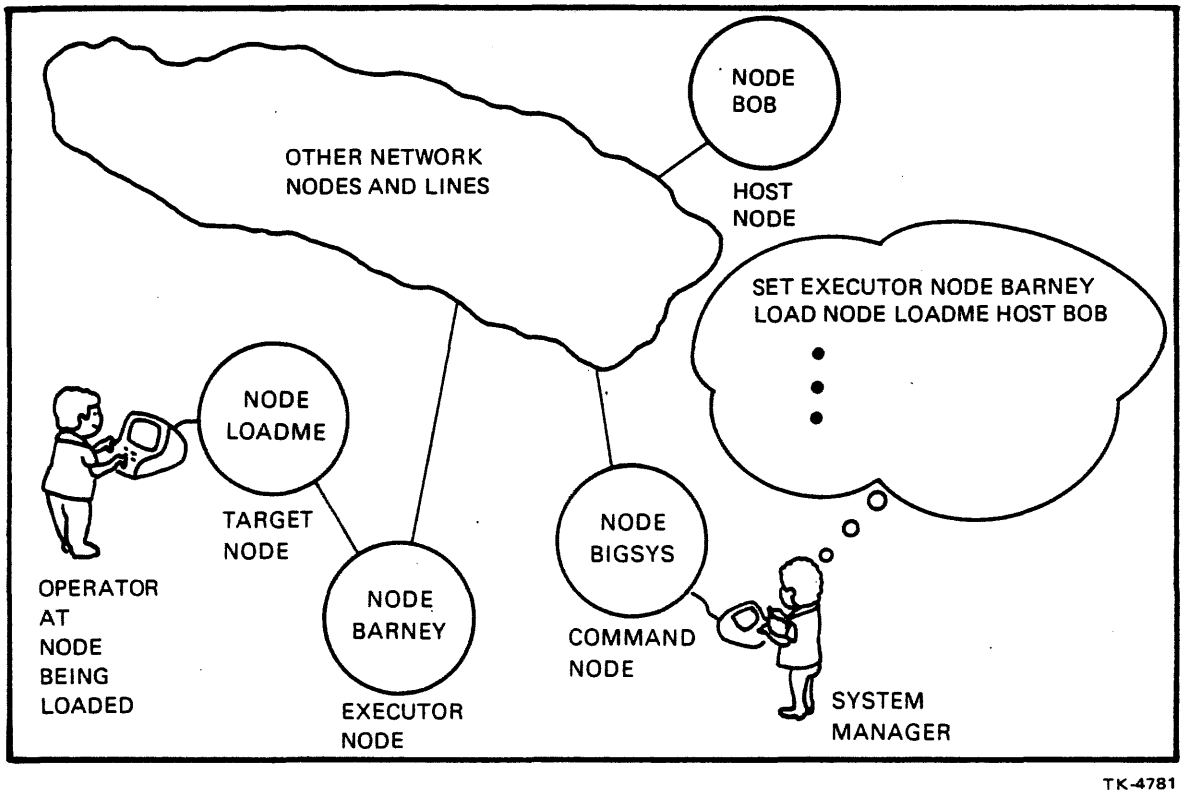


Figure 3 Down-Line System Loading Using
Network Management (NCP Interface)

DECnet LAYERS AND PROTOCOLS

- Layers are defined in the DNA
- Each layer is responsible for certain functions logically different from those supported by another layer
- Layers communicate across a network by exchanging messages according to the rules specified by a corresponding protocol
- Protocol is a set of formal rules representing a layer's logic and communication
- Protocol is transparent across a network
- Implementation of a layer, as described by a particular protocol, is logically transparent in a network
- Currently defined layers and corresponding protocols:
  - User Layer -- User-written protocol
  - Network Management Layer -- NICE protocol
  - Network Applications Layer -- DAP protocol
  - End Communications Layer -- ECL Protocol
  - Routing Layer -- Routing protocol
  - Data Link Layer -- DDCMP protocol
  - Physical Link Layer

COMMUNICATION IN A NETWORK

- Horizontal communication -- Communication between the same layers on different nodes (Figure 4)
  - It is defined by a protocol's rules
  - It is transparent across a network
  - Protocol exchanges with control messages on a time/event basis
- Vertical communication -- Communication between different layers at the same node (Figure 5)
  - It is dependent on implementation
  - Data from the higher layer ~~are~~ is "enveloped" by control information from the lower layer(s) before leaving the source node and "deenveloped" on the destination node (Figures 6 and 7)

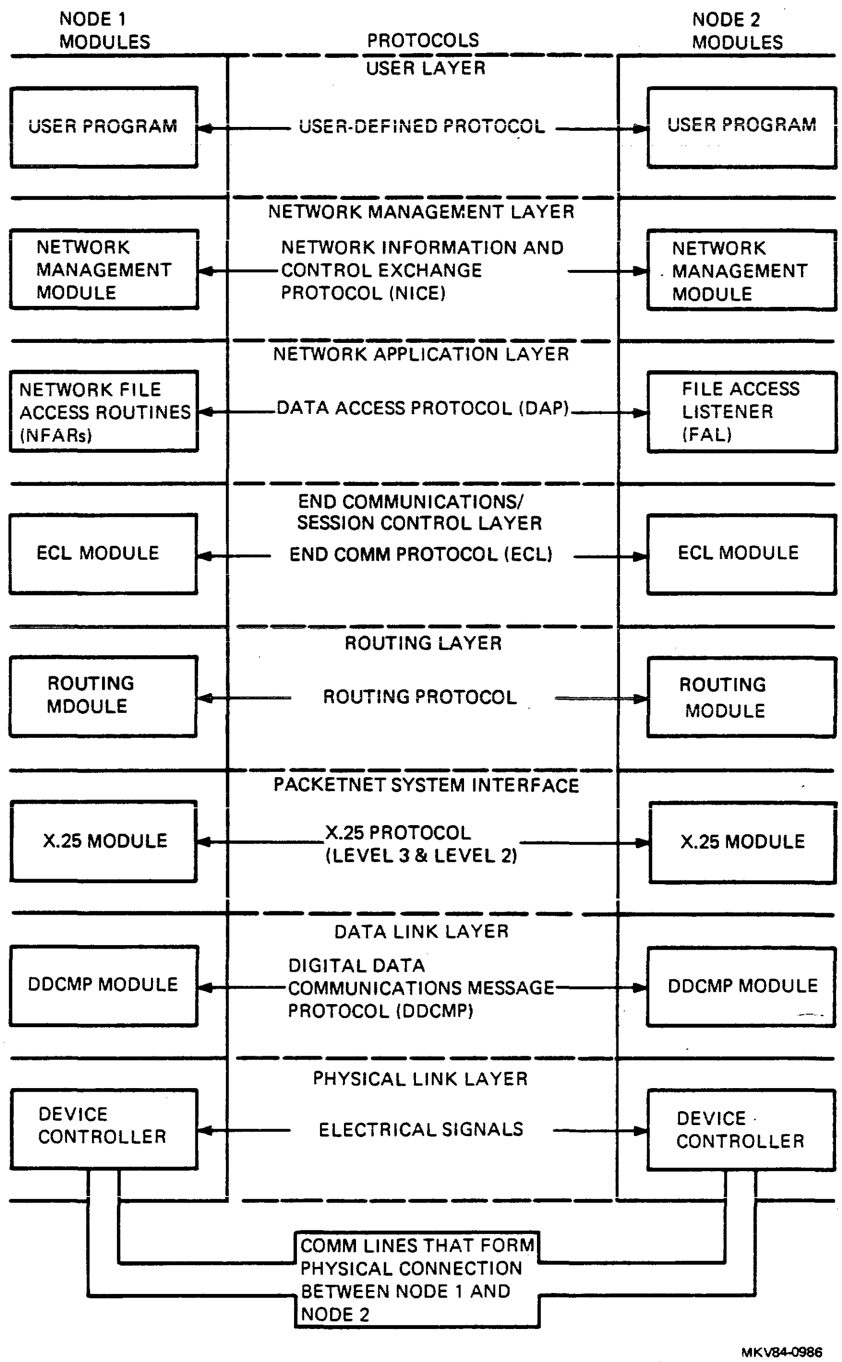


Figure 4 Horizontal Communication in a Network

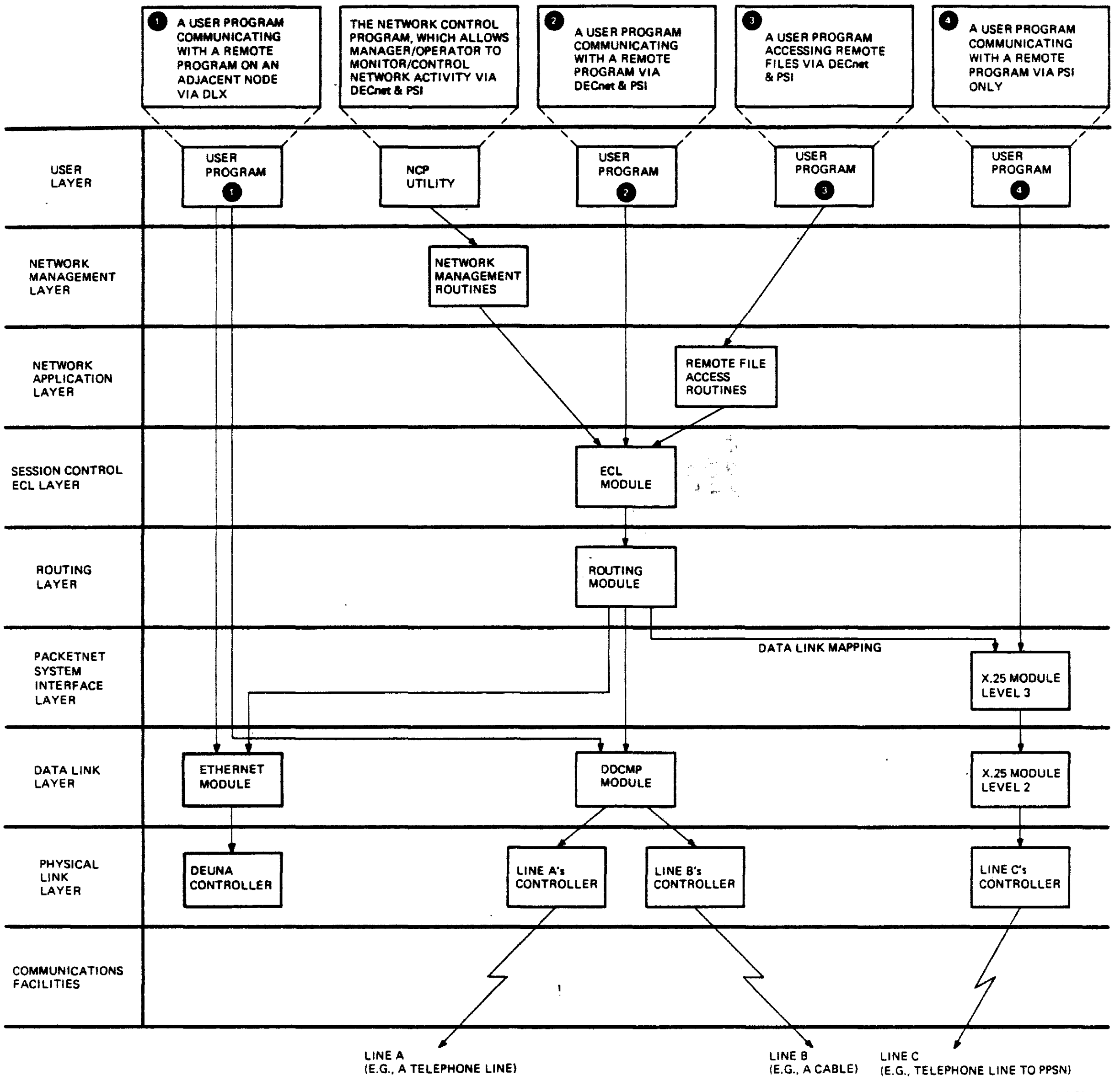


Figure 5 Vertical Communication on a Network Node

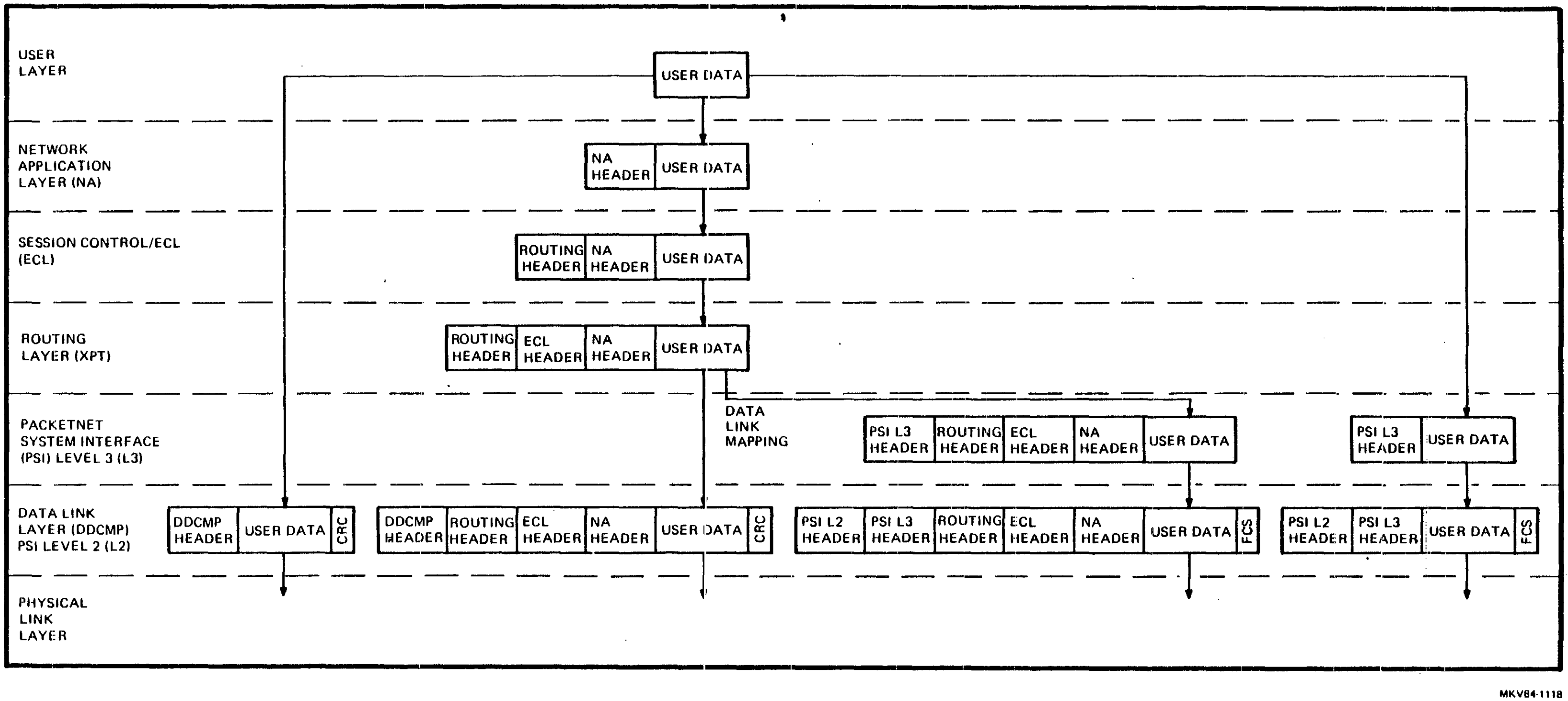
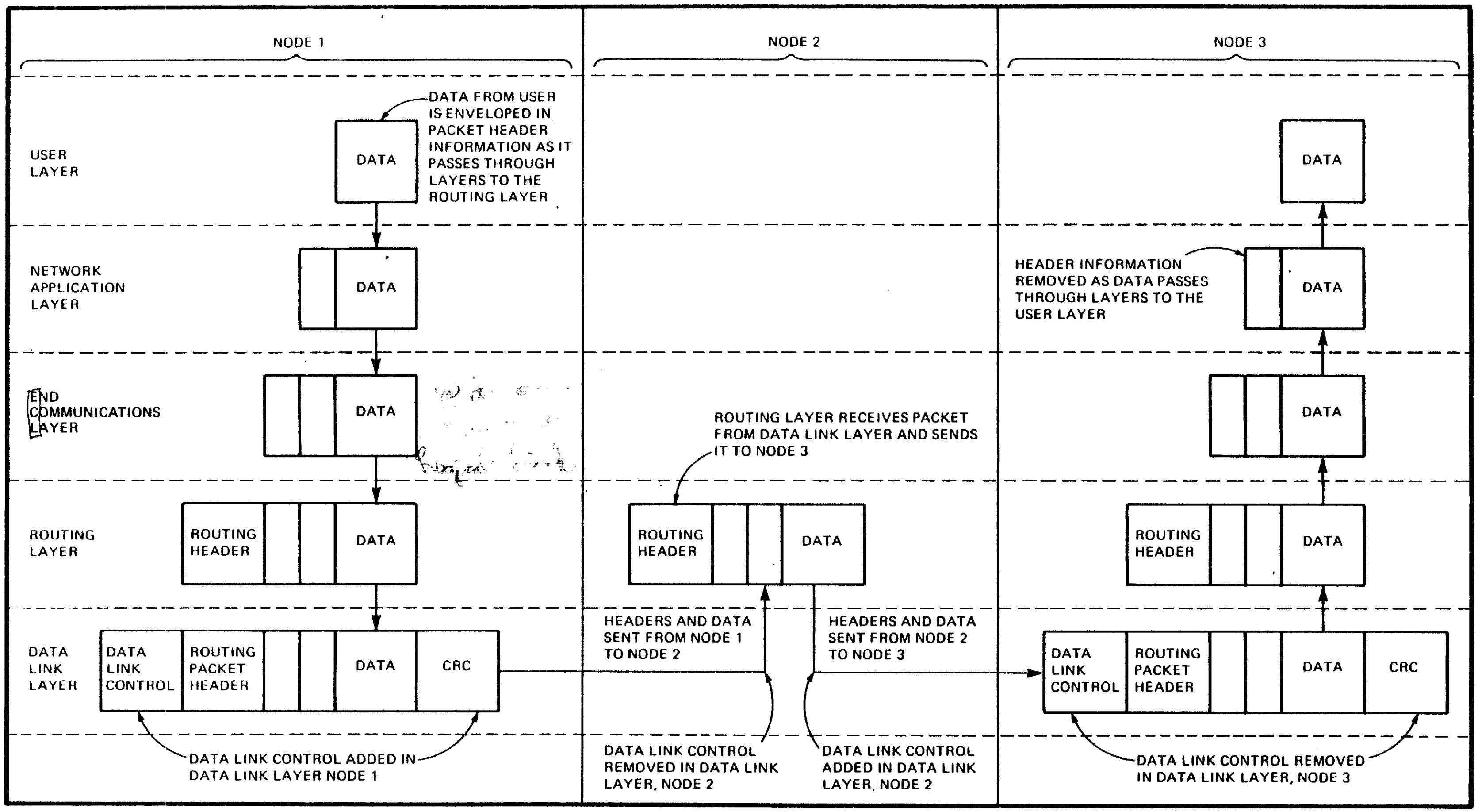


Figure 6 Enveloping User Data by Protocols



MKV84-1117

Figure 7 Data Flow from Node 1 to Node 3 (DECnet Portion of the Network)

DECnet INTERFACES

- User Interface (task-to-task communications, RMS)
- Network Interface (routing, multipoint, Ethernet)

User Interface

- Logical link by using the End Communications Layer
- Virtual circuit by using PSI
- Direct line access by using DLX facility
- User has full control over this interface

Network Interface

- User must be logged in under a privileged User Identification Code (UIC) to access this interface
- It is primarily controlled by a network

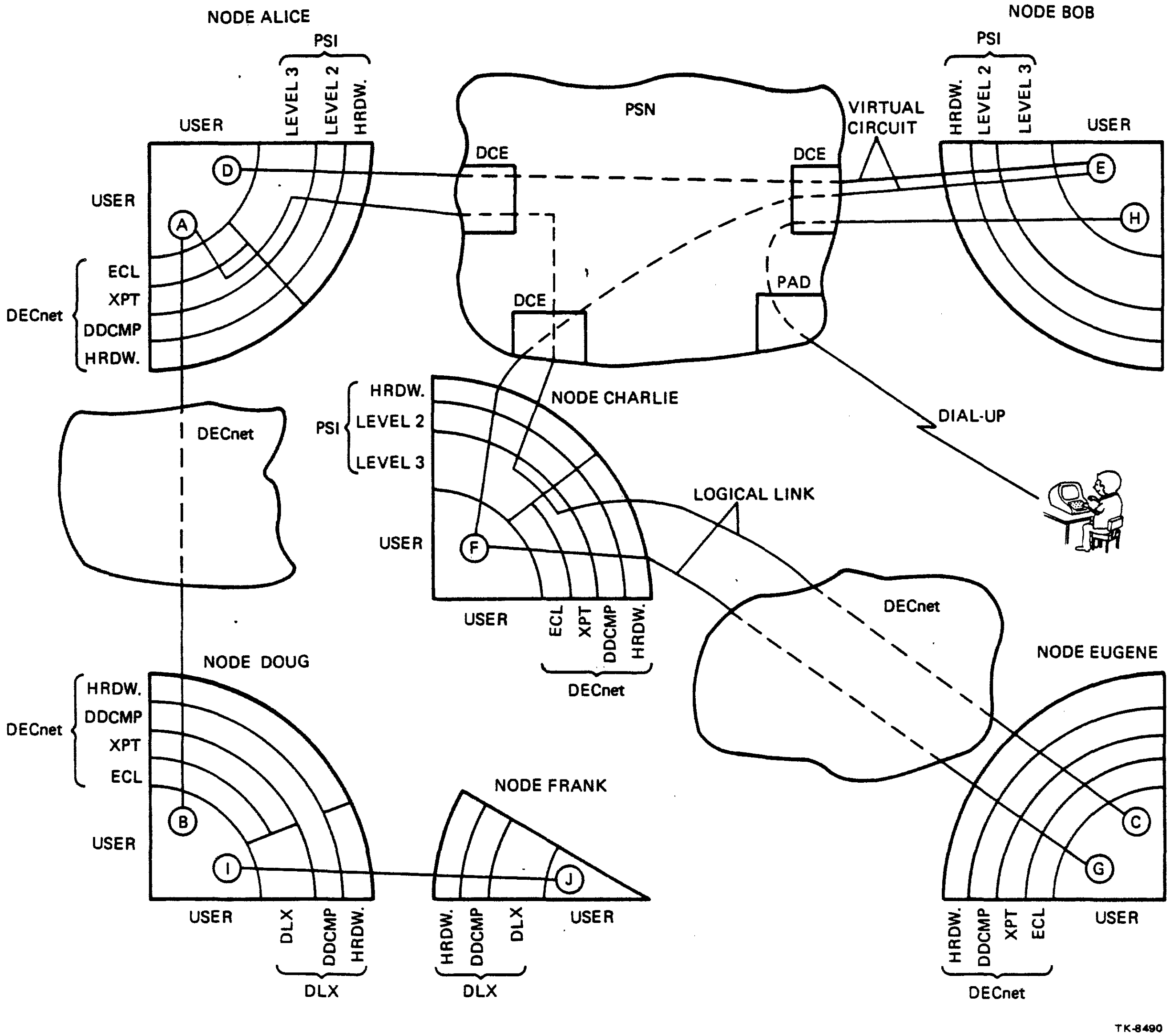


Figure 8 Task Interaction in a Network

Logical Link

- Virtual path between two user processes (tasks) in the network
- It is logically full-duplex
- Flow control is provided by the ECL layer
- User creates and accesses the logical link using a set of DECnet calls
- Usually more than one logical link can be created in a user process (task)

Virtual Circuit

- Virtual path between two user processes (tasks) in the network
- It is logically full-duplex
- Flow control is provided by the PSI layer
- User creates and accesses a virtual circuit using a set of PSI calls
- Usually more than one virtual circuit can be created/activated in a user process (task)

Direct Line Access

- It can be used in communications only between adjacent nodes
- Usually it is used for some specific applications (down-line system loading, loopback testing, and so on)
- User interface is simple, not sophisticated

Notes on Figure 8

1. Consider six nodes -- Alice, Bob, Charlie, Doug, Eugene, and Frank:

   - Nodes Alice and Charlie are DECnet and PSI nodes
   - Bob is PSI node only
   - Doug and Eugene are DECnet nodes only
   - Frank is DLX node only

2. Consider the following ten tasks (processes) -- A, B, C, D, E, F, G, H, I, and J:

   - A and D -- node Alice
   - B and I -- node Doug
   - C and G -- node Eugene
   - E and H -- node Bob
   - F -- node Charlie
   - J -- node Frank

3. A task can have more than one logical link and/or virtual circuit active at the same time (Notes 4 and 5).

4. Tasks with active logical links (DECnet):

   - A (node Alice) ⟷ B (node Doug)
   - A (node Alice) ⟷ C (node Eugene)
   - F (node Charlie) ⟷ G (node Eugene)

5. Tasks with active virtual circuits (PSI):

   - F (node Charlie) ⟷ E (node Bob)
   - D (node Alice) ⟷ E (node Bob)

6. Tasks I (node Doug) and J (node Frank) are DLX-only tasks.

7. Task H (node Bob) is not a network task; the user accesses it by using the PAD facility.

8. Data Circuit Terminating Equipment (DCE); Packet Assembler/Deassembler (PAD) facility.

9. PAD allows the start/stop terminal to access the remote computer by using the Public Packet Switching Network (PPSN).

10. For connections that go by using the PPSN, the user is paying for the number of packets (amount of data) being transferred. For connections that go by using the telephone network, the user is paying for distance and time.

Logical Link Addressing

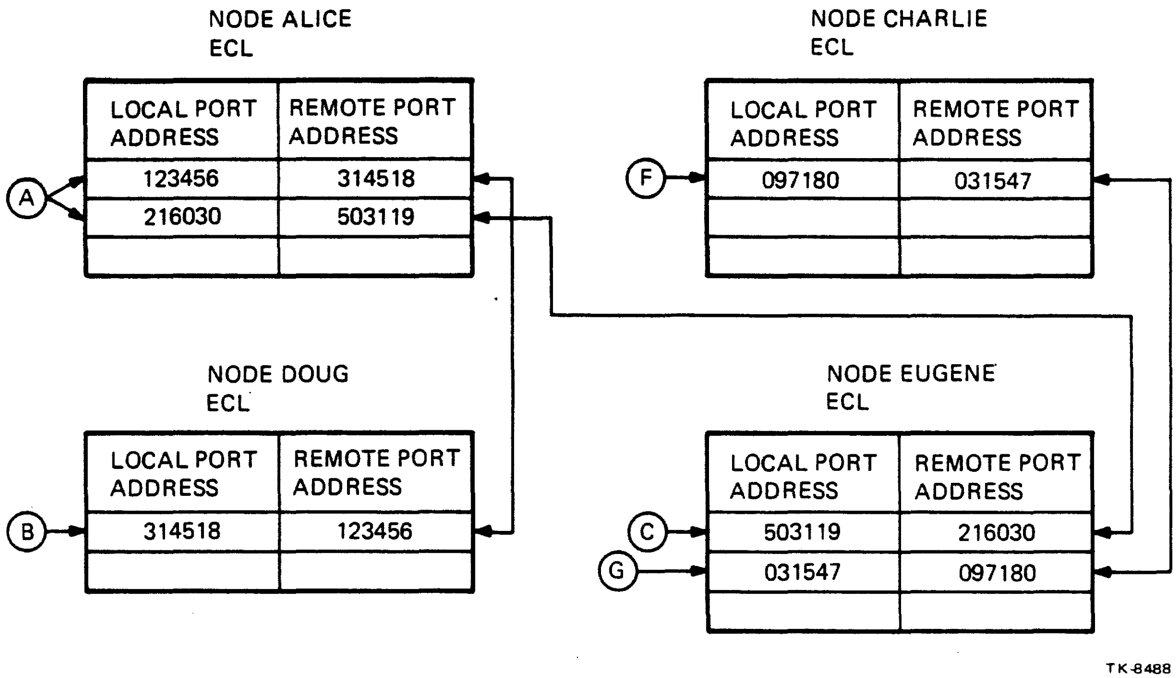


Figure 9 Details of Logical Link Addressing

Notes on Figure 9

1. The names of the nodes and tasks correspond to the names in Figure 8.

2. The logical link is identified by two port addresses assigned by ECLs during the establishing logical link procedure.

3. Each port address is 16 bits long, so each logical link is identified by 32 bits.

Virtual Circuit Addressing

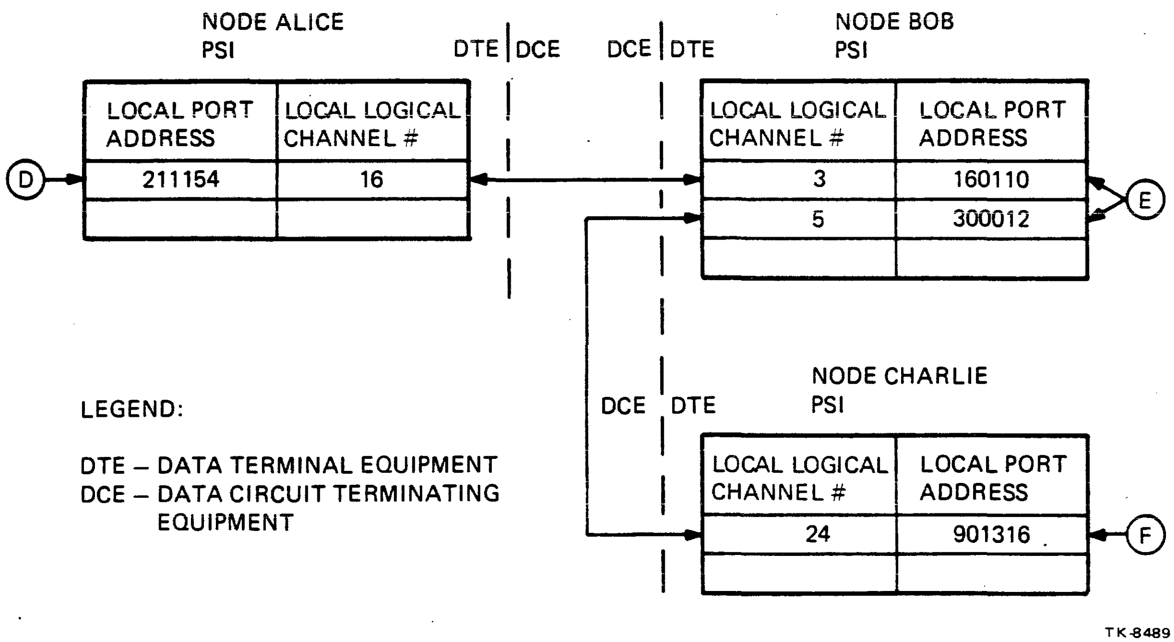


Figure 10 Details of Virtual Circuit Addressing

Notes on Figure 10

1. The names of the nodes and tasks correspond to the names in Figure 8.

2. The initiating side allocates a port and a logical channel number before sending a packet to a local DCE.

3. When the packet arrives to a remote DCE, the DCE allocates a logical channel number and gives the packet to the remote DTE; the remote DTE then allocates a port.

4. The virtual circuit is identified by the Logical Channel Numbers (LCNs) at each end.

Typical DECnet Calls for a Logical Link

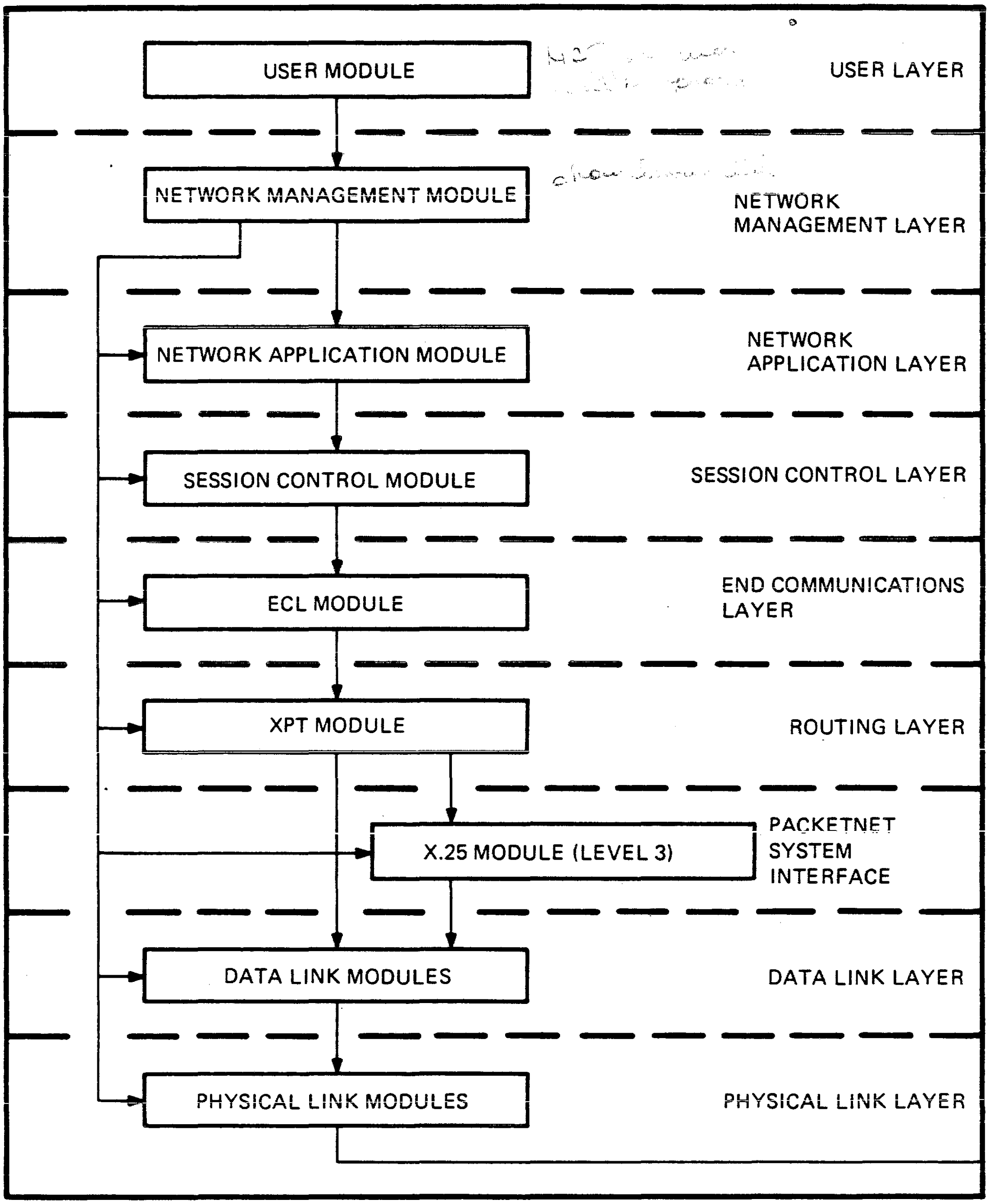
- Requesting a logical link
- Receiving a logical link request
- Accepting or rejecting a logical link request
- Sending data
- Receiving data
- Sending interrupt data
- Receiving interrupt data
- Terminating a logical link

Typical PSI Calls for a Virtual Circuit

- Requesting a virtual circuit
- Receiving a virtual circuit request
- Accepting or rejecting a virtual circuit request
- Sending data
- Receiving data
- Sending interrupt data
- Receiving interrupt data
- Terminating a virtual circuit
- Resetting a virtual circuit

Typical DLX Calls

- Controlling the line (open, close, hangup)
- Sending a message
- Receiving a message



TK-8484

Figure 11 Network Management Interface

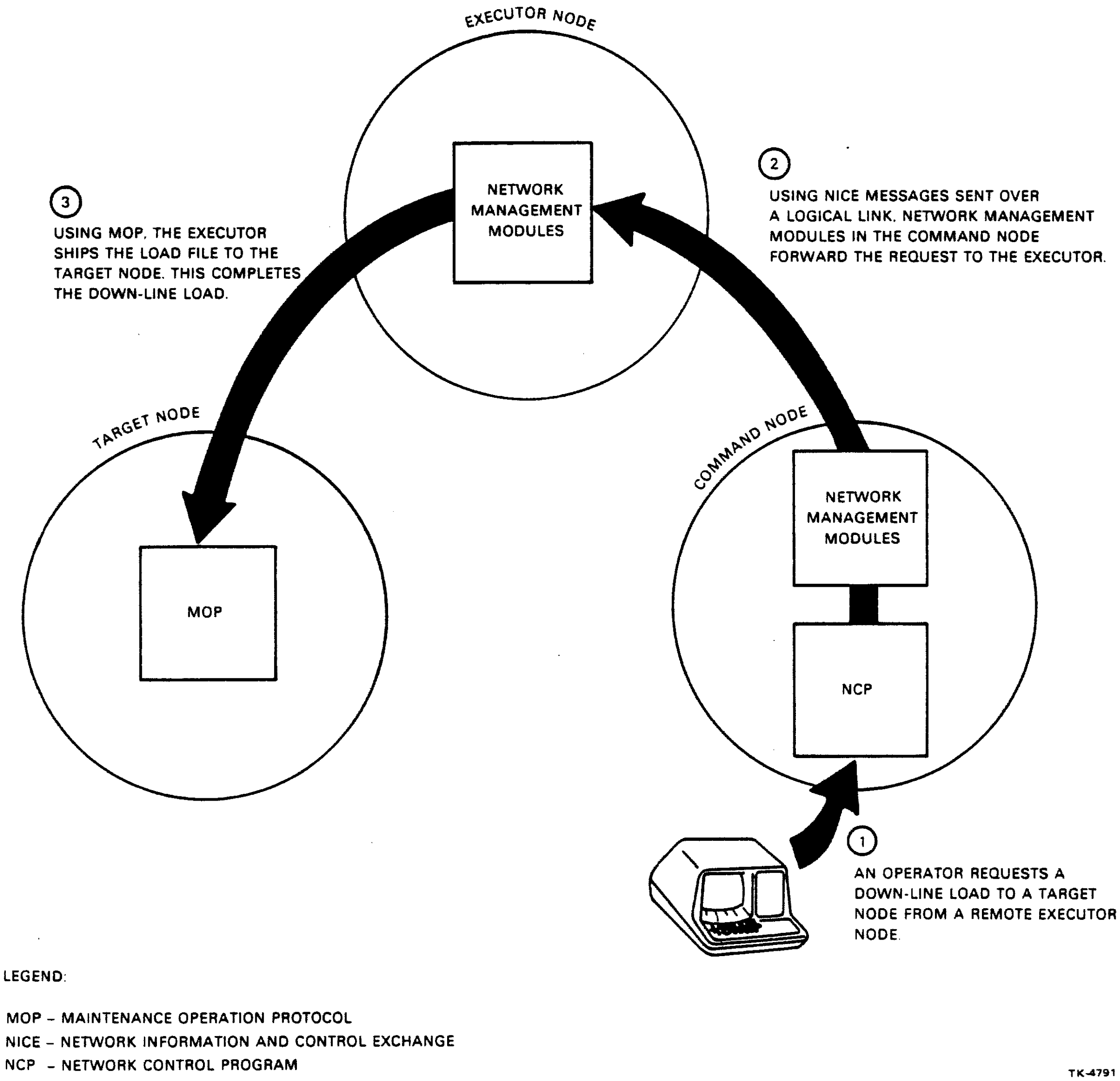


Figure 12 Down-Line Loading Initiated by a Command Node

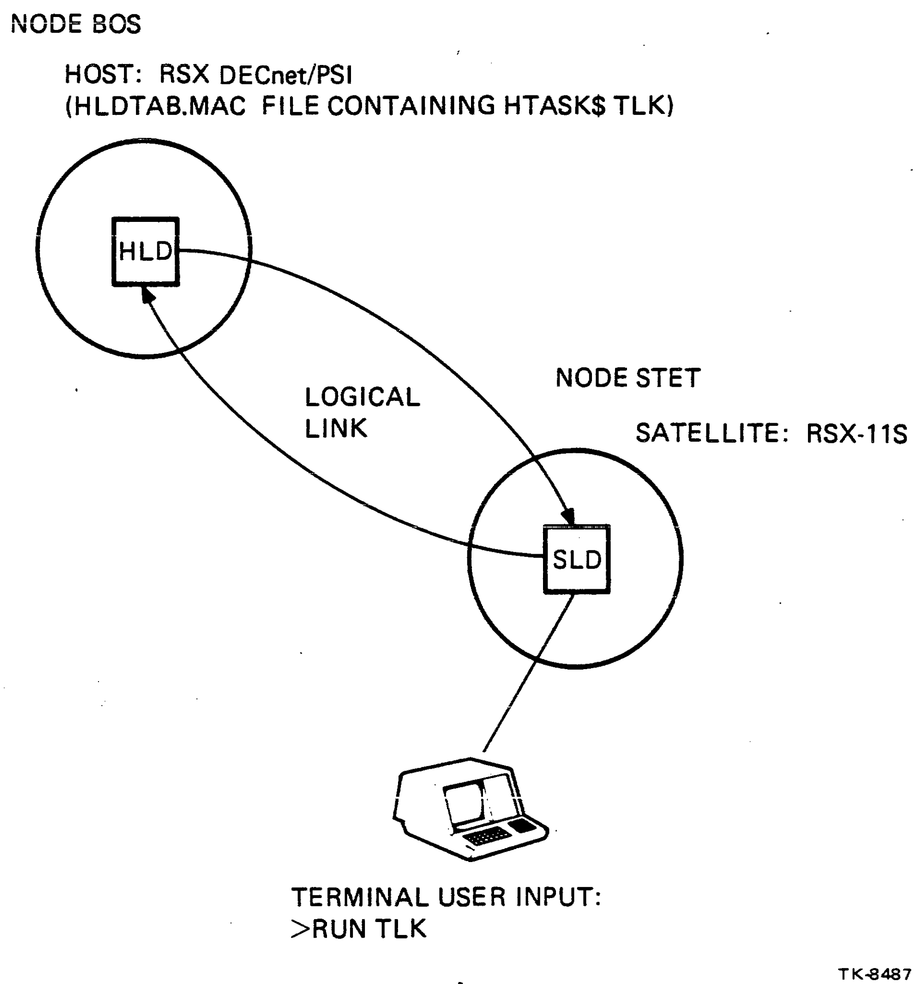


Figure 13 Down-Line Task Loading

ROUTING

Types of Nodes

- Routing IV node -- A Phase IV node that allows for packets to be routed through its lines (route-through traffic); it communicates with a node of any type. 1023

- Nonrouting IV node -- A Phase IV node that has only one physical line that connects it to the network; it does not allow for route-through traffic. It communicates with a routing or nonrouting node. 1023

- Routing III node -- A Phase III node that allows for packets to be routed through its lines (route-through traffic); it communicates with up to 255[redacted] nodes of any type. This node cannot be located on an Ethernet.

- Nonrouting III node -- A Phase III node that has only one physical line that connects it to the network; it communicates with routing or nonrouting nodes whose node addresses are below [redacted].255 This node cannot be located on an Ethernet.

Figure 14 presents a possible topology for a DECnet network.

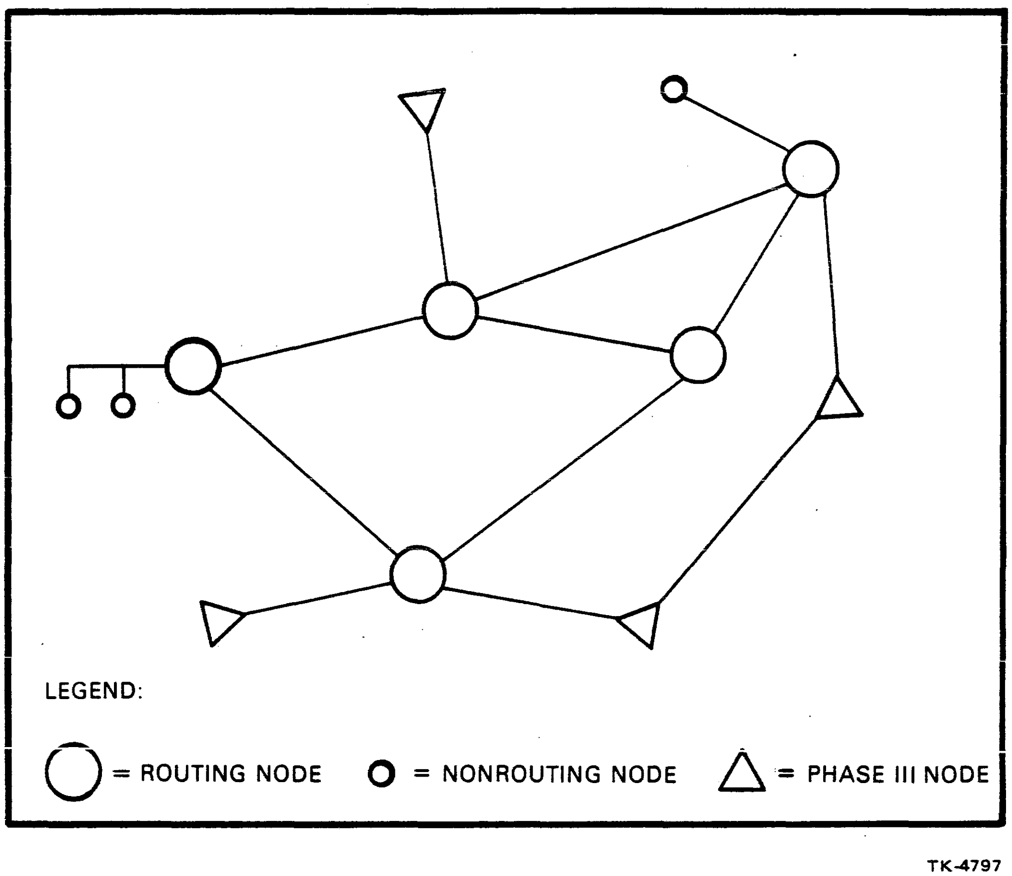
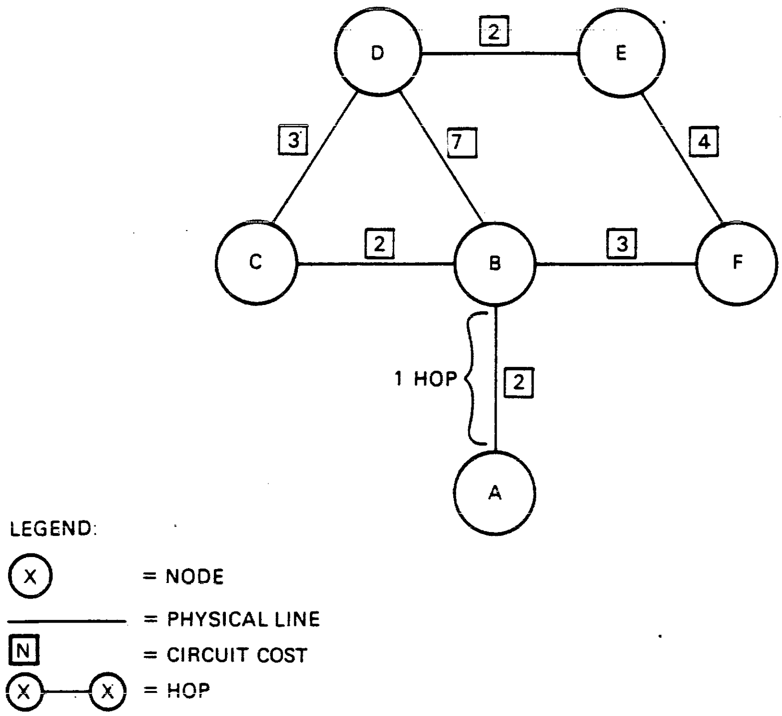


Figure 14 Example of a Network Routing Topology

Routing Terminology

- Hop -- The logical distance between two adjacent nodes
- Path -- The route a packet takes from a source node to a destination node
- Path length -- The sum of hops along the path
- Cost -- An arbitrary integer value assigned to a circuit
- Path cost -- The sum of costs along the path
- Reachable node -- The destination node to which a usable path exists in a network

Cost is used in the routing algorithm to determine the best path for a packet. The transport routes packets on the path of least cost, even if it is not the path with the fewest hops.



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 + 3 + 4 + 2 = 11	4 HOPS

*7 IS THE LOWEST PATH COST; NODE A THEREFORE ROUTES THE PACKET TO NODE D VIA THIS PATH.

TK-4784

Figure 15 Routing Terms

MULTIPOINT CAPABILITIES

A multipoint line is a single communications line connected to more than two nodes. (A line connecting only two nodes is a point-to-point line.)

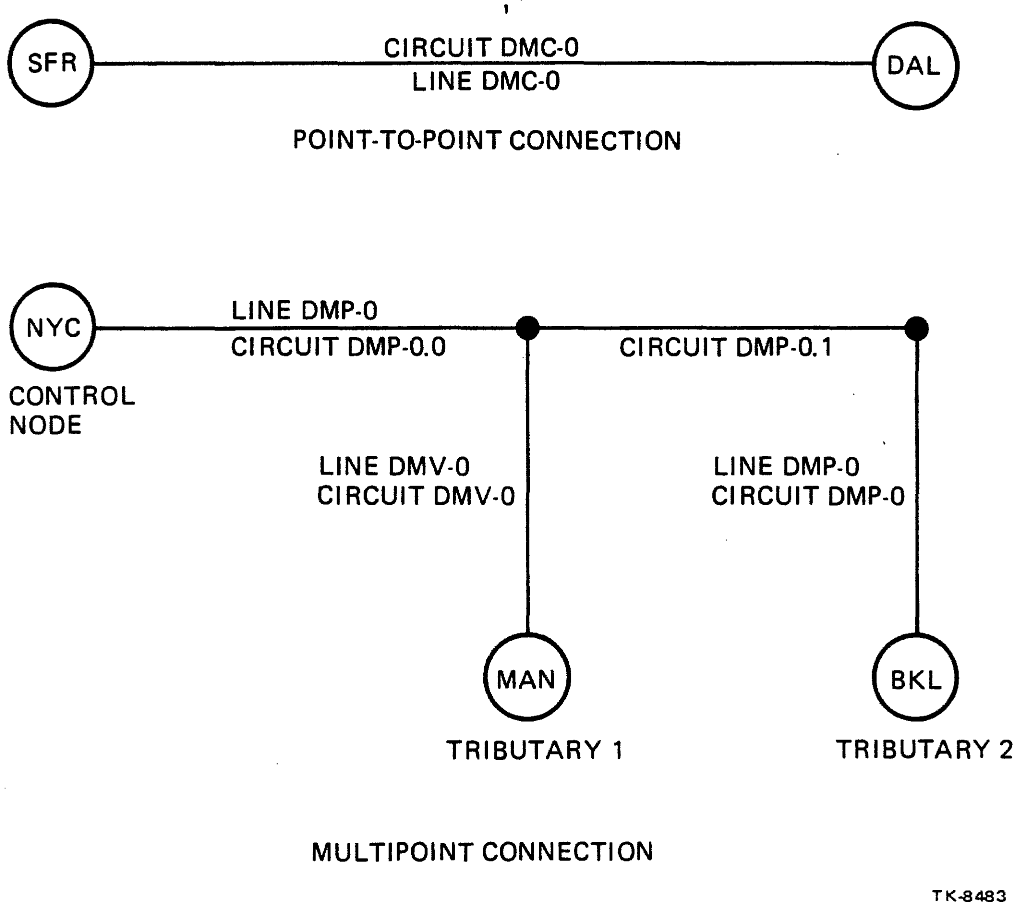


Figure 16 Circuits and Lines

ETHERNET

Ethernet is a local area network that provides a communications facility for high-speed data exchange among computers and other digital devices located within a moderately sized geographic area. It is intended primarily for use in such areas as office automation, distributed data processing, terminal access, and other situations requiring economical connection to a local communication medium carrying traffic at high-peak data rates.

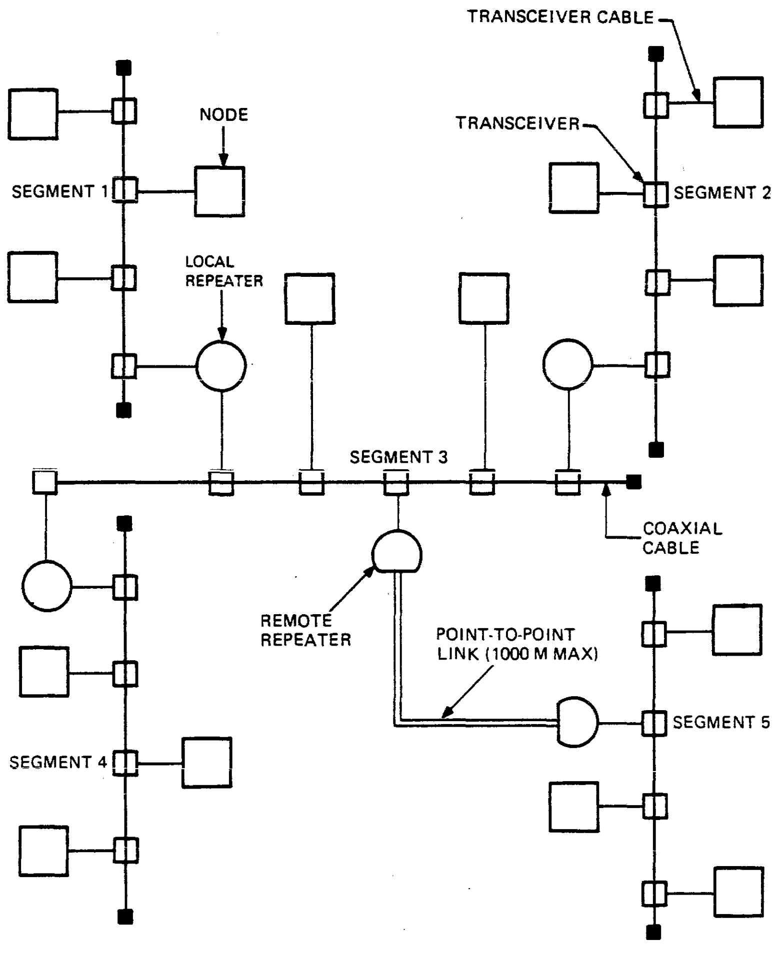


Figure 17 Typical Large-Scale Ethernet Configuration

EXERCISES

1. Match the available network features to an appropriate product. Each item may be used once, more than once, or not at all.

Features

a. Ethernet _____

b. Down-line system loading (host) ____

c. Network management ____

d. Task-to-task communication ____

e. Direct line access ____

f. Network command terminal ____

g. Remote task control ____

h. Remote file access ____

i. Multipoint ____

j. Terminal communication ____

k. Remote terminal access via X.25 network _____

l. File transfer _____

m. PSN access via X.25 ____

n. Down-line system loading (satellite) ____

Products

1. DECnet-11M, Version 4.1
2. DECnet-11M-PLUS, Version 2.1
3. DECnet-11S, Version 4.1
4. RSX-11M PSI, Version 2.Ø
5. RSX-11M-PLUS PSI, Version 2.Ø

2. Select the best answers for the following statements.

   1. The Ethernet Local Area Network is supported under:

      a. Phase III DECnet

      b. Version 2.Ø of PSI

      c. Phase III and IV implementations of DECnet

      d. None of the above

   2. Down-line system loading is used with:

      a. The Packetnet System Interface

      b. The RSX-11S operating system

      c. The communications server product

      d. Micro PDP-11 systems

   3. The DECnet-RSX product is considered a:

      a. Layered product

      b. Part of the RSX-11 operating system

      c. Separate operating system

      d. All of the above

SOLUTIONS

1. Match the available network features to an appropriate product. Each item may be used once, more than once, or not at all.

Features

a. Ethernet 1,2,3

b. Down-line system loading (host) 1,2

c. Network management 1,2,3,4,5

d. Task-to-task communication 1,2,3,4,5

e. Direct line access 1,2,3

f. Network command terminal 1,2,3

g. Remote task control 1,2,3

h. Remote file access 1,2,3

i. Multipoint 1,2,3

j. Terminal communication 1,2,3

k. Remote terminal access via X.25 network 4,5

l. File transfer 1,2,3

m. PSN access via X.25 4,5

n. Down-line system loading (satellite) 3

Products

1. DECnet-11M, Version 4.1
2. DECnet-11M-PLUS, Version 2.1
3. DECnet-11S, Version 4.1
4. RSX-11M PSI, Version 2.Ø
5. RSX-11M-PLUS PSI, Version 2.Ø

2. Select the best answers for the following statements.

   1. The Ethernet Local Area Network is supported under:

      a. Phase III DECnet

      b. Version 2.0 of PSI

      c. Phase III and IV implementations of DECnet

      d. None of the above

   2. Down-line system loading is used with:

      a. The Packetnet System Interface

      b. The RSX-11S operating system

      c. The communications server product

      d. Micro PDP-11 systems

   3. The DECnet-RSX product is considered a:

      a. Layered product

      b. Part of the RSX-11 operating system

      c. Separate operating system

      d. All of the above

4. A DECnet-RSX node can exist in the following configurations:

   a. DECnet only

   b. PSI only

   c. DECnet and PSI

   d. All of the above

5. When user data from a DECnet task is sent over the PPSN, the following headers will be in the packet:

   a. ECL header

   b. DDCMP header

   c. X.25 Level 3

   d. Routing header

CONTENTS

FIGURES

TABLES

EXAMPLES

INTRODUCTION

This module presents the user interface in detail.

The user interface for DECnet includes a set of utilities and programming facilities. Each utility described in this module is implemented as a pair of tasks (requester and server) that communicate by using a logical link. Any server task must be installed on the remote node before a requester can actually access it.

The utilities discussed in this module are:

- Terminal Communications (TLK) Utility -- Allows the user to send single messages and text files to a remote terminal.
- Network File Transfer (NFT) Utility -- Provides functions such as accessing a directory located on a remote node, transferring files between nodes, executing a command file on a remote node, and so on.
- File Transfer Spooler (FTS) Utility -- Spools NFT-like requests to a local RSX-11 queue manager. The queue manager is responsible for giving such a network job to an FTS despooler, which establishes a logical link to a remote file access server.
- Remote Command Terminal (RMT) Utility -- Logically accesses the remote DECnet-11M/M-PLUS node as a terminal user.

Programming facilities for DECnet include:

- **Task-to-task communication** -- A user creates his own requester and server tasks that establish a logical link to one another. Such a facility is also available for PSI (tasks create a virtual circuit rather than a logical link) and DLX (two tasks communicate with each other through a preassigned physical line).
- **Remote file access** -- A user creates only a requester task to communicate by using a logical link with the standard remote file access server.
- **Remote task control** -- A user writes a requester task to establish a logical link to a standard task control server.

This pagination is correct if there is

 ← no half-title pg.

right-hand pg. odd-numbered. pg.

OBJECTIVES

To be successful with the user interface, the user must be able to:

- Use the command syntax for TLK, NFT, FTS, and RMT fluently.
- Identify key concepts for task-to-task communication programming for logical links and virtual circuits.
- List the languages supported for the interface.
- Interpret the programming examples in the reference literature.
- Program simple requester and server tasks for task-to-task communication over a logical link.

RESOURCES

1. <u>DECnet-RSX Guide to User Utilities</u>
2. <u>DECnet Programmer's Reference Manual</u>

Video Screen Dialogue

To support the TLK video screen, the IO.WBT,IO.RPR and IO.ATA terminal functions must all be supported by your target system. The video mode option need not be supported by the target node; the messages appear on the remote terminal like a normal dialogue session.

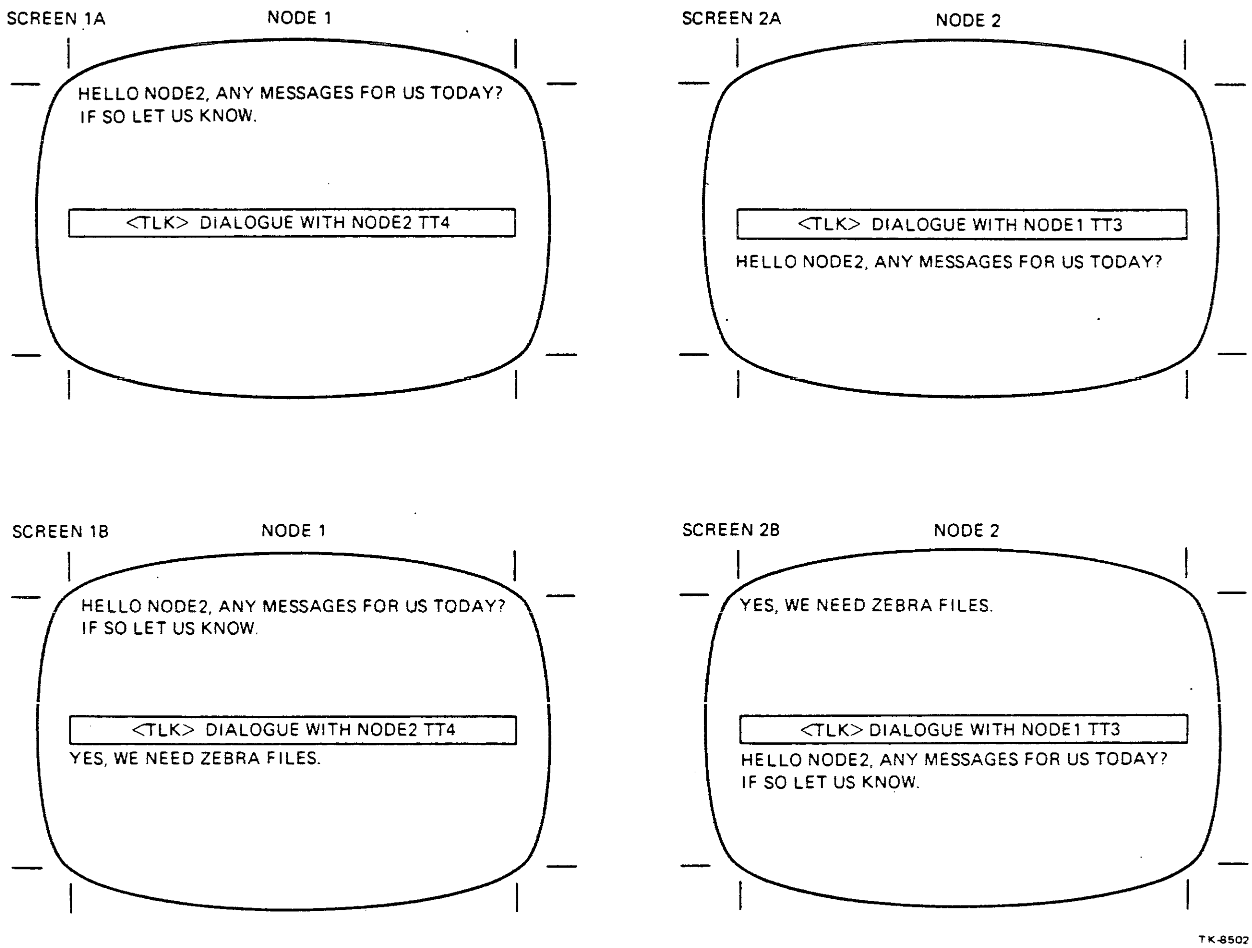


Figure 1 TLK Usage in Video Screen Dialogue Mode

Command File Support

In Example 3:

- The contents of TAPMOU.CMD are displayed on the local terminal if the Trace Switch (/TR) is specified.
- Indirect command files are not supported by DECnet/S.

```
>TLK @TAPMOU/TR

TLK> NODEA::TT2: 'MOUNT A DECTAPE PLEASE.
TLK> TT4: 'THIS IS A LOCAL MESSAGE.
TLK> NODEB:: 'TO NODEB'S CONSOLE TERMINAL.
```

Example 3 Executing the TLK Single Message Command File

In Example 4:

- TLK allows you to execute two types of command files for dialogue mode messages:
  - Command files that contain the actual list of dialogue messages
  - Command files that execute other command files
- If a command file uses dialogue mode; dialogue mode is nonvideo regardless of the terminal type.

File DISTLIS.CMD contains the lines:

```
NODER::TT3:
@NOVACAN.CMD
```

File NOVACAN.CMD contains the messages:

```
ALL MORNING FLIGHTS TO MULGRAVE,
NOVA SCOTIA HAVE BEEN CANCELED
CALL OUR CANADIAN OFFICE AFTER
6 P.M.  TO OBTAIN INFORMATION ON
TOMORROW'S FLIGHTS.
```

When executing DISTLIS.CMD, NOVACAN.CMD is also executed.

Example 4 Executing the TLK Dialogue Command File

Command Switches

TLK allows you to specify /TR for displaying the contents of a command file on the local terminal (Example 4).

Exit with Status

TLK exits with a status code that may be useful in these circumstances:

- An indirect command file
- A batch stream job (for RSX-11M-PLUS systems only)
- When spawned by a parent task or connected to a parent task

There are two exit status codes that TLK can issue:

Status	Code
TLK exited without error	1 (EX$SUC)
TLK exited with an error	2 (EX$ERR)

NOTE

If any command in an indirect command file fails, an exit with a status code 2 (EX$ERR) is issued when TLK exits.

You can test the status code returned by TLK and, on the basis of the results, specify alternative procedures.

TLK Components

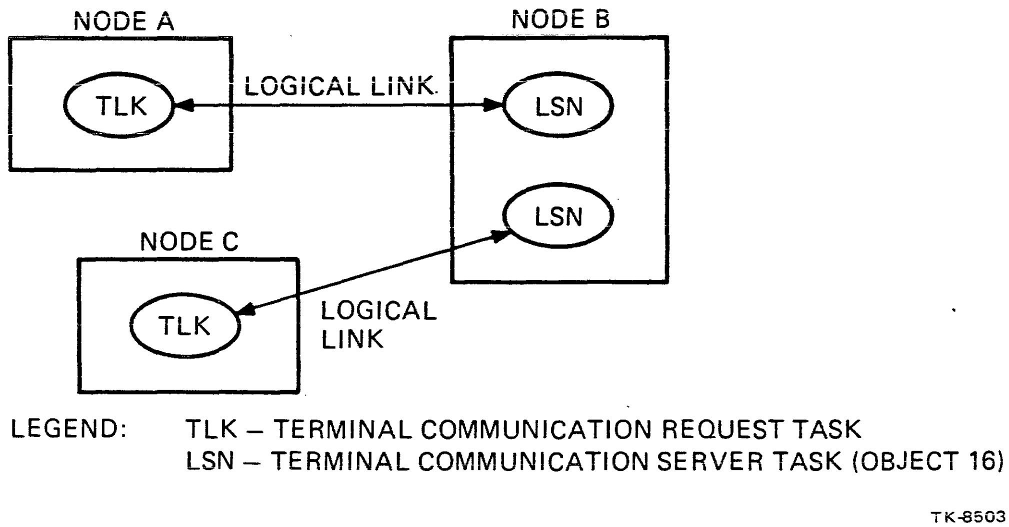


Figure 2 TLK Utility Components

Network File Transfer (NFT) Utility

- NFT features
  - Transfer files between two nodes
  - Delete files on a remote node
  - Execute command files stored on a remote node
  - Transfer local command files to a remote node for submission to the remote command file processor or batch system for execution and subsequent deletion
  - List remote node directories
  - Spool files to a line printer
  - Append files to an existing file
  - Rename a file
  - Change a file's protection status
- Node specification
- File specification
  - Defaults
  - Foreign files
  - Wildcards
- File transfer types
  - Single-file transfer
  - File transfer with concatenation
  - Multiple-file transfer
- Command switches
- Exit with status

NFT Examples

Examples 5 through 9 are explained individually on the following pages:

```
NFT> NODEX/[200,30]/DECNET/10::DT0:[40,10]MAGIC.MAC;3=-
NFT> NODEY"DECNET DEMO"::SY0:[DDP]MYSTIC.MAC;2
```

Example 5 Single-File Transfer

```
NFT> FAL.LOG=BERGIL/MIL/RITCH::"MIL$DISK:FAL.LOG;2"
```

Example 6 Single-File Transfer Using a Foreign File

```
NFT> ELROND/FOO/BAR::DB0:[100,1]=DM1:[*,10]FIL%%.MAC
```

Example 7 Multiple-File Transfer Using Wildcards

```
NFT> WASH::DB0:[200,200]NAMES.CMD/EX
```

Example 8 Using the /EX Switch for Executing a Command File

```
>NFT @filename
```

Example 9 Using a Command File Containing the NFT Command

we need for " " with NFT [illegible]

Node Specification

A node specification consists of the name of the node and optional access control information for that node, followed by two colons:

```
nodename[access control]::
```

Access Control Information

Access control information consists of three position-dependent fields appended to a node name and delimited by slashes (/) or spaces ( ):

```
nodename/userid/passwd/[accnt]::

     or

nodename"userid passwd [accnt]"::
```

where:

userid	is a 1- to 16-character string identifying the user at the remote system
passwd	is a 1- to 8-character string password needed to gain access to the remote file system
accnt	is a 1- to 16-character string used to specify an account at the remote system. This is an optional parameter and only implemented on systems with accounting installed.

Examples of valid node specifications appear as:

```
ELROND/[7,7]/SECRET/WORD::
SAM/5,10/LEFT::
BOSS/EVERY/ONE::
HALDIR"DECNET FOO"::
```

File Specification

- Defaults
- Foreign files
- Wildcards

NFT commands operate on file descriptors that identify files in the network. A file descriptor consists of a node specification and a file specification. File descriptors are in the form:

[node specification::][file specification]

Depending upon its position in a command string, a file specification is either a source or a destination file specification. Furthermore, the node names define the file specification as local or remote. Therefore, NFT files can fall into four categories:

- Local destination
- Remote destination
- Local source
- Remote source

Table 1 Summary of DECnet-RSX NFT File Descriptor Defaults

Field	Conditions	Default Value
nodename	The file is the first or only file in the input list or the output list.	Local node name.
	The file is a subsequent file in a given list.	Preceding node name specified in the list (including access control information).
userid	Userid is preassigned using an alias node name.	Userid value specified with the alias node name.
	A node name with access control information is specified to NFT using the Default Switch (/DF).	Value of userid given for the node name.
passwd	The passwd value is preassigned using an alias node name.	Passwd value specified with the alias node name.
	A node name with access control information is specified to NFT using the /DF switch.	Value of passwd given for the node name.
accnt	The accounting number is preassigned using an alias node name.	Accnt value specified with the alias node name.
	A node name with access control information is specified to NFT using the /DF switch.	Value of accnt given for the node name.

Table 1 Summary of DECnet-RSX NFT File Descriptor Defaults (Cont)

Field	Conditions	Default Value
dev	The file is the first or only file in an input list or an output list associated with a particular node.	SY device associated with the access control given with the specified node name. If no node name is given, it defaults to the user's current SY device.
	The file is a subsequent file in a given list, as previously defined.	Preceding device specified in the list.
	A device is specified to NFT using the /DF switch.	Device specified in the /DF command.
uic	The file is the first or the only file in an input list or an output list associated with a particular node.	Uic associated with the access control given with the specified node name. If no node name is given, it defaults to the user's current Uic.
filename	The file is the first or only file in the input list.	None
	The file is a subsequent file in the input list.	Preceding file name specified in the list.
	The file is an output file.	Name of the corresponding input file.
type	The file is the first or only file in the input list.	None
	The file is a subsequent file in the input list.	Previous type specified in the list.
	The file is an output file.	Type of corresponding input file.

Table 1 Summary of DECnet-RSX NFT File Descriptor Defaults (Cont)

Field	Conditions	Default Value
ver	The version number is omitted for any input file.	Highest version of file.
	The version number is omitted for any output file.	If a file name or type was specified for the output file, the highest version number is used. If no file name or type was specified for the output file, the version number of the corresponding input file is used.

Foreign Files

Files that reside on non-RSX nodes are referred to as foreign files. Since foreign files must use syntax that is compatible with the systems on which they are located, the file specification should be enclosed in double quotes (" "). This directs NFT to transmit the file specification to the foreign node without checking its syntax or applying defaults for missing fields. Note that using double quotes may also be used to override defaults.

For example, the following command line causes a file transfer from a VAX node to the local node. You must specify an output file name, since NFT does not decode the input file name when surrounded by quotes.

```
NFT> ZOOK"RYAN FOOBAR"::=NANOOK::"SYS$SYSROOT:[SYSMGR]ACCOUNT.DAT;2"
```

Example 1Ø One-File Transfer Using a Foreign File

Wildcards

An asterisk inserted in a field means that any value is accepted within that field. Allowable wildcard field specifiers are listed.

In addition, NFT permits wildcard specifiers for part of a field in the file name and file type, as defined:

1. An asterisk (*) in a portion of a field means that part of the field can be replaced with any characters of any length (including none). Examples are:

SOME*.TYP will accept:	SOME.TYP SOMETHG.TYP SOMEBODY.TYP SOMEBZ.TYP

2. A percent sign (%) in a character position means that any one character is accepted in that position (nulls are not permitted). Examples are:

MAIL%.MAI will accept:	MAIL1.MAI MAIL2.MAI MAILX.MAI
IMV%I%%.TXT will accept:	IMV1I22.TXT IMV2I30.TXT IMV3I45.TXT

```
NFT> ELROND/FOO/BAR::DB0:[100,1]=DM1:[*,10]FIL%%.MAC
```

Example 11 File Transfer Using Wildcards

File Transfer Types

- Single-file transfers
- File transfer with concatenation
- Multiple-file transfer

```
NFT> NODEX/[200,30]/DECNET/10::DR0:[40,10]MAGIC.MAC;3=-
NFT> NODEY/200,10/PASME/10::SY0:[30,60]MYSTIC.MAC;2
```

Example 12 Single-File Transfer

```
NFT> NODEA::DR1:[50,10]SUM.TIM=NODEY::DK0:[50,10]-
NFT> SAM.LIT,ONE.WAY
```

Example 13 File Transfer with Concatenation

```
NFT> NODEA::=NODEB/[300,50]/MULTI::-
NFT> DY0:BEE.FIL/AS,CEE.FIL,DEE.FIL
```

Example 14 Multiple-File Transfer

Command Switches

Table 2 Summary of NFT Switches

Switch	Description
/AP	The Append switch permits files to be added to the end of an existing file.
/AS	The ASCII Record Mode switch causes NFT to transfer a file in ASCII record mode.
/AT	The Attributes switch causes a directory to be listed that contains file names and specific attributes associated with each file.
/AX	The Automatic Transfer switch causes NFT to use its default algorithms in selecting the mode to transfer a file to a remote node.
/BK	The Block Mode Transfer switch.
/BR	The Brief Listing switch causes a directory to be listed that contains only the names of files on a node.
/DE	The File Deletion switch allows you to delete one or more files.
/DF	The Default switch allows you to set certain default values for source and destination files.
/EX	The Execute switch causes a command file stored on a remote or local node to be executed.
/FU	The Full Directory Listing switch causes a full directory to be listed for files on a remote node.
/HE	The Help switch causes a descriptive message about NFT switches, subswitches, and commands to be displayed.
/ID	The Identify switch causes the version of NFT to transfer a file in image record mode.
/IM	The Image Record Mode switch causes NFT to transfer a file in image record mode.

Table 2 Summary of NFT Switches (Cont)

Switch	Description
/LI	The Listing switch causes a directory to be listed that contains the name, size, date of creation, and contiguous indicator of files on a remote node.
/LO	The Log switch is a qualifying switch that causes the names of files (copied, deleted, and so on) to be displayed as the operation is performed.
/NV	The New Version switch specifies that an output file will have the same or latest version number, whichever is higher.
/PR	The Protection switch allows a file's protection status to be set or changed.
/QU	The Query switch is a qualifying switch that questions you regarding each file operation in a command.
/RC	The Record Mode Transfer switch directs NFT to transfer a file one record at a time, instead of using block mode.
/RE	The Rename switch allows the name of a file to be changed.
/SB	The Submit switch causes a command file to be submitted for execution on a local or remote node.
/SP	The Spool switch causes files to be spooled to a line printer for printing.
/SU	The Supersede switch allows the contents of an existing file to be replaced with new input and a new file ID, while retaining the existing name and version number.

Exit with Status

NFT exits with a status code that may be useful in these circumstances:

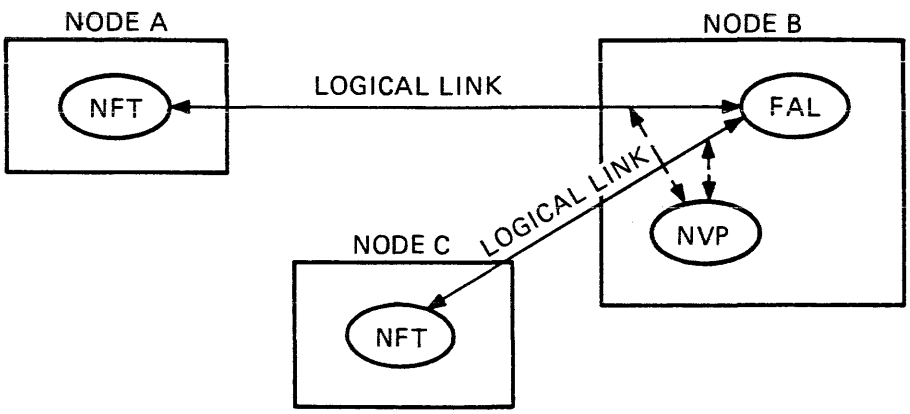
- An indirect command file
- A batch stream job (for RSX-11M-PLUS systems only)
- When spawned by a parent task or connected to a parent task

There are two exit-with-status codes that NFT can issue:

Status	Code
NFT exited without error	1 (EX$SUC)
NFT exited with an error	2 (EX$ERR)

A status code of 2 (EX$ERR) is returned whether you incur a syntax error or an operational error. Also, if any command in an indirect command file fails, an exit-with-status code of 2 (EX$ERR) is issued when NFT exits.

NFT Components



LEGEND:
NFT – NETWORK FILE TRANSFER REQUEST TASK
FAL – NETWORK FILE TRANSFER SERVER TASK (OBJECT 17)
NVP – NETWORK VERIFICATION PROCESSOR TASK (CHECKS INCOMING REQUESTS FOR VALID ACCOUNTS AND PASSWORDS BEFORE GIVING THEM TO FAL)

TK-8504

Figure 3 NFT Utility Components

File Transfer Spooler (FTS) Utility

- FTS features

  FTS has a set of file transfer and manipulation capabilities that are the same as those provided by NFT:

  - Transfer files between two nodes
  - Delete files
  - Execute command files stored on a remote node
  - Spool files to a line printer
  - Append files to an existing file
  - Transfer local command files to a remote node for submission to the remote command file processor, execution, and subsequent deletion

  ok as is

  In addition, FTS provides the following control functions not available in NFT:

  - Queue and process user requests according to user-supplied dates and times
  - List pending user requests
  - Kill pending user requests
  - Log user request completion

- Node verification
- File specification
- File transfer types
  - Single-file transfer
  - File transfer with concatenation
  - Multiple-file transfer
- Command switches
- Function codes (/LI switch) Show queue
- Completion return (/LO switch)

FTS Features

```
FTS> =DENVER::DR2:[200,200]INVENTORY.LST;4
```

Example 15 Simple Transfer Request from a Remote Node

```
FTS> DALLAS::/LI
FTS> USER REQUEST LISTING    28-Jun-80    11:30
Job #          Type/Priority    Options

141954         TLR/50           /IM/LO
Queued 28-Jun-80 10:45          Active
Source File - DBO:[224,1]FICA.LST;1
Destination File - DALLAS::DK1:[200,200]TAXES.LST

382077         TLR/50           /AF:28-Jun-80:11:35/IM/LO
Queued 28-Jun-80 10:50          Tried 1 Time
Source File - DBO:[224,1]STATE.LST;1
Destination File - DALLAS::DK1:[200,200]TAXES.LST
Log File - DBO:DALLAS.LOG
```

Example 16 Print User Requests Involving a Specified Node

Command Switches

Table 3 Summary of FTS Switches

Switch	Description
/AF*	The After switch permits a user request to be executed after a specified date and time.
/AP	The Append switch permits files to be added to the end of an existing file.
/AS	The ASCII Record Mode switch causes FTS to transfer a file in ASCII record mode.
/DE	The File Deletion switch allows you to delete one or more files.
/DF	The Default switch allows you to set certain default values for source and destination files.
/ED*	The Edit switch inhibits queuing of user requests for the current user.
/EX	The Execute switch causes a command file stored on a remote or local node to be executed.
/HE	The Help switch causes a descriptive message about FTS switches, subswitches, and commands to be displayed.
/ID	The Identify switch causes the version of FTS to transfer a file in image record mode.
/IM	The Image Record Mode switch causes FTS to transfer a file in image record mode.
/KI*	The Kill switch causes a request to be removed from the FTS queue.
/LI*	The Listing switch causes any pending user requests to be listed.
/LO*	The Log switch causes a completion message to be logged for a serviced request.

Table 3 Summary of FTS Switches (Cont)

Switch	Description
/PR*	The Priority switch causes a specified priority to be assigned to the user request.
/SB	The Submit switch causes a command file to be submitted for execution on a local or remote node.
/SE*	The Sequence switch causes FTS to process command lines in the order they are input.
/SP	The Spool switch causes files to be spooled to a line printer for printing.

* Switches supported by FTS that are not supported by NFT or that have functions different from those of NFT.

Function Codes (/LI Switch)

Table 4 Summary of FTS Function Codes

Function Code	Description
CLL	Concatenate local files to a local file.
CLR	Concatenate local files to a remote file.
CRL	Concatenate remote files to a local file.
CRR	Concatenate a remote file to a remote file.
DL	Delete file(s) on a local node.
DR	Delete file(s) on a remote node.
EL	Execute command file(s) issued for the local node.
ER	Execute command file(s) issued for a remote node.
PL	Print a file on the local node.
PR	Print a file on a remote node.
SLL	Submit a command file from the local node to the local node.
SLR	Submit a command file from the local node to a remote node.
SRL	Submit a command file from a remote node to the local node.
SRR	Submit a command file from a remote node to a remote node.
TLL	Transfer local file(s) to local file(s).
TLR	Transfer local file(s) to remote file(s).
TRL	Transfer remote file(s) to local file(s).
TRR	Transfer remote file(s) to remote file(s).

Completion Return (/LO Switch)

The Log switch causes FTS to return a completion message into your file area after processing your request. (A separate message is logged for each file in a multiple file transfer.) FTS logs completion messages as a network generation default and stores messages in a user log file. The /LO switch allows you to redefine the user log in which messages are to be placed.

In Example 18, the first command logs a message by default and uses the default user log file specification, SYØ:FTSSYS.LOG. The next command defines the user log file with the specification PAYROLL.LOG. The third command line suspends logging for this request only with the /NOLO switch. The fourth and fifth commands illustrate the use of logical device names in the user log file specification. FTS directs the completion message to the console pseudodevice (CO:) in the fourth command and to the user's terminal (TI:) in the fifth command.

```
FTS> =DALLAS::DR1:[2ØØ,2ØØ]TAXES.LST
FTS> DENVER::SYØ:[2ØØ,2ØØ]=PAYROLL.CBL,.LST/LO:PAYROLL.LOG
FTS> BOSTON::DRØ:[2ØØ,2ØØ]=MISC.DOC;4/NOLO:
FTS> GENEVA::DUØ:[1ØØ,1ØØ]NEWSOFT.MAC=RELEASE.MAC/LO:CO:
FTS> LONDON::DB2:[1ØØ,1ØØ]=REALTIM.MAC/LO:TI:
```

Example 17 Using the /LO Switch

Example 19 illustrates a completion message for a remote-to-remote transfer that terminated in an error. The error code, NØ34Ø7, indicates that the network rejected the connection (see Section 4 of the DECnet-RSX Guide to User Utilities for additional information on /LO error returns).

```
FTS> BOSTON::DL2:[224,223]=DENVER::DUØ:[2ØØ,1]PEN.INV

BOSTON TRR JOB=217526
Time: Ø5-AUG-83  1Ø:56:Ø2  11:2Ø:32
USER: [2ØØ,1] Blocks:Ø
Status: Error=BOSTON NØ34Ø7 Discarded
File: BOSTON::DL2:[224,223]PEN.INV
INPUT:DENVER::DUØ:[2ØØ,1]PEN.INV
```

Example 18 Remote-to-Remote Transfer

```
KRYPTN      TRL            Job=344533
Time:       3-AUG-83       13:56:02 to 14:10:32
User:       [305,524]      Blocks:368.
Status:     Success
File:       DB2:[305,324]FOUND.MEM
Input:      KRYPTN::USER:[RYAN]FOUND.MEM

BERGIL      TRL            Job=446415
Time:       3-SEP-83       1:02:03 to 1:10:32
User:       [305,524]      Blocks:87.
Status:     Error = BERGIL 040040 Discarded
File:       BERGIL::[ARGUS]PLAN.DOC
Input:      DB0:[305,324]PLAN.DOC
```

Example 19 FTSSYS.LOG File

FTS Components

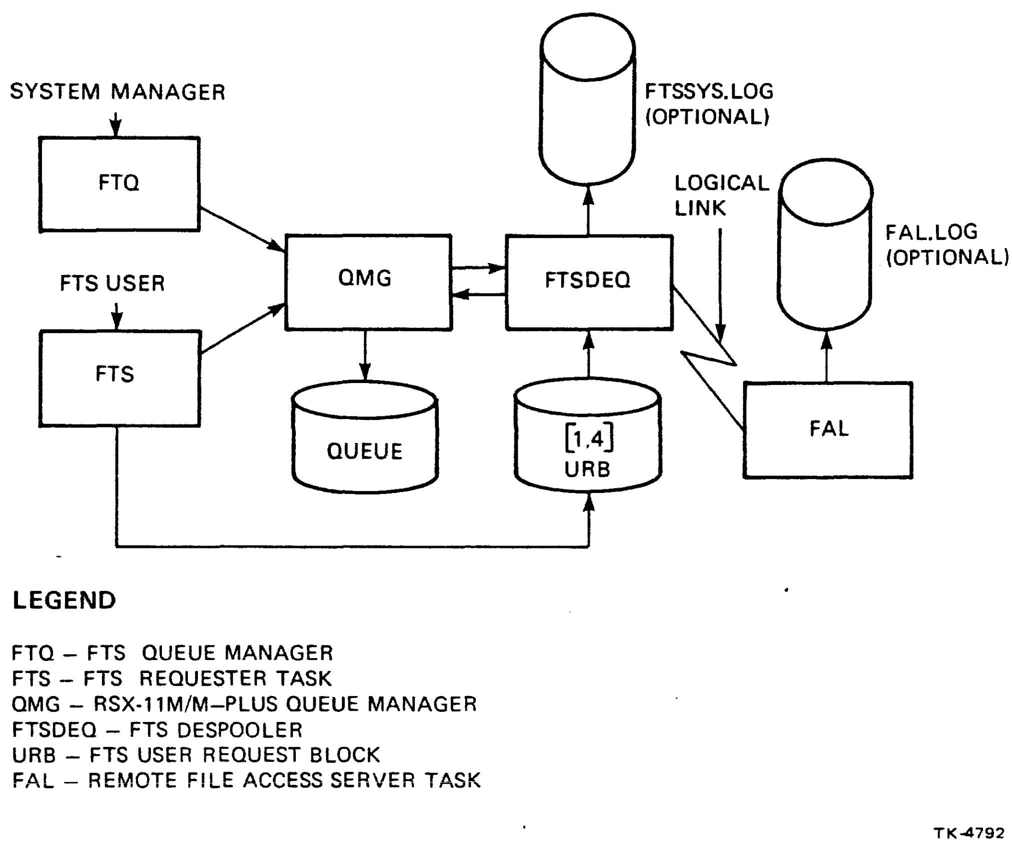


Figure 4 FTS Components

Network Command Terminal (RMT) Utility

RMT Features

- Logically connects a terminal to a DECnet/M/M-PLUS node (host)
- QIO functions supported for RMT on a host

Example 2Ø shows the designated file directory from the host node on your terminal.

```
>
>RMT
Host: ELROND

Connected to "ELROND", System type = RSX-11M-PLUS
System ID: DISTRIBUTED SYSTEMS


>HEL RYAN
Password:

RSX-11M-PLUS V2.1 BL15    [1,54] System       ELROND
Ø7-NOV-83 11:1Ø Logged on Terminal HTØ:

Good Morning


>PIP
PIP> /LI


Directory DBØ:[3Ø5,33Ø]
7-NOV-83 11:1Ø

MAIL.TMP;4          11.            28-DEC-82 14:45

Total of 11./11. blocks in 1. file

PIP>^Z
>
```

Example 2Ø Invoking and Running RMT

RMT handles most control characters (CTRL/O, CTRL/S, CTRL/Q, and CTRL/R) according to standard RSX-11 conventions. However, CTRL/C causes RMT to prompt for input (RMT>) and then pass to MCR on the host system. This procedure differs from conventional RSX-11 usage where CTRL/C directly initiates a local CLI prompt (for example, MCR>).

```
>MAC @FOOBLD
CTRL/C
RMT>ACT
...SYS
...MCR
ACTH3
MACH3
```

Example 21 CTRL/C Handling by RMT

When you finish remote terminal operations, type BYE to log off the host system. Typing BYE automatically disconnects you from the host node and returns control to the local operating system. (If you wish to stay connected to the host node, you can specify BYE/HOLD. You then have to type EXIT RMT to disconnect from the host.)

```
>BYE
>
Have a good afternoon
Ø5-OCT-83  13:1Ø HT2: Logged off ZIRCON

RMT -- Control returned to node _CASTOR::

>
```

Example 22 Exit from RMT Session

If your terminal is externally forced to log off the local system without exiting from the RMT task first, RMT terminates your connection with the host node.

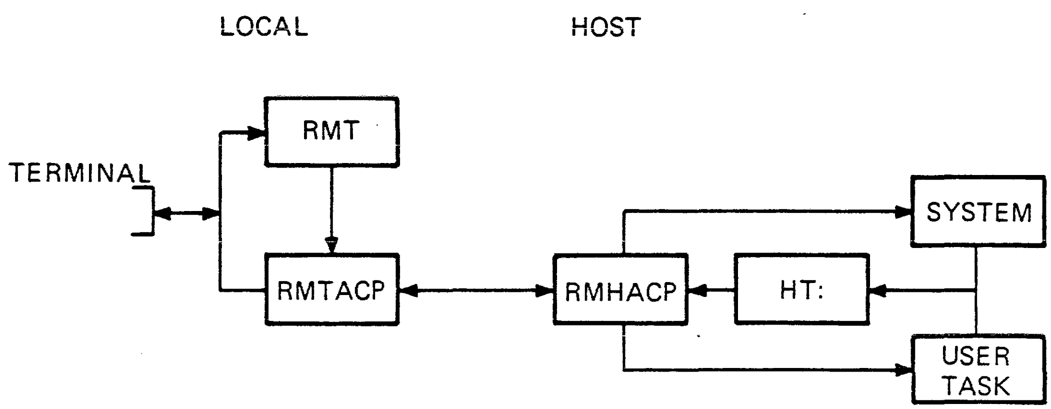
Table 5 QIO Functions Supported for RMT on a Host

Function	Description	Comment
Standard Functions		
IO.ATT	Attach device	On TT: drivers, ATT and DET occur synchronously without fail. On HT: drivers, there is a time lag across the network, so you must specify an event flag to ensure that the task is synchronous with the function. Check for error returns; failures can occur.
IO.DET	Detach device	
IO.KIL	Cancel I/O requests	Same as for TT: driver. If the connection is aborted, it may be because the host system is low on DSR (pool).
IO.RLB	Read logical block (read typed input into buffer)	Same as for TT: driver.
IO.RVB	Read virtual block (read typed input into buffer)	Subfunction bits are not cleared as they are with TT: drivers.
IO.WLB	Write logical block (print buffer contents)	Same as for TT: driver.
IO.WVB	Write virtual block (print buffer contents)	Subfunction bits are not cleared as they are with TT: drivers.
Device-Specific Functions		
IO.ATA	Attach device; specify unsolicited input character AST	See previous comments for IO.ATT.
IO.CCO	Cancel CTRL/O (if in effect; then write logical block)	Same as for TT: driver.

Table 5 QIO Functions Supported for RMT on a Host (Cont)

Function	Description	Comment
SF.GMC	Get multiple characteristics	Refer to the RSX-11M M-PLUS I/O Drivers Reference Manual for details.
IO.GTS	Get terminal support	Same as for TT: driver.
IO.RAL	Read logical block; pass all bits	Same as for TT: driver.
IO.RNE	Read logical block; do not echo	Same as for TT: driver.
IO.RPR	Read logical block after prompt	Same as for TT: driver.
IO.RST	Read logical block ended by special terminators	Same as for TT: driver.
IO.RTT	Read logical block ended by specified special terminator	Same as for TT: driver.
SF.SMC	Set multiple characteristics	Refer to the RSX-11M/ M-PLUS I/O Drivers Reference Manual for details.
IO.WAL	Write logical block; pass all bits	Same as for TT: driver.
IO.WBT	Write logical block; break through any I/O conditions at terminal	Same as for TT: driver except that a WBT message may be broken into two or more pieces. For example, if a BRO program on the host writes a WBT message while you are typing input, the message may be broken into pieces (and your input will be retyped between pieces).

RMT Components



LEGEND: RMT – REMOTE "M/M-PLUS" TERMINAL REQUEST TASK
RMTACP – LOCAL REQUEST PROCESSOR TASK
RMHACP – HOST SERVER TASK (OBJECT 23)
HT: – NETWORK COMMAND TERMINAL DRIVER

TK-5255

Figure 5 RMT Components

PROGRAMMING

Key Concepts

- Task-to-task communications requires two tasks, one on the local computer and one on the remote computer. The programmer must write both tasks.

- Remote File Access requires the user to write only one task. The task links the Network File Access Routines (NFARs) into the user task image. The File Access Listener (FAL) is the remote task the user task communicates with. Even the local files can be accessed using NFARs.

- Task Control also requires the user to write only one task. The task links the Task Control Subroutines to talk to the Task Control Listener (TCL). A remote task can be run, scheduled at periodic intervals, aborted, and the scheduling canceled.

- Direct Line Access provides a direct interface to the Data Link Control Protocol (DDCMP) OR ETHERNET and the driver. This is useful if all the routing, flow control, multiplexing, functionality, and so on are not needed. Flow control and routing, provided by the Routing and the End Control layers are not required. The user must write two tasks talking at both ends of the line directly.

- The Packetnet Switching Interface (PSI) provides only a task-to-task communication facility. The user must write both tasks.

DECnet-RSX Intertask Programming Concepts

Intertask communications concepts include:

BASIC

- Assigning Logical Unit Numbers (LUN)
- Establishing an active network task
- Building a connect block
- Establishing a logical link
- Getting data from the network data queue
- Sending and receiving messages
- Sending interrupt messages
- Terminating activity on a logical link
- Closing a network connection

Assigning Logical Unit Numbers

The following calls use a LUN assigned to the network data queue (MAILBOX): have to assign @ least 1 LUN for mailbox

OPN$	Access the network
SPA$	Specify a user AST routine
GND$	Get network data
REJ$	Reject a logical link request
CLS$	End a task's network operations
GLN$	Get local node information

You can assign the LUN by defining it either as the global symbol .MBXLU in your program or as a parameter for the macro call. The .MBXLU definition and the macro call parameter definition are mutually exclusive (.MBXLU is referenced only if the LUN argument is left blank in the macro parameter block).

NOTE

You should define a particular network data queue LUN in one place only.

If you use the .MBXLU definition, assign the LUN at assembly time or at task build time using one of the following command techniques:

- To include the LUN locally in your source code, include the following in each source module:

  .MBXLU=x

  The variable x is an integer representing the LUN.

- Each source module in the user task must have the same integer x defined for .MBXLU; otherwise, the macros will complete with a privilege error (IE.PRI). A programmer can define a global definition (==) by including the following statement in a single source module:

  .MBXLU==x

  This statement causes the task builder to define references to .MBXLU in all modules of the program to the value x.

- If you want to defer definition of .MBXLU to task build time, issue the following task build option:

  GBLDEF=.MBXLU:x

  This option instructs the task builder to define all global references to .MBXLU as the value of x.

These three command techniques for defining .MBXLU are mutually exclusive. If you do not use any of these procedures, the task builder returns an undefined reference warning message. If you run the task and ignore the warning, the six macro calls (OPN$, SPA$, GND$, REJ$, CLS$, and GLN$) will be rejected by the operating system with a directive status error indication an invalid LUN. The task builder causes undefined references to default to Ø. .MBXLU cannot be defined with a Ø value because Ø is an invalid LUN.

All network LUNs are released and the task's logical links are aborted when you issue a CLS$ call to terminate network operations for a MACRO-11 task. The CLS$ call can be issued in any of the three CLS$ formats - CLS[W]$, CLS[W]$E, and CLS[W]$S.

If a LUN is not assigned to NS, any network directive will be returned with a status of illegal function code.

Establishing an Active Network Task

Before any task can exchange data using intertask communication calls, the task must be an active network task. A task is active if it is running and it has issued an open (OPNx) call. An OPNx call establishes a network data queue for a task and connects the task to the network.

Building a Connect Block

Before the source task can issue a request to connect to another task, it must build a connect block. A connect block contains a destination descriptor, a source descriptor, access control information, and (optional) user-supplied data.

Destination Descriptor

When one task communicates with another task, the tasks are considered to be two objects communicating. There are two kinds of objects: named objects and numbered objects. The destination descriptor identifies the destination task either by task name or by object type number.

Named objects are installed user-defined tasks that are referenced by name when a connection is requested. The object type numeric identifier for user tasks referred to by name is Ø.

Numbered objects are installed user-defined tasks and installed DECnet tasks that are referred to by object type number when a connection is requested. The object type numeric identifier for these tasks referred to by number ranges from 1 to 255. Numbers 1 to 127 are reserved for DECnet tasks. Numbers 128 to 255 are reserved for user tasks. See Table 6 for more information about object definitions.

Table 6 Object Types

Object Type		
Octal	Decimal	Process Type
ØØØ	ØØØ	General task, user process *
ØØ1	ØØ1	File Access Listener (FAL/DAP), Version 1
ØØ2	ØØ2	Unit Record Services (URSs)
ØØ3	ØØ3	Application Terminal Services (ATS)
ØØ4	ØØ4	Command Terminal Services (CTS)
ØØ5	ØØ5	RSX-11M Remote Task Control (TLC) Utility, Version 1
ØØ6	ØØ6	Operator services interface
ØØ7	ØØ7	Node resource manager
Ø1Ø	ØØ8	IBM 327Ø-BSC Gateway
Ø11	ØØ9	IBM 278Ø-BSC Gateway
Ø12	Ø1Ø	IBM 379Ø-BSC Gateway
Ø13	Ø11	TPS application
Ø14	Ø12	RT-11 (CTS-3ØØ) Dibol application
Ø15	Ø13	TOPS-2Ø terminal handler
Ø16	Ø14	TOPS-2Ø remote spooler
Ø17	Ø15	RSX-11M Remote Task Control (TLC) Utility, Version 2
Ø2Ø	Ø16	TLK Utility (LSN)
Ø21	Ø17	File Access Listener (FAL/DAP), Version 4 or later

*Descriptor Format = 1

Table 6 Object Types (Cont)

Object Type		
Octal	Decimal	Process Type
022	018	RSX-11S Host Loader (HLD) Utility
023	019	Network Information and Control Exchange (NICE)
024	020	RSTS/E media transfer program (NETCPY)
025	021	RSTS/E-to-RSTS/E network command terminal handler
026	022	Mail listener (DECnet-based electronic mail system)
027	023	Network command terminal handler (host side)
030	024	Network command terminal handler (terminal side)
031	025	Loopback Mirror (MIR)
032	026	Event Receiver (EVR)
033	027	VAX/VMS personal message utility
034	028	File Transfer Spooler (FTS)
035	029	PHONE utility
036	030	Distributed Data Management Facility (DDMF)
037	031	X.25 Gateway access
040-076	032-062	Reserved for DECnet use
077	063	DECnet Test Tool (DTR)
100-177	064-127	Reserved for DECnet use
200-377	128-255	Reserved for customer use

Source Descriptor

The source descriptor contains information supplied by the DECnet software on the source node. The source descriptor contains the source node name and either the source object number or the source task name if it is a named object. The target task can use this information to determine if it wants to establish communications.

Access Control Information

Access control information contains arguments that define the user's access rights at the remote node. Access control verification is performed according to the conventions of the target system. If the target node is equipped to do so, it verifies access control information before the connect request is passed to the target task.

Table 7 Verification Options

Verification Option	Meaning
ON	Verifies the source program's user identification and password against the node's system account file. If there is not an exact match, the connect request is rejected. The target program does not receive the request.
INSPECT	Verifies the source program's user identification and password just as for the ON option, except that the connect request is forwarded to the target program regardless of the outcome. The verification module tells the target program whether or not the source program checked out against the account file.
OFF	Does not verify. The target program receives all connect requests.

```
NCP> SET OBJECT 135 NAME FOO$$
NCP> SET OBJECT 135 VERIFICATION INSPECT
```

Example 23 Declaring a Task as a Network Object and Turning on Access Verification for that Object

Optional Data Messages

When the source task issues a connect request, you can include a data message of up to 16 characters in the connect block. If the Connect (CONx) call contains the location and length of a block of user data, the source node appends that block to the connect block.

Establishing a Logical Link

To exchange data, a logical link must be established between two active network tasks. A logical link is a path between two cooperating tasks. These tasks must agree to communicate. When the link is established, a user task can send and receive messages.

The task requesting to establish a logical link is called the source task. The other task is called the target task. This distinction applies only during the connection sequence. Once the logical link is established, the terms source and target have no significance, since both tasks have equal access to the logical link.

Tasks at either end of the link must specify a LUN for the link. The LUN is the number each task assigns to the logical link so that the link can be associated by the tasks and the network. The tasks at both ends of the link do not have to use the same LUN for a link.

Getting Data from the Network Data Queue

Once the task is connected to the network, it has a network data queue. The software on the connected task's node places all incoming connect request messages, interrupt messages user disconnect messages, user abort messages, and network abort messages on the task's network data queue. To get these messages, the task must issue a Get Network Data (GNDx) call. A task should begin monitoring its network data queue as soon as the open call is completed successfully.

The GNDx call ordinarily returns the first message in the queue on a first-in, first-out basis. However, the GNDx call has the following options:

- Remove the first message on the queue and place it in the message buffer.
- Remove the first message of a specified type for any logical link and place it in the message buffer.
- Remove the first message of a specified type for a specified logical link and place it in the message buffer.
- Determine the type, length, and associated logical link or any message on the queue without removing it from the queue. This allows you to assign an appropriate buffer size in a subsequent GNDx call that performs one of the four options.

Sending and Receiving Messages

Once a logical link has been established between two tasks, both tasks can send and receive messages. DECnet distinguishes between data and nondata messages. Data messages are delivered directly to a buffer provided by the receiving task. Nondata messages are unsolicited high priority messages that inform the receiving task of some event such as an interrupt or disconnect request.

To send a data message, a task issues a Send (SNDx) call. In the send call, specify the LUN assigned in the connect or accept call. Also specify the location and length of the data message buffer. A send call completes when the receiving node acknowledges to the sending node that it received a message correctly.

To receive a data message, a task issues a Receive (RECx) call. In the receive call, specify the LUN assigned in the connect or accept call. Also specify the location and the length of the data message buffer. A receive call completes when the data message is placed in the specified data message buffer. If the data message buffer is not large enough, the receive call completes with a data overrun condition and excess data is lost. In the case of overrun, the I/O status indicates this. To receive the next data message, another receive call is required.

To send high priority nondata message, a task issues an Abort (ABTx), Disconnect (DSCx), or Interrupt (XMIx) call.

To receive a high priority nondata message, a task issues a GNDx call.

Sending Interrupt Messages

A task can send interrupt messages to another task. Usually an interrupt message informs the receiving task of some unusual event in the sending task. An XMIx call can be up to 16 bytes long. In the interrupt call, specify the LUN assigned in the connect or accept call. Also specify the location and length of the message buffer.

An interrupt call completes when the receiving node acknowledges to the sending node that it has received the message. The receiving node software places the interrupt message on the receiving task's network data queue. The receiving task must issue a get network data call to remove the message from the queue and place it in the task's message buffer.

A task can have only one interrupt message outstanding on a logical link. Until the call completes, any subsequent attempt to send another interrupt message on that same link is returned with a specific error code in the I/O status word.

Terminating Activity on a Logical Link

Any task can terminate activity on a logical link at any time. To do so, you can issue a disconnect call or an abort call. A DSCx call terminates transmissions and interrupts have been sent. An ABTx call disconnects the logical link immediately, regardless of any message queued for transmission. The receiving node software places the termination message on the receiving task's network data queue. The receiving task must issue a GNDx call to retrieve the message.

In both disconnect and abort calls, you can specify the location and length of a user data buffer message for the receiving task. The message can be up to 16 bytes long.

In the disconnect call, specify the LUN assigned in the connect or accept call. When a disconnect call is issued, the software causes all pending transmits for the task issuing the disconnect call to complete before disconnecting the logical link; during this time, the task issuing the disconnect call continues to receive messages. When the last message is transmitted, any remaining receive calls are completed with an abort condition. When the link is disconnected, the LUN is released. A task can use that LUN in subsequent connect or accept calls.

In the abort call, specify the LUN assigned in the connect or accept call. When an abort call is issued, the software immediately aborts all pending transmits and receives and disconnects the link. The LUN is released and a task can use that LUN in subsequent connect or accept calls.

Closing a Network Connection

To close a task's network connection, issue a Close (CLSx) call. The close call informs the software that the task no longer requires network services. This causes the software to purge the task's network data queue. Any active LUNs are deactivated and released for use if the task subsequently issues an Open (OPNx) call.

If there is data in the task's network data queue when the close call is issued the following can occur:

- If the terminating task's network data queue contains any connect requests, the terminating task receives them if it subsequently issues an open call within a short period of time.
- Any other type of data in the terminating task's network data queue is discarded (for example, interrupt, disconnect, and abort messages).

Table 8 Task-to-Task Message Flow

Local Node			Remote Node
User Task	ECL	ECL	User Task
CON$ →			
	CI →		
	←	CI ACK	
		←	ACC$
	←	Connect Confirm	
		←	REC$
	←	LS-Data Request	
	LS-ACK →		
Data →			
←			Data ACK
DSC$ →			
	DSC-Request →		
	←	DSC-Confirm	

Table 9 Network Call Summary for Task-to-Task Communication

DECnet Calls for Logical Link (LL)		DECnet Calls for Virtual Circuit (VC)		Description
MACRO-11	FORTRAN 77, COBOL, BASIC-PLUS-2	MACRO-11	FORTRAN 77	
Accessing/Deaccessing the Network's Facilities				
OPN$	OPNNT	OPX$	OPXNW	Declare a network task and open the task's mailbox
CLS$	CLSNT	CLX$	CLXNW	End a task's network activity and close the task's mailbox
Controlling a Logical Link/Virtual Circuit				
CON$	CONNT	CAL$	CALNW	Connect request for LL; set up request for SVC
DSC$	DSCNT	CLR$	CLRNW	Disconnect request for LL; clear request for SVC
ACC$	ACCNT	CAC$	CACNW	Accept the connect request for the LL; set up the request for the SVC
REJ$	REJNT	CRJ$	CRJNW	Reject the connect request for LL; set up the request for SVC
		CTR$	CTRNW	Transfer a set up request for SVC sent by a remote to another task
ABT$	ABTNT			Abort request for existing LL
		RES$	RESNW	Reset VC; confirm receipt of a reset

Table 9 Network Call Summary for Task-to-Task Communication (Cont)

DECnet Calls for Logical Link (LL)		DECnet Calls for Virtual Circuit (VC)		Description
MACRO-11	FORTRAN 77, COBOL, BASIC-PLUS-2	MACRO-11	FORTRAN 77	
Accessing a Logical Link/Virtual Circuit from/into a Task				
CONB$$	BACC, BFMTØ, BFMT1	NIBB$$, NIBF$$	—	Building a connect block, access control block, destination descriptor, network information block
		APV$	APVNW	Access a PVC
		DPV$	DPVNW	Deaccess a PVC
Sending and Receiving Data on Logical Link/Virtual Circuit				
SND$	SNDNT	XMT$	XMTNW	Transmit data
REC$	RECNT	RCV$	RCVNW	Receive data
XMI$	XMINT	INT$	INTNW	Transmit an interrupt data
		ICF$	ICFNW	Interrupt receipt confirm
Handling Unsolicited Data for Logical Link/Virtual Circuit				
GND$	GNDNT	GNX$	GNXNW	Get unsolicited data
SPA$	—	SPX$	—	Specify an AST routine for handling unsolicited data
—	WAITNT	—	WAITNW	Suspend the task

line up with text

Draw long dashes centered in columns

Table 9 Network Call Summary for Task-to-Task Communication (Cont)

DECnet Calls for Logical Link (LL)		DECnet Calls for Virtual Circuit (VC)		Description
MACRO-11	FORTRAN 77, COBOL, BASIC-PLUS-2	MACRO-11	FORTRAN 77	
Miscellaneous				
GLN$	GLNNT			Get local node information
		TDN$	TDNNW	Translate DTE name

Legend

LL - Logical Link

VC - Virtual Circuit

SVC - Switched VC

PVC - Permanent VC

Notes on MACRO-11 Programming

1. Macro types:

   - Build-type macros - OPN[W]$
   - Execute-type macros - OPN[W]$E
   - Stack-type macros - OPN[W]$S

```
LABEL:   XXX$ LUN,EFN,STAT,AST,<pl,p2,.....pn>

         ;Create a parameter block

         ;for the call designated

         ;by XXX$ using a BUILD-

         ;type macro.
XXX[W]$E LABEL

         ;Issue an EXECUTE-type macro

         ;referencing the parameter

         ;block created for LABEL.
XXX[W]$S LABEL,,,,,,<pl,p2>

         ;Issue an EXECUTE-type macro

         ;and override the parameter

         ;list arguments Pl and P2.
XXX[W]$S #LUN,#EFN,#STAT,#AST<#pl,p2>

         ;Issue a STACK-type macro, create

         ;a parameter block on the stack

         ;and execute the call.
```

Example 24 Macro Types for Programming

2. Event flags 17-22 are used by FORTRAN 77, COBOL, and BASIC-PLUS-2 DECnet calls. Do not assign these event flag numbers to your MACRO-11 calls if you are using high-level language calls in the same program.

3. When using the wait option [W] in MACRO-11 calls, you must assign an event flag. If the flag is not specified, the call completes as a normal asynchronous call.

4. Task must be assembled with NETLIB.MLB and/or PSI.MLB:

   ```
   MAC> TASKA,TASKA/-SP=[1,1]NETLIB/LB,[300,300]TASKA
   ```

5. See the examples in Section 2.3.16 of the DECnet-RSX Programmer's Reference Manual for additional information.

Notes on FORTRAN 77, COBOL, and BASIC-PLUS-2 Programming

1. Event flags 17-22 are reserved and must not be used for calls.

2. Task must be built with NETFOR.OLB, for example:

```
TKB> TASKB,TASKB=TASKB,[1,1]NETFOR/LB,[1,1]FOROTS/LB
```

3. COBOL and BASIC-PLUS-2 use NETFOR.OLB, so to issue calls do as follows:

```
FORTRAN 77      - CALL OPNNT[W] (P1,P2,P3,...)
COBOL           - CALL "OPNNT[W]" USING P1,P2,P3,...
BASIC-PLUS-2    - CALL OPNNT[W] BY REF (P1,P2,P3,...)
```

   where P1,P2,P3, ... are the call's parameters.

4. COBOL and BASIC-PLUS-2 tasks must be built as follows:

```
TKB> TASKC,TASKC=TASKC,[1,1]NETFOR/LB,[1,1]COBLIB/LB
TKB> TASKD,TASKD=TASKD,[1,1]NETFOR/LB,[1,1]BASIC2/LB
```

5. See the examples in Sections 3.7.18 through 3.7.23 of the DECnet-RSX Programmer's Reference Manual for additional information.

Remote File Access

Table 10 Network Call Summary for Remote File Access

DECnet calls for Logical Link (LL)	Description
FORTRAN 77, COBOL, BASIC-PLUS-2	
ACONFW	To set record and file access options
ATTNFW	To set extended attributes
OPWNFW	Create a sequential file
OPANFW	Open a file for appending records
OPMNFW	Open and modify a sequential file
OPRNFW	Open a file for reading
OPUNFW	Open and update a sequential file
GETNFW	Read a single record
PUTNFW	Write a single record
CLSNFW	Close a file
DELNFW	Delete a file
EXENFW	Execute a command file
PRGNFW	Discard an open file
SPLNFW	Spool the file
SUBNFW	Submit a command file for execution

NOTE
Destination task for Remote File Access is FAL (Object 17).

Notes on Remote File Access Programming

These calls are implemented by subroutines. The network open call, OPNNT[W], and the network close call, CLWNT[W], are also used in remote file access operations. You must always issue OPNNT[W] first because this routine allows your task to access the network. Issue CLSNT[W] last to close your task's access to the network.

Each open subroutine creates a DECnet logical link to the node where the file resides and then creates and opens the file. You must use the same LUN to open, write, and close the file. This LUN must be one not in use.

The maximum data record limit is 256 bytes. To override this default adjust the global symbol $NFRSZ to a higher number.

1. All file access calls are synchronous. Therefore they do not return to the user until the operation completes.

2. The following default parameters can be changed at taskbuild time:

Event Flags:	GBLDEF=.TREF:value (default is 17.) GBLDEF=.RCEF:value (default is 18.)
Buffering Level:	GBLDEF=$NFNMB:buffering level (default is 2.)
Maximum Record Size:	GBLDEF=$NFRSZ:record size *(default is 256. bytes)

   Buffer Space Allocation:

   EXTSCT=$$FSR1:value (no default)

3. Task must be built with NETFOR.OLB, for example:

   ```
   TKB> TASKE,TASKE=TASKE,[1,1]NETFOR/LB,[1,1]FOROTS/LB
   ```

4. See the examples in Sections 3.8.19 through 3.8.24 of the DECnet-RSX Programmer's Reference Manual for additional information.

*

Remote Task Control

Table 11 Network Call Summary for Remote Task Control

DECnet Calls for Logical Link (LL)	Description
FORTRAN 77	
RUNNCW	Execute an installed task
ABONCW	Abort an executed task of cancel a scheduled task
BACUSR	Build access control information block

NOTE

Destination task for Remote Task Control is TCL (Object 15).

Notes on Remote Task Control Programming

1. All calls are synchronous and pass control back to the user task only after the operation completes.

2. Task must be built with NETFOR.OLB, for example:

   TKB> TASKF,TASKF=TASKF,[1,1]NETFOR/LB,[1,1]FOROTS/LB

3. See the examples in Section 3.9.6 of the DECnet-RSX Programmer's Reference Manual for additional information.

Direct Line Access

To use DLX, you issue queued I/O (QIO) calls to the NX: device. The DLX interface can be used to communicate over all devices supported by DECnet-RSX.

You can use DLX QIO$ to communicate between your program and a program on an adjacent node using the DECnet-RSX or non-DECnet based system that has similar capabilities. In task-to-task communication between adjacent nodes, DLX significantly improves network performance in terms of CPU and line usage. You can build your own user level protocol that best suits the application.

Table 12 Network Call Summary for Direct Line Access

Direct Line Access (DLX) Calls	Description Non-Ethernet	Ethernet
MACRO-11		
IO.XOP	Open a line	Open an Ethernet device
IO.XSC	Not applicable	Set characteristics
IO.XIN	Initialize the line	Not applicable
IO.XTM	Transmit a message	Transmit a message
IO.XRC	Receive a message	Receive a message
IO.XHG	Hang up a line	Not applicable
IO.XCL	Close the line	Not applicable

Notes on Direct Line Access Programming

1. Special considerations for Ethernet users:

   - All messages on the Ethernet must include a destination address (48-bit) and a protocol type (16-bit).

   - Ethernet may be opened in three different modes (defined in EPMDF$ in NETLIB.MLB):

     - Exclusive
     - Default
     - Normal

   - Ethernet users should refer to Section 4.2 in the DECnet-RSX Programmer's Reference Manual for additional information.

2. All tasks must be assembled with NETLIB.MLB, for example:

   ```
   MAC> TASKG,TASKG/-SP=[1,1]NETLIB/ML,[300,311]TASKG
   ```

3. See the examples in Section 4.3.9 through 4.3.10 of the DECnet-RSX Programmer's Reference Manual for additional information.

APPENDIX ~~A~~ Programs

There is only 1 appendix ~~therefore~~ "A" is not needed.

```
                        .TITLE  SEN10
        ;*****************************************************************
        ;
        ; THIS EXAMPLE WILL :
        ;     SEND 10 DATA MESSAGES WITH THE FORMAT 'THIS IS MESSAGE N'
        ;     ACCEPT A SHORT MESSAGE FROM THE INITIATING TERMINAL
        ;     AND SEND THIS MESSAGE OUT AS AN 'INTERRUPT MESSAGE'.
        ;
        ;*****************************************************************
                .MCALL  OPNW$S,CONW$S,SNDW$S,CONB$$,ALUN$C,QIOW$C
                .MCALL  EXIT$S,MRKT$C,WTSE$C,CLEF$C,SETF$C,QIO$C
                .MCALL  DSCW$S,XMIW$S,ASTX$S
        ;
        ;  DATA AREA
        ;
        MESN:   .ASCII  /THIS IS MESSAGE /      ; MESSAGE TO BE TRANSMITTED
NUM:    .ASCII  /0/                     ; MESSAGE NUMBER
        NN=.-MESN
PRMPT:  .ASCII  /MSG:/                  ; PROMPT FOR INTERRUPT MESSAGE
        .EVEN
IOSTN:  .BLKW   2                       ; COMPLETION STATUS FOR NETWORK
BUFF:   .BLKB   16.                     ; INTERRUPT MESSAGE BUFFER
IOSTB:  .BLKW   2                       ; COMPLETION STATUS FOR BUFFER
CNT:    .WORD   0                       ; NUM OF CHAR IN INTERRUPT MESS
ERRCNT: .WORD   0                       ; ERROR COUNT
IOSB:   .BLKW   1                       ; I/O STATUS
;
        .EVEN
CONBL:  CONB$$  ELROND,0,1,<REC10>      ; CONNECT REQUEST BLOCK
;
;  CODE
;
        .EVEN
START:  CLR     ERRCNT                  ; INITIALIZE ERROR COUNT TO ZERO
        CLEF$C  5                       ; CLEAR EVENT FLAG USED TO MAKE SURE
                                        ; INTERRUPT MESSAGE ACCEPTED PRIOR
                                        ; TO EXIT
        MOVB    #60,NUM                 ; INITIALIZE MESSAGE NUM TO ZERO
        ALUN$C  1,NS                    ; ASSIGN LUN 1 FOR NETWORK DATA QUEUE
        ALUN$C  2,NS                    ; ASSIGN LUN 2 FOR LOGICAL LINK
        OPNW$S  #1,#1,#IOSTN            ; CREATE THE NETWORK DATA QUEUE
        TSTB    IOSTN                   ; TEST FOR ERRORS
        BGT     OK1
        JMP     ERR1
OK1:    CONW$S  #2,#2,#IOSTN,,<#CONBL>  ; CREATE LOGICAL LINK TO 'REC10'
        TSTB    IOSTN                   ; TEST FOR ERRORS
        BLE     ERR2
        QIO$C   IO.RPR,5,,,IOSTB,TRMAST,<BUFF,16.,,PRMPT,4> ; ACCEPT
                                        ; INTERRUPT MESSAGE FROM TERMINAL
                                        ; (USE AST)[16 CHAR MAX]
        TST     $DSW                    ; TEST FOR ERRORS
        BLT     ERR3
        MOV     #10.,R0                 ; SET LOOP COUNTER TO 10
LOOP:   SNDW$S  #2,#2,#IOSTN, ,<#MESN,#NN> ; SEND MESSAGE
        TSTB    IOSTN                   ; TEST FOR ERRORS
        BLE     ERR4
        INCB    NUM                     ; UPDATE MESSAGE NUMBER
        SOB     R0,LOOP                 ; LOOP IF MORE TO SEND
;
        WTSE$C  5                       ; MAKE SURE TERMINAL MESSAGE
                                        ; HAS BEEN ENTERED
                                        ; BEFORE EXITING
;
        DSCW$S  #2,#2,#IOSTN            ; DISCONNECT NETWORK
```

Example 25 Task-to-Task Communication (Sheet 1 of 5)

right-hand pg.
odd-numbered pg.

```
;
        EXIT$S                          ; EXIT
;
;  TERMINAL AST ROUTINE
;
TRMAST: MOV     (SP)+,IOSB              ; POP STACK
        MOV     IOSTB+2,CNT             ; OBTAIN NUMBER OF CHARACTERS
        XMIW$S  #2,#3,#IOSTN, ,<#BUFF,CNT>; TRANSMIT INTERRUPT MESSAGE
                                        ; (NOTE USE OF EF 3 INSTEAD OF
                                        ; EF 2 - AVOID COMPETITION)
        TSTB    IOSTN                   ; TEST FOR ERRORS
        BLE     ERR5
        SETF$C  5                       ; SET EVENT FLAGE TO INDICATE
                                        ; INTERRUPT MESSAGE SENT
        ASTX$S                          ; AST EXIT
;
;  ERROR HANDLING - A SAMPLE DEBUGGING TECHNIQUE
;
ERR5:   INC     ERRCNT                  ; DETERMINE
ERR4:   INC     ERRCNT                  ; WHICH
ERR3:   INC     ERRCNT                  ; ERROR
ERR2:   INC     ERRCNT                  ; OCCURRED
ERR1:   INC     ERRCNT
        MOV     ERRCNT,R1               ; R1 CONTAINS THE ERROR NUMBER
        MOV     $DSW,R2                 ; R2 CONTAINS THE DIRECTIVE STATUS WORI
        MOV     IOSTN,R3                ; R3 CONTAINS THE FIRST I/O STATUS WORI
        MOV     IOSTN+2,R4              ; R4 CONTAINS THE 2ND I/O STATUS WORD
        IOT                             ; ABORT - DUMP THE REGISTERS
;
;
;
        .END    START
```

Example 25 Task-to-Task Communication (Sheet 2 of 5)

```
        .TITLE  REC10
;*************************************************************
;
;  THIS EXAMPLE WILL:
;       ACCEPT SHORT MESSAGES FROM THE SENDER TASK 'SND10'
;       PRINT THE MESSAGES ON THE CONSOLE DEVICE (CO:)
;       DISCONNECT AND EXIT GRACEFULLY.
;
;*************************************************************
        .MCALL  OPNW$S,SPAW$S,RECW$S,GNDW$S,ACCW$S,CLSW$S,NETDF$
        .MCALL  QIOW$S,ALUN$C,CLEF$C,WTSE$C,SETF$C,ASTX$S,EXIT$S
        NETDF$
;
;  DATA AREA
;
BUF1:   .BLKB   25.                     ; BUFFER FOR USER MESSAGES
        .EVEN
BUF2:   .BLKB   N.CBL                   ; BUFFER FOR NETWORK MESSAGES
IOST:   .BLKW   2                       ; COMPLETION STATUS FOR NETWORK
IOST1:  .BLKW   2                       ; COMP. STAT. FOR GET NET DATA
IOST2:  .BLKW   2                       ; COMP. STAT. FOR ACCEPT CONNECT
IOSB:   .BLKW   1                       ; I/O STATUS
ERRCNT: .WORD   0                       ; ERROR COUNT
CNT:    .WORD   0                       ; USER MESSAGE CHAR COUNT
CNTB:   .BLKB   2                       ; INTERRUPT MESSAGE CHAR COUNT
FLAG:   .WORD   0                       ; DISCONNECT FLAG
        .EVEN
;
;  CODE
;
START:  CLR     ERRCNT                  ; INITIALIZE ERROR COUNT TO ZERO
;       CLEF$C  10.                     ; CLEAR EVENT FLAG USED TO MAKE
                                        ; SURE CONNECT HAS OCCURRED
        ALUN$C  1,NS                    ; ASSIGN LUN 1 FOR NETWORK DATA QUEUE
        ALUN$C  2,NS                    ; ASSIGN LUN 2 FOR LOGICAL LINK
        OPNW$S  #1,#1,#IOST             ; CREATE THE NETWORK DATA QUEUE (mailbox)
        TSTB    IOST                    ; TEST FOR ERRORS
        BLE     ERR1
        SPAW$S  #1,#1,#IOST,#CMPAST,<#NETAST> ; SPECIFY AST HANDLING
        TSTB    IOST                    ; TEST FOR ERRORS
        BLE     ERR2
        WTSE$C  10.                     ; WAIT TO MAKE SURE CONNECT
                                        ; HAS OCCURRED
LOOP:   RECW$S  #2,#2,#IOST,,<#BUF1,#25.>; RECEIVE UP TO 25 CHARS
        TSTB    IOST                    ; TEST FOR ERRORS
        BLE     ERR3
        MOV     IOST+2,CNT              ; OBTAIN CHARACTER COUNT
        QIOW$S  #IO.WLB,#5,#5,,,,<#BUF1,CNT,#40>; TYPE MESSAGE
                                        ; ON TERMINAL
        TST     FLAG                    ; HAS DISCONNECT OCCURRED?
        BEQ     LOOP                    ; NO, POST ANOTHER RECEIVE
        CLSW$S  #1,#1,#IOST2            ; CLOSE NETWORK
        TSTB    IOST2                   ; TEST FOR ERRORS
        BLE     ERR5
        EXIT$S                          ; PROGRAM EXIT
        BR      LOOP
```

Example 25 Task-to-Task Communication (Sheet 3 of 5)

```
;
;  ERROR HANDLING - A SAMPLE DEBUGGING TECHNIQUE
;
ERR6:   INC     ERRCNT
ERR5:   INC     ERRCNT
ERR4:   INC     ERRCNT
ERR3:   INC     ERRCNT
ERR2:   INC     ERRCNT
ERR1:   INC     ERRCNT
        MOV     ERRCNT,R1                       ; R1 = ERROR NUMBER
        MOV     $DSW,R2                         ; R2 = DIRECTIVE STATUS WORD
        MOV     IOST,R3                         ; R3 = I/O STATUS BLOCK (1ST WORD)
        MOV     IOST+2,R4                       ; R4 = I/O STATUS BLOCK (2ND WORD)
        IOT                                     ; ABORT - DUMP REGISTERS
;
;
;  AST HANDLING FOR DATA IN NETWORK DATA QUEUE
;
CMPAST: MOV     (SP)+,IOSB                      ; SAVE SPA$ I/O STATUS BLOCK ADDR
        MOV     R0,-(SP)                        ; SAVE R0
        MOV     IOSB,R0                         ; GET I/O STATUS BLOCK ADDRESS
        CMPB    #IS.SUC,(R0)                    ; SUCCESSFUL?
        BEQ     OKA
        JMP     OUT
OKA:    MOV     2(R0),R0                        ; GET CURRENT NETWORK DATA COUNT
        BNE     OKB
        JMP     OUT
OKB:    BR      GET
NETAST: MOV     R0,-(SP)                        ; SAVE R0
        MOV     #1,R0                           ; SET NETWORK DATA COUNT TO 1
GET:    GNDW$S  #1,#1,#IOST1,,<#BUF2,#N.CBL> ; GET NETWORK DATA
        BCS     OUT                             ; CARRY BIT SET - ERROR
        CMPB    #IS.SUC,IOST1                   ; SUCCESSFUL?
        BNE     OUT
        CMPB    #NT.CON,IOST1+1                 ; CHECK IF CONNECT REQUEST
        BNE     OTHER
        ACCW$S  #2,#2,#IOST2,,<#BUF2>           ;  ACCEPT CONNECTION
        TSTB    IOST2                           ; TEST FOR ERRORS
        BLE     ERR4
        SETF$C  10.                             ; SET EVENT FLAG TO INDICATE
                                                ; CONNECT HAS OCCURRED
        BR      NEXT
OTHER:  CMPB    #NT.DSC,IOST1+1                 ; CHECK IF DISCONNECT REQUEST
        BNE     OTHR2
        MOV     #1,FLAG                         ; SET DISCONNECT FLAG
        BR      NEXT                            ; GO BACK TO MAIN ROUTINE
;
;
OTHR2:  CMPB    #NT.INT,IOST1+1                 ; CHECK IF INTERRUPT MESSAGE
        BEQ     OKC
        JMP     ERR6                            ; NOT A EXPECTED COMMAND
OKC:    MOVB    IOST1+2,CNTB                    ; OBTAIN CHARACTER COUNT
        QIOW$S  #IO.WLB,#5,#3, , , ,<#BUF2,CNTB,#40> ; TYPE INTERRUPT MESSAGE
                                                        ; (NOTE USE OF EF 3
```

Example 25 Task-to-Task Communication (Sheet 4 of 5)

```
                                                  ; INSTEAD OF EF 5)
NEXT:   NOP
        DEC     R0                                ; CHECK IF MORE DATA
        BEQ     OUT
        JMP     GET
OUT:    MOV     (SP)+,R0                          ; RESTORE R0
        ASTX$S                                    ; AST EXIT
;
;
;
        .END    START
>
```

Example 25 Task-to-Task Communication (Sheet 5 of 5)

```
      INTEGER*2 MLTYP,RECSIZ,SNDSIZ,OPNLUN,CONLUN,MESNUM,XMITS,NDLEN,TSKLEN
      INTEGER*2 IOST(2),MSTAT(3)
      BYTE ERRMES(2),TSKNAM(6),CONBLK(72),NDNAM(6),DEFNOD(6),DEFTSK(6)
      BYTE SNDBUF(50),RECBUF(10)
      LOGICAL*1 STAT,IMMED
      DATA DEFNOD/'M','A','S','T','E','R'/
      DATA DEFTSK/'R','E','C','V','E','R'/

C
C  INITIALIZE CONSTANTS
C
      IMMED=.TRUE.                    !* SET IMMED TO TRUE FOR GNDNTW
      OPNLUN=1                        !* NETWORK OPNNT LUN
      CONLUN=2                        !* COUNT LUN FOR THE
                                      !* LOGICAL LINK
      XMITS=20                        !* THE NUMBER OF INQUIRIES
                                      !* TO BE SENT TO THE REMOTE NODE
      SNDSIZ=50                       !* THE SIZE OF THE THE MESSAGES TO
                                      !* BE SENT TO THE REMOTE NODE
      RECSIZ=10                       !* THE SIZE OF THE MESSAGES TO
                                      !* BE RECEIVED

C
C  GET THE NODE AND TASK NAMES
C

4     TYPE 300                        !* ASK FOR NODE-NAME
      READ(5,310) (NDNAM(NDLEN),NDLEN=1,6) !* GET THE NAME
      DO 5 NDLEN=6,1,-1               !* LOOP TO FIND LENGTH OF NAME
      IF (NDNAM(NDLEN).NE.' ') GOTO 6 !* IF NOT A SPACE, GET TASK-NAME
5     CONTINUE
      DO 50 I=1,6
50    NDNAM(I)=DEFNOD(I)              !* DEFAULT NODE NAME 'MASTER'
      NDLEN=6                         !* LENGTH OF DEFAULT NAME

6     TYPE 320                        !* ASK FOR THE TASK-NAME
      READ(5,310) (TSKNAM(TSKLEN),TSKLEN=1,6) !* GET IT
      DO 7 TSKLEN=6,1,-1              !* TSKLEN IS LENGH OF TASK-NAME
      IF (TSKNAM(TSKLEN).NE.' ') GOTO 8 !* IF NOT SPACE, ACCESS NETWORK
7     CONTINUE
      DO 60 I=1,6
60    TSKNAM(I)=DEFTSK(I)             !* DEFAULT TASK NAME 'RECVER'
      TSKLEN=6                        !* LENGTH OF DEFAULT NAME

C
C  ACCESS NETWORK
C
8     CALL    OPNNTW(OPNLUN,IOST,MSTAT)
      IF      (IOST(1).NE.1)GOTO 100  !* IF FAILURE JUST EXIT
C
C  BUILD A FORMAT 2 CONNECT BLOCK
C
      CALL    BFMT1(STAT,CONBLK,NDLEN,NDNAM,,TSKLEN,TSKNAM)
      IF      (STAT)GOTO 10           !* IF SUCCESS GO ON
      TYPE    200                     !* ELSE TYPE OUT A FAILURE
                                      !* NOTIFICATION
      GOTO    90                      !* AND EXIT
C
C  CONNECT TO THE TASK ON THE REMOTE NODE
C
10    CALL    CONNTW(CONLUN,IOST,CONBLK)
      IF      (IOST(1).EQ.1)GOTO 15   !* IF SUCCESS TELL HIM
      TYPE    240,IOST                !* ELSE PRINT STATUS BLOCK
      GOTO    90                      !* DEACCESS THE NETWORK
```

Example 26 Remote Task Control
(Sheet 1 of 3)

```
                                             !* AND EXIT
15       TYPE    220                         !* PRINT CONNECT CONFIRMATION
                                             !* NETWORK AND EXIT
C
C   SEND AND RECEIVE MESSAGES TO AND FROM THE REMOTE NODE
C
         DO      40 MESNUM=1,XMITS
C
C   FIRST GET ANY ERROR MESSAGES SENT FROM THE OTHER SIDE VIA
C   INTERUPT MESSAGES
         IF      (MSTAT(1).EQ.0)GOTO 20   !* IF MSTAT(1)=0 NO MESSAGES
                                             !* ARE THERE
         CALL    GNDNTW(IOST,MLTYP,2,ERRMES,,IMMED,2) !* GET THE MESSAGE
         IF      (IOST(1).NE.1)GOTO 20   !* IF WE COULDN'T GET THE MESSAGE
                                             !* JUST IGNORE IT
         TYPE    210,ERRMES(1)               !* PRINT OUT THE MESSAGE
C
C   SEND THE INQUIRY
C
20       CALL    SNDNTW(CONLUN,IOST,SNDSIZ,SNDBUF)
         IF      (IOST(1).EQ.1)GOTO 30   !* IF SUCCESS CONTINUE
         TYPE    210,MESNUM                  !* OTHERWISE TYPE OUT AN
                                             !* ERROR MESSAGE
         GOTO    40                          !* AND START A NEW MESSAGE
C
C   RECEIVE THE RESPONSE FROM THE REMOTE NODE
C
30       CALL    RECNTW(CONLUN,IOST,RECSIZ,RECBUF)
         IF      (IOST(1).EQ.1)GOTO 40   !*IF SUCCESS CONTINUE
         TYPE    210,MESNUM                  !* OTHERWISE TYPE OUT AN
                                             !* ERROR MESSAGE
40       CONTINUE
C
C   DISCONNECT THE LINK
C
         TYPE    230                         !* PRINT OUT DISCONNECT MESSAGE
         CALL    DSCNTW(CONLUN,IOST)
C
C   COME HERE TO DEACCESS THE NETWORK AND EXIT
C
90       CALL    CLSNTW
100      STOP    'END OF PROGRAM EXECUTION'
C
C   FORMAT STATEMENTS
C
200      FORMAT  (' ERROR BUILDING CONNECT BLOCK')
210      FORMAT  (' ERROR ON INQUIRY  ',I3)
220      FORMAT  (' LINK ENABLED')
230      FORMAT  (' LINK DISABLED')
240      FORMAT  (' CONNECT FAIL: IOST= ',I3)
300      FORMAT  (' PLEASE ENTER NODE-NAME <MASTER>: ',$)
310      FORMAT  (6A1)
320      FORMAT  (' PLEASE ENTER TASK-NAME <RECVER>: ',$)
         END
         INTEGER*2 OPNLUN,MLTYP,INDEX,ACCLUN,NUMBER,NUMMES
         INTEGER*2 RECSIZ,SNDSIZ,INTSIZ
         INTEGER*2 MSTAT(3),IOST(2),IOST1(2),IOST2(2)
         BYTE RECBUF(50),SNDDAT(10),MLBX(98),INTMES(2)
C
C   INITIALIZE CONSTANTS
C
         OPNLUN=1                 !* NETWORK OPNNT LUN
         ACCLUN=2                 !* ACCNT LUN FOR THE LOGICAL LINK
         RECSIZ=50                !* SIZE OF DATA BUFFER TO BE RECEIVED
         INTSIZ=2                 !* SIZE OF INTERRUPT DATA BUFFER TO SEND
         NUMMES=0                 !* NUMBER OF MESSAGES RECEIVED
```

Example 26 Remote Task Control
(Sheet 2 of 3)

```
      SNDSIZ=10                !* NUMBER OF BYTES TO SEND BACK
C
C  ACCESS NETWORK
C
      CALL     OPNNTW(OPNLUN,IOST,MSTAT)
      IF       (IOST(1).NE.1)GOTO 100        !* IF FAILURE JUST EXIT
      IF       (MSTAT(1).EQ.0)GOTO 40        !* IF NOTHING ON MAILBOX
                                             !* JUST CLOSE AND EXIT
10    CALL     GNDNT(IOST1,MLTYP,98,MLBX)    !* ISSUE A GET NETWORK DATA
20    CALL     WAITNT(INDEX,IOST1,IOST2)     !* WAIT FOR A COMPLETION
      IF       (INDEX.EQ.2)GOTO 50           !* IF INDEX=2 A RECEIVE HAS
                                             !* BEEN COMPLETED
C
C  NETWORK DATA HAS BEEN RECEIVED
C
      IF       (IOST1(1).NE.1)GOTO 40        !* IF GNDNT FAILED JUST
                                             !* CLOSE AND EXIT
      IF       (MLTYP.GE.3)GOTO 40           !* IF MULTYP>=3 THE LINK HAS
                                             !* BEEN BROKEN
      IF       (MLTYP.EQ.2)GOTO 10           !* IF MLTYP=2 WE'VE RECEIVED
                                             !* AN INTERRUPT MESSAGE, JUST
                                             !* ISSUE A NEW GNDNT
C
C  WE'VE RECEIVED A CONNECT REQUEST - ISSUE AN ACCEPT
C
      CALL     ACCNTW(ACCLUN,IOST,MLBX)
      IF       (IOST(1).NE.1)GOTO 10         !* IF FAILURE ISSUE A NEW
                                             !* GNDNT
C
C  ISSUE A RECEIVE TO PICK UP DATA
C
30    CALL     RECNT(ACCLUN,IOST2,RECSIZ,RECBUF)
      GOTO      10                           !* ISSUE A NEW GNDNT
                                             !* AND WAIT FOR A COMPLETION
C
C  WE COME HERE UPON RECEIVING A DISCONNECT OR ABORT
C
40    CALL     CLSNTW                        !* DEACCESS THE NETWORK
      GOTO     100                           !* AND EXIT
C
C  WE COME HERE IF WE RECEIVE AN INQUIRY
C
50    NUMMES=NUMMES+1                        !* INCREMENT THE MESSAGE
                                             !* COUNT
      IF       (IOST2(1).EQ.1)GOTO 60        !* IF IOST2(1)-1 ALL'S O.K.
C
C  IF THERE WAS AN ERROR SEND BACK AN INTERRUPT MESSAGE WITH
C  MESSAGE NUMBER
C
      INTMES(1)=NUMMES                       !* SEND THE MESSAGE NUMBER
      CALL     XMINT(ACCLUN,IOST,INTSIZ,INTMES)
      GOTO     70                            !* GO ISSUE A NEW RECEIVE
C
C  HERE THE USER CAN LOOK AT THE DATA RECEIVED IN RECBUF AND RESPOND
C  BY PLACING THE REQUESTED INFORMATION INTO SNDDAT
C
C
C  SEND BACK THE DATA AND ISSUE A NEW RECNT
C
60    CALL     SNDNTW(ACCLUN,IOST,SNDSIZ,SNDDAT)
70    CALL     RECNT(ACCLUN,IOST2,RECSIZ,RECBUF)
      GOTO     20                            !* WAIT FOR A COMPLETION
C
C  EXIT PROGRAM
C
100   STOP     'END OF PROGRAM EXECUTION'    !* HALT THE PROGRAM
```

Example 26 Remote Task Control
(Sheet 3 of 3)

```
C
C       THIS PROGRAM IS A VARIATION OF THE PROGRAM THAT APPEARS IN THE
C       RSX DECNET PROGRAMMER'S REFERENCE MANUAL
C
        BYTE NODE(7),BUFFER(128),IDENT(30),ICHAR(3)
        INTEGER ISTAT(2),JSTAT(2),KSTAT(2),LSTAT(2),MSTAT(3),NSTAT(2)
        DATA ICHAR/'A','V','F'/
C
C       GET USER ID, PASSWORD AND ACCOUNT
C
        TYPE 666
666     FORMAT ('$ENTER USER I.D.: ')
        ACCEPT 779,ICNT,(IDENT(I),I=1,ICNT)
        IDENT(ICNT+1)=0
        TYPE 667
667     FORMAT ('$ENTER PASSWORD: ')
        ACCEPT 779,ICNT9,(IDENT(I),I=ICNT+2,ICNT+2+ICNT9-1)
        IDENT(ICNT+2+ICNT9)=0
        TYPE 668
668     FORMAT ('$ENTER ACCOUNT NUMBER: ')
        K=ICNT+1+ICNT9+2
        ACCEPT 779,ICNT8,(IDENT(I),I=K,K+ICNT8-1)
779     FORMAT (Q,10A1)
        TYPE *, 'IDENT = ',(IDENT(I),I=1,15)
        TYPE *, (IDENT(I),I=16,30)
C
C       GET REMOTE NODE NAME LOCAL AND REMOTE FILE SPECS
C
95      TYPE 100
100     FORMAT ('$ENTER REMOTE NODE NAME (6 CHAR. MAX.):')
        ACCEPT 110,ICNT3,(NODE(I),I=1,ICNT3)
110     FORMAT (Q,6A1)
        IF (ICNT3-6) 115,115,95
115     TYPE *,'NODE NAME =',(NODE(I),I=1,7)
        TYPE 120
120     FORMAT ('$ENTER FILE SPEC. OF REMOTE FILE FOR APPEND:')
        ACCEPT 130,ICNT1,(BUFFER(I),I=1,ICNT1)
130     FORMAT (Q,64A1)
        BUFFER(ICNT1+1)=0
        TYPE 140
140     FORMAT ('$ENTER FILE SPEC. OF LOCAL FILE TO BE APPENDED:')
        ACCEPT 150,ICNT2,(BUFFER(I),I=64,63+ICNT2)
150     FORMAT (Q,64A1)
        BUFFER(ICNT2+64)=0
C
C       CREATE NETWORK MAILBOX FOR ONLY ONE LINK
C
        CALL OPNNTW (2,LSTAT,MSTAT,1)
        IF (LSTAT(1)-1) 907,160,907
C
C       OPEN LOCAL AND REMOTE FILES
C
160     OPEN (UNIT=4,NAME=BUFFER(64),TYPE='OLD',ERR=901)
        CALL OPANFW (1,ISTAT,NODE,IDENT,BUFFER,ICHAR,LENGTH,IBLOCK)
        IF (ISTAT(1)-1) 908,200,908
C
C       READ RECORDS FROM LOCAL FILE AND WRITE THEM TO REMOTE FILE
C
200     READ (4,210,END=300,ERR=902) ICNT3,(BUFFER(I),I=1,ICNT3)
210     FORMAT (Q,128A1)
        CALL PUTNFW (1,JSTAT,ICNT3,BUFFER)
        IF (JSTAT(1)-1) 903,200,903
C
C       EOF FOUND -- CLOSE BOTH FILES AND NET
C
```

Example 27 Remote File Access
(Sheet 1 of 5)

```
300     CLOSE (UNIT=4,ERR=904)
        CALL CLSNFW (1,KSTAT)
        IF (KSTAT(1)-1) 905,310,905
310     CALL CLSNTW (NSTAT)
        IF (NSTAT(1)-1) 906,320,906
320     STOP 'APPEND O.K.'
C
C       ERROR HALTS
C
901     TYPE *, (BUFFER(I),I=1,63)
        TYPE *, (BUFFER(I),I=64,128)
        STOP 'CAN NOT OPEN LOCAL FILE'
902     STOP 'READ ERROR FROM LOCAL FILE'
903     TYPE *,'STATUS =',JSTAT(1)
        STOP 'WRITE ERROR FROM REMOTE FILE'
904     STOP 'CAN NOT CLOSE LOCAL FILE'
905     TYPE *,'STATUS =',KSTAT(1)
        STOP 'CAN NOT CLOSE REMOTE FILE'
906     STOP 'CAN NOT CLOSE NETWORK'
907     TYPE *,'STATUS =', LSTAT(1)
        STOP 'MAILBOX CREATION ERROR'
908     TYPE *,'STATUS =' ,ISTAT(1),ISTAT(2)
        STOP 'CAN NOT OPEN REMOTE FILE'
C
        END
C       read from a remote file and write to a remote file usins
C       decnet fortran remote file access support

C       declare the necessary data structures

        DIMENSION       IARRAY(256),ISTAT(2),IBLK(1),LNTH(1)
        COMMON          ISTAT
        LOGICAL*1 IDINFO(13),ICHARS(3)
        LOGICAL EOF
        DATA EOF/.FALSE./

C       ASCIZ strinss for user ID, password and account number.

        DATA IDINFO/'J','O','E',0,'P','R'.'I','V',0,'4','0','4',0/

C       array containins mode, record type and carriase control information

        DATA ICHARS/'I','V','T'/

c       declare network task

        CALL    OPNNTW(7,ISTAT,,2)
        CALL    CKSTAT

C       open two files, one for input and one for output.
C       both of these files exist on a remote node.

C       open file for input

        CALL    OPRNFW(1,ISTAT,'IASNOD',IDINFO,'
        [133,224]NET.TST',ICHARS,LNTH,)
        CALL    CKSTAT

C       open file for output

        CALL    OPWNFW(2,ISTAT,'IASNOD',IDINFO,'
        [133,224]NEWNET.TST',ICHARS,LNTH,)
        CALL    CKSTAT

C       once the files are successfully opened, we may transfer records.
C       the file associated with lun 1 is opened for readins, the
```

Example 27 Remote File Access
(Sheet 2 of 5)

```
C       file on lun 2 is opened for writing.

C       transfer files
        DO 30 I=1,100,1

C       get a record from the input file.

        CALL    GETNFW(1,ISTAT,256,IARRAY)
C       status code 050047 is end of file
        IF (ISTAT(1) .NE. 1 .AND. ISTAT(2) .NE. "050047) GO TO 40
        LNTH(1) = ISTAT(2)

C       check for end of file

        IF (ISTAT(1) .NE. 1 .AND. ISTAT(2) .EQ. "050047) EOF=.TRUE.
        IF      (EOF .EQ. .TRUE.) GO TO 50
        CALL    PUTNFW(2,ISTAT,LNTH,IARRAY)
        IF (ISTAT(1) .NE. 1 .AND. ISTAT(2) .NE. "050047) GO TO 40
30      CONTINUE
40      PRINT 41
41      FORMAT (1H1,'READ OR WRITE ERROR OCCURRED')

C       close files
C       now that files have been transferred we may close
C       the files

50      DO 55 I=1,2,1
        CALL    CLSNFW(I,ISTAT)
55      CONTINUE

        IF      (EOF .NE. .TRUE.) GO TO 60
        PRINT   57
57      FORMAT(1H1,'END OF FILE REACHED, FILES CLOSED')

60      STOP
        END

        SUBROUTINE      CKSTAT
        DIMENSION       ISTAT(2)
        COMMON          ISTAT

        IF      (ISTAT(1) .EQ.1) GO TO 10
5       FORMAT(1H1,'OPEN ERROR')
        PRINT   5
        STOP
10      RETURN
        END
```

Example 27 Remote File Access
(Sheet 3 of 5)

```
C
C  RUNABO.FTN
C
C        - USES THE DECNET TASK CONTROL CALLS TO RUN OR ABORT A
C          TASK ON THE SPECIFIED NODE (LOCAL OR REMOTE).
C        - TASK 'TCL'  MUST BE INSTALLED AT THE TARGET NODE.s
C        - SAMPLE COMMANDS TO DEVELOP THIS TASK FOLLOW
C
C              >FOR RUNABO,RUNABO=RUNABO
C              >TKB RUNABO,RUNABO=RUNABO,[1,1]NETFOR/LB
C              >INS RUNABO/TASK=...CTL
C
      LOGICAL*1 ANSWER,RUN,ABO,TARTSK(6),TARNOD(6),PASSWD(8),USERID(16)
      LOGICAL*1 ACCNT(16)
      INTEGER STATUS(2),STAT
      INTEGER*2 MSTAT(3)
      DATA RUN/'R'/,ABO/'A'/
C
C  CREATE NETWORK DATA QUEUE
C
      CALL OPNNTW(1,STATUS,MSTAT)
      IF (STATUS(1) .NE. 1) WRITE(5,8)STATUS(1)
C
C  PROMPT FOR TARGET NODE, AND TARGET TASK.
C
 10   WRITE(5,1)
  1   FORMAT(5X,$'TARGET NODE ? :')
      READ(5,2,END=999)ICNT1,TARNOD
  2   FORMAT(Q,16A1)
C
      WRITE(5,3)
  3   FORMAT(5X,$'TARGET TASK ? :')
      READ(5,2,END=999)ICNT2,TARTSK
C
C PROMPT FOR ACCESS CONTROL INFORMATION
C
      WRITE(5,50)
 50   FORMAT(5X,$'TARGET USER I.D. ?:')
      READ (5,2,END=999)ICNT4,USERID
C
      WRITE(5,4)
  4   FORMAT(5X,$'TARGET PASSWORD ?:')
      READ (5,2,END=999)ICNT3,PASSWD
C
      WRITE(5,11)
 11   FORMAT(5X,$'TARGET ACCOUNT NUMBER ?:')
      READ (5,2,END=999)ICNT5,ACCNT
C
      WRITE(5,6)
  6   FORMAT(5X,$'RUN (R) OR ABORT (A) ? :')
      READ(5,7,END=999)ANSWER
  7   FORMAT(A1)
```

Example 27 Remote File Access
(Sheet 4 of 5)

```
C
C  DECIDE WHETHER TO CALL BACUSR
C
        IF (ICNT4 .EQ. 0 .AND. ICNT5 .EQ. 0) GO TO 70
        CALL BACUSR (STAT,ICNT4,USERID,ICNT5,ACCNT)
        IF (STAT .EQ. .TRUE.) GO TO 70
        WRITE(5,80)STAT
  80    FORMAT (' ERROR BUILDING CONNECT BLOCK ')
        GOTO 10
C
C  DECIDE WHETHER TO RUN OR ABORT THE TASK.
C
  70    IF (ANSWER .EQ. ABO) GOTO 20
        IF (ANSWER .EQ. RUN) GOTO 30
        GOTO 999
C
C  ABORT THE TASK, AND PRINT STATUS.
  20    WRITE (5,100)
 100    FORMAT (5X,$'IF A PRIVILEGED PASSWORD IS USED, ENTER 0. IDENT ?:')
        READ (5,110)IDENT
 110    FORMAT(I6)
        CALL ABONCW (2,STATUS,ICNT1,TARNOD,ICNT3,PASSWD,ICNT2,TARTSK,IDENT)
C
        WRITE(5,8)STATUS(1)
   8    FORMAT(' STATUS = ',I7)
        GOTO 10
C
C  RUN THE TASK, AND PRINT STATUS.
C
  30    CALL RUNNCW(2,STATUS,ICNT1,TARNOD,ICNT3,PASSWD,ICNT2,TARTSK,IDENT)
        WRITE (5,8)STATUS(1)
        IF (STATUS(1) .EQ. 1) WRITE (5,90) IDENT
  90    FORMAT (' THE IDENT IS ',I6)


        GOTO 10
 999    STOP
        END
>
```

Example 27 Remote File Access
(Sheet 5 of 5)

(74 Blank Pg)

Should be 18pt

EXERCISES

diff. type size ~~from~~ following head! ?

Match the following DECnet utilities and RSX supported languages to the appropriate functions that they can perform/support. Each item may be used once, more than once, or not at all.

1. RMT
2. FTS
3. NFT
4. TLK
5. FORTRAN-77
6. MACRO-11
7. LSN
8. COBOL
9. FAL
10. NS:

a. ______ Used to transfer files in a spooled environment.

b. ______ Used for terminal-to-terminal communication on the local node.

c. ______ Supports only task-to-task and DLX communication.

d. ______ Is the general purpose server task for remote file access.

e. ______ Allows remote terminal communication between two DECnet-RSX nodes.

f. ______ Is the programming interface for all task-to-task communication.

g. ______ Can list remote directories immediately to the TI: device.

h. ______ Is the requester task for the LSN task.

i. ______ Will allow remote task control.

j. ______ Allows input from a command file.

right-hand pg.

odd-numbered pg.

Should be 18pt.

SOLUTIONS diff type size?

Match the following DECnet utilities and RSX supported languages to the appropriate functions that they can perform/support. Each item may be used once, more than once, or not at all.

1. RMT
2. FTS
3. NFT
4. TLK
5. FORTRAN-77
6. MACRO-11
7. LSN
8. COBOL
9. FAL
10. NS:

a. 2 Used to transfer files in a spooled environment.

b. 4 Used for terminal-to-terminal communication on the local node.

c. 6 Supports only task-to-task and DLX communication.

d. 9 Is the general purpose server task for remote file access.

e. 1 Allows remote terminal communication between two DECnet-RSX nodes.

f. 10 Is the programming interface for all task-to-task communication.

g. 3 Can list remote directories immediately to the TI: device.

h. 4 Is the requester task for the LSN task.

i. 5 Will allow remote task control.

j. 2,3,4 Allows input from a command file.

Answers are to be underlined

LAB EXERCISES diff. type size? Should be 18pt.

1. Use the TLK utility to send messages to a remote terminal using the following modes of operation.

   a. Single message mode

   b. Dialogue mode (use video screen option, when possible)

   c. Command file mode

2. Establish a TLK dialogue mode session with a remote terminal, then try to use command file mode from both ends (local and remote) of the logical link. Check if it works from the remote terminal.

3. Create a command file which contains the MCR BRO request and execute it using NFT on:

   a. The local node

   b. Remote DECnet-11M node

   c. Remote DECnet-11M-PLUS node

4. Repeat Lab Exercise 3 using FTS with /LO and /AF switches. Analyze the FTSSYS.LOG file for possible errors.

5. Use RMT to establish connection to a remote DECnet-11M/M-PLUS node; log onto the remote system. Run DEV/LOG (MCR) or SHO DEV (DCL) displays to see which HT: device you are using. Run RMD (MCR) or SHO SYS (DCL) displays to see the available memory allocation. Note how often the display is updated (the slower the communication path between the local node and the remote node, the longer the time intervals are).

6. Write two tasks in MACRO-11 or FORTRAN 77 that establish a logical link to each other. (Use optional data fields in CON and ACC calls to send short user messages. Display those messages on the local and remote terminals to make sure that the logical link was actually established.)

7. Enhance the two tasks you have created in Lab Exercises 6 to include SND, REC, and XMI calls. Use corresponding examples in reference literature for additional information.

SOLUTIONS TO LAB EXERCISES

diff type size? Should be 18pt.

1.

   a. See example 1

   b. See example 2,3

   c. See example 4

2. TLK cannot communicate with remote terminals (HT:)

3.

   a. NFT>COMMAND.CMD/EX

   b. NFT>NØ11M::COMMAND.CMD/EX

   c. NFT>NØ11MP::COMMAND.CMD/EX

NOTE

The MCM task uses TTO: as a default; to execute the command procedure, be sure this terminal is logged in under a privileged account.

4.

   a. FTS>COMMAND/CMD/EX/AF:DD-MMM-YY:HH:MM/LO:LOG.FIL

   b. FTS>NO11M::COMMAND.CMD/EX/AF:DD-MMM-YY:HH:MM/LO

   c. FTS>NO11MP::COMMAND.CMD/EX/AF::DD-MMM-YY:HH:MM/LO

5. RMT
   Host:NODE

6. Example Program Listings in the Appendix 6

7. Same as 6

Solutions are to be circled?

text cannot be moved up by number? No. Then OK as is.

CONTENTS

FIGURES

TABLES

EXAMPLES

INTRODUCTION

DECnet software has an extensive Network Management interface that allows a user to configure, monitor, and test the network. This interface allows a network manager to access the Network Management Layer on the local or remote node by assigning an EXECUTOR to that node.

This module details the Network Management interface. For comparison, information on systems other than DECnet-RSX is included.

OBJECTIVES

To manage a network, the specialist must be able to:

- Use Network Control Program (NCP) commands fluently.
- Bring network circuits up and down.
- Change node, circuit, and line parameters.
- Execute node, circuit, and line loopback tests.
- Retrieve error information from counters displays and the event logging facility, and correspond those to the network performance.
- Execute performance tests.
- Retrieve information from the Network Information Display (NTD).

RESOURCES

<u>DECnet-RSX System Manager's Guide</u>

NETWORK MANAGEMENT LAYER

Figure 1 shows the location and interfaces for the Network Management Layer in DIGITAL Network Architecture (DNA).

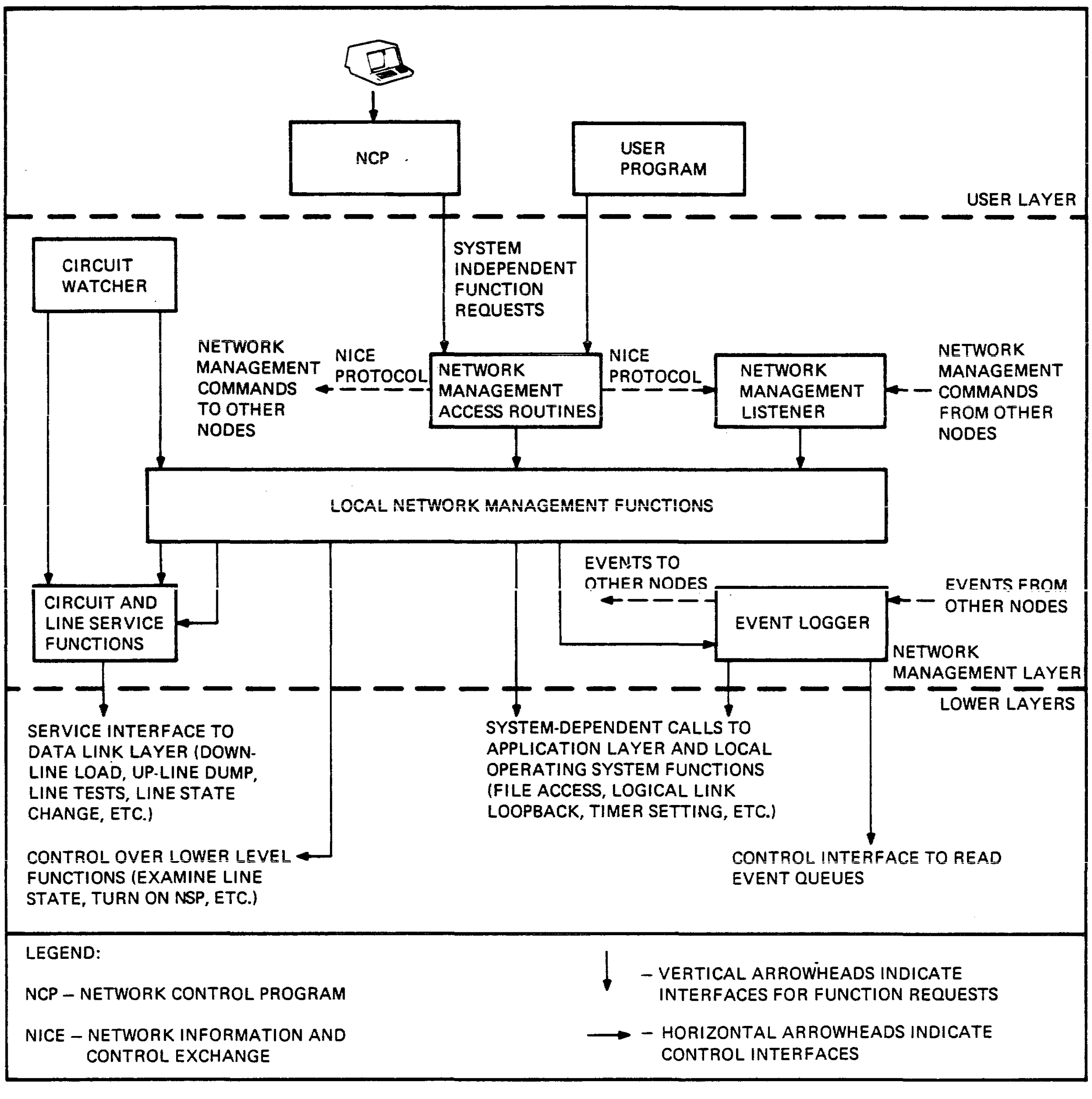


Figure 1 Network Management Layer

BASIC FUNCTIONS

- Configuring a DECnet and/or Packetnet System Interface (PSI) node
- Configuring a network
- Testing nodes and a network
- Controlling node and network operations
- Inspecting and monitoring nodes
- Monitoring network activity

Configuring a DECnet and/or PSI Node

- PREGEN, NETGEN procedures
- Configuration databases
  - Node database - CETAB.MAC file
  - Down-line system loading database
  - Down-line task loading database
- Utilities
  - Network Control Program (NCP)
  - Configuration File Editor (CFE)
  - Virtual Network Processor (VNP)

Configuring a Network

- Down-line system loading
- Modifying routing parameters
- Modifying line and circuit parameters
- Utilities
  - NCP
  - CFE
  - VNP

Testing Nodes and a Network

- Loopback test
- NTEST procedure
- NCP Utility

Controlling Node and Network Operations

- Passwords and access control information
- Down-line task loading and overlaying/up-line checkpointing
- File transfer spooling management
- Object specification
- Utilities
  - NCP
  - CFE
  - VNP
  - FTS Queue Manager (FTQ)
  - Host Task Loader (HLD)
  - Satellite Task Loader (SLD)

Inspecting and Monitoring Nodes and a Network

- Node, circuit, and line counters
- Event logging
- Performance testing
- Trace (PSI only)
- Network dump analysis
- Utilities
  - NCP
  - Network Display (NTD)
  - Data Test Send/Data Test Receive (DTS/DTR) for DECnet
  - Remote Command Terminal Access (RMT/RVT*/RRS*/HOST*)
  - Network Dump Analyzer (NDA)
  - Dump of the Running CEX (CEDUMP)

*Utility is not supported on DECnet-RSX.

NETWORK MANAGEMENT UTILITIES

Table 1 Summary of Network Management and Supporting Utilities

Utility	Operating System						
	VAX/VMS	RSX-11M /M-PLUS	RSX-11S	IAS	RT	RSTS/E	TOPS-20
Network Control Program (NCP)	X	X	X(1)	X	X	X	X
Configuration File Editor (CFE)		X	X	X			
Configuration File Editor (NETGEN)							X
Virtual Network Processor (VNP)		X	X				
Virtual Network Processor (SYSDPY)							X
Network Display Program (NTD)		X	X	X			
Data Test Send/Receive for DECnet (DTS/DTR)	X	X	X	X	X	X	X
Data Test Send/Receive for PSI (XTS/XTR)	X	X					X
Remote M/M-PLUS Command Terminal (RMT)		X	X	X	X(2)		
Remote VAX/VMS Command Terminal (RVT)		X(2)			X(2)		
Remote RSTS/E Command Terminal (RST)		X(2)					

Table 1 Summary of Network Management and Supporting Utilities (Cont)

	Operating System						
Utility	VAX/VMS	RSX-11M /M-PLUS	RSX-11S	IAS	RT	RSTS/E	TOPS-20
Remote TOPS-20 Command Terminal (HOST)		X(2)					X
Remote Command Terminal (NET)						X	
Remote Command Terminal (SET HOST)	X						
System Dump Analyzer (SDA)	X						
Crash Dump Analyzer (CDA)		X	X	X			
System Dump Analyzer (MCBDA)							X
Network Dump Analyzer (NDA)	X(3)	X	X	X			X(3)

Notes

1. Provides a subset of functionality
2. Nonsupported feature
3. Part of SDA or MCBDA

Table 2 NCP Functions and Keywords

	Operating System						
NCP Functions	VAX/VMS	RSX-11M /M-PLUS	RSX-11S	IAS	RT	RSTS/E	TOPS-20
Loading/Unloading (SET/CLEAR/LOAD/ TRIGGER/TELL)							
Load/Unload a Network Software	X	X		X	X	X	X
Load/Unload a Network Line	X	X		X	X	X	X
Load/Unload a Network Circuit	X	X				X	X
Change Line Status/Values	X	X	X(1)	X	X	X	X
Change Circuit Status/Values	X	X	X(1)	X(2)	X(2)	X	X
Load/Unload a Network Process	X	X		X	X	X	X
Trigger the Boot-strap of a Target Node	X	X		X			X
Down-Line System Loading	X	X		X			X
Control Functions (SET/TELL/LOOP)							
Multipoint Lines	X	X			X(3)	X	
Remote Command Execution	X	X		X	X	X	X
Setting and Clearing Executor Parameters	X	X		X	X	X	X

Table 2 NCP Functions and Keywords (Cont)

	Operating System						
NCP Functions	VAX/VMS	RSX-11M /M-PLUS	RSX-11S	IAS	RT	RSTS/E	TOPS-2Ø
Control Functions (SET/TELL/LOOP) (cont)							
Setting and Clearing Passwords	X	X	X	X	X	X	X
Routing	X	X	X	X		X	X
Event Logging	X	X		X		X	X
Loopback	X	X	X	X	X	X	X
Displays (SHOW)							
Line, Circuit, and Node Information	X	X	X	X(2)	X(2)	X	X
Network Process Information		X		X	X		
Network Objects	X	X		X	X	X	X
System Information	X	X	X	X	X	X	X
Logging Information	X	X	X(1)	X		X	X
Counters Information	X	X	X	X	X	X	X
Line Counters	X	X	X	X	X	X	X
Circuit Counters	X	X	X			X	X
Node and Executor Counters	X	X	X(1)	X	X	X	X

Table 2 NCP Functions and Keywords (Cont)

NCP Functions	Operating System						
	VAX/VMS	RSX-11M /M-PLUS	RSX-11S	IAS	RT	RSTS/E	TOPS-20
Node Name Information (SET/CLEAR/ZERO)							
Changing a Node Name	X	X		X	X	X	X
Alias Name Information		X		X	X	X	X

Notes

1. Provides a subset of functionality
2. Concept of a circuit is not implemented
3. Tributary only

```
>
>
>
>
>NCP
NCP>SET HOST ADDR 35
    SET >>HOST ADDR 35
NCP -- Set not accepted, unrecognized command
NCP>
NCP>SET HOST NODE 35
    SET >>HOST NODE 35
NCP -- Set not accepted, unrecognized command
NCP>HELP SET
The format of the SET command is:

   SET component component-options

Component types are:

   ALIAS
   CIRCUIT
   EXECUTOR
   LINE
   LOGGING
   NODE
   OBJECT
   PROCESS
   SYSTEM

For more help use:

   HELP component-type
   HELP SET component-type
```

Example 1 NCP HELP Session for Setting Host Node
(Sheet 1 of 2)

```
NCP>HEK\K\LP SET EXECUTOR
Use the SET EXECUTOR command to create or modify executor node parameters.
The format is:

   SET EXECUTOR parameters

Parameter types are:

        HOST                               STATE
        NODE                               TRANSMIT PASSWORD
        RECEIVE PASSWORD                   VERIFICATION [STATE]
        ROUTING TIMER
        SEGMENT BUFFER SIZE

For more help use:

   HELP EXECUTOR
   HELP SET EXECUTOR parameter-type
NCP>SET EXECUTOR HOST 35
NCP>^Z
>
>
>
>
```

Example 1 NCP HELP Session for Setting Host Node
(Sheet 2 of 2)

DATABASES AFFECTED BY NETWORK MANAGEMENT UTILITIES

Network software has two databases:

- Permanent (file)
- Volatile (core)

If the volatile database is affected by the network management utility, the effect of such change is known and can be used immediately. But if the system or network is restarted, those changes are lost. When changes are made to the permanent database, they become known to the system and can be used only after restarting the system or network.

Table 3 Volatile and Permanent Databases and Corresponding Utilities

		Databases for Systems						
Utility	Command	VAX/ VMS	RSX-11M /M-PLUS	RSX- 11S	IAS	RT	RSTS/E	TOPS-2Ø
NCP	SET/CLEAR SHOW/ZERO	Vol.	Vol.	Vol.	Vol.	Vol.	Vol.	Vol.
NCP	DEFINE/ PURGE/LIST	Perm.				Perm.	Perm.	Perm.
CFE	DEFINE/ PURGE/LIST		Perm.	Perm. (1)	Perm.			
NETGEN	DEFINE/ PURGE/LIST							Perm.
VNP	SET/CLEAR/ SHOW	Vol. (4)	Vol. (2),(3)	Vol. (4)				
NETGEN	DEFINE/ PURGE/LIST							Vol. (4)

Notes

1. Used on the host system (VAX/VMS, RSX-11M/M-PLUS, IAS) to modify the permanent database for RSX-11S system image before running the Virtual Network Processor (VNP).

2. Used to modify the RSX-11M/S system image to activate a network on the system image file.

3. VNP cannot be used against the RSX-11M-PLUS system image.

4. VNP (NETGEN) in the host (VAX/VMS, RSX-11M/M-PLUS, TOPS-2Ø) is used to modify the RSX-11S system image.

NODE, CIRCUIT, AND LINE IDENTIFICATION

- A DECnet node is identified by node address, which is unique in the network, and by name, which is local to the node (executor) and/or to a user (alias). The maximum node address is 1023.

- A PSI node is identified by a 12-character Data Terminal Equipment (DTE) address (supplied by the PPSN vendor), and by name, which is local to the node. The first 4 characters in the DTE address is an international code for a PPSN.

  Example:

  3110-TELENET
  2342-PSS

- A circuit is a logical path between two adjacent nodes or DTEs. A DECnet circuit is identified as:

  devicetype-controllernumber[-unitnumber].[tributary-number]

  Example:

  DMC-1
  DMP-1.3
  DLM-0.1
  KDP-0-0.3

- A PSI Permanent Virtual Circuit (PVC) is identified by name.

  Example:

  ABCDEF
  DLM-1.0 (Special case for DECnet using PSI PVC)

- A line is a physical path between two adjacent nodes. It is identified as:

  devicetype-controllernumber[-unitnumber]

  Example:

  DUP-Ø

  SDP-1

  KMX-Ø-1

  DMP-1

DECnet Node Types

- Executor -- A node that a given NCP command is executed on
- Command -- A node that an NCP command is initiated from
- Host -- A node that a particular application depends on (for example, remote command terminal, down-line system loading, down-line task loading, and so on)
- Target -- A node that the host addresses to

DECnet and PSI Types

- Local -- A node that a particular procedure is initiated from (operator command, setting up Switched Virtual Circuit (SVC), and so on)
- Remote -- A node that the local request addresses to

EXECUTOR, CIRCUIT, AND LINE STATES

- Executor states
  - ON -- Normal network activity is allowed
  - OFF -- No network activity is allowed
  - SHUT -- Temporary state before the OFF state; only existing network activity is allowed (for example, existing logical links are still active, but no new logical link is allowed)
- Circuit states (Table 4)
- Line states
  - ON -- Line is active
  - OFF -- Line is inactive (stopped)
  - SERVICE -- Line is in maintenance mode
  - CLEARED -- Line is not loaded

Table 4 Circuit States and Substates

State	Substate	Meaning
OFF	None	The circuit is not usable.
ON	- RUNNING	The circuit is in normal use.
	- STARTING	The circuit is in the DDCMP initialization cycle.
	- REFLECTING	The circuit is in use for passive loopback testing.
	- AUTOSERVICE	The circuit is reserved for automatic service use.
	- AUTOLOADING	The circuit is in use for automatic loading.
	- AUTODUMPING	The circuit is in use for automatic dumping.
	- AUTOTRIGGERING	The circuit is in use for automatic triggering.
	- LOADING	The circuit is in use for loading.
	- DUMPING	The circuit is in use for dumping.
	- LOOPING	The circuit is in use for active circuit loopback testing.
	- TRIGGERING	The circuit is in use for triggering.
	- SYNCHRONIZING	The circuit is trying to synchronize with its adjacency.

Table 4 Circuit States and Substates (Cont)

State	Substate	Meaning
SERVICE	- IDLE	The circuit is reserved for an active service function.
	- REFLECTING	The circuit is in use for passive loopback testing.
	- LOADING	The circuit is in use for loading.
	- DUMPING	The circuit is in use for dumping.
	- LOOPING	The circuit is in use for loopback testing.
	- TRIGGERING	The circuit in use for triggering.
CLEARED		The corresponding line is not loaded.

EXAMPLES OF NCP USAGE FOR NODE, CIRCUIT, AND LINE

All NCP SHOW commands cause output to be displayed at the user's terminal. However, you can direct output to any output file at the local node. To do so, append the keyword TO followed by the output file specification. For example, to direct output to the file NANOOK.LST, type:

```
NCP> SHOW NODE NANOOK TO NANOOK.LST
```

If the specified output file does not already exist, NCP creates a new file. If the specified file already exists, NCP appends the output to that file. NCP does not create a new version of the existing output file.

Active nodes summary as of 1-NOV-83 13:51:16

Executor node = 62 (CASTOR)

State = On, Identification = RSX-11M-PLUS V2.1

Remote Node	State	Active Links	Delay	Circuit	Next Node
1 (VORTEX)	Reachable	0	4	UNA-0	1 (VORTEX)
21 (ORAC)	Reachable	0	4	UNA-0	21 (ORAC)
30 (PHOEBE)	Reachable			UNA-0	30 (PHOEBE)
41 (SADE)	Reachable			UNA-0	41 (SADE)
42 (GNAT)	Reachable			UNA-0	152 (ALGOL)
43 (SHALOM)	Reachable			UNA-0	151 (KOBAL)
44 (WEBSTR)	Reachable			UNA-0	44 (WEBSTR)
45 (MARAT)	Reachable			UNA-0	45 (MARAT)
47 (BANZAI)	Reachable			UNA-0	47 (BANZAI)
60 (KERMIT)	Reachable	0	4	UNA-0	60 (KERMIT)
61 (QUASAR)	Reachable	0	4	UNA-0	61 (QUASAR)
65 (RSXHUB)	Reachable			UNA-0	95 (BEING)
66 (THEFLY)	Reachable			UNA-0	66 (THEFLY)
67 (RUNE)	Reachable	0	4	UNA-0	67 (RUNE)
68 (BAXTER)	Reachable			UNA-0	68 (BAXTER)
69 (ADVISE)	Reachable			UNA-0	69 (ADVISE)
70 (GUIDO)	Reachable			UNA-0	70 (GUIDO)
71 (VAXUUM)	Reachable			UNA-0	71 (VAXUUM)
76 (SPIT20)	Reachable			UNA-0	185 (EIFFEL)
77 (ZK20FE)	Reachable			UNA-0	185 (EIFFEL)
81 (TITAN)	Reachable			UNA-0	81 (TITAN)
90 (METEOR)	Reachable	0	10	UNA-0	90 (METEOR)
94 (ROYAL)	Reachable			UNA-0	94 (ROYAL)
95 (BEING)	Reachable	0	4	UNA-0	95 (BEING)
100 (METOO)	Reachable			UNA-0	100 (METOO)
102 (LEGO)	Reachable			UNA-0	102 (LEGO)
103 (POSSUM)	Reachable			UNA-0	177 (MAGIC)
105 (ATOM)	Reachable			UNA-0	213 (WOMBAT)
108 (BOOKIE)	Reachable			UNA-0	108 (BOOKIE)

Example 2 NCP> SHOW ACTIVE NODES

NOTE

If your node is a nonrouting node and you wish to determine which nodes are reachable, you must have the SHOW ACTIVE NODE command executed at an adjacent full-routing node.

```
Circuit summary as of 1-NOV-83 13:52:12

Circuit = UNA-0

   State = On
```

Example 3 NCP> SHOW CIRCUIT UNA-0

```
Known lines summary as of 1-NOV-83 13:51:46

Line                State

UNA-0               On
PCL-0               Cleared
DMC-0               Cleared
DMC-1               Cleared
```

Example 4 NCP> SHOW KNOWN LINES

```
All aliases summary as of 1-NOV-83 13:52:33

Alias
Name     Scope              Destination

A        Terminal TT63:     ALIEN/PETTENGILL/...
M        Terminal TT63:     QUASAR/PETTENGILL/...
K        Terminal TT63:     KERMIT/PETTENGILL/...
R        Terminal TT63:     RUNE/PETTENGILL/...
CASTOR   Global             CASTOR/DECNET/...
DOC      Global             DOC/DECNET/...
KERMIT   Global             KERMIT/DECNET/...
POLLUX   Global             POLLUX/DECNET/...
QUASAR   Global             QUASAR/DECNET/...
RSXHUB   Global             RSXHUB/DECNET/...
RUNE     Global             RUNE/DECNET/...
```

Example 5 NCP> SHOW ALL ALIASES

Global aliases are aliases set by the system manager that may be used by all tasks and users. Local aliases are those set by nonprivileged user and apply only at the user's terminal.

LOOPBACK TESTING

There are two types of loopback tests for the DECnet:

- Node level loopback test
- Circuit level loopback test

Figures 2 and 3 illustrate the node and circuit level loopback test components.

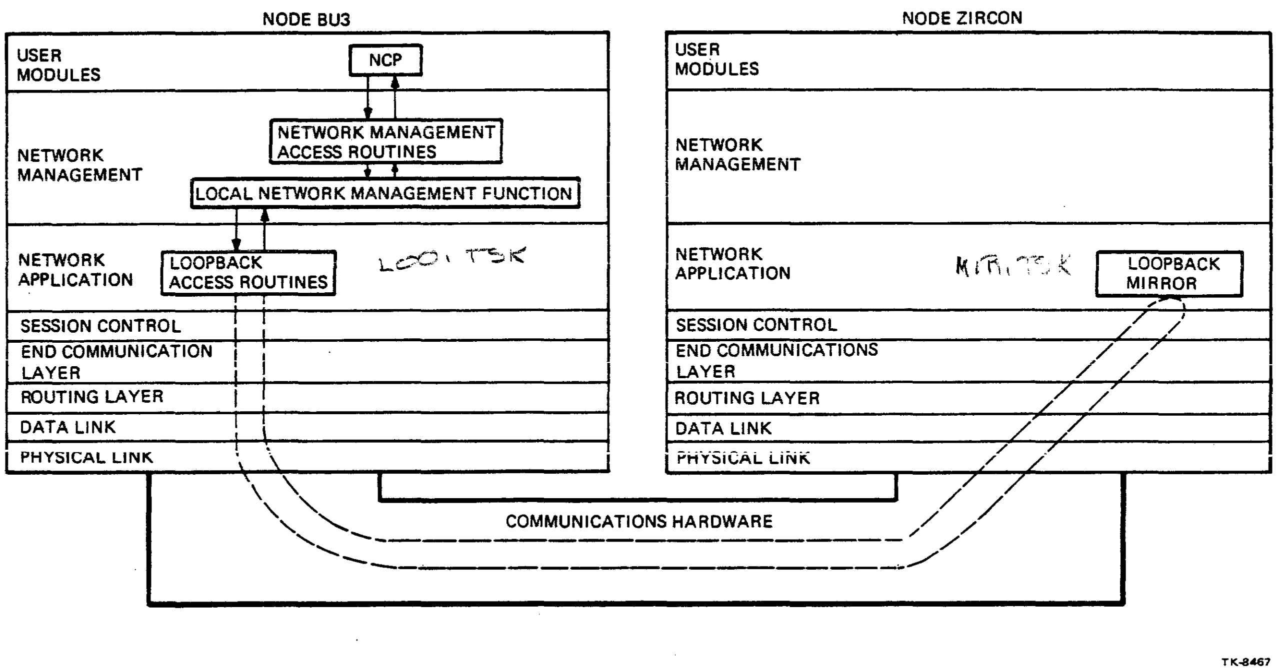


Figure 2 Node Level Loopback Test Components
(NCP> LOOP NODE ZIRCON)

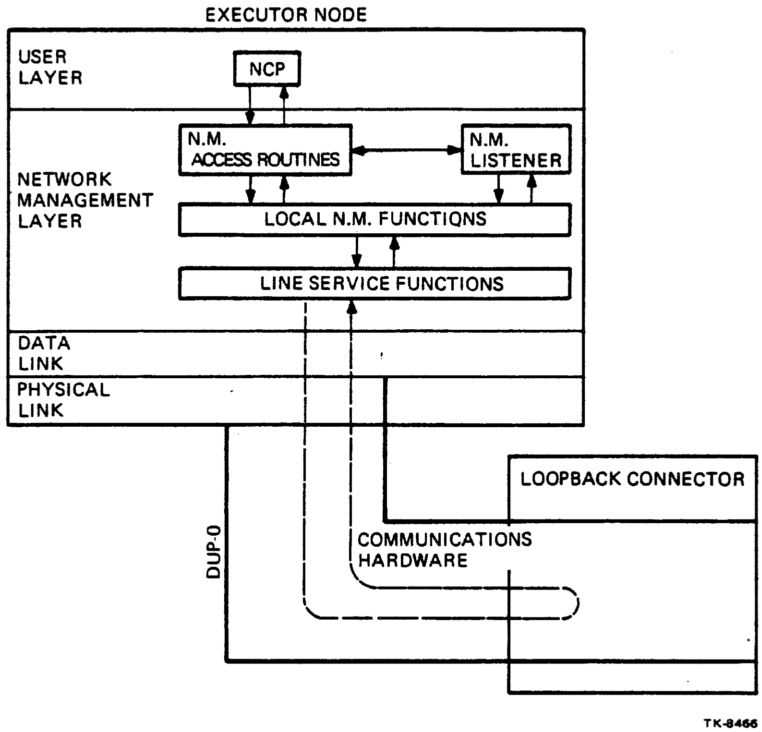


Figure 3 Circuit Level Loopback Test Components (NCP> LOOP CIRCUIT DMC-Ø)

NOTE

Because of physical buffer limitations a DMC-11 device cannot successfully do a loop circuit directly. The user must do the following:

1. NCP> SET NODE XXX CIRCUIT DMC-Ø

   NCP> LOOP NODE XXX

2. NCP> LOOP CIRCUIT DMC-Ø COUNT "<5Ø" (any number less than 5Ø)

EVENT LOGGING AND MONITORING

Figure 4 shows the event logging interface in DECnet.

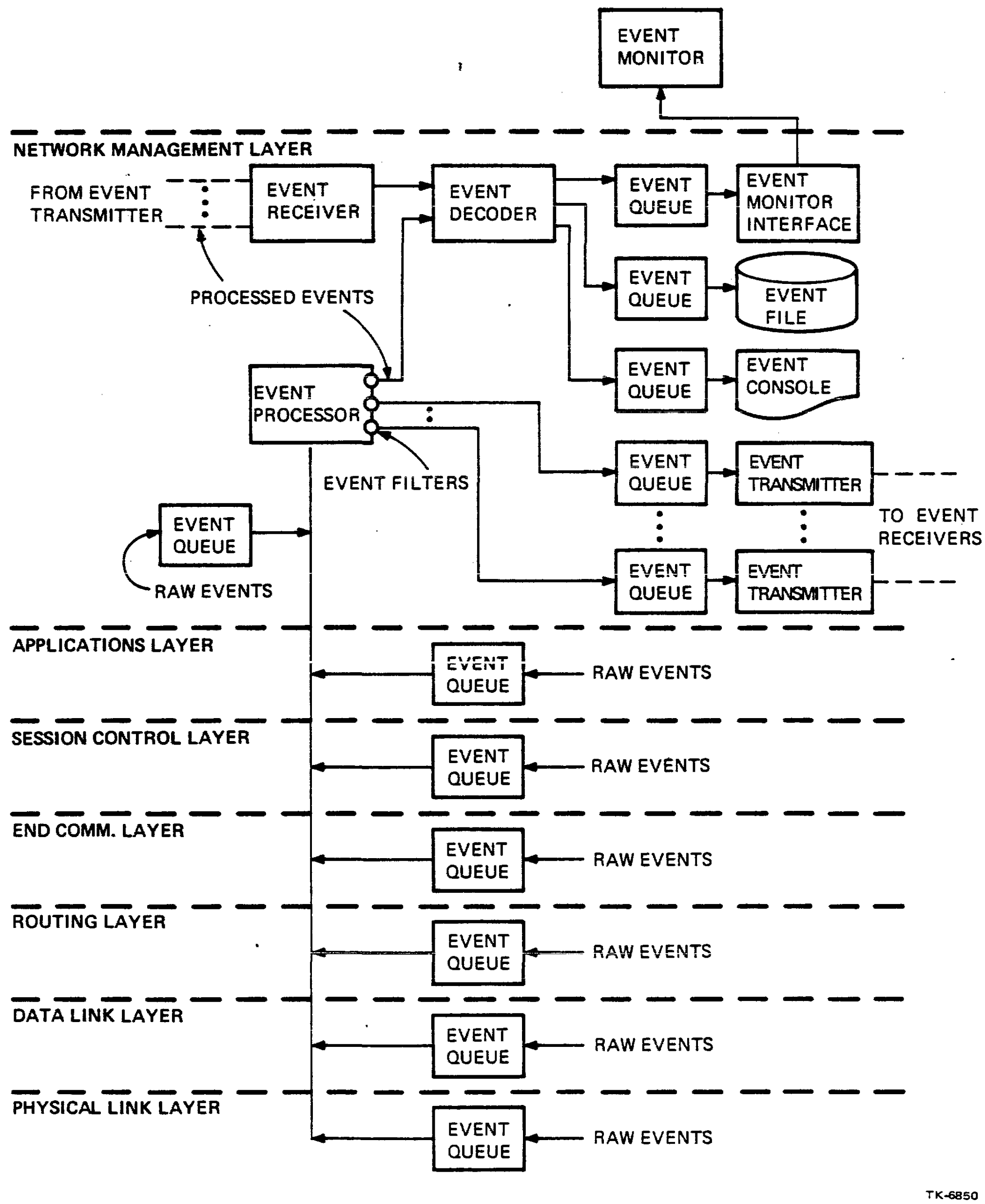


Figure 4 Event Logging Architectural Model

Table 5 lists the supported event classes.

Table 5 Network Event Classes

Event Class	Source of the Event
0	Network Management Layer
1*	Applications Layer
2	Session Control Layer
3	Network Services Layer
4	Routing Layer
5	Data Link Layer
6	Physical Link Layer
7-31*	Reserved for other common classes
32-63*	Reserved for RSTS/E systems
64-95	RSX-11 system specific
96-127*	Reserved for TOPS-20 systems
128-159*	Reserved for VMS systems
160-479*	Reserved for future use

*DECnet-RSX does not log events for these classes. However, processed events in these classes from other remote nodes are logged.

NOTE

For additional information on network events, refer to Appendices C and D of the DECnet-RSX System Manager's Guide.

Example 6 shows some events as they appear on the system console.

```
>
>
>
>
>

>


 Event type 4.7, Circuit down - circuit fault
Occurred 23-NOV-83 14:27:20 on node   65 (RSXHUB)
Circuit DMC-1
Line synchronization lost


 Event type 4.7, Circuit down - circuit fault
Occurred 23-NOV-83 14:43:54 on node   65 (RSXHUB)
Circuit DMC-1
Adjacency listener receive timeout



 Event type 4.7, Circuit down - circuit fault
Occurred 23-NOV-83 19:58:26 on node   65 (RSXHUB)
Circuit DMC-0
Adjacency listener receive timeout

 Event type 4.7, Circuit down - circuit fault
Occurred 28-NOV-83 08:22:18 on node   65 (RSXHUB)
Circuit DMC-1
Adjacency listener receive timeout

 Event type 4.7, Circuit down - circuit fault
Occurred 28-NOV-83 11:54:41 on node   65 (RSXHUB)
Circuit DMC-2
Line synchronization lost

 Event type 4.7, Circuit down - circuit fault
Occurred 28-NOV-83 11:54:51 on node   65 (RSXHUB)
Circuit DMC-2
```

Example 6 Event Logging on CO:

NODE, CIRCUIT, AND LINE COUNTERS

Some events, when detected, appear in the node, circuit, and line counters.

```
Active nodes counters as of 1-NOV-83 13:53:08

Executor node = 62  (CASTOR)

      39106  Seconds since last zeroed
      50489  Bytes received
      50376  Bytes sent
       6897  Messages received
       6967  Messages sent
         36  Connects received
         36  Connects sent
          0  Response timeouts
          0  Received connect resource errors
          8  Total maximum logical links active
          9  Aged packet loss
          0  Node unreachable packet loss
          0  Node out-of-range packet loss
          1  Oversized packet loss
          0  Packet format error
          0  Partial routing update loss
          0  Verification reject
          4  Node maximum logical links active
          0  Total received connect resource errors

Remote node = 21 (ORAC)

     >65534  Seconds since last zeroed
    1722088  Bytes received
       6169  Bytes sent
       6988  Messages received
       6976  Messages sent
         11  Connects received
          2  Connects sent
          0  Response timeouts
          0  Received connect resource errors
          2  Node maximum logical links active

Remote node = 30 (PHOEBE)

   No information

Remote node = 41 (SADE)

   No information

Remote node = 42 (GNAT)

   No information
```

Example 7 NCP> SHOW ACTIVE NODE COUNTERS

```
Active circuits counters as of 1-NOV-83 13:53:35

Circuit = UNA-0

      39438  Seconds since last zeroed
      94005  Terminating packets received
     100626  Originating packets sent
          0  Corruption loss
        752  Transit packets received
        751  Transit packets sent
          0  Transit congestion loss
          0  Circuit down
          0  Initialization failure
   56993662  Bytes received
    6314583  Bytes sent
     574818  Data blocks received
     113205  Data blocks sent
          0  User buffer unavailable
```

Example 8 NCP> SHOW ACTIVE CIRCUIT COUNTERS

```
Line counters as of 1-NOV-83 13:53:58

Line = UNA-0

      39452  Seconds since last zeroed
   62896709  Bytes received
    7689281  Bytes sent
   55791183  Multicast bytes received
     578286  Data blocks received
     113355  Data blocks sent
     483518  Multicast blocks received
       2209  Blocks sent, initially deferred
        276  Blocks sent, single collision
        140  Blocks sent, multiple collision
          1  Send failure, including:
                Frame too long
          0  Collision detect check failure
          9  Receive failure, including:
                Block check error
                Framing error
       2812  Unrecognized frame destination
          0  Data overrun
          0  System buffer unavailable
```

Example 9 NCP> SHOW LINE UNA-0 COUNTERS

Performance Analysis and Testing NTD Facility

The Network Information Display (NTD) program allows you to observe and evaluate allocation of the network resources on the designated node (local or remote) on a periodic basis. Example 1Ø shows an NTD display.

```
    >
    >
    >
    >NTD                                 (3)
(1) 15-DEC-83 10:44:41 (2) Node: ZIRCON(  231) ZIRCON_ENCRUSTED_TWEEZERS
                  (6) (7)    RSX-11M, Non-Routing Node, DECnet V4.0
(4) LNKS (5) 2 / 10 XXXXXXXX!------------------------------   Alloc Fails:
(10) CCB     4 / 10 XXXXXXXXXXXXXXXX!-----------------------   (9) 0
(11) SDB     0 / 11 ----------------------------------------       0
(12) LDB     4 /  9 XXXXXXXXXXXXX     !---------------------       0 : (13) 0
                                     (8)
          Known Circuits:                           Network Tasks:
 Circuit  Cost Size  Adj Node/Status          Task   TI   Links MBX XMT RCV

(15) DMC-0  (16) 5  576  RSXHUB(   65)    (18) NTD...  COO: (19) 1   0   0   0
              (17)    (14)                     ...NTD  TTO:      1   0   0   1
```

Example 1Ø NTD Display for a Node

NTD displays the following information:

1. Correct date and time
2. Node name and address
3. Node identification
4. Logical links
5. Number of resources currently in use
6. Total number of resources generated in the system
7. Number of resources in use (scaled)
8. Highest resource usage recorded
9. Number of allocation failures for each resource type
10. Communication control buffers
11. Small data buffers
12. Large data buffers
13. Total allocation failures
14. Remote adjacent node name(s) and address(es)
15. Circuit(s) in or available for use
16. Known cost to the adjacent node
17. Maximum message size
18. Currently active network tasks
19. Information about currently active network tasks

DTS/DTR Tests

A popular technique used for performance analysis is Data Test Send/Data Test Receive (DTS/DTR) testing.

DTS/DTR are performance utilities supplied on the DECnet Distribution Kit. They allow you to perform the following types of tests:

- Connection
- Data
- Disconnection
- Interrupt

Example 11 demonstrates DTS/DTR testing from a VAX/VMS node to an RSX node. For an example of DTS/DTR testing between RSX nodes, see the section on Testing the DECnet-RSX Software in the Installation and Checkout module.

In the example, note that the default value for the /SPEED qualifier is 1 million bps.

NOTE

DTS/DTR must be installed on both systems for the test to work successfully.

```
$
$
$ @NETICP

                NETICP.COM

        DECnet Installation Checkout Procedure

Remote node for test                         (node name): ZIRCON
Type of remote node (VAX, RSX, IAS, RT, RSTS, DEC20): RSX
Speed of line for statistics                      (baud): 56000

        DTS/DTR tests exercise the basic NSP level capabilities of a DECnet
        product by requesting common task-to-task communication functions.
        We will first test DTS to DTR on the local node and then test it
        to the remote node.


DTS Version 2.00 initiated on 14-DEC-1983 15:31:05.46
CONNECT/TYPE=ACCEPT/NORETURN
%DTS-I-CINREJ, connect aborted, nsp reason is 00000004
DISCONNECT/TYPE=ABORT/RETURN=STANDARD
%DTS-I-CINREJ, connect aborted, nsp reason is 00000004
DATA/TYPE=SINK/SIZE=128/SECONDS=10
%DTS-I-CINREJ, connect aborted, nsp reason is 00000004

Test parameters:
   Test duration (sec)  10
   Target nodename      ""
   Line speed (baud)    1000000
   Message size (bytes) 128

Summary statistics:
   Total messages XMIT  0        RECV    0
   Total bytes XMIT     0
   Messages per second  0.0
   Bytes per second     0
   Line thruput (baud)  0
   % Line utilization   0.0
EXIT

DTS terminated on 14-DEC-1983 15:31:27.49

DTS Version 2.00 initiated on 14-DEC-1983 15:31:29.01
/NODENAME=ZIRCON/SPEED=56000/STATISTICS/PRINT/NODISPLAY-
-!
-!      Connect tests
-!
CONNECT/TYPE=ACCEPT/NORETURN
%DTS-S-NORMAL, normal successful completion
CONNECT/TYPE=ACCEPT/RETURN=STANDARD
%DTS-S-NORMAL, normal successful completion
CONNECT/TYPE=ACCEPT/RETURN=RECEIVED
%DTS-S-NORMAL, normal successful completion
CONNECT/TYPE=REJECT/NORETURN
%DTS-S-NORMAL, normal successful completion
CONNECT/TYPE=REJECT/RETURN=STANDARD
%DTS-S-NORMAL, normal successful completion
CONNECT/TYPE=REJECT/RETURN=RECEIVED
%DTS-S-NORMAL, normal successful completion
-!
-!      Disconnect tests
-!
DISCONNECT/TYPE=SYNCHRONOUS/NORETURN
```

Example 11 DTS/DTR Test Between DECnet/VAX and DECnet-11M
(Sheet 1 of 4)

```
%DTS-S-NORMAL, normal successful completion
DISCONNECT/TYPE=SYNCHRONOUS/RETURN=STANDARD
%DTS-S-NORMAL, normal successful completion
DISCONNECT/TYPE=SYNCHRONOUS/RETURN=RECEIVED
%DTS-S-NORMAL, normal successful completion
DISCONNECT/TYPE=ABORT/NORETURN
%DTS-S-NORMAL, normal successful completion
DISCONNECT/TYPE=ABORT/RETURN=STANDARD
%DTS-S-NORMAL, normal successful completion
DISCONNECT/TYPE=ABORT/RETURN=RECEIVED
%DTS-S-NORMAL, normal successful completion
-!
-!      Data tests
-!
DATA/TYPE=SINK/SIZE=128/SECONDS=10
%DTS-S-NORMAL, normal successful completion

Test parameters:
   Test duration (sec)  10
   Target nodename      "ZIRCON"
   Line speed (baud)    56000
   Message size (bytes) 128

Summary statistics:
   Total messages XMIT  107      RECV     0
   Total bytes XMIT     13696
   Messages per second  10.7
   Bytes per second     1370
   Line thruput (baud)  10957
   % Line utilization   19.5
DATA/TYPE=SEQUENCE/SIZE=128/SECONDS=10
%DTS-S-NORMAL, normal successful completion

Test parameters:
   Test duration (sec)  10
   Target nodename      "ZIRCON"
   Line speed (baud)    56000
   Message size (bytes) 128

Summary statistics:
   Total messages XMIT  108      RECV     0
   Total bytes XMIT     13824
   Messages per second  10.8
   Bytes per second     1382
   Line thruput (baud)  11059
   % Line utilization   19.7
DATA/TYPE=PATTERN/SIZE=128/SECONDS=10
%DTS-S-NORMAL, normal successful completion

Test parameters:
   Test duration (sec)  10
   Target nodename      "ZIRCON"
   Line speed (baud)    56000
   Message size (bytes) 128

Summary statistics:
   Total messages XMIT  101      RECV     0
   Total bytes XMIT     12928
   Messages per second  10.1
   Bytes per second     1293
   Line thruput (baud)  10342
   % Line utilization   18.4
DATA/TYPE=ECHO/SIZE=128/SECONDS=10
%DTS-I-BADDATLEN, not enough data sent/received

Test parameters:
```

Example 11 DTS/DTR Test Between DECnet/VAX and DECnet-11M
(Sheet 2 of 4)

```
      Test duration (sec)  10
      Target nodename      'ZIRCON'
      Line speed (baud)    56000
      Message size (bytes) 128

   Summary statistics:
      Total messages XMIT  61      RECV    53
      Total bytes XMIT     14592
      Messages per second  11.4
      Bytes per second     1459
      Line thruput (baud)  11674
      % Line utilization   20.8
   -!
   -!      Interrupt tests
   -!
   INTERRUPT/TYPE=SINK/SIZE=16/SECONDS=10
   %DTS-S-NORMAL, normal successful completion

   Test parameters:
      Test duration (sec)  10
      Target nodename      'ZIRCON'
      Line speed (baud)    56000
      Message size (bytes) 16

   Summary statistics:
      Total messages XMIT  141     RECV    0
      Total bytes XMIT     2256
      Messages per second  14.1
      Bytes per second     226
      Line thruput (baud)  1805
      % Line utilization   3.2
   INTERRUPT/TYPE=SEQUENCE/SIZE=16/SECONDS=10
   %DTS-S-NORMAL, normal successful completion

   Test parameters:
      Test duration (sec)  10
      Target nodename      'ZIRCON'
      Line speed (baud)    56000
      Message size (bytes) 16

   Summary statistics:
      Total messages XMIT  134     RECV    0
      Total bytes XMIT     2144
      Messages per second  13.4
      Bytes per second     214
      Line thruput (baud)  1715
      %.Line utilization   3.0
   INTERRUPT/TYPE=PATTERN/SIZE=16/SECONDS=10
   %DTS-S-NORMAL, normal successful completion

   Test parameters:
      Test duration (sec)  10
      Target nodename      'ZIRCON'
      Line speed (baud)    56000
      Message size (bytes) 16

   Summary statistics:
      Total messages XMIT  135     RECV    0
      Total bytes XMIT     2160
      Messages per second  13.5
      Bytes per second     216
      Line thruput (baud)  1728
      % Line utilization   3.0
   INTERRUPT/TYPE=ECHO/SIZE=16/SECONDS=10
   %DTS-S-NORMAL, normal successful completion
```

Example 11 DTS/DTR Test Between DECnet/VAX and DECnet-11M
(Sheet 3 of 4)

```
Test parameters:
   Test duration (sec)  10
   Target nodename      "ZIRCON"
   Line speed (baud)    56000
   Message size (bytes) 16

Summary statistics:
   Total messages XMIT  81       RECV    80
   Total bytes XMIT     2576
   Messages per second  16.1
   Bytes per second     258
   Line thruput (baud)  2061
   % Line utilization   3.6
EXIT

DTS terminated on 14-DEC-1983 15:33:34.33

        Perform some simple file procedures with DCL.

        Create a file and copy it to the remote node.
        Copy it back and compare the file after the round trip to
        see if there were any changes. The file is copied back with
        both COPY and TYPE/OUT=.
        The file is deleted on the remote node after the test.
```

Example 11 DTS/DTR Test Between DECnet/VAX and DECnet-11M (Sheet 4 of 4)

EXERCISES

Select the best answers for each of the following statements.

1. When the Executor is assigned to a remote node, a logical link connection is established from the:

   a. Local Network Management Layer and the remote Network Management Layer

   b. Local End Communication Layer and the remote End Communication Layer

   c. Local Network Applications Layer and the remote Network Applications Layer

   d. No Logical Link is created

2. NCP allows the user to modify the:

   a. Volatile DECnet down-line system loading database

   b. Permanent and volatile DECnet database

   c. Volatile DECnet database only

   d. Permanent DECnet database only

3. The maximum node address that can be assigned on a Phase IV DECnet node is:

   a. 1024

   b. 255

   c. 1023

   d. 2048

4. The Data Test Send and Data Test Receive tasks (DTS/DTR) will operate:

   a. Only between two DECnet-RSX nodes

   b. Between any DECnet nodes

   c. Only between a DECnet-RSX node and a DECnet/VAX node

   d. None of the above

5. The loopback node name is used to:

   a. Replace circuit-level loopback tests with node-level loopback tests

   b. Replace the current name of the host node

   c. Replace line-level loopback tests with node-level loopback tests

   d. Simulate two nodes when there is only one

6. The loopback mirror is used to:

   a. Record all loopback tests in a file

   b. Reflect back the loopback requests only for node-level loopback tests

   c. Give a current status of all circuits, and lines on the host node

   d. a and b only

7. All network events are classified by:

   a. Priority of the task issuing the event

   b. Class of event

   c. Operating system currently running

   d. Network Layer issuing the event

8. Events detected on a node can be dispatched to:

   a. The console device

   b. The event logger data file

   c. A remote node

   d. A monitor task

9. The NCP command to stop all activities on a DECnet-RSX node is:

   a. SET NODE STOP

   b. SET EXECUTOR STATE STOP

   c. SET EXECUTOR STATE OFF

   d. SET NODE OFF

10. The CFE command to create a new node entry into the database is:

    a. SET NODE ROGER ADDRESS 300

    b. DEFINE NODE 300 NAME ROGER

    c. DEFINE ADDRESS 300 NAME IS ROGER

    d. SET NAME ROGER ADDRESS 300

SOLUTIONS

Select the best answers for each of the following statements.

1. When the Executor is assigned to a remote node, a logical link connection is established from the:

   a. Local Network Management Layer and the remote Network Management Layer

   b. Local End Communication Layer and the remote End Communication Layer

   c. Local Network Applications Layer and the remote Network Applications Layer

   d. No Logical Link is created

2. NCP allows the user to modify the:

   a. Volatile DECnet down-line system loading database

   b. Permanent and volatile DECnet database

   c. Volatile DECnet database only

   d. Permanent DECnet database only

3. The maximum node address that can be assigned on a Phase IV DECnet node is:

   a. 1024

   b. 255

   c. 1023

   d. 2048

4. The Data Test Send and Data Test Receive tasks (DTS/DTR) will operate:

   a. Only between two DECnet-RSX nodes

   b. Between any DECnet nodes

   c. Only between a DECnet-RSX node and a DECnet/VAX node

   d. None of the above

5. The loopback node name is used to:

   a. Replace circuit-level loopback tests with node-level loopback tests

   b. Replace the current name of the host node

   c. Replace line-level loopback tests with node-level loopback tests

   d. Simulate two nodes when there is only one

6. The loopback mirror is used to:

   a. Record all loopback tests in a file

   b. Reflect back the loopback requests only for node-level loopback tests

   c. Give a current status of all circuits, and lines on the host node.

   d. a and b only

7. All network events are classified by:

   a. Priority of the task issuing the event

   b. Class of event

   c. Operating system currently running

   d. Network Layer issuing the event

8. Events detected on a node can be dispatched to:

   a. The console device

   b. The event logger data file

   c. A remote node

   d. A monitor task

9. The NCP command to stop all activities on a DECnet-RSX node is:

   a. SET NODE STOP

   b. SET EXECUTOR STATE STOP

   c. SET EXECUTOR STATE OFF

   d. SET NODE OFF

10. The CFE command to create a new node entry into the database is:

   a. SET NODE ROGER ADDRESS 300

   b. DEFINE NODE 300 NAME ROGER

   c. DEFINE ADDRESS 300 NAME IS ROGER

   d. SET NAME ROGER ADDRESS 300

LAB EXERCISES

1. Using NCP, display the following information:

   a. Executor characteristics

   b. System characteristics

   c. Circuit status

   d. Known nodes

   e. Active line characteristics

   f. Remote node known circuit counters

2. Given a network configuration, perform the following tests:

   a. Node level loopback test

   b. Circuit level loopback test

   c. Line level loopback test

   Display node, circuit, and line counters. Analyze for possible errors.

3. Run an NTD display and analyze the display data. How many logical links are active? What is the path cost to remote node? What circuit is used to get there? Which network tasks have receive outstanding? Repeat the same operation for the remote DECnet-RSX node.

4. Perform a DTS/DTR test for a given configuration.

5. Look at the event logging outputs on your system console and analyze the information displayed. How many times did the circuits go down or up? How many accesses to the node were rejected?

SOLUTIONS TO LAB EXERCISES

1.

a. NCP SHO EXE CHAR

b. NCP SHO SYS CHAR

c. NCP SHO KNOWN CIR STA

d. NCP SHO KNOWN NODES

e. NCP SHO ACT LINES

f. NCP SHO NODE XXX COUNT

2.

a. NCP ZERO EXE COUNT
NCP LOOP EXEC
NCP SHO EXEC COUNT

b. NCP ZERO CIR XXX-N COUNT
NCP LOOP CIR XXX-N
NCP SHO CIR XXX-N COUNT

c. NCP ZERO LINE XXX-N COUNT
NCP LOOP LINE XXX-N
NCP SHO LINE XXX-N COUNT

3.

a. NTD

b. NTD NODE::

4. NTEST

5. Example 6

CONTENTS

FIGURES

TABLES

INTRODUCTION

The Packetnet System Interface (PSI) is designed to follow the recommendations produced by International Telephone and Telegraph Consultative Committee (CCITT). Many European countries and Canada already provide Packetnet service. In the U.S., common carriers such as TELENET and TYMNET also provide such service.

The major advantages of Packetnet include cost saving for medium traffic, rates independent of distance, flexibility, network management provided by the common carriers, and an ability to communicate with other vendors' equipment.

In this module, the basic concepts behind Packetnet are discussed. In the appendixes, the details of the protocols used are presented.

OBJECTIVES

Upon completion of this module, the specialist should be able to:

- Specify the differences between X.25, X.3, X.28, and X.29 protocols.
- Define X.25 Level 1, Level 2, and Level 3.
- List the advantages provided by Packetnet.

RESOURCES

1. Martin, James. Computer Networks and Distributed Processing. Englewood Cliffs, NJ: Prentice-Hall, 1981.
2. DNA X.25 Packet Level Specification
3. DNA X.25 Frame Level Specification

REASONS FOR PACKET SWITCHING

- International standard recommended by CCITT. The standards are known as the X series.
- Countries such as France (TRANSPAC), Canada (DATAPAC), the United Kingdom (PSS), Germany (DATEX-P), Holland, (DN1), and others are or will be providing Public Data Network service based on these recommendations.
- Public Packet Switching improves the efficiency of a transmission line by sharing the line among many users, thus reducing the amount of time the line is idle.
- Cost does not depend on connection time and distance as in dial-up or leased lines.
- Cost is made up of a fixed portion plus a variable portion depending on the volume of data.
- Public Data Networks cover a wide area and have extensive backup (alternative routine) capability. Network management is done at Network Control Centers. The user does not need to worry about managing the network.
- Reliability and flexibility are expected to be high.
- Users can interface different vendors' equipment as long as they adhere to these standards.
- Terminals, even asynchronous dumb terminals, can be connected to the networks.

CONCEPTS IN PACKET SWITCHING

X.25 Recommendation

- The end user does not need to know anything about the internal mechanism used in Public Packet Switched Networks (PPSN).

- The X.25 recommendation specifies the interface between the Data Terminal Equipment (DTE) and the Data Circuit Terminating Equipment (DCE). Figure 1 illustrates this relationship.

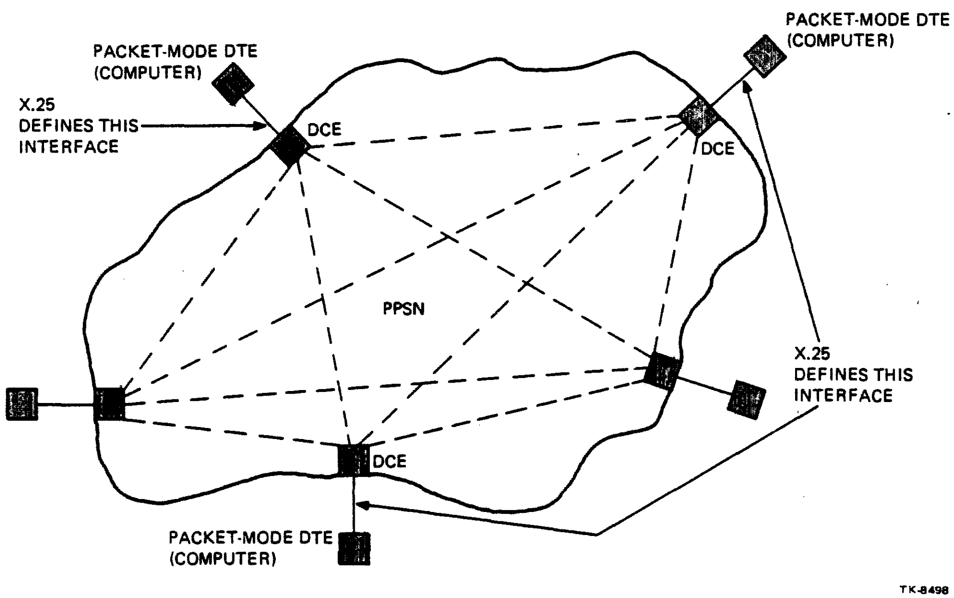


Figure 1 DTE/DCE Relationship

- The X.3, X.28, and X.29 recommendations together specify how to control asynchronous terminals connected directly to the network. Together, these protocols define the Interactive Terminal Interface (ITI).

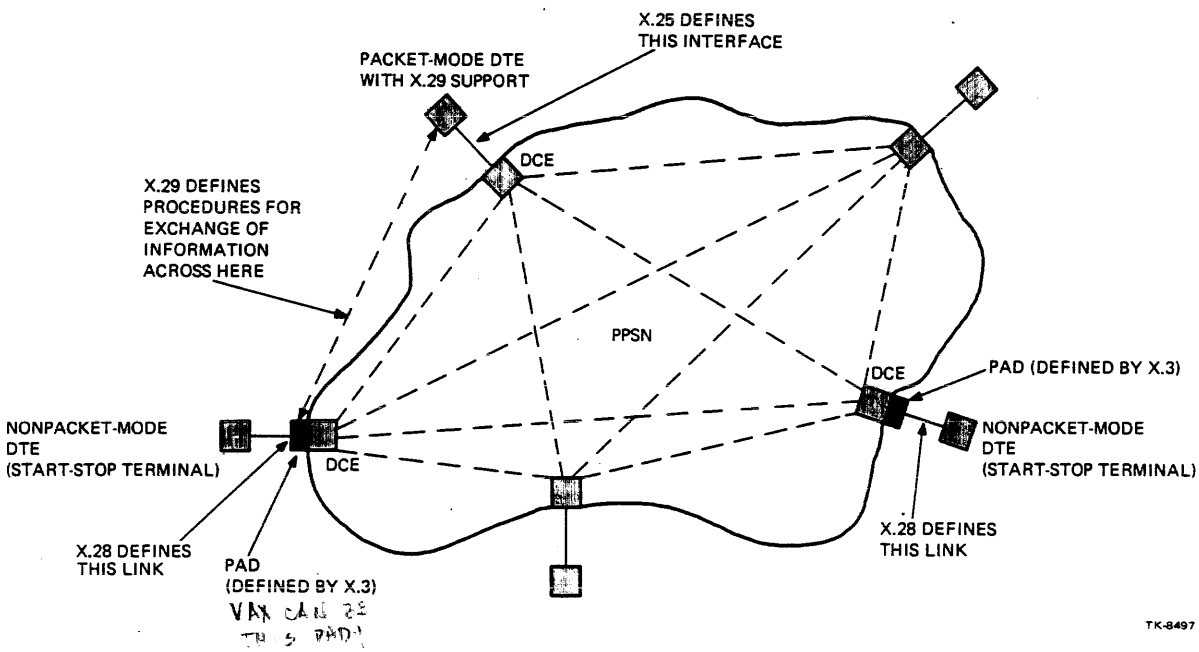


Figure 2 X.3, X.28, and X.29 Protocols

Notes on Figure 2

1. X.3 -- Packet Assembly/Disassembly (PAD) facility in a public data network.

2. X.28 -- DTE/DCE interface for start-stop mode DTE accessing the PAD facility in a public data network situated in the same country.

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

X.25 Levels

The X.25 specification consists of three levels:

- Level 1 -- Hardware characteristics for the DTE/DCE interface (X.21-bis or RS-232-C).
  - Defines physical characteristics
  - Defines electrical characteristics
  - Defines operational function of the interchange circuits
  - Defines fault indications
- Level 2 -- Data Link Control procedure for the DTE/DCE interface Higher Level Data Link Control (HDLC). (Refer to Appendix A for details.)
  - Frame structure
  - Error detection and retransmission
  - Link level flow control
- Level 3 -- Packet procedure across the DTE/DCE interface (X.25). (Refer to Appendix B for details.)
  - Packet structure
  - Virtual call establishment and clearing
  - Virtual circuit flow control
  - Error detection and recovery

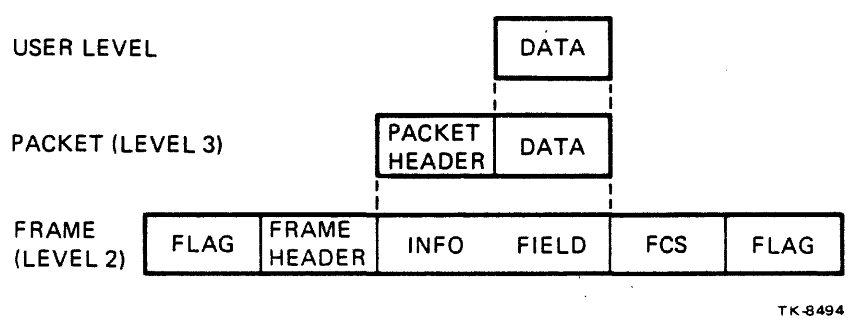


Figure 3 Protocol Layering in X.25

Virtual Circuits

The concept of the virtual circuit is necessary in understanding X.25. When a computer is to be added to the PPSN, it is given a unique DTE address. It is also given a range of logical channel numbers.

When a program on a local DTE wants to communicate with another program on a remote DTE, it requests the use of a free logical channel number. A CALL REQUEST packet is sent from the local DTE.

If the call is accepted by the remote program, a CALL ACCEPTED packet is sent. The packet contains a free logical channel number from the remote DTE. The numeric value of the logical channel number from the local DTE may be different from that of the remote DTE.

A virtual circuit now exists and data exchange may proceed. The packets sent on this virtual circuit contain the logical channel number between the local DTE and DCE. The packets arrive at the remote DTE with the logical channel number between the remote DCE and DTE.

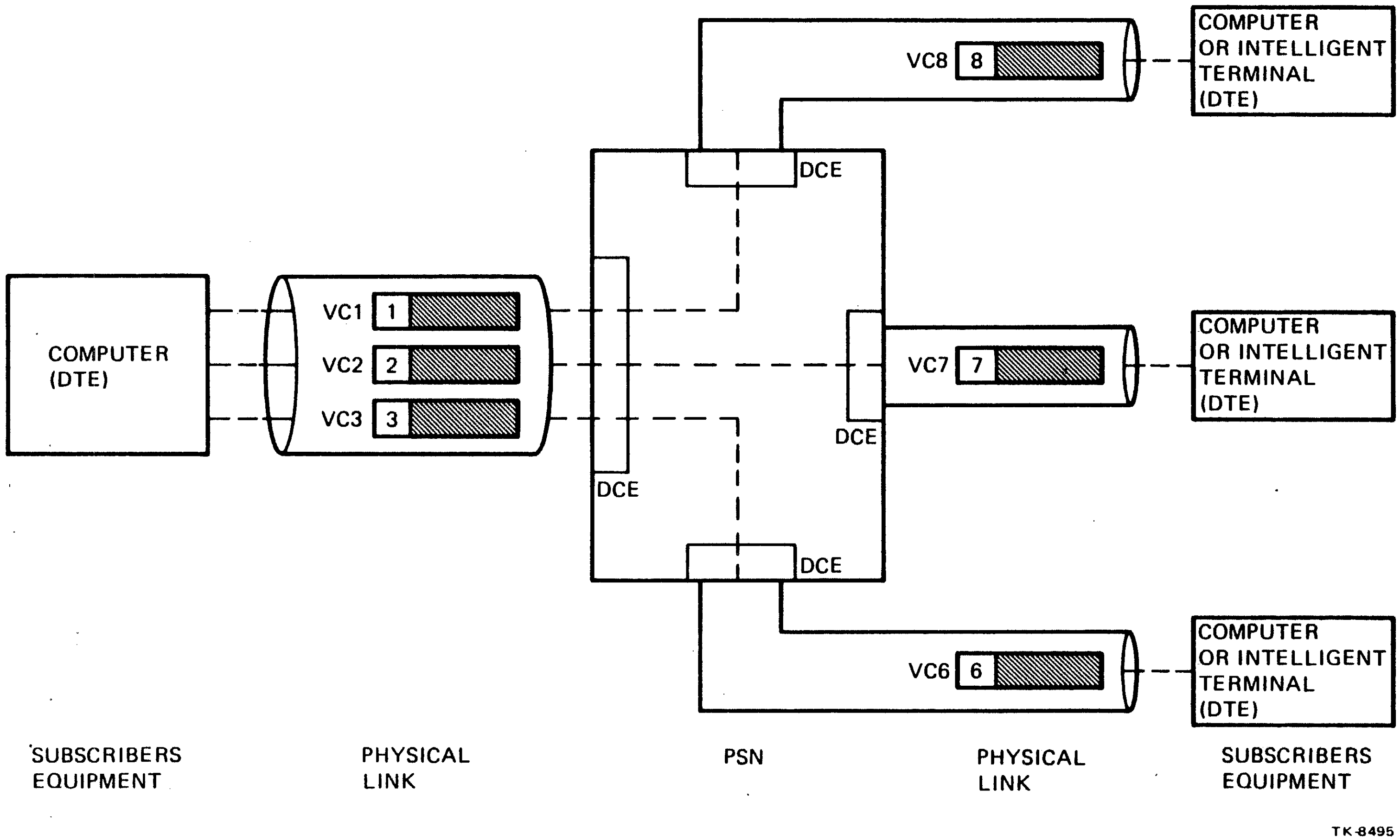


Figure 4 Virtual Circuits and Logical Channel Numbers

Other X.25 Facilities

- Other facilities provided by the Public Packet Switched Systems normally include:
  - Permanent virtual circuits -- No setting up sequence required
  - Flow control (by using window size) to increase throughput and reduce the change of congestion. (Refer to Appendix B for further details.)
  - Interrupts -- One byte of user-defined data
  - Resets and restarts -- Reinitialize one or all circuits
- Other optional features may include:
  - Closed user group
  - Reverse charge
  - Incoming call redirection

APPENDIX A
Data Link Control Protocol

Objectives of HDLC

- To protect data transmission from errors.
- The data field can be any number of bits and any pattern of bits.
- The protocol should be able to handle point-to-point, multipoint, or looped physical links (not necessarily true on X.25 subnets).
- The protocol should be able to handle half- or full-duplex links.
- The protocol should be as efficient as possible.

Frame Structure

FLAG	ADDRESS	CONTROL	INFORMATION	FRAME CHECK	FLAG
01111110	8 BITS	8 BITS	(VARIABLE)	16 BITS	01111110

TK-6842

Figure 5 HDLC Frame Structure

Notes on Figure 5

1. Flag -- Used to denote the start/end of data.
2. Address -- Used by the primary station to address the secondaries and by the secondary to identify their responses.
3. Control -- Identifies the frame as Informational (I), Supervisory (S), or Unnumbered (U).
4. Information -- The data to be transmitted.

Frame Format

- Informational frame (I) -- An I frame is used to transmit data and may also acknowledge previous correctly received I frames.

- Supervisory frame (S) -- An S frame is used to perform channel control functions such as acknowledging correctly received I frames, requesting retransmission of I frames, or indicating temporary unavailability for receiving I frames.

  There are three supervisory frames:

  - RR -- Receive Ready
  - RNR -- Receive Not Ready
  - REJ -- Reject

- Unnumbered frame (U) -- A U frame is used to provide additional link control functions. Normal routine operations use only I frames and S frames. Some U frames are:

  - SABM -- Set Asynchronous Balanced Mode (LAPB only)
  - SARM -- Set Asynchronous Response Mode (LAP only)
  - UA -- Unnumbered Acknowledgment
  - DM -- Disconnect Mode (LAPB only)
  - DISC -- Disconnect
  - CMDR -- Command Reject (LAP only)
  - FRMR -- Frame Reject (LAPB only)

NOTE

RSX PSI, Version 2.0 supports only LAPB.

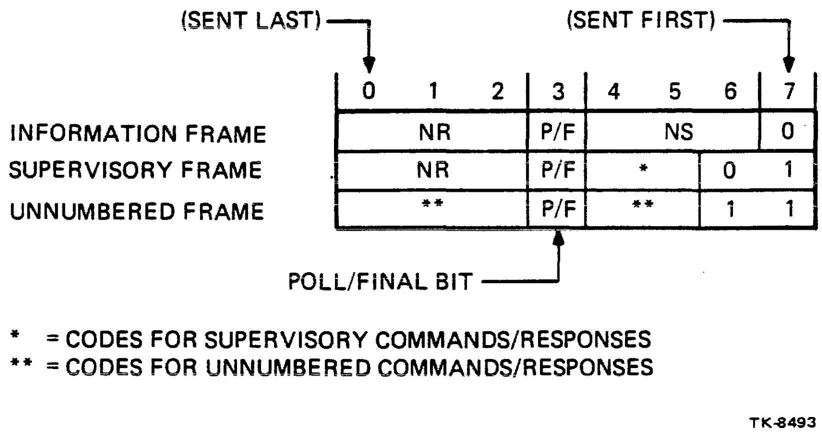


Figure 6 Control Field Formats

Example of Data Exchange

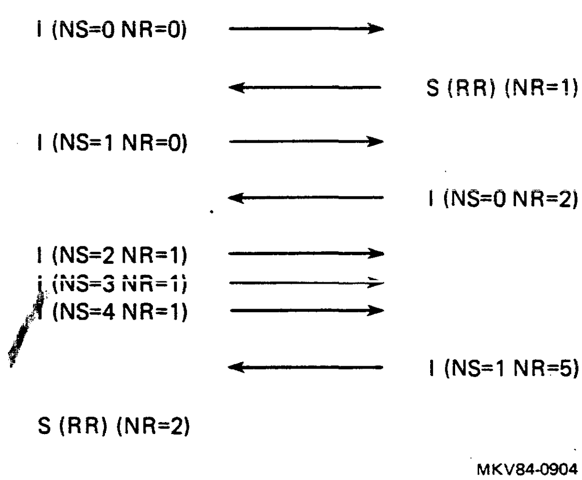


Figure 7 Data Exchange Using the HDLC Protocol

Link Control Procedure

Two sets of procedures for link control are defined in the X.25 recommendation as follows:

- LAP -- The DTE/DCE interface is defined as operating in two-way simultaneous Asynchronous Response Mode (ARM). That is, both DTE and DCE perform primary and secondary functions.

- LAPB -- The DTE/DCE interface is defined as operating in two-way Asynchronous Balanced Mode (ABM). This is described in detail.

Using the concept of a finite-state machine, the DTE/DCE link can be in one of the following states:

- LINK UP -- The link setup procedure has been completed and link is in the information transfer state.

- LINK DOWN -- The link is operational but waiting for the link setup procedure to be initiated by the DCE.

- LINK FAILURE -- The link is considered dead by either the DCE or the DTE, and the link disconnection procedure has not been completed.

In addition, the DTE/DCE link can be in one of the following transient states:

- LINK SETUP -- The DTE has initiated a request for setting up the link.
- LINK DISCONNECT -- Either the DTE or the DCE has initiated a request for disconnecting the link.
- LINK RESET -- The DTE or the DCE has initiated a request for resetting the line.

Link Synchronization

The Flag (Ø111111Ø) is also used for synchronization. To synchronize a channel, the receiver searches for the unique flag pattern (Ø111111Ø).

Recognizing Data Size

- The flag byte Ø111111Ø is used.
- Every message begins and ends with a flag byte.
- A zero bit is inserted after every five 1-bits in the data so that the flag byte pattern can never occur by chance.
- Example of this technique:

Original Pattern	Transmitted Pattern with Zeroes Inserted	Received Pattern with Zeros Removed
1Ø1Ø1Ø1Ø1Ø1Ø	1Ø1Ø1Ø1Ø1Ø1Ø	1Ø1Ø1Ø1Ø1Ø1Ø
111111111111	1111Ø1111111Ø11	11111~~Ø~~111111~~Ø~~11
Ø11111Ø	Ø11110Ø	Ø11111~~Ø~~Ø
Ø111111Ø	Ø1111Ø1Ø	Ø11111~~Ø~~1Ø

APPENDIX B
X.25 Level 3 Packet Procedure

General

- Level 3 defines the packet level procedure. This procedure enables DTEs to communicate with one another across the network.
- The maximum size of a packet is determined by the PPSN (usually 128 bytes).
- The general procedure for a switched virtual circuit may be treated in three parts:
  - Setting up a virtual circuit
  - Exchanging data
  - Clearing a virtual circuit
- Note that the format of the data exchanged within a virtual circuit is not specified within X.25. The data may contain any other user-defined protocol plus any user data. For example, between two DIGITAL systems, the NSP protocol may be used.

Packet Structure and Types

The general format is shown in Figure 8.

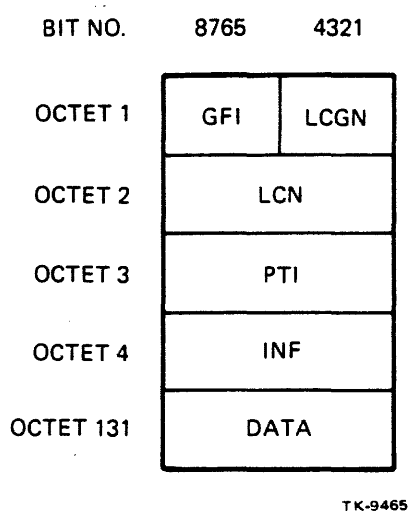


Figure 8 General Packet Format for the X.25 Level 3 Protocol

Notes on Figure 8

1. GFI (General Format Identifier) -- A four-bit binary-coded field that identifies the general type of packet group.

2. LCGN (Logical Channel Group Number) -- Depending upon the PPSN, LCGN may identify a particular DTE, a customer, or class of service. The LCGN is zero in RESTART, RESTART CONFIRMATION, and DIAGNOSTIC packets.

3. LCN (Logical Channel Number) -- This identifies the logical channel. The LCN appears in every packet except restart packets. LCN is zero in RESTART, RESTART CONFIRMATION, and DIAGNOSTIC packets.

4. PTI (Packet Type Identifier) -- This defines the packet type.

5. INF (General Information) -- This contains information such as the DTE addressed, and clearing, resetting, and restarting causes. There is no information field in data packets.

6. DATA -- This contains user data (data packets only).

Table 1 X.25 Packet Types

Packet Type		PTI
From DCE to DTE	**From DTE to DCE**	87654321
Call Setup and Clearing		
Incoming Call	Call Request	00001011
Call Connected	Call Accepted	00001111
Clear Indication	Clear Request	00010011
DCE Clear Confirmation	DTE Clear Confirmation	00010111
Data and Interrupt		
DCE Data	DTE Data	XXXXXXX0
DCE Interrupt	DTE Interrupt	00100011
DCE INT Confirmation	DTE INT Confirmation	00100111
Datagram*		
DCE Datagram	DTE Datagram	XXXXXXX0
Datagram Service Signal		XXXXXXX0
Flow Control and Reset		
DCE RR (Mod. 8)	DTE RR (Mod. 8)	XXX00001
DCE RNR (Mod. 8)	DTE RNR (Mod. 8)	XXX00101
	DTE REJ (Mod. 8)*	XXX01001
DCE RR (Mod. 128)*	DTE RR (Mod.128*	00000001
DCE RNR (Mod. 128)*	DTE RNR (Mod. 128)*	00000101
	DTE REJ (Mod. 128)*	00001001
Reset Indication	Reset Request	00011011
DCE Reset Confirmation	DTE Reset Confirmation	00011111
Restart		
Restart Indication	Restart Request	11111011
DCE Restart Confirmation	DTE Restart Confirmation	11111111
Diagnostic		
Diagnostic*		11110001

X - Indicates that the bit may be set to either 0 or 1.

* - Indicates that this packet is not necessarily available on every network.

Message Exchange

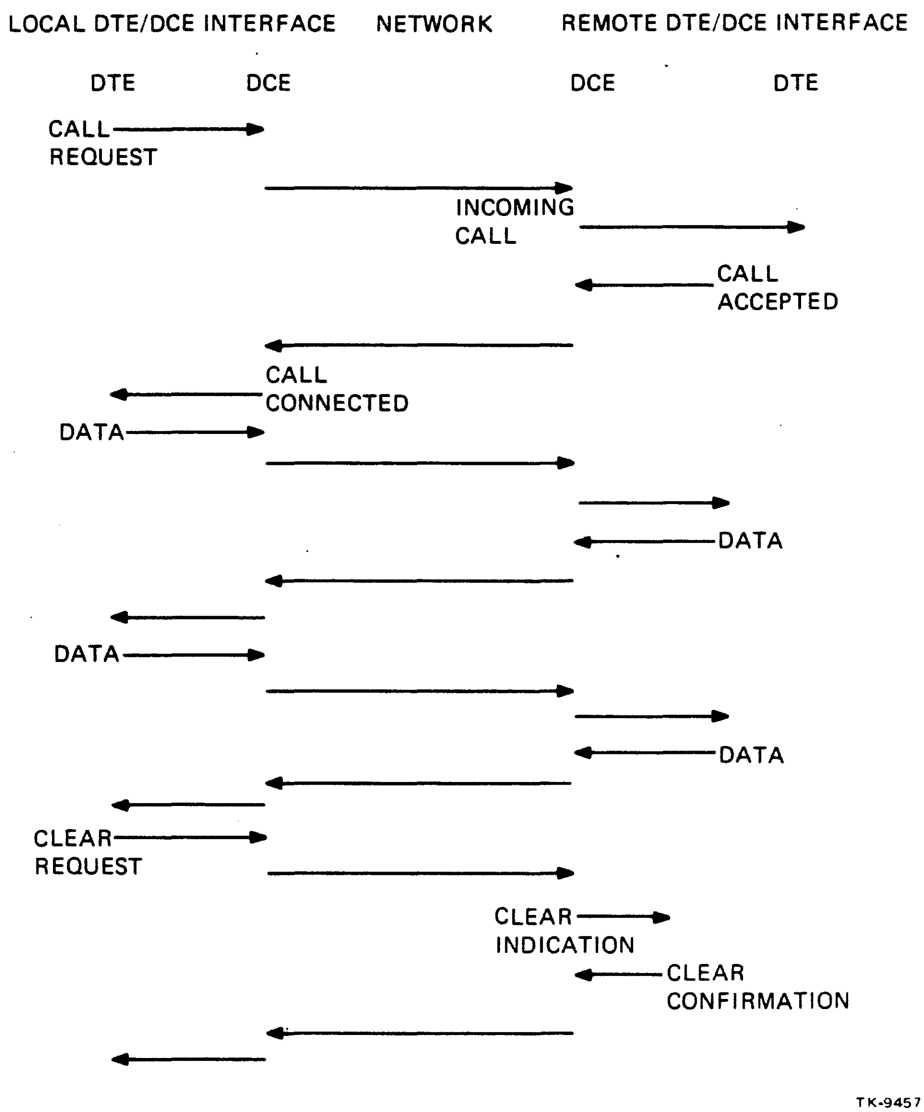


Figure 9 A Typical Call Sequence

Flow Control

- Flow control for control packets:
  - Control packets are defined in pairs (for example, Call Request and Call Connected packets; Clear Request and Clear Confirmation packets).
  - After a DTE sends a Call Request packet, it can do nothing on that particular virtual circuit until it has received a Call Connected or a Clear Request packet.
  - This method controls the flow of packets by not allowing any further packets to be sent on a particular virtual circuit until the previous action is acknowledged or completed.

- Flow control for data packets:
  - Each data packet is sequentially numbered in either module 8 or module 128 as specified by the PPSN.
  - Concept of window is used. The window size can be from 1 to 7 or from 1 to 127. Only data packets with a number within the window can be sent.

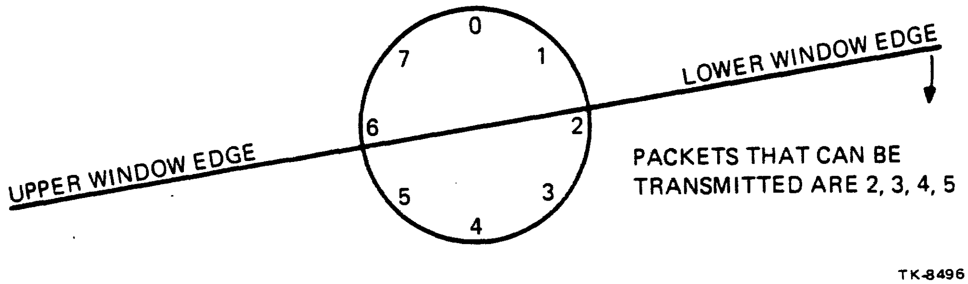


Figure 10 Window Operations

Notes on Figure 10

1. The window size = 4.

2. The last receive sequence number is 2. Hence the lower window size is 2 and the upper window size is 6 (2 + 4). The packets that can be sent are 2, 3, 4, and 5.

3. If no acknowledgment is received, the window cannot be turned and no more messages can go out.

4. If an acknowledgment containing 4 as the receive sequence number is received while packets 2, 3, 4, and 5 are sent, a window turn takes place. The lower window size becomes 4 and the upper window size becomes 0 (4 + 4 Module 8).

5. Packets 4, 5, 6, and 7 are within the window and can be sent.

EXERCISES

1. Match the following PPSN names to the appropriate countries.

a.	PSS	____	1.	United States
b.	TELENET	____	2.	West Germany
c.	DATAPAC	____	3.	United Kingdom
d.	TYMNET	____	4.	Holland
e.	TRANSPAC	____	5.	France
f.	DATEX-P	____	6.	Italy
			7.	Canada

2. Select the best answer for each.

   1. The X.25 recommendation defines the:

      a. Interface between the two DTEs

      b. Interface between the two DCEs

      c. Routing algorithm in the PPSN

      d. None of the above

   2. The term PAD stands for:

      a. Packet Assembler Device

      b. Principle Address of Destination

      c. Packet Assembler/Disassembler

      d. Primary Access Device

3. The X.29 recommendation defines the:

   a. Interface between a terminal and DTE

   b. Interface between a DCE and DTE

   c. Interface between a DTE and a modem

   d. None of the above

4. Level 2 of the X.25 protocol:

   a. Defines the Data Link

   b. Defines the Virtual Circuit

   c. Defines the Logical Link

   d. b and c

5. Level 3 of X.25 provides:

   a. Data integrity over the Logical Link

   b. Flow control over the Virtual Circuit

   c. Defines the permanent Virtual Circuit

   d. Defines the hardware interface

SOLUTIONS

1. Match the following PPSN names to the appropriate countries.

a.	PSS	3	1.	United States
b.	TELENET	1	2.	West Germany
c.	DATAPAC	7	3.	United Kingdom
d.	TYMNET	1	4.	Holland
e.	TRANSPAC	5	5.	France
f.	DATEX-P	2	6.	Italy
			7.	Canada

2. Select the best answer for each.

   1. The X.25 recommendation defines the:

      a. Interface between the two DTEs

      b. Interface between the two DCEs

      c. Routing algorithm in the PPSN

      d. None of the above

   2. The term PAD stands for:

      a. Packet Assembler Device

      b. Principle Address of Destination

      c. Packet Assembler/Disassembler

      d. Primary Access Device

3. The X.29 recommendation defines the:

   a. Interface between a terminal and DTE

   b. Interface between a DCE and DTE

   c. Interface between a DTE and a modem

   d. None of the above

4. Level 2 of the X.25 protocol:

   a. Defines the Data Link

   b. Defines the Virtual Circuit

   c. Defines the Logical Link

   d. b and c

5. Level 3 of X.25 provides:

   a. Data integrity over the Logical Link

   b. Flow control over the Virtual Circuit

   c. Defines the permanent Virtual Circuit

   d. Defines the hardware interface

FIGURES

TABLES

EXAMPLES

INTRODUCTION

Before the DECnet-RSX software can run on the RSX-11 system, the following must be performed:

- SYSGEN -- For including DECnet-RSX related considerations such as assigning the partition for the Communications Executive (CEX), increasing the available vector area to allow for the communications hardware to be installed, including certain features for a terminal driver, and so on.
- PREGEN -- For copying the DECnet-RSX distribution to the disk media from which from which the actual NETGEN occurs.
- NETGEN -- For generating the DECnet-RSX software.
- NETINS -- For bringing up the DECnet-RSX software.
- NTEST -- For testing the DECnet-RSX software.

This module covers these procedures in detail.

OBJECTIVES

To install the DECnet-RSX software, the specialist must be able to:

- List the SYSGEN requirements for DECnet-RSX.
- List the DECnet and PSI major components.
- Identify the features of the major components.
- Perform the NETGEN procedure.
- Perform the NTEST procedure.

RESOURCES

1. <u>DECnet-RSX Network Generation and Installation Guide</u>
2. <u>DECnet-RSX System Managers Guide</u>

COMPONENTS

There are three categories of components:

- NET components support the Communications Executive and provide for general communications functions. These components include:
  - NET Communications Executive components
  - NET Communications Executive support components
  - NET System Management utilities
- DEC components are specific to DECnet. They include:
  - DECnet communication components
  - DECnet Network Management support components
  - DECnet satellite support
  - DECnet file utilities
  - DECnet network terminal and control utilities
- PSI (Packetnet System Interface) components allow DECnet to communicate with a Public Packet Switching Network (PPSN) using the X.25 protocol. These components include:
  - Required PSI communications components
  - Optional PSI components

    PSI components are included in your system only if you have purchased a license for the PSI product.

Table 1 is a list of individual components. The table shows whether the component is required (R), always built but not absolutely required (A), optional (O), or never used (N) for your RSX-11M, M-PLUS, or (S) system. In very small systems, you may choose not to install tasks marked "A" as a way of saving memory.

The table provides a reference to the section where the dialogue for a given component is presented. The table also tells whether a component is a task or a process. Note that the Communication Executive is neither a task or a process, but is rather an extension of the RSX executive.

Table 1 DECnet Components

		Task or Process	System M	M+	S	Notes
CEX	Communications Executive	-	R	R	R	
AUX	Auxiliary Process	P	R	R	R	
DDM	Device Driver Modules	P	R	R	R	
NET Communications Executive Support Components						
NTINIT	Network Initializer	T	R	R	R	
NTL	Network Loader	T	R	R	R	
KMCL	KMC Microcode Loader	T	R	R	R	1
MLD	General Microcode Loader	T	R	R	R	2
EVL/EVC	Network Event Logger	P&T	O	O	O	
DLX	Direct Line Access	P	O	O	O	3
NET System Management Utilities						
NCP	Network Control Program	T	R	R	R	
NMDRV	Network Management Device Driver	P	R	R	A	4
NMVACP	Network Management Volatile ACP	T	R	R	N	
CFE	Configuration File Editor	T	A	A	A	
NDA	Network Crash Dump Analyzer	T	A	A	A	
VNP	Virtual Network Processor	T	A	N	A	
DECnet Communications Components						
ECL	Network Services and Process Driver	P	R	R	R	
XPT	Transport Task	P	R	R	R	
NETACP	Network ACP	T	R	R	R	
DCP DDCMP	Digital Communication Process	P	R	R	R	5
EPM	Ethernet Protocol Manager	P	R	R	R	6
NETFOR.OLB	DECnet Library for Higher Level Languages	-	O	O	O	
NETLIB.MLB	MACRO-11 MACRO Definition Library	-	O	O	O	

Table 1 DECnet Components (Cont)

		Task or Process	System M	M+	S	Notes
DECnet Network Management Support Components						
NICE	Network Information and Control Exchange	T	O	O	O	
EVR	Event Logging Receiver	T	O	O	O	
NTD	Network Display Utility	T	O	O	O	
NTDEMO	Network Display Server	T	O	O	O	
LIN	Link Watcher	T	O	A	N	
LOO	Loopback Tester	T	A	A	N	
MIR	Loopback Mirror	T	A	A	A	
RCP	Routing Control Processor	T	R	R	R	7
NVP	Network Verification Program	T	O	O	O	
DECnet Satellite Support Components						
DLL	Down-Line System Loader	T	O	O	N	
DUM	Up-Line System Dumper	T	O	O	N	
CCR	Console Carrier Requester	T	O	O	N	
HLD	Host Task Loader	T	O	O	N	
SLD	Satellite Task Loader	T	N	N	O	
DECnet File Utilities						
NFT	Network File Transfer Utility	T	O	O	N	
FTS	File Transfer Spooler	T	O	O	N	
FAL	File Access Listener	T	O	O	O	
MCM	Command-File/Batch-File Submission Task	T	O	O	N	
DECnet Network Terminal and Control Utilities						
RMT/RMTACP	Remote Terminal Utility RMTACP	T	O	O	O	
RMHACP/HTDRV	Remote Terminal Host ACP and Host Driver	T	O	O	O	
NCT	Network Control Terminal Task	T	O	O	O	
TLK	Interterminal Message Utility	T	O	O	O	
LSN	TLK Server Task	T	O	O	O	
TCL	Remote Task Control Utility	T	O	O	O	

Notes

1. KMCL is required for all 11M/11M-PLUS systems if you specify KMC or KMS devices and specify powerfail recovery support. KMCL is required on 11S systems that support KMC or KMS devices.

2. MLD is required for all systems if you specify Ethernet devices.

3. DLX is required if you specify certain operations (see Section 2.4.1.5 in the DECnet-RSX Network Generation and Installation Guide for more information).

4. NMDRV is always built only if EVL is specified.

5. DCP is required for all systems if you specify devices that require software DDCMP support.

6. EPM is required for all systems if you specify Ethernet devices.

7. RCP is required only for routing type nodes.

Tasks and Processes

There are two types of components: tasks and processes. Tasks are the normal tasks on RSX (for example, ready-to-run core images that are scheduled to run by the RSX Executive). Processes are special units of code that are scheduled and run by the Communications Executive. The Communications Executive is an extension of the RSX Executive and performs special services for communications products such as DECnet. Processes are largely invisible to the RSX Executive; they are totally self-contained within DECnet. Processes are core-resident, are no more than 4K in size, and perform time-critical DECnet functions.

In several cases, individual processes are paired with associated tasks that are larger and assist with less time-critical functions. These tasks are called Ancillary Control Processors (ACPs). For example, NSDRV, which is the process that creates and maintains logical links, has an associated ACP, NETACP, which assists it with its less time-critical processing.

HOST SYSTEM CONFIGURATION

To run NETGEN, the host system must be either RSX-11M, Version 4.1 or RSX-11M-PLUS, Version 2.1.

The host must have a minimum amount of available disk space. In addition to the disk used for the operating system, 1 RL01/02 satisfies this minimum. With larger-capacity disks, fewer drives are required. With an RM02/03, RM05, RM80, RP06 or RP07 disk one disk is usually sufficient to store the operating system and all the files necessary to perform NETGEN.

TARGET SYSTEM CONFIGURATION

The target system must be either RSX-11M, Version 4.1; RSX-11M-PLUS, Version 2.1; or RSX-11S, Version 4.1.

The target system must be a mapped system with a minimum of 16K words of memory dedicated to the DECnet software. Additional memory is required for user-written network tasks and any DECnet utilities.

The target system must have at least one communications device to satisfy the minimum hardware requirements.

DEVICES SUPPORTED BY DECnet

The following devices are supported by DECnet:

Table 2 Device Support for RSX-DECnet PSI

UNIBUS Devices		Q-Bus Devices
Asynchronous Line Support		
	DL-11	DLV-11
	DZ-11	DZV-11
	KMC-11/DZ-11	
Synchronous Line Support		
	DMP-11	DMV-11
	DUP-11	DPV-11
	DU-11	DUV-11
	DMC-11	
	DMR-11	
	DV-11	
	KMC-11/DUP-11	
Parallel Line Support		
	PCL-11	
Ethernet Line Support		
	DEUNA	DEQNA
Supported for PSI		
	DUP-11	DPV-11
	KMS-11	

PREPARING FOR NETGEN

The logical procedure for creating a DECnet node (for RSX-11M systems only) consists of these steps:

1. Performing SYSGEN
2. Defining partitions (by editing SYSVMR.CMD) M/S
3. Performing PREGEN
4. Performing NETGEN
5. Using NETINS.CMD to install components in the correct partitions (see Chapter 8 of the DECnet-RSX Generation and Installation Guide).

System Sizes

The way you configure your network depends on your system's memory size. For the purposes of NETGEN, your system can be classified into small, medium, and large, defined as follows:

Small = Under 8ØK words of memory

Medium = 8ØK to 124K words of memory

Large = Over 124K words of memory

NOTE

These definitions are based on light network loading and are somewhat arbitrary. Available memory is most important, and this is altered both by the specific complement of network components you select and by your other software and application programs.

These definitions of small, medium, and large systems should serve only as a guide. To get a more accurate system capacity, you need to determine the amount of available memory in GEN after the basic network and your application are running. (See Section 3.1.6 of the DECnet-RSX Generation and Installation Guide for more information.)

RSX-11M Systems

For RSX-11M systems, the partition layout you select varies depending upon the size of your system and your intended uses. In all cases, you must add the partition CEXPAR.

RSX-11S Systems

Because RSX-11S systems are often special-purpose dedicated systems, it is difficult to make any specific rules for determining partition layout. You need to add the partition CEXPAR.

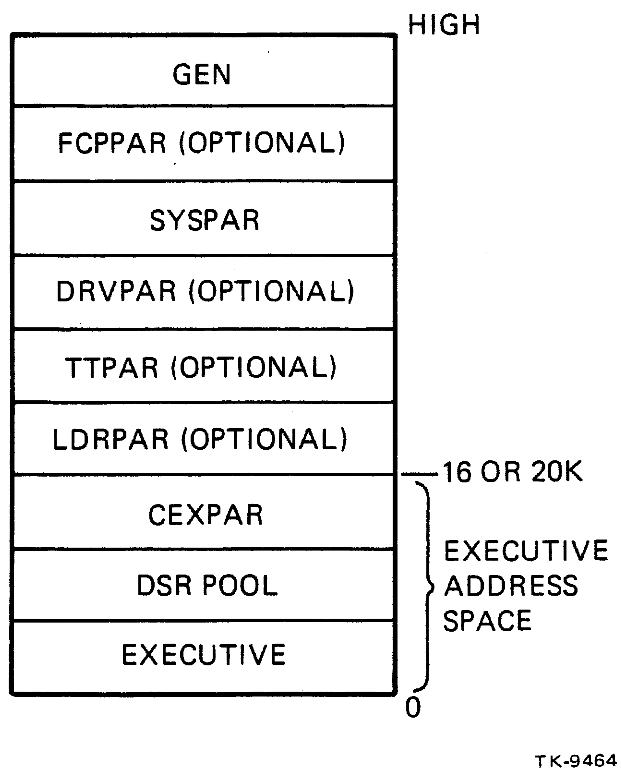


Figure 1 Partition Layout after SYSGEN (With CEXPAR Added) for RSX-11M/S

RSX-11M-PLUS

For RSX-11M-PLUS systems, no special partitions are necessary.

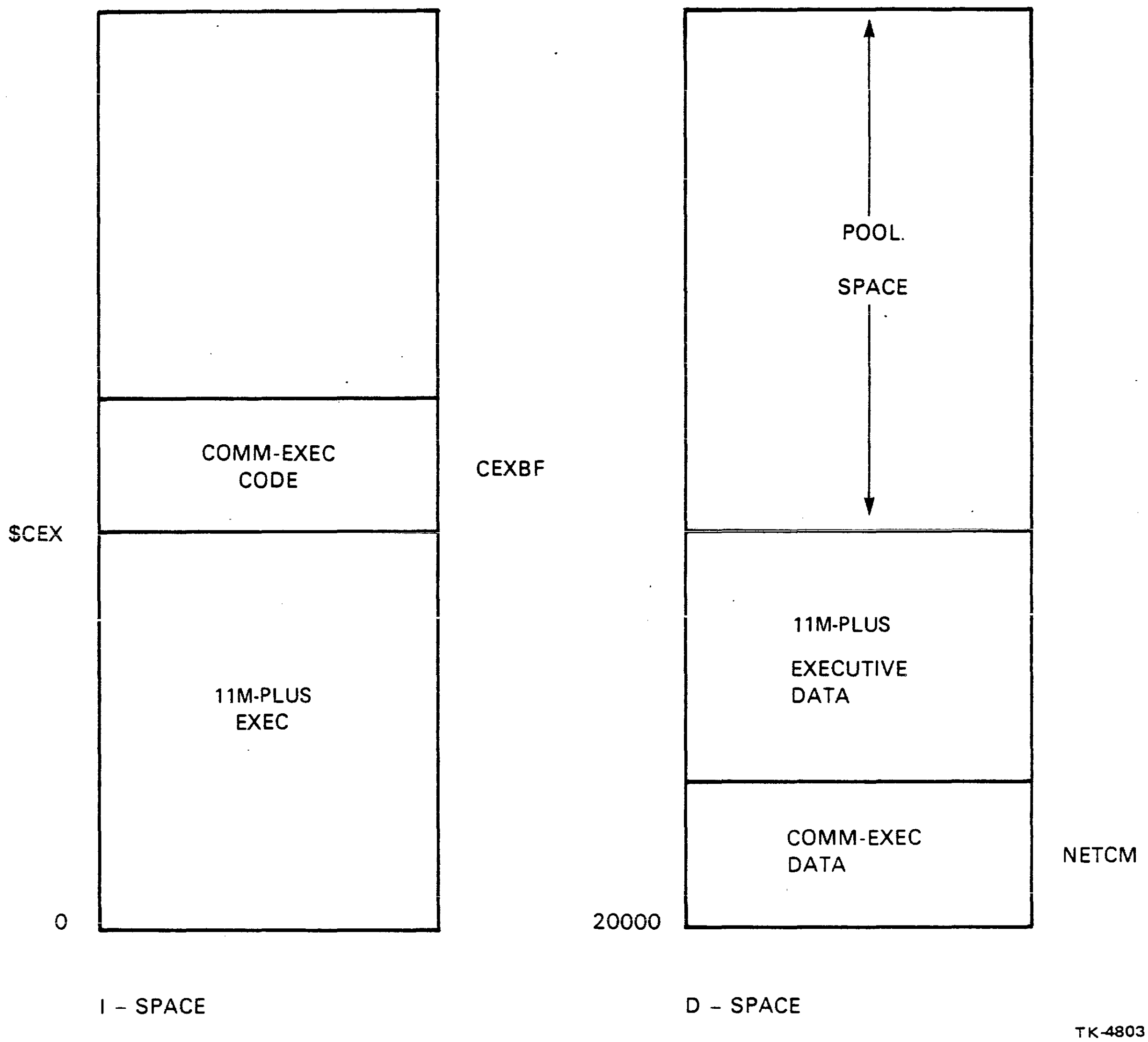


Figure 2 RSX-11M-PLUS Communications Executive Implementation

Small RSX-11M Systems

For small RSX-11M systems, the partition layout shown in Figure 3 is recommended.

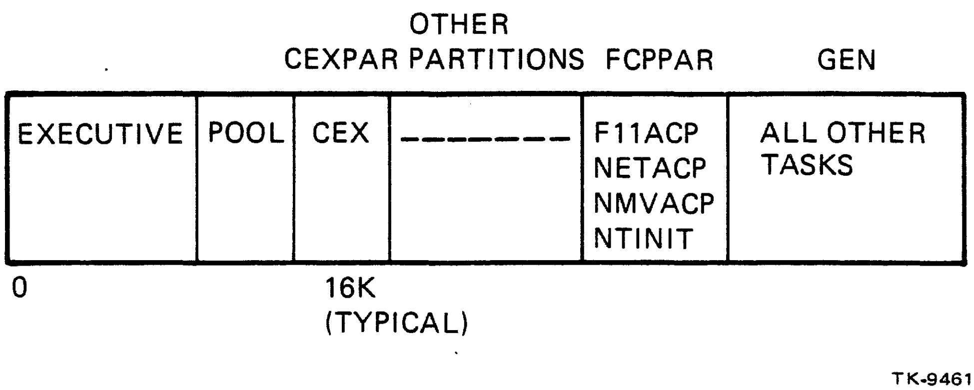


Figure 3 Partition Layout for Small RSX-11M Systems

Here, NETACP, NMVACP, and NTINIT are installed in FCPPAR, where they run along with F11ACP, the Files-11 ACP. NETACP must be generated as checkpointable. You should size FCPPAR to accommodate the largest task in the partition; this often is NETACP. You can arbitrarily choose a size for FCPPAR that is too large (7K words) and then reduce the size appropriately after NETGEN, when you know the actual sizes. All other network and user tasks are installed in GEN.

Medium-Sized RSX-11M Systems

For medium size RSX-11M systems, the partition layout in Figure 4 is recommended.

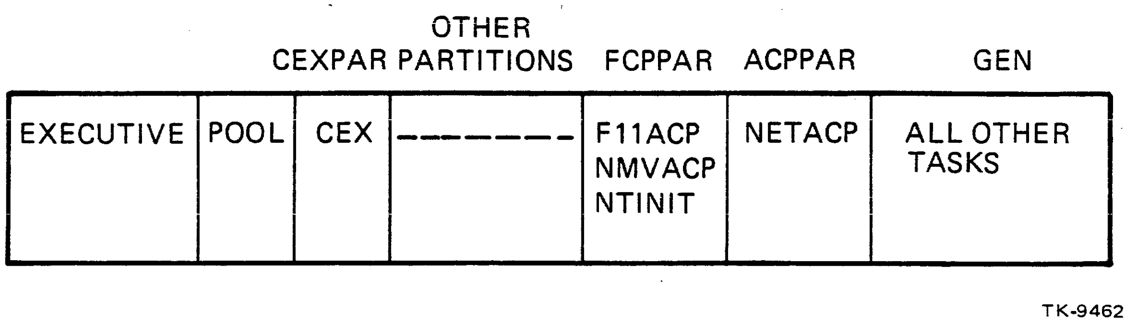


Figure 4 Partition Layout for Medium-Sized RSX-11M Systems

Here, NETACP has been removed from the FCPPAR partition and placed in its own partition.

The advantage of removing NETACP from the FCPPAR partition in this way is that both NETACP and F11ACP are very active tasks, and this layout avoids needlessly swapping them in and out of memory. The result is increased performance. This layout has all the other advantages of the recommended layout for small RSX-11M systems.

Large RSX-11M Systems

Figure 5 assumes that you decided to put F11ACP (the FILES-11 ACP) in its own partition, FCPPAR. This is the default during SYSGEN. All DECnet tasks run in the system-controlled partition GEN.

		CEXPAR	OTHER PARTITIONS	FCPPAR	GEN
EXECUTIVE	POOL	CEX	-----	F11ACP	ALL TASKS

TK-9443

Figure 5 Typical Partition Layout

SYSGEN

- SYSGEN support for DECnet
- RSX and DECnet communications devices
- Standard function system (RSX-11M/S only)
- System controlled partitions (RSX-11M/S only)
- SYSGEN component parameters
- Large or Small Executive (RSX-11M/S only)
- Editing SYSVMR.CMD (RSX-11M/S only)
- Adding CEXPAR (RSX-11M/S only)
- Adding other partitions (RSX-11M/S only)

SYSGEN Support for DECnet

To create the proper support for DECnet on the system you are generating, answer YES to the following questions in the Executive Options section of SYSGEN:

For RSX-11M/S Systems:

```
>* 32.  Include support for communications products?  [Y/N]:
```

For RSX-11M-PLUS Systems:

```
>* CE120 Do you want support for communications products

>* (such as DECnet)?  [Y/N D:N]:
```

In both cases, answering YES causes all of the essential DECnet support features to be added to the RSX operating system being generated.

DECnet Communications Devices

Because the RSX-11M/S/M-PLUS Executive does not need to know about hardware that will be used as DECnet communications devices, and because no communications device can be shared between DECnet and the Executive, do not specify any DECnet communications devices when answering questions in the Target Configuration section of SYSGEN. However, when you generate your RSX system, you must specify how much space is to be allocated for interrupt vectors on your target system. The space must be sufficient to include the interrupt vectors belonging to DECnet devices. To specify properly, answer the following question with the highest vector address of any devices on your system, including the DECnet communications hardware.

For RSX-11M/S Systems:

```
>* 14.  Higest interrupt vector:
```

For RSX-11M-PLUS Systems:

```
>* CP9632 What is the highest interrupt vector address?
```

These vector addresses are available as a result of hardware installation.

For this question, do not specify the default, since the value that SYSGEN computes in this case would not take into account DECnet communication devices. For example on RSX-11M and RSX-11S systems, increase the value by octal 20 if your DECnet system includes any of the following character interrupt devices: DL, DLV, DPV, DU, DUP, DUV. This is necessary because, for these devices, the transmitter section of the device driver may interrupt the receiver interrupt handler. When such an interrupt occurs, the interrupted values are stored on the kernel stack. The extra 20 bytes make room for these values. For RSX-11M-PLUS systems, 20 bytes are added automatically.

In answering this question for your operating system, leave enough space for devices you have or will have soon, as the Executive is built above the address you supply, and overwrites any vectors in its space. For all systems, answering 774 guarantees that the value is sufficiently high, but usually results in some loss of available memory (memory that would otherwise be used as system pool).

Standard Function System (RSX-11M Only)

For RSX-11M systems, Question 9 at the beginning of SYSGEN asks:

```
>* 9.  Do you want a Standard Function System ?  [Y/N]:
```

If you answer the question YES and select the standard function system, you automatically receive the following options (others may be included):

- System-controlled partitions
- Checkpointing (but without the system checkpointing file)
- Full-duplex terminal driver (with all options)
- Large (2ØK) Executive

NOTE

If you choose the standard function system, you are not able to dynamically change your maximum node address, since the standard function system does not support the required system checkpointing file.

Have to rebuild RCP1.TSK

Pre-SYSGENed System (RSX-11M-PLUS Only)

If you choose the pre-SYSGENed system available for RSX-11M-PLUS systems, you must edit one symbol found in the RSXMC.MAC file on UIC [11,10]. This is because the target processor ~~symbol, R$$TPR, is~~ not properly set up to define the real processor type of your target system. Using an editor, search for the symbol in the file. Edit this symbol to read:

```
R$$TPR="nn
```

Where nn equals the number of the processor, such as 44 for a PDP-11/44. This change is required prior to NETGEN; no rebuilding of the operating system is required.

System-Controlled Partitions (RSX-11M/S Only)

As a general rule, you should request system-controlled partitions for RSX-11M/S systems. This is specified in Question 33 of the Executive Options section of SYSGEN:

```
>* 33. system controlled partitions? [Y/N]:
```

System-controlled partitions allow most network tasks to be loaded into one partition (GEN), and automatically swapped in and out of memory. The only situation where you might not want to use system-controlled partitions is in a dedicated system in which one task is loaded into each partition, thus ensuring that the task is always in memory. Each such partition should be sized exactly for the particular task. Such a system would save the small amount of memory used by the Executive to handle system-controlled partitions.

SYSGEN Component Parameters

The following parameters that you specify in SYSGEN do not affect the overall performance of DECnet, but only that of specific DECnet components.

Changing Number of Nodes (RSX-11M Only)

If you want to be able to change the number of nodes in your network without rebuilding the Routing Control Processor (RCP) task, select dynamic checkpoint allocation support. Question 49 in the Executive Options section of SYSGEN asks:

```
>* 49.  Checkpointing:  A-Yes B-With system ckpnt.  file [S]:
```

B - can change max node address in routing node. - A [illegible]

Answering B enables dynamic checkpoint support. (You should always select checkpointing in some form, that is, either A or B.)

Support for TLK (RSX-11M/S Only)

Question 1 of the Terminal Driver Options section of SYSGEN asks:

```
>* 1.  Terminal driver desired (A/B/C/D, * prints table) [S]:
```

If you enter an asterisk (*), SYSGEN prints the following table:

```
Terminal drivers:

A.  - Tailorable half-duplex

B.  - Tailorable half-duplex (user oriented)

C.  - Full-duplex

D.  - Baseline half-duplex
```

satisfactory

Answering A, B, or C produces satisfactory results. Answering B automatically provides the necessary support for both versions of TLK. If you answer A or C, you must select the specific driver features desired.

If you want either mode of TLK (dialogue mode or single-message mode), you must answer YES to the following question:

```
>* 8.  Breakthrough write?  [Y/N]:
```

If you want the dialogue mode of TLK, you must answer YES to the following question:

> 17. Read after prompt? [Y/N]:

If you want the video dialogue mode to TLK, and if you answered A or C to Question 1, you must answer YES to the following question in the same section of SYSGEN:

>* 6. Unsolicited input character AST? [Y/N]:

Asynchronous System Traps (ASTs) are software interrupts. Answering YES to this question allows your terminal to process characters typed on the keyboard at the same time characters are being printed on the screen. Support for unsolicited input character ASTs is automatically included if you answer B to question 1. For more information on TLK, see the <u>DECnet-RSX Guide to User Utilities</u>.

Support for PSI X.29 (RSX-11M Only)

If you want support for PSI, you must answer C to the terminal driver question (Question 1) under Support for TLK. PSI's X29ACP requires the full-duplex driver.

Also answer YES to Question 12:

>* 12. Set multiple characteristics? [Y/N]:

The result is to create MCR support for SET commands relating to terminals, such as:

SET /VT1ØØ=TI:

Queued File Access Requests

FTS allows file access requests (file transfers, and so on) to be queued and processed in sequence. If you want FTS, the RSX-11M/M-PLUS Queue Manager (QMG) must be built during SYSGEN.

For RSX-11M systems, the following question in the Executive Options section of SYSGEN must be answered YES.

need you FTS?

```
>* 35.  Queue manager and queued print spoolers?  [Y/N]:
```

For RSX-11M-PLUS systems, answer YES to the following questions:

```
>* CE180 Do you want to include the Queue Manager?  [Y/N D:N]:
```

(For more information on FTS, see the DECnet-RSX Guide to User Utilities.)

Large or Small Executive (RSX-11M/S Only)

The RSX Executive for an RSX-11M or RSX-11S system can be built in either 16K or 2ØK of address space. In the Executive Option section of phase 1, SYSGEN asks the following question:

```
>* 22.  Large (2ØK) Executive?  [Y/N]:
```

If you answer NO to this question you get the 16K Executive.

more pool.

This question determines the amount of Executive address space allocated to the Executive plus DSR (pool). The size of the Executive itself is determined solely by your answers to questions about Executive options. The space left over is allocated to pool. This allocation, however, can be altered by editing the SYSVMR.CMD file.

Deciding which size Executive to choose depends upon your type of system, its size, and its intended use. In general, for medium to large RSX-11M systems, you want the 2ØK Executive because the extra pool is useful. You have enough memory so that the 4K allocated to pool does not significantly affect your ability to run tasks.

On small RSX-11M systems, you need to make a trade-off decision between the need for pool and the need for extra task space. Often you may decide to use the smaller Executive.

Editing SYSVMR.CMD (RSX-11M/S Only)

For RSX-11M and RSX-11S systems, it is necessary to add at least one partition, CEXPAR, to the system in order to run DECnet. For medium size RSX-11M systems, you may also wish to add another partition. These partitions are added by editing the SYSVMR.CMD command file.

In Phase 2 of SYSGEN, VMR Question 1 asks:

```
>* 1.  Edit SYSVMR.CMD?  [Y/N]:
```

Answer this YES, and SYSGEN pauses and allows you to edit the file. You can answer NO (or press return <RET>) and edit the file later.

Answer no & wait til finished, then use EDT [illegible] [illegible]

Adding CEXPAR (RSX-11M/S Only)

For RSX-11M/S systems, the necessary partition (CEXPAR) is incorporated by adding a SET /MAIN=CEXPAR line to the SYSVMR.CMD file.

The beginning of the SYSVMR.CMD file from an RSX-11M system looks like this:

```
RSX11M
SET /POOL=*
SET /MAIN=EXCOM1:*:200:COM
INS EXCOM1
SET /MAIN=EXCOM2:*:200:COM
INS EXCOM2
SET /MAIN=LDRPAR:*:26:TASK
INS LDR
FIX LDR...
SET /MAIN=TTPAR:*:400:TASK
LOA TT:
SET /MAIN=DRVPAR:*:*:SYS
```

You should add a line to the SYSVMR.CMD file between the SET /POOL= command and the SET /MAIN command for the first partition shown. To reflect the base address at which CEXPAR loads, change the value of the SET /POOL= line. The effect of these changes is that CEXPAR loads immediately above the RSX Executive and pool.

The format of the two lines is:

SET /POOL=<u>base address</u> 1128

SET /MAIN=CEXPAR:*:<u>size</u>:COM

CEXPAR:*:20:COM

The asterisk (*) here means, load at the next available location. The word COM signifies that this is a common block partition. The base address, which is also the top of the pool, and size are shown (default values):

Executive (in words)	CEX Size(1)	Base Address(2)	Par Size(2)
2ØK	large	112Ø	6Ø
2ØK	small	114Ø	4Ø
16K	large	72Ø	6Ø
16K	small	74Ø	6Ø

- CEX size is determined as follows: CEX is automatically generated in one of two sizes by NETGEN. For a DECnet-only (non-DECnet-PSI) system with only one line, NETGEN generates a small CEX. In all other cases, NETGEN generates a large CEX.

- The base address and size are shown in octal bytes. Note that the final two digits of the address and size are omitted in this table, because that is the convention employed by VMR. Thus, an octal byte address of 112ØØØ is expressed as 112Ø. (In NETGEN however, you must express the last 2 digits.)

PREGEN

PREGEN is a preliminary procedure you must perform before generating a network. It takes your required and optional distribution media and merges them, so they can be used by the NETGEN procedure. PREGEN can run for either a standard system, or for a small dual-disk (RL01/02) system. The same procedure is used for both.

Media Used During PREGEN

A standard PREGEN is performed for systems other than small dual-disk systems. A standard PREGEN can include the following distribution kits (disks or magnetic media):

- Network kit
- DECnet kit
- PSI kit (optional)

Each kit contains command files and object files. The command files become the corresponding parts of NETGEN: NET, DEC, and PSI (optional). The object files build the appropriate NET, DEC, or PSI components.

Thus, from the distribution kits, PREGEN generates the following logical groups of files:

File	**Contents**
NETGEN command files	Command files from all available kits
Network object files	Object files for the NET procedure of NETGEN
DECnet object files	Object files for the DEC procedure of NETGEN
PSI object files	Object files for the PSI procedure of NETGEN

PREGEN Dialogue

For distribution kits in the form of disks or tapes follow the procedures listed in Sections 3.3.4 through 3.3.9 of the DECnet-RSX Network Generation and Installation Guide.

NETGEN Procedures

Network Generation consists of the following procedures:

- NET -- Communications Executive and common components
- DEC -- DECnet
- PSI -- Packetnet System Interface (optional)

The NET procedure of NETGEN asks questions regarding the Communications Executive (CEX) and command components that are always built in the configuration of the network. The Communications Executive contains the general routines that support both DECnet and PSI.

The DEC section of NETGEN asks DECnet specific questions. These questions are in addition to NET questions. NETGEN then builds the routines which, when combined with NET, allow the system to run DECnet.

The PSI section of NETGEN asks questions regarding PSI (for users who have purchased a license for this product). PSI allows you to access an X.25 Public Packet Switching Network (PPSN). Both DECnet and PSI can run concurrently on the same processor (assuming sufficient system resources such as memory).

NETGEN Sections

Each procedure within NETGEN (NET, DEC, or PSI) consists of sections. Each section is a logical grouping of related questions. A section is introduced with a header such as:

```
>;=======================================================
>; NET - Section    2 - Define the Target System
>;=======================================================
```

This header identifies the procedure of NETGEN (NET here), the section number, and provides a brief description of the purpose of the section.

At the end of a section, NETGEN asks an End of Section (EOS) question, which gives you the option of repeating the section to correct any errors that may have been made in that section.

NETGEN Questions

The first question in each section is numbered Ø1.ØØ, the second Ø2.ØØ, and so on. The sequence continues in this manner to the end of the section. Questions that have subordinate questions are numbered Ø1.Ø1, Ø1.Ø2 and so on.

NETGEN Operating Modes

The NETGEN procedures can be run in four operating modes:

- Question and Answer Mode
- Restore Mode
- Dry Run Mode
- Component Mode

When you start NETGEN, you begin in Question and Answer Mode; you can then introduce other modes as appropriate. The other three modes are special cases of Question and Answer Mode. Restore Mode and Dry Run Mode can be invoked in any combination with Question and Answer Mode. Component Mode, however, is incompatible with either Restore Mode or Dry Run Mode.

The modes you select remain in effect through the entire generation procedure. If, for example, the procedure is started in Component Mode, it runs in Component Mode throughout.

User Responses and Defaults

For the most part, your responses to NETGEN questions consist simply of answers to the questions. In certain cases, however, you may want to:

- Use the ESCAPE key to obtain explanatory text
- Respond to end-of-section and break questions
- Take advantage of NETGEN defaults

Table 3 Summary of NETGEN Response

Response Type	NETGEN Displays:	You enter:
YES/NO	[D=N] [Y/N]:	Y to indicate YES, N to indicate NO (or press <RET> for default NO).
Numeric Octal	[O R:m-n]:	An octal number in the range of m through n.
	[O R:m-n D:d]:	An octal number in the range of m through n (or press <RET> to indicate that the default (d) should be used).
Decimal	[D R:m.-n.]:	A decimal number in the range of m through n.
	[D R:m.-n. D:d.]:	A decimal number in the range of m through n (or press <RET> to indicate that the default (d.) should be used). You do not need to include a decimal point after the number you enter.
Character String	[S]:	A character string of any length.
	[f, D:d] [S]:	A character string in the form f (such as ddu:), (or press <RET> to indicate that the default (d) should be used).
	[S R:m.-n.]:	A character string from m through n characters in length.

Output from NETGEN

Table 4 lists the disk areas where the files generated by NETGEN reside on the target disk. Group code x refers to the group code for NETGEN output that you specify in NET. The dduu: for Listing and Map files may represent the NL: device or any disk in the system.

Table 4 Summary of NETGEN Output Files

UIC on the Target Disk		
Files	DECnet-11M 11M-PLUS	DECnet-11S
Libraries	[1,1]	[1,1]
NETHLP.ULB (Help file for NCP)	[1,2]	[1,2]
Command files and work files NETINS.CMD NETCFE.CMD NETCFG.TXT	[x,1]	[x,1]
Saved response files (GEN and RES types)		
Prefix files (used to specify conditional assembly parameters) NETGEN copy of RSXMC.MAC	[x,10]	[x,10]
Object files and libraries, assembly and task build command files	[x,24]	[x,24]
Listing files	dduu:[x,34]	dduu:[x,34]
Map files	dduu:[x,34]	dduu:[x,34]
Tasks and processes CETAB.MAC NETGEN copy of RSX11M.STB ([x,54]) or RSX11S.STB ([x,64])	[x,54]	[x,64]

NETGEN Files

The following is a description of the NETGEN files:

- NETINS.CMD -- The Network Installation file is the file invoked to install DECnet.

- NETCFE.CMD -- The Network CFE Command file is an indirect command file used as input to CFE. This file provides a record of the CETAB.MAC configuration as created by NETGEN. It can also be used as a template for making changes to CETAB.MAC. In this case, you edit the file to reflect the appropriate changes, and then use the file as an indirect command file to CFE.

- NETCFG.TXT -- The Network Configuration file provides information in text form on the configuration of devices, processes, and CEX products on the target system.

- CETAB.MAC -- The Communication Executive Database file contains the database used to load the configuration defined for the node. It is one of the most important outputs of NETGEN.

GENERATION EXAMPLES

```
>
>
>
>
>
>
>
>PIP NETGEN.CMD/LI

Directory DR3:[137,10]
13-DEC-83 14:02

NETGEN.CMD;1        22.        13-DEC-83 13:29

Total of 22./22. blocks in 1. file

>
>
>
>@NETGEN
>;
>; ====================================================================
>;  NET - RSX-11M-PLUS CEX System Generation Procedure
>;        Started at 14:02:18 on 13-DEC-83
>; ====================================================================
>;
>; Copyright (C) 1981, 1982, 1983 by
>; Digital Equipment Corporation, Maynard, Mass.
>;
>; Generate a CEX System, Version 02.00, for your RSX-11M-PLUS target system.
>;
>; The following CEX Products may be generated
>;      DECnet
>;
>; The distribution kits must be up  to  the  current  patch  level.   If
>; necessary, you should stop the  generation  and  apply  the  necessary
>; patches.
>;
>; The logical device name LB: should already be assigned to  the  device
>; which contains the library files for the system  you  are  generating.
>; These files are EXELIB.OLB, EXEMC.MLB,  RSXMAC.SML,  SYSLIB.OLB,  and,
>; for RSX-11S, 11SLIB.OLB.
>;
>; The logical device names IN:, OU:, LS: and MP: should not be  assigned
>; to any devices before starting the generation.  These  logical  device
>; names are used by NETGEN to refer to various  disks  used  during  the
>; generation.
>;
>;
>; <EOS>  Do you want to:
>*       <RET>-Continue, E-Exit [S]:
```

Example 1 NETGEN Dialogue (Sheet 1 of 17)

```
>;
>;
>; ====================================================================
>;  NET - Section  1 - General Initialization
>; ====================================================================
>;
>* 01.00 Do you want to see the NETGEN notes/cautions [D=N]? [Y/N]: Y
>;
>; Note the following:
>; -------------------
>;
>; . If you have not already performed a SYSGEN for your  target  system,
>;   you must do so before continuing.
>;
>; . The  privileged  tasks  generated  during  NETGEN  use configuration
>;   dependent  parameters  produced  by  SYSGEN.   For  RSX-11M/S/M-PLUS
>;   systems, the RSXMC.MAC file and the RSX11M.STB (or RSX11S.STB)  file
>;   for the target system are necessary.   If they are  not  already  on
>;   the target device under the proper UIC, NETGEN will move  them  from
>;   the user's system device to the target device.
>;
>; . If your target device contains useful files, it is recommended  that
>;   you back it up before  continuing,  as  the  device  will  be  write
>;   enabled during the NETGEN sequence.
>;
>; . The proper system libraries for the target system  must  be  on  the
>;   logical device LB: under the UIC [1,1] while you are performing your
>;   NETGEN.  These libraries are:
>;
>;        EXELIB.OLB, EXEMC.MLB, RSXMAC.SML, and SYSLIB.OLB.
>;
>;   In addition, for an RSX-11S NETGEN, the 11SLIB.OLB file is required.
>;
>; . If your target system supports the Queue Manager (QMG), and you wish
>;   to generate network tasks which may use  this  facility,  the  Queue
>;   Manager object library must be  on  the  logical  device  LB:  under
>;   UIC [1,24] while you are performing your NETGEN.  This file is:
>;
>;      QMG.OLB
>;
>; . If you wish to generate network tasks which require support for  RMS
>;   file  access, you must have the RMS files on your logical LB: device
>;   prior  to  performing  your NETGEN. NETGEN will look for these files
>;   under the UIC determined by sysgen. The following are the files that
>;   NETGEN will look for:
>;
>;      RMSLIB.OLB, RMS11X.ODL and RMS12X.ODL.
>;      RMSRES.TSK and RMSRES.STB (if resident libraries are supported).
>;
```

Example 1 NETGEN Dialogue (Sheet 2 of 17)

```
>;
>; . If  your  target  system  supports a Memory Resident or a Supervisor
>;   Mode  FCS  library,  and  you  wish  the  network  tasks to use this
>;   library,  the  FCS  library  files must be on the logical device LB:
>;   under  UIC  [1,1]  while you are performing your NETGEN. These files
>;   are:
>;
>;      FCSRES.TSK and FCSRES.STB (Memory resident FCS library files).
>;      FCSFSL.TSK and FCSFSL.STB (Supervisor Mode FCS library files).
>;
>; . The following installed tasks are required on your host system while
>;   you are performing your NETGEN:
>;
>;      PIP, LBR, MAC, and TKB for RSX-11M/S/M-PLUS NETGENs.
>;      STK (slow task builder) for RSX-11M/M-PLUS NETGENs.
>;
>;   If these tasks are not already installed, and you are logged in as a
>;   privileged user, NETGEN will install them automatically.  If TKB and
>;   and  MAC  are  installed,  you may want to remove them and reinstall
>;   them with an increment of 3000. to 15000. bytes so the build section
>;   of NETGEN will run faster.
>;
>; . The following installed tasks are required on your host system while
>;   you are performing your NETGEN, under the following conditions:
>;
>;      LOA and UNL if device drivers require loading.
>;      MOU and DMO if disks require mounting.
>;      UFD if the required UICs do not already exist.
>;
>;   If these tasks are not already installed, and you are logged in as a
>;   privileged user, NETGEN will install them automatically.
>;
>; . If  you  are  generating  more than one network onto the same target
>;   disk,  the  network  help  file ([1,2]NETHLP.ULB) and  the  language
>;   libraries  ([1,1]NETFOR.OLB and [1,1]NETLIB.MLB)  will  be  replaced
>;   each  time  NETGEN  is run.  You may want to save these files before
>;   doing  the next NETGEN. This is of particular interest if one of the
>;   networks  is  for  RSX-11S  and  the other network is for RSX-11M or
>;   RSX-11M-PLUS.
>;
>; . For questions that can be answered YES or  NO,  the  default  is  NO
>;   unless  otherwise  specified.  For questions with  defaults,  the
>;   default is  produced  by  pressing  a  carriage  return  (<RET>)  in
>;   response to the  question.
>;
>; . Additional explanatory text for each question  may  be  obtained  by
>;   hitting the ESCAPE key.
>;
>; . Unless otherwise specified, it is possible to interrupt a series  of
>;   questions by typing <CTRL/Z>.  You will then be  given  the  options
>;   of either restarting the section from the top, temporarily  stopping
>;   the generation, or terminating the generation.
>;
>* 02.00 Target system device [dduu, D=SY00:] [S]: DR3:
>* 03.00 Listing/map device [dduu, D=None] [S]:
>* 04.00 UIC Group Code for NETGEN output [O R:1-377 D:5]: 300
>:
```

Example 1 NETGEN Dialogue (Sheet 3 of 17)

```
>; Checking for required library files, tasks, and UICs.
>;
>;
>; Creating UFD for DR03:[300,001]
>; Creating UFD for DR03:[300,010]
>;
>* 07.00 User ID for saving new responses [D=None] [S R:0.-30.]: 13DECRYAN
>* 08.00 Is this generation to be a dry run [D=N]? [Y/N]: N
>* 09.00 Do you want a standard function network [D=N]? [Y/N]:
>;
>; Answering  YES to this question will result in NETGEN asking a minimum
>; number of questions. A full compliment of components will be generated
>; and most optional parameters will be filled in with predefined values.
>; If a value has not been defined for a parameter,  NETGEN will ask  the
>; required questions.
>;
>* 09.00 Do you want a standard function network [D=N]? [Y/N]: N
>* 10.00 Should all components be generated [D=N]? [Y/N]: Y
>* 11.00 Should old files be deleted [D=N]? [Y/N]: Y
>;
>; <EOS>  Do you want to:
>*        <RET>-Continue, R-Repeat section, P-Pause, E-Exit [S]:
>;
>;
>; =====================================================================
>;  NET - Section  2 - Define the target system
>; =====================================================================
>;
>* 02.00 RSXMC.MAC location (ddu:[g,m], D=DR03:[011,010]) [S]:
>;
>; DR03:[011,010]RSXMC.MAC is being copied to DR03:[300,010]RSXMC.MAC
>;
>; DR03:[300,010]RSXMC.MAC is being scanned to define your target system.
>; This may take up to several minutes.
>;
>; The target is an RSX-11M-PLUS system, with...
>;      A 20K-Executive
>;      Kernel data space enabled
>;      Extended memory support (more than 124K words of memory)
>;      UMR support
>;      Multi-user protection
>;      Extended instruction set (EIS)
>;      Powerfail recovery support
>;      Dynamic task checkpoint allocation
>;      PLAS support
>;      Queue Manager (QMG) support
>;
>; Checking for required library files and tasks.
>;
>;
>; Creating UFD for DR03:[300,024]
>; Creating UFD for DR03:[300,054]
>;
>* 04.00 RSX11M.STB location (ddu:[g,m], D=DR03:[001,054]) [S]:
>;
```

Example 1 NETGEN Dialogue (Sheet 4 of 17)

```
>; DR03:[001,054]RSX11M.STB is being copied to DR03:[300,054]RSX11M.STB
>;
>* 05.00 Should tasks link to the Supervisor Mode FCS library [D=N]? [Y/N]:
>* 06.00 Should tasks link to the Memory Resident FCS library [D=N]? [Y/N]:
>;
>; The DECnet CEX Product will be generated.
>;
>; Routing or non-routing nodes can be generated from your distibution kit.
>;
>;
>; <EOS>  Do you want to:
>*        <RET>-Continue, R-Repeat section, P-Pause, E-Exit [S]:
>;
>;
>; ====================================================================
>;  NET - Section  3 - Define the system lines
>; ====================================================================
>;
>* 01.00 Device Driver Process name [<RET>=Done] [S R:0-3]:
>;
>; The Device Driver Module  (DDM)  Process  name  identifies  a  process
>; which controls an I/O device.  Enter a carriage return when there  are
>; no more devices to support.
>;
>; The legal DECnet device driver process names are:
>;
>; Name Physical device          Name    Physical device
>; ---- ---------------          ----    ---------------
>; DMC  DMC11 or DMR11           DL      DL11
>; DMP  DMP11                    DZ      DZ11
>; DU   DU11                     KDZ     KMC11/DZ11
>; DUP  DUP11                    PCL     PCL11
>; DV   DV11
>; KDP  KMC11/DUP11
>; UNA  DEUNA
>;
>* 01.00 Device Driver Process name [<RET>=Done] [S R:0-3]: DMC
>* 02.00 How many DMC controllers are there [D R:1.-14. D:1.]:
>;
>* 03.01 CSR address for DMC-0 [O R:160000-177777 D:177777]: 160120
>* 03.02 Vector address for DMC-0 [O R:0-774 D:0]: 330
>* 03.03 Device priority for DMC-0 [O R:4-6 D:5]: 5
>;
>* 04.00 Is DMC-0 a FULL or HALF duplex line [D=FULL] [S]:
>* 04.01 Line speed for DMC-0 [D R:50.-56000. D:56000.]:
>* 04.07 Set the state for DMC-0 ON when loading the network [D=N]? [Y/N]: YES
```

Example 1 NETGEN Dialogue (Sheet 5 of 17)

1) For timing in Commodities,

2) For the cost associated with the line.

```
>;
>* 01.00 Device Driver Process name [<RET>=Done] [S R:0-3]: DUP
>* 02.00 How many DUP controllers are there [D R:1.-14. D:1.]:
>;
>* 03.01 CSR address for DUP-0 [O R:160000-177777 D:177777]: 160070
>* 03.02 Vector address for DUP-0 [O R:0-774 D:0]: 320
>* 03.03 Device priority for DUP-0 [O R:4-6 D:5]:
>;
>* 04.00 Is DUP-0 a FULL or HALF duplex line [D=FULL] [S]:
>* 04.01 Line speed for DUP-0 [D R:50.-9600. D:9600.]:
>* 04.07 Set the state for DUP-0 ON when loading the network [D=N]? [Y/N]: Y
>* 05.00 Is DUP-0 multipoint [D=N]? [Y/N]: N
>;
>* 01.00 Device Driver Process name [<RET>=Done] [S R:0-3]:
>;
>;
>; <EOS>  Do you want to:
>*        <RET>-Continue, R-Repeat section, P-Pause, E-Exit [S]:
>;
>;
>; ======================================================================
>;  NET - Section  4 - Define the CEX System
>; ======================================================================
>;
>; Creating build files for CEX, the communications executive.
>;
>; Attempting to determine the value of $CEX in the RSX-11M-PLUS Exec.
>;
>; Space for CEX (the Communications Executive) was generated within  the
>; RSX-11M-PLUS Executive at address 66560.
>;
>; Network events will be logged.
>;
>; Creating build files for STCRC, the CRC16 calculation routine.
>;
>* 03.00 Does the target system have a KG-11 [D=N]? [Y/N]: N
>;
>; Creating build files for CETAB, the CEX configuration tables.
>;
>; Creating build files for the DDMs, the device driver modules.
>;
>;       DMC
>;       DUP
>;
>;
>; <EOS>  Do you want to:
>*        <RET>-Continue, R-Repeat section, P-Pause, E-Exit [S]:
>;
```

Example 1 NETGEN Dialogue (Sheet 6 of 17)

```
>;
>; ======================================================================
>;  NET - Section  5 - Define the Comm Exec Support Components
>; ======================================================================
>;
>; Creating build files for NTINIT, the network initializer.
>;
>; Creating build files for NTL, the network loader.
>;
>; Creating build files for EVL/EVC, the event logger and collector.
>;
>* 06.01 How many Event Buffers should EVL use [D R:4.-50. D:6.]: 10
>* 06.02 Should EVC support file event logging [D=N]? [Y/N]: N
>* 06.03 Should EVC support monitor event logging [D=N]? [Y/N]: N
>* 06.04 Should EVC support remote event logging [D=N]? [Y/N]: N
>;
>;
>; <EOS>  Do you want to:
>*        <RET>-Continue, R-Repeat section, P-Pause, E-Exit [S]:
>;
>;
>; ======================================================================
>;  NET - Section  6 - Define the System Management Utilities
>; ======================================================================
>;
>; Creating build files for NCP, the network control program.
>;
>; Creating build files for NMVACP  and  NMDRV,  the  Network  Management
>; Volatile ACP and driver.
>;
>; Creating build files for CFE, the configuration file editor.
>;
>; Creating build files for NDA, the network dump analyzer.
>;
>;
>; <EOS>  Do you want to:
>*        <RET>-Continue, R-Repeat section, P-Pause, E-Exit [S]:
>;
>;
>; ======================================================================
>;  NET - Section  7 - Define the CEX Products
>; ======================================================================
>;
>; Questions concerning the following CEX Products will now be asked.
>;
>;      DECnet
```

Example 1 NETGEN Dialogue (Sheet 7 of 17)

```
>;
>;
>; ======================================================================
>;  DEC - DECnet CEX Product Generation Procedure
>;        Starting questions at 14:13:11 on 13-DEC-83
>; ======================================================================
>;
>; Copyright (C) 1981, 1982, 1983 by
>; Digital Equipment Corporation, Maynard, Mass.
>;
>; Generate the DECnet CEX Product for your RSX-11M-PLUS target
>; System.
>;
>; ======================================================================
>;  DEC - Section  1 - Define the target and remote nodes
>; ======================================================================
>;
>* 01.00 What is the target node name [S R:0-6]: ET
>* 02.00 What is the target node address [D R:1.-1023.]: 6
>* 03.00 Target node ID [D=None] [S R:0.-32.]: ET_The Extraterrestrial (sic)
>* 04.00 Do you want to generate a routing node [D=N]? [Y/N]: y
>* 05.00 Highest node address in the network [D R:6.-1023. D:6.]: 50
>;
>; To  include  support  for  products  layered  on DECnet you must first
>; include the required extended network support.
>;
>* 06.00 Do you want to include this extended network support [D=N]? [Y/N]:
>;
>; Answer this question YES, to include the extended network support for
>; the following options:
>;
>;   - Products layered on DECnet:
>;       - X.25 Gateway Access
>;       - SNA  Gateway Access
>;
>; If  you  do not  include the extended network support, you will not be
>; able  to  include  these  features  in  your  network.  Including this
>; support  will  cause  ECL and NETACP to increase in size by a total of
>; 1300. words.
>;
>; You  may  add  the  extended  network  support  later  by performing a
>; Component Mode generation.
```

Example 1 NETGEN Dialogue (Sheet 8 of 17)

```
>;
>* 06.00 Do you want to include this extended network support [D=N]? [Y/N]: Y
>* 07.00 Remote node name [<RET>=Done] [S R:0-6]: ZORRO
>* 07.01 Remote node address [D R:1.-50. D:1.]: 7
>;
>* 07.00 Remote node name [<RET>=Done] [S R:0-6]: CHICO
>* 07.01 Remote node address [D R:1.-50. D:8.]: 16
>;
>* 07.00 Remote node name [<RET>=Done] [S R:0-6]: GRUMPY
>* 07.01 Remote node address [D R:1.-50. D:17.]: 8
>;
>* 07.00 Remote node name [<RET>=Done] [S R:0-6]:
>;
>; The DECnet MACRO user library and FORTRAN/COBOL/BASIC+2  library  will
>; be included.
>;
>; The DECnet MACRO user library will  be  placed  on  your  target  disk
>; as 'DR03:[1,1]NETLIB.MLB'.
>;
>; The DECnet FORTRAN/COBOL/BASIC+2 object library will be placed on your
>; target disk  as 'DR03:[1,1]NETFOR.OLB'.
>;
>; If necessary, please move them to  the  library  disk  (LB:)  on  your
>; target system, if this is different from your target disk (DR03:).
>;
>;
>; <EOS>  Do you want to:
>*        <RET>-Continue, R-Repeat section, P-Pause, E-Exit [S]:
>;
>;
>; ====================================================================
>;  DEC - Section  2 - Define the DECnet Communications Components
>; ====================================================================
>;
>; Creating build files for XPT, the transport process.
>;
>; Creating build files for RCP, the routing event processing task.
>;
>; Creating build files for ECL, the network services process and driver.
>;
>; Creating build files for NETACP, the network services ACP.
>;
>* 05.01 Should NETACP be checkpointable [D=N]? [Y/N]: Y
>;
>; Creating build files for DCP, the DDCMP line protocol process.
>;
>;
>; <EOS>  Do you want to:
>*        <RET>-Continue, R-Repeat section, P-Pause, E-Exit [S]:
>;
```

Example 1 NETGEN Dialogue (Sheet 9 of 17)

```
>;
>;
>; ======================================================================
>;  DEC - Section  3 - Define the DECnet Network Management Components
>; ======================================================================
>;
>; Creating build files for NICE, the  network  information  and  control
>; server task.
>;
>; Creating build files for EVR, the network event receiver task.
>;
>* 03.01 Include file event logging [D=N]? [Y/N]:
>;
>; File logging allows EVR to append event data to a  system  wide  event
>; log file.  Answer this question YES if you wish to include support for
>; this option, and NO otherwise.
>;
>* 03.01 Include file event logging [D=N]? [Y/N]: N
>* 03.02 Include monitor event logging [D=N]? [Y/N]:
>;
>; Monitor logging allows EVR to hand event data to a user  written  task
>; for later processing.  Answer this question YES if you wish to include
>; support for this option, and NO otherwise.
>;
>* 03.02 Include monitor event logging [D=N]? [Y/N]: N
>;
>; Creating build files for NTD, the node state display task.
>;
>; Creating build files for NTDEMO, the node state display server task.
>;
>; Creating build files for LIN, the link watcher task.
>;
>; Creating build files for LOO, the loop test sender task.
>;
>; Creating build files for MIR, the loop test mirror task.
>;
>; Creating build files for NVP, the network connect verification task.
>;
>;
>; <EOS>  Do you want to:
>*        <RET>-Continue, R-Repeat section, P-Pause, E-Exit [S]:
>;
>;
>; ======================================================================
>;  DEC - Section  4 - Define the DECnet Satellite Support Components
>; ======================================================================
>;
>* 02.00 Do you want the Satellite Support Components [D=N]? [Y/N]:
>;
>; The DECnet Satellite Support Components are:
>;
>;       DLL     - The down-line system loader, host component
>;       DUM     - The up-line system dumper, host component
>;       HLD     - The down-line task loader, host component
```

Example 1 NETGEN Dialogue (Sheet 10 of 17)

```
>;
>; If  you  answer  YES  to  this question,  all of the Satellite Support
>; Components will be generated.  If you answer this question NO, you can
>; add these components by doing a Component Mode Generation.
>;
>* 02.00 Do you want the Satellite Support Components [D=N]? [Y/N]: Y
>;
>; Creating build files  for  DLL,  the  down-line  system  loader,  host
>; component.
>;
>; Creating  build  files  for  DUM,  the  up-line  system  dumper,  host
>; component.
>;
>; Creating  build  files  for  HLD,  the  down-line  task  loader,  host
>; component.
>;
>* 06.01 Number of incoming connections to support [D R:1.-32. D:4.]:
>;
>; Enter the number of simultaneous incoming connections that HLD  should
>; accept.  This will limit the number of down-line task loads which  can
>; be performed at the same time.
>;
>* 06.01 Number of incoming connections to support [D R:1.-32. D:4.]: 2
>;
>; To generate the HLD data base which describes the tasks  to  down-line
>; load, you may execute the HLDDAT.CMD command  procedure  at  any  time
>; after your network generation is complete.  This command file will  be
>; located as follows.
>;
>;      DR03:[300,001]HLDDAT.CMD
>;
>;
>; <EOS>  Do you want to:
>*        <RET>-Continue, R-Repeat section, P-Pause, E-Exit [S]:
>;
>;
>; ==================================================================
>;  DEC - Section  5 - Define the DECnet File Utilities
>; ==================================================================
>;
>; Creating build files  for  NFT, the network file access user task.
>;
>; Creating  build  files  for  FTS,  the  network  file  access  spooler
>; user task.
>;
>; Creating build files for FAL, the network file access server.
>;
>;
>; FAL will support RMS file access.
>; FAL will be a multi-copy object.
>; FAL will not be overlayed.
>* 04.03 Do you want to use Supervisor mode RMSRES [D=N]? [Y/N]:
>* 04.07 User data buffer size [D R:260.-1024. D:512.]:
>;
```

Example 1 NETGEN Dialogue (Sheet 11 of 17)

```
>; Creating build files for NCM, the network command or batch file
>; submission server.
>;
>* 05.01 Should requests be queued to Batch [D=N]? [Y/N]:
>;
>; If you answer this question YES, NFT Submit (/SB) and Execute (/EX)
>; requests will be entered into the Batch queue for execution.
>;
>; If you answer NO, the requests will be sent to the Indirect Command
>; File processor (AT.).
>;
>* 05.01 Should requests be queued to Batch [D=N]? [Y/N]: N
>;
>;
>; <EOS>  Do you want to:
>*        <RET>-Continue, R-Repeat section, P-Pause, E-Exit [S]:
>;
>;
>; ======================================================================
>;  DEC - Section  6 - Define the DECnet Terminal and Control Utilities
>; ======================================================================
>;
>; Creating build files for RMT and RMTACP, the remote network terminal
>; task and ACP.
>;
>* 02.01 Maximum number of simultaneous RMT users [D R:1.-15. D:4.]: 8
>;
>; Creating build files for HT: and RMHACP, the remote network terminal
>; driver and ACP.
>;
>* 03.01 Number of incoming connections to support [D R:1.-16. D:4.]:
>;
>; Enter the number of simultaneous incoming connections that RMHACP
>; should accept. This will limit the number of network terminal users
>; that can log onto your system at the same time.
>;
>* 03.01 Number of incoming connections to support [D R:1.-16. D:4.]:
>;
>; Creating build files for TLK, the remote talk user task.
>;

>; Creating build files for LSN, the remote talk server task.
>;
>; Creating build files for TCL, the remote task control server task.
>;
>;
>; <EOS>  Do you want to:
>*        <RET>-Continue, R-Repeat section, P-Pause, E-Exit [S]:
>;
```

Example 1 NETGEN Dialogue (Sheet 12 of 17)

```
>;
>; .
>; ===================================================================
>;  DEC - DECnet CEX Product Generation Procedure
>;        DECnet question/answer section completed at 14:23:01 on 13-DEC-83
>; ===================================================================
>;
>; --
>; ===================================================================
>;  NET - Section  8 - Complete the CEX System Definitions
>; ===================================================================
>;
>; Creating build files for DLX, the Direct Line Access process.
>;
>* 02.00 What is the Large Data Buffer (LDB) size [D R:192.-1484. D:292.]:
>;
>; <EOS>  Do you want to:
>*        <RET>-Continue, R-Repeat section, P-Pause, E-Exit [S]:
>;
>;
>; ===================================================================
>;  NET - Section  9 - Build the CEX System at 14:23:50 on 13-DEC-83
>; ===================================================================
>;
>; All questions have now been asked and the selected components will now
>; be built.  This may take from one to three  hours,  depending  on  the
>; selection of components and the system you are running on.
>;
>PIP DR03:[300,054]CETAB.MAC/PU/NM
>PIP DR03:[300,054]AUX.DAT/PU/NM
>;
>SET /UIC=[1,2]
>LBR OU:NETHLP.ULB/CR:10:::UNI
>PIP OU:NETHLP.ULB/PU/NM
>SET /UIC=[300,24]
>SET /UIC=[1,2]
>LBR OU:[1,2]NCP.HLP=IN:[1,2]NETHLP.HLB/EX:NC1
>LBR OU:[1,2]NCPCLE.HLP=IN:[1,2]NETHLP.HLB/EX:NC1CLE
>LBR OU:[1,2]NCPLOA.HLP=IN:[1,2]NETHLP.HLB/EX:NC1LOA
>LBR OU:[1,2]NCPLOO.HLP=IN:[1,2]NETHLP.HLB/EX:NC1LOO
>LBR OU:[1,2]NCPSET.HLP=IN:[1,2]NETHLP.HLB/EX:NC1SET
>LBR OU:[1,2]NCPSHO.HLP=IN:[1,2]NETHLP.HLB/EX:NC1SHO
>LBR OU:[1,2]NCPTRI.HLP=IN:[1,2]NETHLP.HLB/EX:NC1TRI
>LBR OU:[1,2]NCPZER.HLP=IN:[1,2]NETHLP.HLB/EX:NC1ZER
>LBR OU:[1,2]NETHLP.ULB=OU:[1,2]NCP.HLP/RP
>LBR OU:[1,2]NETHLP.ULB=OU:[1,2]NCPCLE.HLP/RP
>LBR OU:[1,2]NETHLP.ULB=OU:[1,2]NCPLOA.HLP/RP
>LBR OU:[1,2]NETHLP.ULB=OU:[1,2]NCPLOO.HLP/RP
>LBR OU:[1,2]NETHLP.ULB=OU:[1,2]NCPSET.HLP/RP
>LBR OU:[1,2]NETHLP.ULB=OU:[1,2]NCPSHO.HLP/RP
>LBR OU:[1,2]NETHLP.ULB=OU:[1,2]NCPTRI.HLP/RP
>LBR OU:[1,2]NETHLP.ULB=OU:[1,2]NCPZER.HLP/RP
>PIP OU:[1,2]NCP.HLP;*/DE/NM,NCPCLE.HLP;*,NCPLOA.HLP;*
>PIP OU:[1,2]NCPLOO.HLP;*/DE/NM,NCPSET.HLP;*,NCPSHO.HLP;*
>PIP OU:[1,2]NCPTRI.HLP;*/DE/NM,NCPZER.HLP;*
>SET /UIC=[300,24]
>SET /UIC=[1,2]
```

Example 1 NETGEN Dialogue (Sheet 13 of 17)

```
>LBR OU:[1,2]CFE.HLP=IN:[1,2]NETHLP.HLB/EX:CF1
>LBR OU:[1,2]CFEDEF.HLP=IN:[1,2]NETHLP.HLB/EX:CF1DEF
>LBR OU:[1,2]CFELIS.HLP=IN:[1,2]NETHLP.HLB/EX:CF1LIS
>LBR OU:[1,2]NETHLP.ULB=OU:[1,2]CFE.HLP/RP
>LBR OU:[1,2]NETHLP.ULB=OU:[1,2]CFEDEF.HLP/RP
>LBR OU:[1,2]NETHLP.ULB=OU:[1,2]CFELIS.HLP/RP
>PIP OU:[1,2]CFE.HLP;*/DE/NM,CFEDEF.HLP;*,CFELIS.HLP;*
>SET /UIC=[300,24]
>SET /UIC=[1,2]
>LBR OU:[1,2]NDA.HLP=IN:[1,2]NETHLP.HLB/EX:NDA
>LBR OU:[1,2]NETHLP.ULB=OU:[1,2]NDA.HLP/RP
>PIP OU:[1,2]NDA.HLP;*/DE/NM
>SET /UIC=[300,24]
>;
>TKB @OU:[300,24]CEXBLD.CMD
>PIP OU:[300,54]CEX.TSK/PU/NM,CEX.STB,AUX.TSK,AUX.STB
>TKB @OU:[300,24]STCRCBLD.CMD
>PIP OU:[300,54]STCRC.TSK/PU/NM,STCRC.STB
>MAC @OU:[300,24]CETABASM.CMD
>PIP OU:[300,24]CETAB.OBJ/PU/NM
>TKB @OU:[300,24]CETABBLD.CMD
>PIP OU:[300,54]CETAB.TSK/PU/NM,CETAB.STB
>TKB @OU:[300,24]NTINITBLD.CMD
>PIP OU:[300,54]NTINIT.TSK/PU/NM
>STK @OU:[300,24]NTLBLD.CMD
>PIP OU:[300,54]NTL.TSK/PU/NM
>MAC @OU:[300,24]EVLASM.CMD
>PIP OU:[300,24]EVLDAT.OBJ/PU/NM
>TKB @OU:[300,24]EVLBLD.CMD
>PIP OU:[300,54]/NV=IN:[131,24]EVL.DAT
>PIP OU:[300,54]EVL.TSK/PU/NM,EVL.STB,EVL.DAT
>TKB @OU:[300,24]EVCBLD.CMD
>PIP OU:[300,54]EVC.TSK/PU/NM
>TKB @OU:[300,24]NCPBLD.CMD
>PIP OU:[300,54]NCP.TSK/PU/NM
>MAC @OU:[300,24]NMVACPASM.CMD
>PIP OU:[300,24]NMDRV.OBJ/PU/NM,NMTAB.OBJ
>STK @OU:[300,24]NMVACPBLD.CMD
>PIP OU:[300,54]NMVACP.TSK/PU/NM,NMDRV.TSK,NMDRV.STB
>PIP OU:[300,54]/NV/CO=IN:[132,54]CFE.TSK
>PIP OU:[300,54]CFE.TSK/PU/NM
>PIP OU:[300,54]/NV/CO=IN:[141,54]NDA.TSK
>PIP OU:[300,54]NDA.TSK/PU/NM
>MAC @OU:[300,24]DLXASM.CMD
>PIP OU:[300,24]DLXTAB.OBJ/PU/NM
>TKB @OU:[300,24]DLXBLD.CMD
>PIP OU:[300,54]DLX.DAT/NV=IN:[131,24]DLX.DAT
>PIP OU:[300,54]DLX.TSK/PU/NM,DLX.STB,DLX.DAT
>PIP OU:[300,54]DLXTAB.TSK/PU/NM,DLXTAB.STB
>;
```

Example 1 NETGEN Dialogue (Sheet 14 of 17)

```
/,
>TKB @OU:[300,24]DMCBLD.CMD
>PIP OU:[300,54]DMC.TSK/PU/NM,DMC.STB
>PIP OU:[300,54]/NV=IN:[131,24]DMC.DAT
>PIP OU:[300,54]DMC.DAT/PU/NM
>TKB @OU:[300,24]DDHARBLD.CMD
>PIP OU:[300,54]DDHAR.TSK/PU/NM,DDHAR.STB
>TKB @OU:[300,24]DUPBLD.CMD
>PIP OU:[300,54]DUP.TSK/PU/NM,DUP.STB
>PIP OU:[300,54]/NV=IN:[131,24]DUP.DAT
>PIP OU:[300,54]DUP.DAT/PU/NM
>SET /UIC=[1,1]
>LBR OU:[1,1]NETLIB/CR:1:0:128.:MAC=IN:[130,10]DLXDF.MAC
>LBR OU:[1,1]NETLIB.MLB=IN:[130,10]EPMDF.MAC,CHRDF.MAC
>LBR OU:[1,1]NETLIB.MLB=IN:[130,10]NETDEF.MAC,NETMAC.MAC
>LBR OU:[1,1]NETLIB.MLB=IN:[130,10]NFAMAC.MAC
>PIP OU:[1,1]NETLIB.MLB/PU
>LBR OU:[1,1]NETFOR.OBS=IN:[133,24]DAPFOR/EX
>LBR OU:[1,1]NETTMP.OBS=IN:[133,24]DAPTRC/EX
>LBR OU:[1,1]NETFOR/CO:1:320.:192.=IN:[134,24]NETFOR.OLB
>LBR OU:[1,1]NETFOR/IN=OU:[1,1]NETFOR.OBS
>LBR OU:[1,1]NETFOR/IN/-EP=OU:[1,1]NETTMP.OBS
>LBR OU:[1,1]NETFOR/DG:$MBLUN:$ASTBL:$ASTLU
Entry points deleted:
$MBLUN
$ASTBL
$ASTLU

>PIP OU:[1,1]NETFOR.OBS;*/DE,NETTMP.OBS;*
>PIP OU:[1,1]NETFOR.OLB/PU
>SET /UIC=[300,024]
>PIP OU:[300,024]/NV=IN:[137,24]NTEST.CMD
>PIP OU:[300,024]NTEST.CMD/PU/NM
>SET /UIC=[300,054]
>PIP OU:[300,054]/NV/CO=IN:[133,54]DTS.TSK,DTR.TSK
>PIP OU:[300,054]DTS.TSK/PU/NM,DTR.TSK
>SET /UIC=[300,024]
>TKB @OU:[300,24]XPTBLD.CMD
>PIP OU:[300,54]XPT.DAT/NV=IN:[131,24]XPT.DAT
>PIP OU:[300,54]XPT.TSK/PU/NM,XPT.STB,XPT.DAT
>TKB @OU:[300,24]RCP1BLD.CMD
>PIP OU:[300,54]RCP1.TSK/PU/NM
>MAC @OU:[300,24]ECLASM.CMD
>PIP OU:[300,24]ECLTAB.OBJ/PU/NM
>TKB @OU:[300,24]ECLBLD.CMD
>PIP OU:[300,54]ECL.DAT/NV=IN:[131,24]ECL.DAT
>PIP OU:[300,54]ECL.TSK/PU/NM,ECL.STB,ECLTAB.TSK,ECLTAB.STB,ECL.DAT
>TKB @OU:[300,24]NETACPBLD.CMD
>PIP OU:[300,54]NETACP.TSK/PU/NM,NETACP.STB
>TKB @OU:[300,24]DCPBLD.CMD
>PIP OU:[300,54]/NV=IN:[131,24]DCP.DAT,DCPST.DAT
>PIP OU:[300,54]DCP.TSK/PU/NM,DCP.STB,DCP.DAT,DCPST.DAT
>TKB @OU:[300,24]NICEBLD.CMD
>PIP OU:[300,54]NICE.TSK/PU/NM
>TKB @OU:[300,24]EVRBLD.CMD
>PIP OU:[300,54]EVR.TSK/PU/NM
```

Example 1 NETGEN Dialogue (Sheet 15 of 17)

```
>TKB @OU:[300,24]NTDBLD.CMD
>PIP OU:[300,54]NTD.TSK/PU/NM
>TKB @OU:[300,24]NTDEMOBLD.CMD
>PIP OU:[300,54]NTDEMO.TSK/PU/NM
>TKB @OU:[300,24]LINBLD.CMD
>PIP OU:[300,54]LIN.TSK/PU/NM
>TKB @OU:[300,24]LOOBLD.CMD
>PIP OU:[300,54]LOO.TSK/PU/NM
>TKB @OU:[300,24]MIRBLD.CMD
>PIP OU:[300,54]MIR.TSK/PU/NM
>TKB @OU:[300,24]NVPBLD.CMD
>PIP OU:[300,54]NVP.TSK/PU/NM
>TKB @OU:[300,24]DLLBLD.CMD
>PIP OU:[300,54]DLL.TSK/PU/NM
>PIP OU:[300,54]/NV/CO=IN:[136,54]SEC*.SYS,TER*.SYS
>PIP OU:[300,54]SEC*.SYS/PU/NM,TER*.SYS
>TKB @OU:[300,24]DUMBLD.CMD
>PIP OU:[300,54]DUM.TSK/PU/NM
>TKB @OU:[300,24]HLDBLD.CMD
>PIP OU:[300,54]HLD.TSK/PU/NM
>PIP OU:[300,1]/NV=IN:[132,24]HLDDAT.CMD
>PIP OU:[300,1]HLDDAT.CMD/PU/NM
>TKB @OU:[300,24]NFTBLD.CMD
>PIP OU:[300,54]NFT.TSK/PU/NM
>SET /UIC=[1,2]
>LBR OU:[1,2]NFT.HLP=IN:[1,2]DECHLP.HLB/EX:NFT
>LBR OU:[1,2]NETHLP.ULB=OU:[1,2]NFT.HLP/RP
>PIP OU:[1,2]NFT.HLP;*/DE/NM
>SET /UIC=[300,24]
>MAC @OU:[300,24]FTSASM.CMD
>TKB @OU:[300,24]FTSBLD.CMD
>PIP OU:[300,54]FTS.TSK/PU/NM,FTSDEQ.TSK,FTSPRM.OBJ
>SET /UIC=[1,2]
>LBR OU:[1,2]FTS.HLP=IN:[1,2]DECHLP.HLB/EX:FTS
>LBR OU:[1,2]NETHLP.ULB=OU:[1,2]FTS.HLP/RP
>PIP OU:[1,2]FTS.HLP;*/DE/NM
>SET /UIC=[300,24]
>MAC @OU:[300,24]FALASM.CMD
>PIP OU:[300,24]FALPRM.OBJ/PU/NM
>TKB @OU:[300,24]FALBLD.CMD
>PIP OU:[300,54]FAL.TSK/PU/NM
>TKB @OU:[300,24]MCMBLD.CMD
>PIP OU:[300,54]MCM.TSK/PU/NM
>MAC @OU:[300,24]RMTASM.CMD
>PIP OU:[300,24]RMTACP.OBJ/PU/NM
>TKB @OU:[300,24]RMTBLD.CMD
>PIP OU:[300,54]RMT.TSK/PU/NM,RMTACP.TSK
>MAC @OU:[300,24]RMHACPASM.CMD
>PIP OU:[300,24]RMHACP.OBJ/PU/NM,HTTAB.OBJ
```

Example 1 NETGEN Dialogue (Sheet 16 of 17)

```
>TKB @OU:[300,24]RMHACPBLD.CMD
>PIP-OU:[300,54]RMHACP.TSK/PU/NM,HTDRV.TSK,HTDRV.STB
>TKB @OU:[300,24]TLKBLD.CMD
>PIP-OU:[300,54]TLK.TSK/PU/NM
>TKB @OU:[300,24]LSNBLD.CMD
>PIP-OU:[300,54]LSN.TSK/PU/NM
>MAC @OU:[300,24]TCLASM.CMD
>PIP-OU:[300,24]TCLPWD.OBJ/PU/NM
>TKB @OU:[300,24]TCLBLD.CMD
>PIP-OU:[300,54]TCL.TSK/PU/NM
>;
>; ===================================================================
>;  NET - Section 10 - Generation Clean Up
>; ===================================================================
>;
>; Creating NETCFG.TXT, the network configuration description file.
>; Creating NETCFE.CMD, the network configuration command file.
>; Creating NETINS.CMD, the network installation command file.
>; Creating NETREM.CMD, the network removal command file.
>;
>;
>; The network HELP file library has been placed on your target disk, as:
>;
>;      DR03:[1,2]NETHLP.ULB
>;
>; This file should be moved to your target systems library device (LB:).
>;
>;
>; ===================================================================
>;  NET - RSX-11M-PLUS CEX System Generation Procedure
>;        Stopped at 15:33:59 on 13-DEC-83
>; ===================================================================
>;
>@ <EOF>
>
>
>
>
>
>
>
>
>
```

Example 1 NETGEN Dialogue (Sheet 17 of 17)

```
>
>@LB:[300,1]NETINS
>* Do you want to install and load the CEX system? [Y/N]: Y
>* Do you want to install and start DECnet? [Y/N]: Y
>* On what device are the network tasks [D=DL0:] [S]:
>* What is the network UIC group code [O R:1-377 D:5]: 300
>INS XX:[300,54]NTINIT
>INS XX:[300,54]NTL
>INS XX:[300,54]EVC
>INS XX:[300,54]NCP
>; INS XX:[300,54]CFE
>INS XX:[300,54]NMVACP
>SET /SYSUIC=[300,54]
>ASN XX:=LB:
>LOA NM:/HIGH
>ASN =LB:
>SET /SYSUIC=[1,54]
>INS XX:[300,54]NETACP
>INS XX:[300,54]NICE
>INS XX:[300,54]EVR
>INS XX:[300,54]NTD
>INS XX:[300,54]NTDEMO
>INS XX:[300,54]LIN
>INS XX:[300,54]LOO
>INS XX:[300,54]MIR
>INS XX:[300,54]NVP
>INS XX:[300,54]NFT
>INS XX:[300,54]FAL
>INS XX:[300,54]MCM
>INS XX:[300,54]RMT
>INS XX:[300,54]RMTACP/CLI=YES
>CLI /INIT=RMTACP/NULL/RESTRICT/CPR="<15><12>/RMT>/"
>SET /SYSUIC=[300,54]
>ASN XX:=LB:
>LOA HT:/HIGH
>SET /SYSUIC=[1,54]
>ASN =LB:
>INS XX:[300,54]RMHACP
>INS XX:[300,54]TLK
>INS XX:[300,54]LSN
>INS XX:[300,54]TCL
>ASN =XX:
>SET /UIC=[5,1]
>NCP SET SYS
>NCP SET EXE STA ON

 Event type 2.0, Local node state change
Occurred 14-DEC-83 09:38:57 on node 2.231 (ZIRCON)
Reason for state change: Operator command, Old node state = Off
New node state = On

>@ <EOF>
>
```

Example 2 Bringing Up DECnet Software

```
>
>
>@LB:[300,24]NTEST
>;
>;========================================================================
>;          *   RSX-11M/M-Plus Network Installation Procedures    *
>;========================================================================
>;
>; Copyright (C) 1979, 1980,1981,1982,1983 by
>; Digital Equipment Corporation, Maynard, Mass.
>;
>; Test procedure performed at 09:56:05 on 14-DEC-83
>;
>; This procedure is designed to allow a user to  perform  the  following
>; tests on a DECnet-11M/M-Plus system:
>;
>;    . Internal Node Level --  Test  the  local  node  without  using any
>;      communication circuits.
>;    . Circuit Level -- Test the local node with a communications line at-
>;      tached to a turnaround/loopback  device or  a modem with  loopback
>;      capabilities.
>;    . Remote Node Level -- Test the local node by attempting to communi-
>;      cate with a Remote Node.
>;
>;
>; The procedure will execute DTS/DTR  tests  and  will  use  NFT/FAL  to
>; transfer and execute files.
>;
>; Before running this procedure, you should have access to the following
>; information:
>;
>;    . The  device and group code of the  UIC  under  which  the  network
>;      tasks are stored.
>;    . The target node name (for local tests,  this  is  the  local  node
>;      name).
>;    . The  necessary  UIC  and  password to access files residing on the
>;      target node, for NFT tests.
>;
>;
>* Do you want to: <CR>-continue E-exit P-pause [S]:
>;
>;========================================================================
>;                        *    Test  Parameters    *
>;========================================================================
>;
>; If you do not choose long dialogue mode,  explanatory  text  for  each
>; question is available by hitting the 'ESCAPE' key.
>;
>* Do you want long dialogue mode? [Y/N]: Y
>;
>; Enter the UIC group code under which the  network  files  were  stored
>; for this node's network.
>;
>* What group UIC is the network stored under [O R:1-377]: 300
>;
>; Specify the device on which the network files are stored.  The default
>; device is 'SY:'.
>;
>* On what device are the network files (DDNN) [S R:0-5]: SY0
>;
>; Answer 'YES' to the next  question if  this is an Internal  Node Test,
>; i.e. the  test  does  not  involve  communications lines and/or remote
>; nodes.
>;
>* Is this an Internal Node Level test? [Y/N]: N
>;
>; If you want to set up your system for the  tests,  for  example  issue
>; some ncp  commands to define a Loopback Node Name  or turn  on a line,
>; this procedure will pause.  When you are ready, continue with the pro-
>; cedure.
>;
>* Do you want to pause to configure your system? [Y/N]: N
>;
>; Enter the target node name ( 1-6 alphanumeric characters). This is the
>; node name of the  remote node. If this  is an Internal Level test,then
>; the Local Node Name should be used. If this is a line level test, then
>; the Loopback Node Name should be used.
>;
>* What is the target node name [S R:1-6]: RSXHUB
>;
>; DTS will now be invoked to perform three tests:
>;        1.) DTS will issue  a connect request to DTR causing
>;            DTR to be loaded, answer the request and task
>;            exit
>;        2.) DTS/DTR will execute a 1 minute pattern test
>;            using a 10 byte message size
>;        3.) DTS/DTR will execute a 1 minute pattern test
```

Example 3 Testing DECnet-RSX (Sheet 1 of 2)

```
>;          using an 1100 byte message size.
>;
>; The 10 byte message size was selected because  this  size  is  smaller
>; than NSP's segment size...thus there  will  be  1  segment  for  each
>; message.  The  1100 byte message size is larger than the segment size;
>; therefore, there will be multi-segments for each  of  these  messages.
>;
>INS SY0:[300,54]DTS/INC=600.
>INS SY0:[300,54]DTR/INC=600.
>PIP DTSTST.CMD;*/DE/NM
>DTS @DTSTST
 LOAD/EXEC DTR VIA CONNECT TEST
DTS -- Test finished at 9:57:12
 PAT/MSG=10/TIME=1M
DTS -- Test started at 9:57:12
DTS -- Test finished at 9:58:12
  1903 Messages sent
   317 Characters/second
  2536 Baud

DTS -- Connect error...status 177667 0
>PIP DTSTST.CMD;*/DE/NM
>;
>;
>;
>; In order to execute NFT  file transfer  tests, you  must  supply valid
>; access control information for the target node RSXHUB.This information
>; is
>;       (1).  A user identification (the format depends on what type  of
>;             system RSXHUB is.
>;       (2).  The password for that user-id.
>;
>* What is the user-id for NFT tests on node RSXHUB [S R:1.-16.]: RYAN
>* What is the corresponding password [S R:0.-8.]: JESSI
>PIP NFTTS1.ZZZ;*/DE/NM,NFTTS2;*,NFTTS3;*,NFTTS4;*,NFTTS5;*
>;
>; NFT will now be invoked. A copy of the command file NTEST.CMD will be
>; transferred to node RSXHUB with the filename TMPNETTST.ZZZ  using  the
>; access control information you specified. NFT  will  then  delete  the
>; file.
>;
>NFT @NFTTS1.ZZZ/TR
NFT>RSXHUB/RYAN/JESSI::TMPNETTST.ZZZ;1=SY0:[300,24]NTEST.CMD
>;
>; The file TMPNETTST.ZZZ will now be deleted using NFT.
>;
>NFT @NFTTS2.ZZZ/TR
NFT>RSXHUB/RYAN/JESSI::TMPNETTST.ZZZ;1/DE
>;
>PIP NFTTS1.ZZZ;*/DE/NM,NFTTS2;*,NFTTS3;*,NFTTS4;*,NFTTS5;*
>;
>; If you have reached this  point  with  no  errors,  then  you  can  be
>; highly confident that the node has been created properly.
>;
>; End of test at 09:58:54
>;
>;=====================================================================
>;                  *    End of Installation Test    *
>;=====================================================================
>@ <EOF>
>
>
>
>
```

Example 3 Testing DECnet-RSX (Sheet 2 of 2)

```
PIP TI:=[300,54]CETAB.MAC
        .TITLE  CETAB
        .IDENT  /V04.0/
        ;
        ; Copyright (C) 1982, 1983 by
        ; Digital Equipment Corporation, Maynard, Mass.
        ;
        .MCALL  SYS$DF
        SYS$DF  <RS$11M,M$$MGE>
        PDV$DF  <AUX,EVL,ECL,XPT,DLX>,<>,<DMC>
        SLT$DF  DMC,DMC,XPT,LF.MFL!LF.ENA!LF.TIM,0,0,,,15.
        LLC$DF  AUX,ZF.LLC!ZF.TIM!ZF.MTM!ZF.MFL,0,0.
        LLC$DF  EVL,ZF.LLC!ZF.TIM!ZF.MFL,0,0.
        LLC$DF  ECL,ZF.LLC!ZF.TIM!ZF.MFL!ZF.COU,0,0,10.,NS,,,,4.,0
        LLC$DF  XPT,ZF.LLC!ZF.TIM!ZF.MFL!ZF.COU,0,1.
        LLC$DF  DLX,ZF.LLC!ZF.TIM!ZF.MFL,0,1.,,NX
        DDM$DF  DMC,ZF.DDM!ZF.DLC!ZF.COU!ZF.MAN,5,0,1.
        CNT$DF  0,310,160070,5,,NX
        UNT$DF  0,177470,1,,5.
        LOG$ST  1
        ROU$DF  0.,1022.,10.,300.,0.,0.,0.,1022.,0.
        NOD$DF  <ZIRCON>,<ZIRCON_ENCRUSTED_TWEEZERS>,2.,231.,2.,231.
        FEA$DF  NS,000627,100000,100000
        BUF$DF  8.,38.,9.,576.,11.,34.,2.,0.
        PAR$DF  ,GEN,17.,TOP,2,9.
        OBJ$DF  0.,,2,0,1
        OBJ$DF  15.,<TCL...>,0,0,1
        OBJ$DF  16.,<LSN$$$>,2,200,5
        OBJ$DF  17.,<FAL...>,0,0,1
        OBJ$DF  18.,<HLD...>,2,0,1
        OBJ$DF  19.,<NIC$$$>,1,200,5
        OBJ$DF  23.,<RMHACP>,2,0,1
        OBJ$DF  25.,<MIR$$$>,2,200,5
        OBJ$DF  26.,<EVR$$$>,2,200,5
        OBJ$DF  63.,<DTR...>,2,0,1
        EVT$DF  0,<311,0,0,0>,CONSOLE
        EVT$DF  200,<3,0,0,0>,CONSOLE
        EVT$DF  300,<4,0,0,0>,CONSOLE
        EVT$DF  400,<137777,5,0,0>,CONSOLE
        EVT$DF  500,<160000,1,0,0>,CONSOLE
        EVT$DF  10000,<6,0,0,0>,CONSOLE
        EVT$DF  10400,<40000,0,0,0>,CONSOLE
        EVT$DF  10500,<1,0,0,0>,CONSOLE
        REM$DF  <VORTEX>,2.,1.
        REM$DF  <KERMIT>,2.,60.
        REM$DF  <CASTOR>,2.,62.
        REM$DF  <RSXHUB>,2.,65.
        REM$DF  <BEING>,2.,95.
        REM$DF  <HARDY>,2.,149.
        REM$DF  <ALIEN>,2.,226.
        REM$DF  <SUPER>,2.,230.
        END$DF  ; last edit by TTO [5,2] on 2-DEC-83 02:21:17
        EXPPDT  <AUX,EVL,ECL,XPT,DLX>,<>,<DMC>
        EXPSLT  <>,<DMC>
        EXPLLT  <AUX,EVL,ECL,XPT,DLX>
        EXPSTT
        EXPPUT  <AUX,EVL,ECL,XPT,DLX>,<>,<DMC>
        .LIST
        .END
>
```

Example 4 CETAB.MAC

```
CETAB   MACRO V05.00  Wednesday 14-Dec-83 10:43  Page 3

                                        .TITLE  CETAB
                                        .IDENT  /V04.0/
                                        ;
                                        ; Copyright (C) 1982, 1983 by
                                        ; Digital Equipment Corporation, Maynard, Mass.
                                        ;
                                        .MCALL  SYS$DF
      000000                            SYS$DF  <RS$11M,MS$MGE>

                                ;
                                ;       PROCESS DESCRIPTOR ADDRESS TABLE
                                ;
                                ;       EACH ENTRY IS THE ADDRESS OF THE CORRESPONDING
                                ;       PROCESS DESCRIPTOR VECTOR ( BELOW ).
                                ;
                                ;               PROCESS PROCESS PROCESS PROCESS
                                ;                NAME    LEVEL   INDEX    MFL
      000000                    $PDVTB::
      000000  000042'           .WORD   AUXPDV  ; AUX      LLC       0.      YES    ; (A) ;
      000002  000062'           .WORD   EVLPDV  ; EVL      LLC       2.      YES    ; (A) ;
      000004  000102'           .WORD   ECLPDV  ; ECL      LLC       4.      YES    ; (A) ;
      000006  000122'           .WORD   XPTPDV  ; XPT      LLC       6.      YES    ; (A) ;
      000010  000146'           .WORD   DLXPDV  ; DLX      LLC       8.      YES    ; (A) ;
      000012  000172'           .WORD   DMCPDV  ; DMC      DDM      10.       NO    ; (A) ;
      000014                    $PDVND::

                                ;
                                ;       SYSTEM LINE TABLE OFFSETS AND FLAG WORD BIT DEFINITIONS
                                ;
      000000                    L.FLG:  .BLKW   1       ; FLAGS WORD
      000002                    L.DDM:  .BLKB   1       ; DDM PROCESS INDEX
      000003                    L.DLC:  .BLKB   1       ; DLC PROCESS INDEX
      000004                    L.DDS:  .BLKW   1       ; DDM LINE TABLE ADDRESS
      000006                    L.DLM:  .BLKW   1       ; DLC LINE TABLE RELOCATION BIAS
      000010                    L.DLS:  .BLKW   1       ; DLC LINE TABLE ( VIRTUAL ) ADDRESS
      000012                    L.CTL:  .BLKB   1       ; CONTROLLER NUMBER
      000013                    L.UNT:  .BLKB   1       ; MULTIPLEXER UNIT NUMBER
      000014                    L.NSTA: .BLKB   1       ; NUMBER OF TRIBUTARIES ON THIS LINE
      000015                    L.COST: .BLKB   1       ; LINE COST FOR ROUTING TASK
      000016                    L.KRBA: .BLKW   1       ; POINTER TO CONTROLLER REQUEST BLOCK
      000020                    L.NMST: .BLKB   1       ; NETWORK MANAGEMENT LINE STATE/SUBSTATE
      000021                    L.OWNR: .BLKB   1       ; LINE OWNER PROCESS INDEX
      000022                    L.MPF:                  ; START OF MULTIPOINT FLAG TABLE
              000022            L.LEN=.                 ; LENGTH OF EACH SLT ENTRY
                                ;
                                ;       FLAG WORD DEFINITIONS
                                ;
              000000            LF.X2P=000000   ; LINE NEEDS PARAMETERS FROM X2P$DF
              000007            LF.BWT=000007   ; BUFFER WAIT QUEUE COUNT
              000010            LF.TIM=000010   ; LINE NEEDS TIMER SERVICE
              000020            LF.MTP=000020   ; LINE IS MULTI-POINT
              000040            LF.SER=000040   ; CIRCUIT HAS SERVICE DISABLED
              000100            LF.MDC=000100   ; LINE NEEDS MODEM CONTROL
              000200            LF.PAC=000200   ; LINE WAS PREVIOUSLY ACTIVE
                                                        ; (BEFORE RECONFIGURATION)
              000400            LF.BRO=000400   ; LINE IS A BROADCAST CHANNEL (ETHERNET)
              001000            LF.LPB=001000   ; LINE IS IN LOOPBACK
```

Example 5 CETAB.LST (Sheet 1 of 7)

```
CETAB   MACRO V05.00  Wednesday 14-Dec-83 10:43  Page 3-1

         002000                  LF.ENA=002000    ; LINE IS TO BE ENABLED AT INITIALIZATION TIME
         004000                  LF.MFL=004000    ; LINE IS MARKED FOR LOAD AT INITIALIZATION TIME
         010000                  LF.REA=010000    ; LINE IS REASSIGNABLE
         020000                  LF.UNL=020000    ; LINE IS MARKED FOR UNLOAD
         040000                  LF.RDY=040000    ; LINE IS READY (DDM,DLC PROCESSES AND LINE TABLES ASSIGNED)
         100000                  LF.ACT=100000    ; LINE IS ACTIVE ( LINE IS ASSIGNED TO AN LLC )
                                 ;
                                 ;       NETWORK MANAGEMENT STATE/SUBSTATE BYTE DEFINITIONS
                                 ;          STATE = LOW 4 BITS
                                 ;          SUBSTATE = HIGH 4 BITS
                                 ;
         000017                  LN.STA=17        ; STATE FIELD BIT MASK
         000360                  LN.SUB=360       ; SUBSTATE FIELD BIT MASK
         000000                  LN.ON=0          ; STATE = ON
         000001                  LN.OFF=1         ; STATE = OFF
         000002                  LN.SER=2         ; STATE = SERVICE
         000003                  LN.OAU=3         ; STATE = ON, AUTOSERVICE
         000004                  LN.OOP=4         ; STATE = ON, OPEN
         000000                  LN.CLO=0         ; SUBSTATE = CLOSED
         000001                  LN.OPE=1         ; SUBSTATE = OPEN, UNSPECIFIED
         000002                  LN.REF=2         ; SUBSTATE = OPEN, REFLECTING
         000003                  LN.LOO=3         ; SUBSTATE = OPEN, LOOPING
         000004                  LN.LOA=4         ; SUBSTATE = OPEN, LOADING
         000005                  LN.DUM=5         ; SUBSTATE = OPEN, DUMPING
         000006                  LN.TRI=6         ; SUBSTATE = OPEN, TRIGGERING
         000000                          .=0
                                 ;
                                 ;       MULTIPOINT FLAG TABLE OFFSETS AND FLAG BYTE DEFINITIONS
                                 ;
000000                           S.FLG:  .BLKB   1       ; FLAGS BYTE
000001                           S.COST: .BLKB   1       ; TRIBUTARY COST FOR ROUTING TASK
000002                           S.NMST: .BLKB   1       ; NETWORK MANAGEMENT TRIBUTARY STATE/SUBSTATE
000003                           S.OWNR: .BLKB   1       ; TRIBUTARY OWNER PROCESS INDEX
000004                           S.LEN:                  ; LENGTH OF TRIBUTARY EXTENSION
                                 ;
                                 ;       FLAG BYTE DEFINITIONS
                                 ;
         000200                  SF.ACT=000200    ; TRIBUTARY IS ACTIVE
         000100                  SF.ENA=000100    ; TRIBUTARY IS MARKED FOR ENABLE AT INITIALIZATION TIME
         000040                  SF.UNL=000040    ; TRIBUTARY IS MARKED FOR UNLOAD
         000040                  SF.MFL=000040    ; TRIBUTARY IS MARKED FOR LOAD (DATA LINK MAPPING ONLY)
         000020                  SF.PAC=000020    ; TRIBUTARY WAS PREVIOUSLY ACTIVE
                                                         ; (BEFORE RECONFIGURATION)
         000010                  SF.REA=000010    ; TRIBUTARY IS REASSIGNABLE
         000004                  SF.LPB=000004    ; TRIBUTARY IS IN LOOPBACK
         000002                  SF.SVC=000002    ; DLM CIRCUIT IS AN SVC
         000001                  SF.SER=000001    ; CIRCUIT HAS SERVICE DISABLED

                                 ;
                                 ;       SYSTEM LINE MAPPING TABLE
                                 ;
000014                           $SLTMB::
000014  000016'                          .WORD   SLT0    ; SYSTEM LINE TABLE 0

                                 ;
                                 ;       SYSTEM LINE TABLE
                                 ;
```

Example 5 CETAB.LST (Sheet 2 of 7)

```
CETAB   MACRO V05.00  Wednesday 14-Dec-83 10:43  Page 3-2

     000016                          $SLTTB::
                                     ;    SLNO DMC-0-0
     000016  006010                  SLT0:   .WORD   6010    ; (D) ; FLAG FOR DMC0
     000020     012                          .BYTE   10.     ; (D) ; DMC OFFSET IN PDVTB::
     000021     012                          .BYTE   10.     ; (D) ; DMC OFFSET IN PDVTB::
     000022  000000                          .WORD   0       ; (I) ; DMC LINE TABLE ADDRESS
     000024  000000                          .WORD   0       ; (I) ; DMC LINE TABLE BIAS
     000026  000000                          .WORD   0       ; (I) ; DMC LINE TABLE ( VIRTUAL ) ADDRESS
     000030     000                          .BYTE   0       ; (D) ; CONTROLLER NUMBER ( DMC-0-0 )
     000031     000                          .BYTE   0       ; (D) ; MULTIPLEXER UNIT NUMBER
     000032     000                          .BYTE   0       ; (A) ; NUMBER OF TRIBUTARIES ON THIS LINE
     000033     001                          .BYTE   1       ; (D) ; LINE COST FOR ROUTING TASK
     000034  000000                          .WORD   0       ; (D) ; POINTER TO CONTROLLER REQUEST BLOCK
     000036  000000                          .WORD   0       ; (D) ; NETWORK MANAGEMENT LINE STATE
             000000                          .REPT   0
                                             .BYTE   0       ; (D) ; MULTIPOINT TRIBUTARY FLAGS
                                             .BYTE   1       ; (D) ; TRIBUTARY ROUTING COST
                                             .WORD   0       ; (D) ; NETWORK MANAGEMENT TRIBUTARY STATE
                                             .ENDR
     000040                          $SLTND::
 54                                  ;
                                     ;       SYSTEM LINE TABLE ( EXTENSION )
                                     ;
                                     ;       THIS TABLE IS INDEXED BY SYSTEM LINE NUMBER AND
                                     ;          ASSOCIATES EACH SYSTEM LINE WITH AN LLC PROCESS
                                     ;          AND A CHANNEL NUMBER WITHIN THAT PROCESS.

     000040                          $LLCTB::
     000040      001      006                .BYTE   1,6             ; (A) ; SLNO: CHANNEL,LLC
 55                                  ;
                                     ;       NO TRIBUTARIES IN THIS CONFIGURATION
                                     ;
 56                                  ;
                                     ;       PROCESS DESCRIPTOR VECTOR OFFSETS AND BIT DEFINITIONS
                                     ;

     000000                          Z.DSP:  .BLKW   1       ; RELOCATION BIAS OF DISPATCH TABLE
     000002                                  .BLKW   1       ; VIRTUAL ADDRESS OF DISPATCH TABLE

     000004                          Z.NAM:  .BLKW   1       ; PROCESS NAME (RAD50)
     000006                          Z.LLN:  .BLKB   1       ; # OF LOGICAL LINES (LLC'S ONLY)
     000007                          Z.SCH:  .BLKB   1       ; PROCESS PRIORITY (COMPLEMENT)
     000010                          Z.FLG:  .BLKW   1       ; FLAGS WORD
     000012                          Z.PCB:  .BLKW   1       ; PCB OF LOADED PROCESS PARTITION
     000014                          Z.AVL:  .BLKW   1       ; POINTER TO FIRST FREE BLOCK IN PROCESS SPACE
             000016                  Z.LEN=  .-Z.DSP         ; LENGTH OF PDV
     000016                          Z.DAT:  .BLKW   1       ; BIAS OF LLC DATA BASE
     000020                          Z.MAP:                  ; DISPLACEMENT OF LLC MAPPING TABLE
                                     ;
                                     ; FLAGS WORD BIT DEFINTIONS
                                     ;
             000000                  ZF.X3P=  0              ; PROCESS REQUIRES PARAMETERS FROM X3P$DF MACRO
             000001                  ZF.DDM=  1              ; PROCESS IS A DDM
             000002                  ZF.DLC=  2              ; PROCESS IS A DLC
             000004                  ZF.LLC=  4              ; PROCESS IS A LLC
             000010                  ZF.MFL=  10             ; PROCESS IS MARKED FOR LOAD ( WITH NS )
             000020                  ZF.KMX=  20             ; PROCESS IS A DDM FOR A KMC DEVICE
```

Example 5 CETAB.LST (Sheet 3 of 7)

```
CETAB   MACRO V05.00  Wednesday 14-Dec-83 10:43  Page 3-3

          000040                   ZF.MUX=  40             ; PROCESS IS A DDM FOR A MULTIPLEXER DEVICE
          000100                   ZF.LMC= 100             ; PROCESS REQUIRES MICRO-CODE TO BE LOADED
          000200                   ZF.TIM= 200             ; PROCESS REQUESTS TIMER SUPPORT
          000400                   ZF.MTM= 400             ; PROCESS REQUIRES TIMER ENTRY ON EVERY ACTIVE PROCESSOR
                                                           ; (RSX-11M+ MULTI-PROCESSOR SYSTEMS ONLY)
          001000                   ZF.COU=1000             ; LLC/DLC PROCESS SUPPORTS COUNTERS
          002000                   ZF.PSE=2000             ; PROCESS IS A PSEUDO DDM/DLC
          004000                   ZF.DIA=4000             ; PROCESS REQUIRES DIAGNOSTICS MICRO-CODE LOAD
          010000                   ZF.SLI=10000            ; PROCESS SUPPORTS SYSTEM LEVEL INTERFACE
          020000                   ZF.MAN=20000            ; PROCESS SUPPORTS NETWORK MANAGEMENT REQUESTS
          040000                   ZF.INI=40000            ; PROCESS SUPPORTS INITIALIZATION AND TERMINATE
          100000                   ZF.DVP=100000    ; PROCESS ALWAYS RUNS AT IT'S PRIORITY
                                   ;
                                   ; CHANNEL TABLE FLAGS
                                   ;
          100000                   ZS.ASN=100000    ; CHANNEL IS FREE TO ASSIGN
          140000                   ZS.BSY=140000    ; CHANNEL IS IN PROCESS OF ASSIGNMENT

                                   ;
                                   ;       AUX - PROCESS DESCRIPTOR VECTOR # 0
                                   ;
                                   ;               THIS IS AN LLC PROCESS
                                   ;
000042                             AUXPDV::
000042  000000                             .WORD   0               ; (I) ; RELOCATION BIAS
000044  000000                             .WORD   0               ; (I) ; DISPATCH TABLE ADDRESS
000046  004640                             .RAD50  /AUX/           ; (D) ; PROCESS NAME
000050     000                             .BYTE   0               ; (R) ; AVAILABLE BYTE
000051     340                             .BYTE   ^C<PRO>&PR7         ; (A) ; PRIORITY 0
000052  000614                             .WORD   614             ; (D) ; FLAG WORD
000054  000000                             .WORD   0               ; (I) ; LOADED PROCESS PCB POINTER
000056  000000                             .WORD   0               ; (R) ; PROCESS FREE SPACE POINTER
000060  000000                             .WORD   0               ; (I) ; VIRTUAL ADDRESS OF LLC DATA BASE
                                   ;
                                   ;       EVL - PROCESS DESCRIPTOR VECTOR # 1
                                   ;
                                   ;               THIS IS AN LLC PROCESS
                                   ;
000062                             EVLPDV::
000062  000000                             .WORD   0               ; (I) ; RELOCATION BIAS
000064  000000                             .WORD   0               ; (I) ; DISPATCH TABLE ADDRESS
000066  021274                             .RAD50  /EVL/           ; (D) ; PROCESS NAME
000070     000                             .BYTE   0               ; (R) ; AVAILABLE BYTE
000071     340                             .BYTE   ^C<PRO>&PR7         ; (A) ; PRIORITY 0
000072  000214                             .WORD   214             ; (D) ; FLAG WORD
000074  000000                             .WORD   0               ; (I) ; LOADED PROCESS PCB POINTER
000076  000000                             .WORD   0               ; (R) ; PROCESS FREE SPACE POINTER
000100  000000                             .WORD   0               ; (I) ; VIRTUAL ADDRESS OF LLC DATA BASE
                                   ;
                                   ;       ECL - PROCESS DESCRIPTOR VECTOR # 2
                                   ;
                                   ;               THIS IS AN LLC PROCESS
                                   ;
000102                             ECLPDV::
000102  000000                             .WORD   0               ; (I) ; RELOCATION BIAS
000104  000000                             .WORD   0               ; (I) ; DISPATCH TABLE ADDRESS
000106  017704                             .RAD50  /ECL/           ; (D) ; PROCESS NAME
```

Example 5 CETAB.LST (Sheet 4 of 7)

```
CETAB   MACRO V05.00  Wednesday 14-Dec-83 10:43  Page 3-4

     000110      000                      .BYTE   0                ; (R) ; AVAILABLE BYTE
     000111      340                      .BYTE   ^C<PRO>$PR7             ; (A) ; PRIORITY 0
     000112   001214                      .WORD   1214             ; (D) ; FLAG WORD
     000114   000000                      .WORD   0                ; (I) ; LOADED PROCESS PCB POINTER
     000116   000000                      .WORD   0                ; (R) ; PROCESS FREE SPACE POINTER
     000120   000000                      .WORD   0                ; (I) ; VIRTUAL ADDRESS OF LLC DATA BASE
                                  ;
                                  ;       XPT - PROCESS DESCRIPTOR VECTOR # 3
                                  ;
                                  ;               THIS IS AN LLC PROCESS
                                  ;
     000122                       XPTPDV::
     000122   000000                      .WORD   0                ; (I) ; RELOCATION BIAS
     000124   000000                      .WORD   0                ; (I) ; DISPATCH TABLE ADDRESS
     000126   114224                      .RAD50  /XPT/            ; (D) ; PROCESS NAME
     000130      002                      .BYTE   2.               ; (A) ; NUMBER OF CHANNELS ALLOCATED FOR XPT
     000131      340                      .BYTE   ^C<PRO>$PR7             ; (A) ; PRIORITY 0
     000132   001214                      .WORD   1214             ; (D) ; FLAG WORD
     000134   000000                      .WORD   0                ; (I) ; LOADED PROCESS PCB POINTER
     000136   000000                      .WORD   0                ; (R) ; PROCESS FREE SPACE POINTER
     000140   000000                      .WORD   0                ; (I) ; VIRTUAL ADDRESS OF LLC DATA BASE
     000142      000     300              .BYTE   0,300            ; (A) ; CHANNEL 0: RESERVED FOR LOOPBACK
     000144      000     000              .BYTE   0,0              ; (A) ; CHANNEL 1: SLNO,TRIBUTARY 0
                                  ;
                                  ;       DLX - PROCESS DESCRIPTOR VECTOR # 4
                                  ;
                                  ;               THIS IS AN LLC PROCESS
                                  ;
     000146                       DLXPDV::
     000146   000000                      .WORD   0                ; (I) ; RELOCATION BIAS
     000150   000000                      .WORD   0                ; (I) ; DISPATCH TABLE ADDRESS
     000152   015370                      .RAD50  /DLX/            ; (D) ; PROCESS NAME
     000154      002                      .BYTE   2.               ; (A) ; NUMBER OF CHANNELS ALLOCATED FOR DLX
     000155      340                      .BYTE   ^C<PRO>$PR7             ; (A) ; PRIORITY 0
     000156   000214                      .WORD   214              ; (D) ; FLAG WORD
     000160   000000                      .WORD   0                ; (I) ; LOADED PROCESS PCB POINTER
     000162   000000                      .WORD   0                ; (R) ; PROCESS FREE SPACE POINTER
     000164   000000                      .WORD   0                ; (I) ; VIRTUAL ADDRESS OF LLC DATA BASE
     000166      000     300              .BYTE   0,300            ; (A) ; CHANNEL 0: RESERVED FOR LOOPBACK
     000170      000     200              .BYTE   0,200            ; (A) ; CHANNEL 1: UNASSIGNED
                                  ;
                                  ;       DMC - PROCESS DESCRIPTOR VECTOR # 5
                                  ;               THIS IS A DDM PROCESS
                                  ;
     000172                       DMCPDV::
     000172   000000                      .WORD   0                ; (I) ; RELOCATION BIAS
     000174   000000                      .WORD   0                ; (I) ; DISPATCH TABLE ADDRESS
     000176   015413                      .RAD50  /DMC/            ; (D) ; PROCESS NAME
     000200      000                      .BYTE   0                ; (R) ; AVAILABLE BYTE
     000201      100                      .BYTE   ^C<PR5>$PR7             ; (A) ; PRIORITY 5
     000202   021003                      .WORD   21003            ; (D) ; FLAG WORD
     000204   000000                      .WORD   0                ; (I) ; LOADED PROCESS PCB POINTER
     000206   000000                      .WORD   0                ; (R) ; PROCESS FREE SPACE POINTER
              000001                      .END
```

Example 5 CETAB.LST (Sheet 5 of 7)

```
CETAB   MACRO V05.00  Wednesday 14-Dec-83 10:43  Page 3-5
Symbol table

AUX    = 000000        ECLNXT= 000001        LN.STA= 000017        P$$LAS= 000000        SOCN   = 000000
AUXCHN= 000000         ECLPDV  000102RG   002 LN.SUB= 000360       P$$LOL= 001130        SODD   = 000005
AUXFL  = 000614        ECLPR  = 000000       LN.TRI= 000006        P$$OFF= 000000        SODL   = 000005
AUXNXT= 000001         EVL    = 000001       L$$ASG= 000000        P$$P45= 000000        SOLL   = 000003
AUXPDV  000042RG    002 EVLCHN= 000000       L$$DRV= 000000        P$$RFL= 000000        SOPD   = 000006
AUXPR  = 000000        EVLFL  = 000214       L$$LDR= 000000        P$$RTY= 000000        SOSF   = 006010
A$$BIO= 000000         EVLNXT= 000001        L$$PTO= 000036        P$$SRF= 000000        SOTY   = 000001
A$$BRT= 000000         EVLPDV  000062RG   002 L.COST  000015       Q$$MGR= 000000        SOUN   = 000000
A$$CHK= 000000         EVLPR  = 000000       L.CTL   000012        Q$$OPT= 000005        TR     = 000002
A$$CLI= 000005         EXITUN= 000001        L.DDM   000002        RDB.NB= 000011        T$$ACR= 000000
A$$CPS= 000000         E$$DVC= 000000        L.DDS   000004        RDB.SZ= 001100        T$$BTW= 000000
A$$NSI= 000000         E$$LOG= 000000        L.DLC   000003        R$$EXV= 000000        T$$BUF= 000000
A$$PRI= 000000         E$$XPR= 000000        L.DLM   000006        R$$LKL= 000001        T$$CCA= 000000
A$$TRP= 000000         F$$LPP= 000000        L.DLS   000010        R$$L11= 000001        T$$CCO= 000000
B$$LV1= 032463         F$$LVL= 000001        L.FLG   000000        R$$NDC= 000005        T$$CPW= 000000
B$$LV2= 020040         G$$EFN= 000000        L.KRBA  000016        R$$NDH= 000226        T$$CTR= 000000
CCB.NB= 000010         G$$TPP= 000000        L.LEN  = 000022       R$$NDL= 000001        T$$CUP= 000000
CCB.SZ= 000046         G$$TSS= 000000        L.MPF   000022        R$$POI= 000000        T$$ESC= 000000
CHAN   = 000001        G$$TTK= 000000        L.NMST  000020        R$$SND= 000000        T$$GMC= 000000
CNT    = 000031        H$$RTZ= 000074        L.NSTA  000014        R$$TPR= 032063        T$$GTS= 000000
CNTLR  = 000000        I$$RAR= 000000        L.OWNR  000021        R$$11M= 000000        T$$HFF= 000000
CTBNUM= 000000         I$$RDN= 000000        L.UNT   000013        SDB.NB= 000013        T$$HLD= 000000
C$$CDA= 000007         K$$AST= 000000        MASTER= 177777        SDB.SZ= 000042        T$$KMG= 000000
C$$CKP= 000004         K$$CNT= 177546        M$$CRB= 000124        SF.ACT= 000200        T$$LWC= 000000
C$$CSR= 174400         K$$CSR= 177546        M$$CRX= 000000        SF.ENA= 000100        T$$RED= 000000
C$$INT= 000000         K$$IEN= 000115        M$$EIS= 000000        SF.LPB= 000004        T$$RNE= 000000
C$$ONS= 000001         K$$LDC= 000001        M$$FCS= 000000        SF.MFL= 000040        T$$RPR= 000000
C$$ORE= 002022         K$$TPS= 000074        M$$MGE= 000000        SF.PAC= 000020        T$$RST= 000000
C$$RSH= 177564         LD$CO  = 000000       M$$MUP= 000000        SF.REA= 000010        T$$RUB= 000000
C$$RUN= 000001         LD$DL  = 000000       M$$NET= 000000        SF.SER= 000001        T$$SMC= 000000
C$$SMT= 000000         LD$TT  = 000000       M$$OVR= 000000        SF.SVC= 000002        T$$SYN= 000000
C$$TTY= 177564         LF.ACT= 100000        NONE   = 177777       SF.UNL= 000040        T$$TRW= 000000
DLX    = 000004        LF.BRO= 000400        N$$LDV= 000001        SLAVE  = 000000       T$$UTB= 000000
DLXCHN= 000001         LF.BWT= 000007        N$$MOV= 000041        SLINE  = 000001       T$$UTO= 000170
DLXFL  = 000214        LF.ENA= 002000        PDN    = 000005       SLN    = 000000       T$$VBF= 000000
DLXNXT= 000001         LF.LPB= 001000        PDVNM  = 000010       SLTTOT= 000001 G      T$$30P= 000000
DLXPDV  000146RG    002 LF.MDC= 000100       PDVTOT= 000006 G      SLTO    000016R   002 UNT    = 000000
DLXPR  = 000000        LF.MFL= 004000        PD$AUX= 000000 G      SSLO   = 000000       V$$CTR= 000604
DMC    = 000005        LF.MTP= 000020        PD$DLX= 000010 G      STATBL= 000000        V$$RSN= 000041
DMCFL  = 021003        LF.PAC= 000200        PD$DMC= 000012 G      STATN  = 000000       XPT    = 000003
DMCMXC= 000000         LF.RDY= 040000        PD$ECL= 000004 G      S$$HFC= 000036        XPTCHN= 000001
DMCPDV  000172RG    002 LF.REA= 010000       PD$EVL= 000002 G      S$$NM1= 044532        XPTFL  = 001214
DMCPR  = 000005        LF.SER= 000040        PD$XPT= 000006 G      S$$NM2= 041522        XPTNXT= 000002
DMCOA  = 000001        LF.TIM= 000010        POINT  = 000001       S$$NM3= 047117        XPTPDV  000122RG    002
D$$IAG= 000000         LF.UNL= 020000        PR0    = 000000       S$$TIM= 000000        XPTPR  = 000000
D$$ISK= 000000         LF.X2P= 000000        PR1    = 000040       S$$TOP= 000000        ZF.COU= 001000
D$$L11= 000002         LN.CLO= 000000        PR2    = 000100       S$$WLK= 000000        ZF.DDM= 000001
D$$PAR= 000000         LN.DUM= 000005        PR3    = 000140       S$$WPC= 000036        ZF.DIA= 004000
D$$SHF= 000000         LN.LOA= 000004        PR4    = 000200       S$$WPR= 000005        ZF.DLC= 000002
D$$WCK= 000000         LN.LOO= 000003        PR5    = 000240       S$$WST= 000000        ZF.DUP= 100000
D$$YNC= 000000         LN.OAU= 000003        PR6    = 000300       S$$YSZ= 007600        ZF.INI= 040000
D$$YNM= 000000         LN.OFF= 000001        PR7    = 000340       S.COST  000001        ZF.KMX= 000020
D$$ZMD= 000000         LN.ON  = 000000       P$$BPR= 000063        S.FLG   000000        ZF.LLC= 000004
D$$Z11= 000001         LN.OOP= 000004        P$$CTL= 000000        S.LEN   000004        ZF.LMC= 000100
ECL    = 000002        LN.OPE= 000001        P$$FRS= 000310        S.NMST  000002        ZF.MAN= 020000
ECLCHN= 000000         LN.REF= 000002        P$$GMX= 000000        S.OWNR  000003        ZF.MFL= 000010
ECLFL  = 001214        LN.SER= 000002        P$$HIL= 003100        SOCH   = 000001       ZF.MTM= 000400
```

Example 5 CETAB.LST (Sheet 6 of 7)

```
CETAB   MACRO V05.00  Wednesday 14-Dec-83 10:43  Page 3-6
Symbol table

ZF.MUX= 000040          ZS.ASN= 100000          Z.FLG    000010          Z.PCB    000012          $PDVND  000014RG     002
ZF.PSE= 002000          ZS.BSY= 140000          Z.LEN  = 000016          Z.SCH    000007          $PDVTB  000000RG     002
ZF.SLI= 010000          Z.AVL   000014          Z.LLN    000006          $DS1   = 000000          $SLTMB  000014RG     002
ZF.TIM= 000200          Z.DAT   000016          Z.MAP    000020          $DT1   = 000000          $SLTND  000040RG     002
ZF.X3P= 000000          Z.DSP   000000          Z.NAM    000004          $LLCTB   000040RG    002 $SLTTB  000016RG     002

. ABS.  000022    000   (RW,I,GBL,ABS,OVR)
        000000    001   (RW,I,LCL,REL,CON)
CEXCOM  000210    002   (RW,I,LCL,REL,CON)
Errors detected:  0

*** Assembler statistics


Work  file  reads: 26
Work  file writes: 31
Size of work file: 20695 Words  ( 81 Pages)
Size of core pool: 20466 Words  ( 78 Pages)
Operating  system: RSX-11M/PLUS

Elapsed time: 00:01:14.01
OU:[300,24]CETAB,LS:[300,34]CETAB.LST/-SP=IN:[130,10]NETLIB/ML,OU:[300,010]RSXMC,[300,54]CETAB
```

Example 5 CETAB.LST (Sheet 7 of 7)

```
>
>
>PAR
CEXPAR 111734 112000 006000 MAIN COM
EXCOM1 111670 120000 014700 MAIN COM
EXCOM2 111624 134700 010200 MAIN COM
LDRPAR 111560 145100 002600 MAIN TASK
TTPAR  111214 147700 040000 MAIN TASK
DRVPAR 110670 207700 002300 MAIN SYS
       110624 207700 002100 SUB  DRIVER -DL:
       110524 212000 000200 SUB  DRIVER -CO:
SYSPAR 110460 212200 010100 MAIN TASK
FCSRES 110414 222300 032000 MAIN COM
FCPPAR 110350 254300 024200 MAIN SYS
       034374 254300 024200 SUB  (F11ACP)
GEN    110304 300500 457300 MAIN SYS
       034264 300500 006300 SUB  (PMT...)
       044400 321700 023200 SUB  (NETACP)
       035000 345100 020000 SUB  (...MCR)
       034454 407000 027200 SUB  (RMDEMO)
NT.DMC 046634 663700 003400 SUB  DYNAMIC
NT.DLX 046524 667300 011100 SUB  DYNAMIC
NT.XPT 046570 700400 010500 SUB  DYNAMIC
NT.ECL 036734 711100 016000 SUB  DYNAMIC
NT.EVL 045020 727100 003000 SUB  DYNAMIC
NT.AUX 045314 732100 002700 SUB  DYNAMIC
POOL.. 041744 735000 020200 SUB  DYNAMIC
       035150 755200 001700 SUB  DRIVER -HT:
       040330 757100 000700 SUB  DRIVER -NM:
>
```

Example 6 Partitions

```
 T\T\
>
>
>TAS
LDR... 13.02  LDRPAR 248. 002600 LB0:-00012506 FIXED
TKTN   05.00  SYSPAR 248. 010100 LB0:-00016376
...RMD 2.00   GEN    225. 027200 LB0:-00017037
RMDEMO 2.00   GEN    225. 027200 LB0:-00017037
F11MSG 13.00  GEN    200. 005700 LB0:-00015422
MTAACP 14.00  GEN    200. 014700 LB0:-00016711
NETACP V04.00 GEN    200. 023200 DL0:-00004161
EVC... V04.00 GEN    198. 027600 DL0:-00022636 CHECKPOINTED
EVR$$$ V04.00 GEN    197. 027500 DL0:-00011221
...DMO 04.00  GEN    160. 014600 LB0:-00014446
MCR... 4.00   SYSPAR 160. 010100 LB0:-00015264
...DCL 2.0    GEN    160. 040000 LB0:-00015671
...MOU 26.00  GEN    160. 037700 LB0:-00015310
...MCR 2.01   GEN    160. 020000 LB0:-00015505
NVP... V04.00 GEN    150. 012700 DL0:-00011302
F11ACP 05.00  FCPPAR 149. 024200 LB0:-00014542
ERRLOG 2.00   GEN    148. 040000 LB0:-00017506
PMT... 2.00   GEN    148. 006300 LB0:-00015242
COT... 2.0    GEN    145. 013300 LB0:-00014465
PMD... 08.00  GEN    140. 015700 LB0:-00016661
SHF... 6.00   SYSPAR 105. 010100 LB0:-00017176
...INS 8.00   GEN    100. 034500 LB0:-00014750
...SAV 04.00  GEN    100. 033100 LB0:-00016550
...UFD 05.00  GEN    100. 005700 LB0:-00015470
...BAD 06.00  GEN    100. 070000 LB0:-00017573
...SEN 12AUG  GEN    100. 104500 SY0:-00035451
MAL$$$ 11RX   GEN    100. 132300 SY0:-00036333
NMVACP V04.00 GEN    100. 027200 DL0:-00025673
LIN$$$ V04.00 GEN    100. 010300 DL0:-00011367
LOO$$$ V04.00 GEN    100. 022300 DL0:-00011170
FAL... V05.00 GEN    100. 143600 DL0:-00011500
RMHACP V04.00 GEN    100. 023500 DL0:-00011724
LSN$$$ V04.00 GEN    100. 020300 DL0:-00013473
RMTACP V04.00 GEN     99. 027700 DL0:-00011653
QMG... 03.00  GEN     75. 031400 LB0:-00015366
PRT... 2.0    GEN     70. 001100 LB0:-00012723
LPO    03.00  GEN     70. 014200 LB0:-00016447
...ACS 3.00   GEN     70. 004500 LB0:-00017477
...BRU 5.31   GEN     70. 175700 LB0:-00027377
...EDT 02.00  GEN     65. 130700 LB0:-00032531
...MAI 24FEB  GEN     65. 137300 SY0:-00035671
...AT. 6.0    GEN     64. 060000 LB0:-00015076
NTL... V04.00 GEN     60. 037500 DL0:-00026152
NTINIT V04.00 GEN     55. 006600 DL0:-00002374
...QUE 03.00  GEN     50. 020100 LB0:-00016277
...PRI 03.00  GEN     50. 020100 LB0:-00016277
...BOO 06.02  GEN     50. 021700 LB0:-00012662
...ELI 1.00   GEN     50. 017000 LB0:-00017552
...MAG 02.00  GEN     50. 031500 LB0:-00015434
...LOA 4.0    GEN     50. 032300 LB0:-00016406
...HEL 2.00   GEN     50. 023500 LB0:-00017355
...BYE 05.00  GEN     50. 012600 LB0:-00017314
...BRO 05.00  GEN     50. 030300 LB0:-00017420
...UNL 4.0    GEN     50. 024500 LB0:-00016632
...PIP 16.00  GEN     50. 040000 LB0:-00004732
...MAC V05.00 GEN     50. 070000 LB0:-00004356
...TKB M40.10 GEN     50. 070000 LB0:-00005123
MAL... 30OCT  GEN     50. 057700 SY0:-00036126
...NCP V04.00 GEN     50. 064400 DL0:-00025237
NICE$$ V04.00 GEN     50. 007600 DL0:-00004207
...NTD V03.00 GEN     50. 037700 DL0:-00010760
```

Example 7 Tasks (Sheet 1 of 2)

```
NTD...  V03.00 GEN      50. 030200 DL0:-00011022
MIR$$$  V04.00 GEN      50. 001600 DL0:-00011215
...NFT  V04.00 GEN      50. 063000 DL0:-00013662
.CMTS.  V04.00 GEN      50. 002100 DL0:-00011353
...RMT  V04.00 GEN      50. 002100 DL0:-00011646
...TLK  V04.00 GEN      50. 042700 DL0:-00013534
TCL...  V04.00 GEN      50. 002500 DL0:-00011765
...LBR  07.00  GEN      50. 040000 LB0:-00004246
```

Example 7 Tasks (Sheet 2 of 2)

```
>
>
>DEV
DL0: Public Mounted Loaded Label=RSXM35 Type=RL02
DL1: Loaded Type=RL02
CO0:    TT0:
HT0: Offline Loaded
HT1: Offline Loaded
HT2: Offline Loaded
HT3: Offline Loaded
NM0: Loaded
NS0:
NX0: Public Loaded
TT0: [5,1] - Logged in  Loaded
TT1: Loaded
TT2: Loaded
TT3: Loaded
TT4: Loaded
TT5: Loaded
TT6: Loaded
TT7: Loaded
TT10: Loaded
TT11: Loaded
NL0:
TI0:
CL0:    TT0:
LB0:    DL0:
SY0:    DL0:
>
```

Example 8 Devices

EXERCISES

Match the network components to the appropriate functions. Each item may be used once, more than once, or not at all.

1. CEX
2. NCP
3. KMCL
4. NSDRV
5. EVL
6. RCP
7. NTL
8. PLI
9. NETACP
1Ø. DLX
11. LAB
12. NTINIT
12. EVC
14. DLM

a. ____ Responsible for loading the network software into core.

b. ____ Provides direct line access interface.

c. ____ Updates the Routing database on routing nodes.

d. ____ Manages the DECnet software.

e. ____ Receives the network events from remote nodes.

f. ____ User interface for the logical link.

g. ____ User interface for the network management.

h. ____ DDCMP software process.

i. ____ Interface between the Routing and PLI.

j. ____ X.25 Level 3 protocol software.

k. ____ X.25 Level 2 protocol software.

SOLUTIONS

Match the network components to the appropriate functions. Each item may be used once, more than once, or not at all.

1. CEX
2. NCP
3. KMCL
4. NSDRV
5. EVL
6. RCP
7. NTL
8. PLI
9. NETACP
10. DLX
11. LAB
12. NTINIT
12. EVC
14. DLM

a. 7 Responsible for loading the network software into core.

b. 10 Provides direct line access interface.

c. 6 Updates the Routing database on routing nodes.

d. 1 Manages the DECnet software.

e. 13 Receives the network events from remote nodes.

f. 4 User interface for the logical link.

g. 2 User interface for the network management.

h. none DDCMP software process.

i. 14 interface between the Routing and PLI.

j. 8 X.25 Level 3 protocol software.

k. 11 X.25 Level 2 protocol software.

LAB EXERCISES

1. Given a network configuration, perform NETGEN on a designated system.

2. Run the NTEST procedure for:

   - Local node
   - Remote node
   - Loopback node

SOLUTIONS TO LAB EXERCISES

1. See NETGEN Example 2

2. See NTEST Example 4

CONTENTS

FIGURES

TABLES

EXAMPLES

INTRODUCTION

This module covers the following areas:

- Down-line system loading/up-line dumping
- Down-line task loading/up-line checkpointing
- Multipoint
- Ethernet

These facilities are provided in the communication hardware (down-line system loading/up-line dumping, multipoint, and Ethernet) as well as in the software. Both software and hardware arrangements are discussed in detail.

To down-line system load we must assume that DECnet is running on the host computer, and there is no operating system running on the satellite. To achieve a successful completion of the loading of the O/S we utilize the Maintenance Operations Protocol (MOP). The MOP protocol, which is a subset of DDCMP, allows the system satellite to be loaded with the O/S and also allows up-line dumping of memory in the event of system failure.

On the other hand, down-line task loading/up-line checkpointing assures that the satellite as well as the host has DECnet software running. This facility is implemented as a logical link between the Host Loader (HLD) and the Satellite Loader (SLD) tasks.

Multipoint allows more than two computers to share a physical line. Polling or time-sharing algorithms are used to identify the node that controls the line at a given time.

Ethernet is a local area network specification developed by Xerox Corporation, Digital Equipment Corporation, and Intel Corporation to provide high-speed exchange of data within a moderate sized geographic area.

OBJECTIVES

Upon completion of this module, the specialist should be able to:

- List the down-line system loading considerations for a host and a satellite.
- Identify the requirements for performing down-line task loading/up-line characteristics.
- Create/update DLL and HLD databases using standard installation procedures.
- List the hardware requirements for down-line system loading (multipoint and Ethernet).
- Explain the basics of polling algorithm and set the corresponding parameters using NCP commands.
- Demonstrate down-line system loading using generated systems.
- Demonstrate down-line task loading using generated systems.

RESOURCES

1. DNA: MOP Protocol Functional Specification
2. DECnet-RSX System Manager's Guide
3. DECnet-RSX Network Installation Guide
4. M9301-YJ Bootstrap Technical Manual
5. M9312 Technical Manual
6. DIGITAL Ethernet Products and Services

SYSTEM DOWN-LINE LOAD

An RSX-11S system image can be down-line loaded. Before attempting to down-line load, you must ensure that nodes, lines, and circuits meet the following requirements:

- The target node must be connected directly to the executor node. The executor node provides the line and circuit-level access.
- The primary loader must be a cooperating program either in the target or in the microcode of the target's device (DMC,DMR,DMP,DMV or UNA). The down-line load operation usually involves loading a series of bootstraps; each requests the next program until the operating system itself is loaded.
- To allow the tertiary loader to complete the load sequence, the RSX-11S system image itself must be 2K words less than the total memory on the target system.
- The executor must have access to the load files. The location of the files either can be specified in the load request or can be a default in the volatile database.
- The target node must be able to recognize the trigger message or must be triggered manually.
- The circuit involved in the load operation must be enabled to perform service functions. It must also be in the ON or SERVICE state.

Table 1 System Down-Line Loading

Host		Satellite
Enter MOP Mode - Only if a communications device on the satellite can be triggered (DMC/DMR/DMP/DMV/UNA/QNA)	→	
		Request program - All communication devices (secondary boot)
Memory loaded with XFER ADDR (entire secondary boot)	←	
	→	Request program (tertiary boot)
Memory loaded without XFER ADDR (first segment of tertiary boot)	←	
.		Request memory load (next segment)
.		.
.		.
Memory load with XFER ADDR (Last segment of tertiary boot)	→	.
		Request program (operating system)
	←	
Memory load without XFER ADDR (first segment of operating system)	→	
.		Request memory load
.	←	(next segment)
.		.
Parameter load with XFER ADDR	→	.
		.
	←	Request memory load (dummy/ACK)
		-Load Complete-

Down-Line Loading on DECnet-RSX

Table 2 Target Initiated Load on DECnet

Host		Target
	BOOT	1. The primary bootstrap is triggered in one of the many possible ways (ROM or loaded into core via tapes, etc.). The MOP 8 message will be sent.
	←	(MOP 8, request secondary boot)
2. Circuit could be in either ON or SERVICE state. XPT detects the message and activates LIN$$$. LIN$$$ gives circuit to DLX interface and calls DLL$$$. By now the MOP 8 message is lost. LIN$$$ sends the MOP 12 message (MOP mode running).		
(MOP 12, MOP mode running)	→	
		3. The target node treats this as a bad message and retransmits the MOP 8 message.
	←	(MOP 8, request secondary boot)
4. This time DLL$$$ is ready to process the MOP 8 message. It uses its database to identify secondary loader for that particular circuit and sends it to the target.		
(MOP Ø, send entire secondary boot)	→	
		5. The secondary boot assumes unmapped system and requests tertiary boot.
	←	(MOP 8, request tertiary boot)

Table 2 Target Initiated Load on DECnet (Cont)

Host		Target
6. DLL$$$ sends the tertiary boot.		
(MOP 2 and MOP Ø, more than one transfer)	→	
		7. The tertiary boot turns on memory management, locates itself to top of core, and requests the operating system.
	←	(MOP 8, request operating system)
8. DLL$$$ sends the operating system. In reality, the tertiary boot is 2K and relocates itself to top of memory. Thus only 122KW of a 124KW system image can be loaded.		
(MOP 2 and MOP Ø, transfer RSX-11S)	→	
9. When load is completed, DLL$$$ records event to EVL; LIN$$$ reassigns the circuit back to XPT interface if previously owned by XPT. Procedure is completed.		

NOTE

If the target is attached by using an Ethernet cable to the host node, the load request passes through the DDM to the Ethernet Protocol Manager (EPM) to LIN$$$. LIN$$$ then gives the circuit to DLX and calls DLL$$$.

Load Initiated by the NCP TRIGGER Command

NCP activates DLL$$$, that sends a MOP mode message (MOP 6) to the DMC/DMR/DMP/DMV/UNA/QNA on the other side. The DMC/DMR/DMP/DMV/UNA/QNA responds by activating the bootstrap ROM and sends the MOP 8 message to request the secondary boot. From this point on, the sequence of events is the same as if the target system initiated the sequence.

Use the TRIGGER command when the target node is an unattended system equipped with an appropriate bootstrap ROM. The executor for a triggered down-line load is not necessarily the same as the node that executed the TRIGGER command. This command triggers the bootstrap mechanism of a node so that it loads itself.

Load Initiated by the NCP LOAD Command

NCP activates DLL$$$, which first picks up information from the volatile database. If parameters have been specified in the LOAD command, these parameters override those in the permanent database. The file specification in the "software type" is sent first. (The default is secondary boot; if successful, continue.) If this fails, the MOP 6 (enter MOP mode) message is sent as in the trigger case. From this point on, the sequence of events is the same as the NCP TRIGGER command.

Use the LOAD command when the target's primary bootstrap ROM is already running. The node that issues a LOAD command is always the executor for the requested down-line load. (Additional details on down-line system load hardware arrangements are provided in Appendix A.)

Host Considerations (for Down-Line System Load)

Both the permanent and the volatile database can include default parameters for down-line loading.

- Use the CFE DEFINE NODE command to set up the permanent database, that is loaded into the volatile database when the system is loaded.
- Use the NCP SET NODE command to establish default information for the target node in the volatile database.

The NCP LOAD and TRIGGER commands default to entries in the volatile database when you omit down-line loading parameters. Any parameter you specify explicitly in a LOAD or TRIGGER command overrides the default. These default parameters are also used for target-initiated down-line loads.

```
>NCP
NCP> SET NODE 258 NAME REMSYS
NCP> SET NODE REMSYS SERVICE CIRCUIT UNA-0
NCP> SET NODE REMSYS SERVICE DEVICE UNA
NCP> SET NODE REMSYS SERVICE VERSION 4
NCP> SET NODE REMSYS SERVICE PASSWORD 0000000000000000
NCP> SET NODE REMSYS HARDWARE ADDRESS AA00030100AB
NCP> SET NODE REMSYS LOAD FILE LB0:[2,54]RSX11S.SYS
NCP> SET NODE REMSYS SECONDARY LOADER LB0:[2,54]SECUNA.SYS
NCP> SET NODE REMSYS TERTIARY LOADER LB0:[2,54]TERUNA.SYS
```

Example 1 Creating a Down-Line System Loading Database (Using NCP)

```
>CFE
Enter filename: LB0:[300,54]CETAB.MAC
CFE> DEFINE NODE 258 NAME REMSYS
CFE> DEFINE NODE REMSYS SERVICE CIRCUIT UNA-0
CFE> DEFINE NODE REMSYS SERVICE DEVICE UNA
CFE> DEFINE NODE REMSYS SERVICE VERSION 4
CFE> DEFINE NODE REMSYS SERVICE PASSWORD 0000000000000000
CFE> DEFINE NODE REMSYS HARDWARE ADDRESS AA00030100AB
CFE> DEFINE NODE REMSYS LOAD FILE LB0:[2,54]RSX11S.SYS
CFE> DEFINE NODE REMSYS SECONDARY LOADER LB0:[2,54]SECUNA.SYS
CFE> DEFINE NODE REMSYS TERTIARY LOADER LB0:[2,54]TERUNA.SYS
```

Modify Permanent D.B

Example 2 Creating a Down-Line System Loading Database (Using CFE)

Table 3 Default Loader Files by Device Type

Device	Secondary Loader	Tertiary Loader
DLV	SECDLV.SYS	TERDLV.SYS
DL11	SECDL.SYS	TERDL.SYS
DMC11	SECDMC.SYS	TERDMC.SYS
DMP	SECDMP.SYS	TERDMP.SYS
DMV	SECDMV.SYS	TERDMV.SYS
DPV	SECDPV.SYS	TERDPV.SYS
DU11	SECDU.SYS	TERDU.SYS
DUV11	SECDUV.SYS	TERDUV.SYS
QNA	SECQNA.SYS	TERQNA.SYS
UNA	SECUNA.SYS	TERUNA.SYS

These files are shipped with the DECnet-11S kit.

RSX-11S System Down-Line Loading Considerations

In order to down-line load an RSX-11S node, you must create a system image file that is either a DECnet-11S node or a stand-alone RSX-11S system (a process control system, for example). To create a DECnet-11S node, use VMR and VNP to create the system image file.

```
>VMR
Enter filename:RSX11S
VMR> SET /POOL=260
VMR> SET /MAIN=MCRPAR:260:63:TASK
VMR> SET /MAIN=CEXPAR:343:41:COM
VMR> INS BASMCR/FIX=YES
VMR> INS [xxx,64]NETACP/CKP=NO/FIX=YES
VMR> INS [xxx,64]NTINIT/FIX=YES
VMR> <Ctrl Z>
>
>VNP
Enter filename:RSX11S
VNP> SET SYSTEM ALL
VNP> SET EXECUTOR STATE ON
VNP> SET LINE DMC-0 ALL
VNP> SET CIR DMC-0 STATE ON
VNP> EXIT
>
>
```

Example 3 Performing VMR and VNP on a RSX-11S System

TASK DOWN-LINE LOAD

Down-line task loading extends nonresident initial load, checkpointing, and overlay support to a DECnet-11S node. These functions are provided by the SLD on the DECnet-11S node and by the HLD on the host DECnet node. The host node in this context is determined by the HOST parameter when you down-line load the target. Neither SLD or HLD has an operator interface. Once the configuration is set up, they operate transparently.

Control is passed to SLD when an installed task has been requested to run (RUN command from a CLI or programmed request from another task). SLD then establishes intertask communication with HLD on the host system to transfer the required task image.

Table 4 Task Down-Line Load Sequence

Satellite	Host
1. Operator types: >RUN TLK	
2. SLD establishes a logical link to HLD.	
	3. HLD looks at the HLD database for task TLK entry. If found, pass it to SLD via logical link.
4. SLD loads task into core and starts it.	

Host Loader

- HLD communicates with SLD on DECnet-11S node to implement down-line task loading.
- Tasks to be down-line loaded to RSX-11S node must be installed in the RSX-11S node but not fixed.
- HLD has a user-defined database to store information about the tasks to be down-line loaded.
- The tasks to be down-line loaded may be general purpose, such as TLK in the previous example. That is, they may be down-line loaded to any node.
- Or the tasks may be node specific, such as a task named AB that is to be run on node XYZ:: only.
- For general purpose tasks to be down-line loaded into different RSX-11S systems (that may have different configurations), the LUN FIXING option must have been chosen at NETGEN of the RSX-11S SLD task. LUN FIXING reinitializes the LUNs after the task has been down-line loaded.

  LUN FIXING is optional for RSX-11S systems where only node specific tasks are down-line loaded.
- Checkpointing and overlaying of a task in the RSX-11S node is also implemented by using the logical link between HLD and SLD. Checkpoint space must be allocated inside the task being down-line loaded (by using the /AL switch during the task build). Only node-specific tasks can be checkpointed back to the host.

Example 4 is an example of generating a down-line task loading database, and Example 5 shows an HLD database.

```
>@LB:[300,1]HLDDAT.CMD
>;
>; ====================================================================
>;  HLDDAT - Create/modify the down-line task load data base for HLD
>; ====================================================================
>;
>;
>; Creating initial copy of the down-line task load data base.
>;
>* Command [S]: HELP
>;
>; Typing <ESCAPE> to any question will provide help on how to answer the
>; question.
>;
>; The following commands are supported.
>;
>;   DEFINE      Create new node and/or task entries.
>;
>;   EXIT        Terminate the session, replacing the old data base
>;               contents, and optionally assemble and build the new data
>;               base.  Same as <CTRL/Z>.
>;
>;   HELP        Print this message.  Same as <ESCAPE>.
>;
>;   LIST        Show all node and task entries.
>;
>;   PURGE       Delete old node and/or task entries.
>;
>;   QUIT        Terminate the session with no changes to  the  old  data
>;               base contents.
>;
>* Command [S]: DEFINE
>;
>* Define new general purpose tasks? [Y/N]: Y
>;
>; Define the next general purpose task.  If finished, type <RET>.
>; ----------------------------------------------------------------
>;
>* Gen    - Task image file (<RET> = done) [S]: DL0:[1,100]UPDATE.TSK
>* Gen    - Task name (Def=UPDATE) [S]: ...UPD
>* Gen    - Unmapped task (Def=NO)? [Y/N]:
>;
>; Define the next general purpose task.  If finished, type <RET>.
>; ----------------------------------------------------------------
>;
>* Gen    - Task image file (<RET> = done) [S]:
>;
>; Define new node entry.
>; ----------------------
>;
>* Name of new node (<RET> = done) [S]: DOWLIN
>;
>; Define the next task for node DOWLIN.  If finished, type <RET>.
>; ------------------------------------------------------------
>;
>* DOWLIN - Task image file (<RET> = done) [S]: UPDATE.TSK
>* DOWLIN - Task name (Def=UPDATE) [S]: ...UP1
>* DOWLIN - Perform LUN fixing (Def=NO)? [Y/N]: Y
>;
>; Define the next task for node DOWLIN.  If finished, type <RET>.
>; ------------------------------------------------------------
```

Example 4 Generating a Down-Line Task Loading Database
(Sheet 1 of 2)

```
>;
>* DOWLIN - Task image file (<RET> = done) [S]:
>;
>; Define new node entry.
>; ----------------------
>;
>* Name of new node (<RET> = done) [S]:
>;
>* Command [S]: EXIT
>;
>* Do you wish to task build the HLD data base now? [Y/N]: N
>;
>; When you wish to build the HLD down-line task load data base, you  may
>; either execute this command file, and issue the EXIT command,  or  you
>; may mount the DECnet object kit on some disk  device  'ddu'  and  then
>; issue the following commands to MCR.
>;
>;      ASN ddu:=IN:
>;      SET /UIC=[300,24]
>;      MAC @HLDTABASM
>;      TKB @HLDTABBLD
>;
>@ <EOF>
>
```

Example 4 Generating a Down-Line Task Loading Database (Sheet 2 of 2)

```
         .TITLE  HLDTAB
.IDENT   /02.001/
;
; COPYRIGHT (C) 1981, 1982 BY
; DIGITAL EQUIPMENT CORPORATION, MAYNARD, MASS.
;
; HLD EXTERNAL TABLE
;
.MCALL   HLDTB$
HLDTB$
HTASK$   ...UPD,<DL0:[1,100]UPDATE.TSK>,MAP
HNODE$   DOWLIN
HTASK$   ...UP1,<UPDATE.TSK>,LUN
.END
```

Example 5 HLD Database

Satellite Task Loader

The SLD must be installed and fixed in the RSX-11S system. For example:

```
>VMR
Enter Filename: RSX11S
VMR> INS [xxx,64]SLD/FIX=YES
VMR> LOA OV:
```

This establishes SLD as the loading task (LDR...) for the RSX-11S executive. RSX-11S task image files are specified during network generation. These files are stored under network UIC on the host system. The network UIC is [xxx,64] for RSX-11S.

Additional Notes on Down-Line Task Loading

- If the RSX-11S operating system is not to be down-line system loaded, you must define the node SLD connects to, using the VNP SET EXECUTOR HOST command; for example:

  ```
  >VNP RSX11S
  VNP> SET EXECUTOR HOST SUPER
  ```

  In this example, SUPER is the name of the host node that HLD resides.

- HLD must be installed on the host system.

UP-LINE DUMPING

You can include certain SET NODE parameters in the permanent or volatile database that allow an adjacent RSX-11S or server node to dump its memory into a file on the local DECnet-11M or DECnet-11M-PLUS node. (A target node can also dump its memory up-line to an adjacent DECnet/VAX node). This procedure is referred to as up-line dumping. It is a valuable tool for crash analysis; that is, programmers can analyze the dump file and determine why the RSX-11S system failed. When it (RSX-11S) detects an impending system failure, the RSX-11S node requests the up-line dump if the appropriate support is selected during the RSX-11S system generation.

Up-Line Dump Requirements

- The target node must be directly connected to the executor node by a physical line. The executor node provides the line and circuit level access.

- The target node must be capable of requesting the up-line dump when it detects a system failure. If the dumping program (NETPAN) does not exist on the target, up-line dumping cannot occur.

- The executor node circuit involved in the dump operation must be enabled and set to ON to perform service functions. This is the default state of the circuit following NETGEN. For example, the following command prepares circuit DMC-0 for up-line dumping.

  ```
  NCP> SET CIRCUIT DMC-0 SERVICE ENABLE STATE ON
  ```

- If the target does not supply the DUMP COUNT value, the executor must have this value in its volatile database. RSX-11S systems built with the NETPAN routine always include the count.

- The executor must have a DUMP FILE entry in the volatile database.

Up-Line Dump Database Parameters

You can define the following up-line dump parameters in either the permanent database, using the CFE DEFINE NODE command, or the volatile database, using the NCP SET NODE command:

- DUMP FILE file -- Specifies the file on the host node that receives the RSX-11S memory dump.
- DUMP ADDRESS address -- Specifies the octal address in the target's memory to begin the dump.
- DUMP COUNT number -- Specifies a default number in decimal of 16-bit words to be dumped.

MULTIPOINT

Basic Concepts

- Three types of multipoint are available:
  - PCL11-B TDM bus
  - DMP11/DMV11 microprocessor-based serial interface. Implements polling algorithm in microcode.
  - Software polling. Implemented in DDCMP software (where applicable).
- Configurations
  - PCL11-B (see Appendix B for details)
  - DMP11/DMV11 (see Appendix C for details)
  - Software polling (see the following list)
- Concepts on software polling
  - There is a control station that contains the polling algorithm code. This station is responsible for selecting the tributary station for communication.
  - There can be up to eight tributary stations. Each listens to the messages from the control station, discarding messages not for itself. It also responds with data or ACK when it receives the polling message from the master.
  - The relationship between the control station and the tributaries can be viewed conceptually as point-to-point connections from the control station to each tributary (see Figure 1).

- The code is implemented at the DDCMP level. The control station polls the tributaries when there is no transmission to any tributary. The tributaries listen to every message from the master and check the STATION ADDRESS field in the DDCMP header to see at whom the message is aimed.

  Thus, this technique is suitable for applications with low CONTROL-TO-TRIBUTARY traffic. It is not recommended if the application involves high CONTROL-TO-TRIBUTARY traffic or high TRIBUTARY-TO-TRIBUTARY traffic.

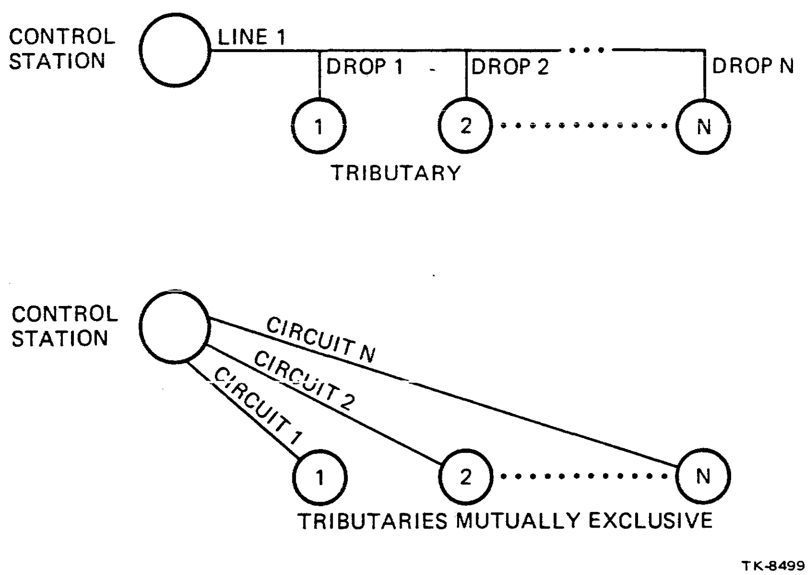


Figure 1 Multipoint Operation

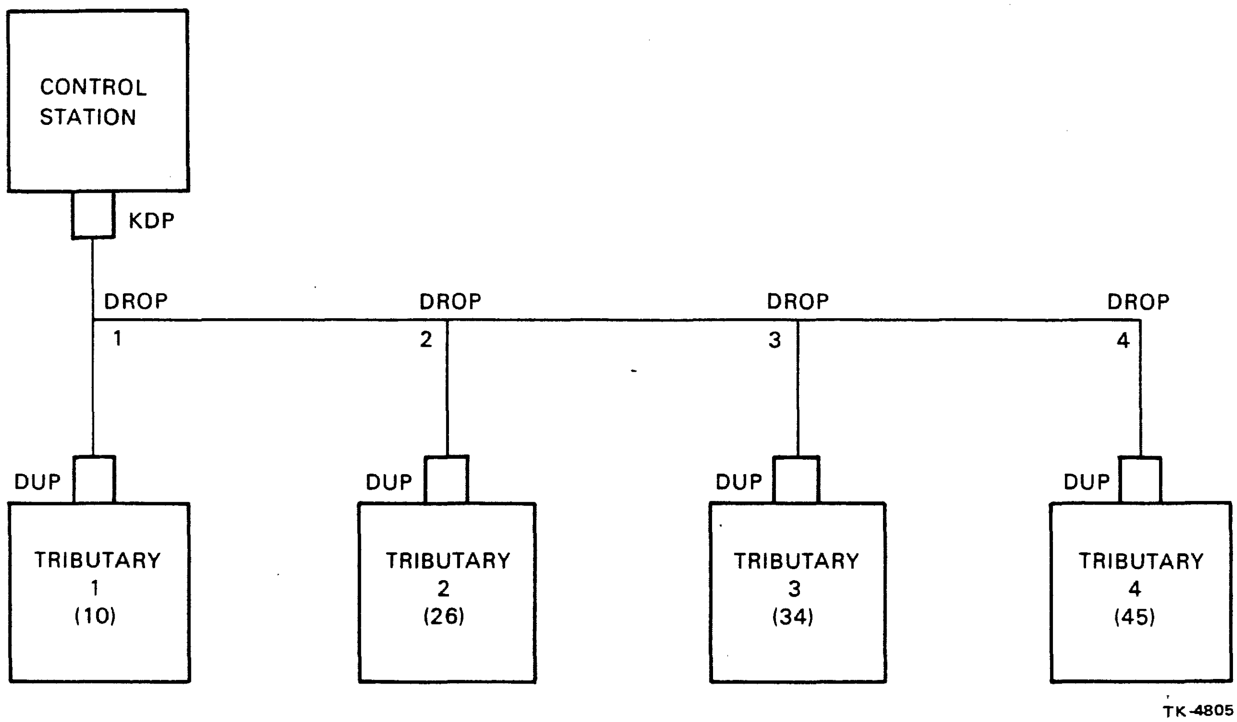


Figure 2 Configuration Example (Software Polling)

The Polling Algorithm

Conceptually, the polling algorithm is the same for software polling and DMP11/DMV11.

- Selection of the tributary by the control station is controlled by two basic mechanisms:
  - Transmission directed to a particular tributary.
  - Use of a polling list to select the tributary.
- A tributary can be:
  - Nonexistent -- Node has not been turned on
  - Active -- A responding tributary
  - Dying -- A tributary that has not responded to the last two consecutive polls
  - Dead -- A tributary that has not responded to the last eight consecutive polls (two caused it to be dying)
- Under NCP/VNP or CFE, the polling rates can be changed by altering:
  - Active Polling Ratio (software polling only) -- The control station polls each active tributary every nth time it goes through the polling list, where n is the active polling ratio for that tributary. This ratio defaults to 1. Thus, the frequency of polling for each station can be varied.
  - Dying and Dead Polling Ratios (software polling only) -- This is on a line basis. There is one dead polling ratio (X) for all tributaries belonging to the line to a control station. The master polls one dead tributary each X times through the polling list (round-robin). The dead polling ratio default setting is 8. The control station polls dying tributaries four times as often as a single dead one (or one-fourth the dead polling ratio).

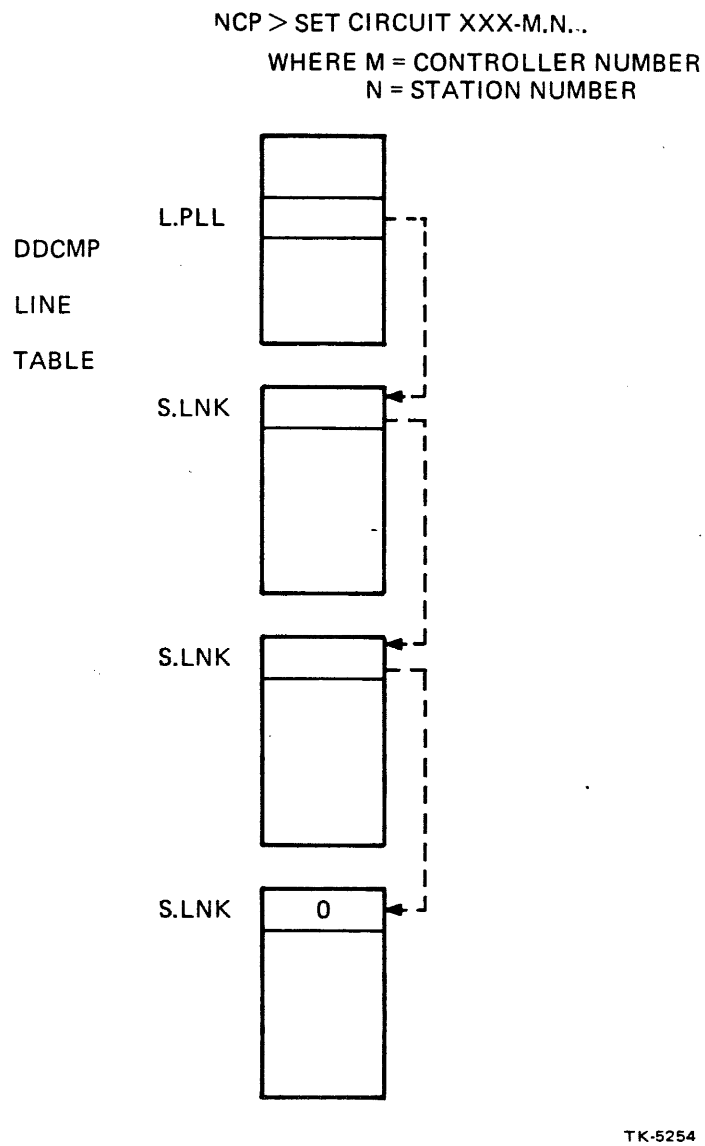


Figure 3 Idle Polling List (All Station Tables)

NCP> SET CIRCUIT XXX-M.N STA ON

where:

M = Controller number
N = Station number

- Set station to ACTIVE polling state
- Set polling count to the MULTIPOINT ACTIVE rate
- Set up to send STRT message
- If first active station on the line, awaken polling

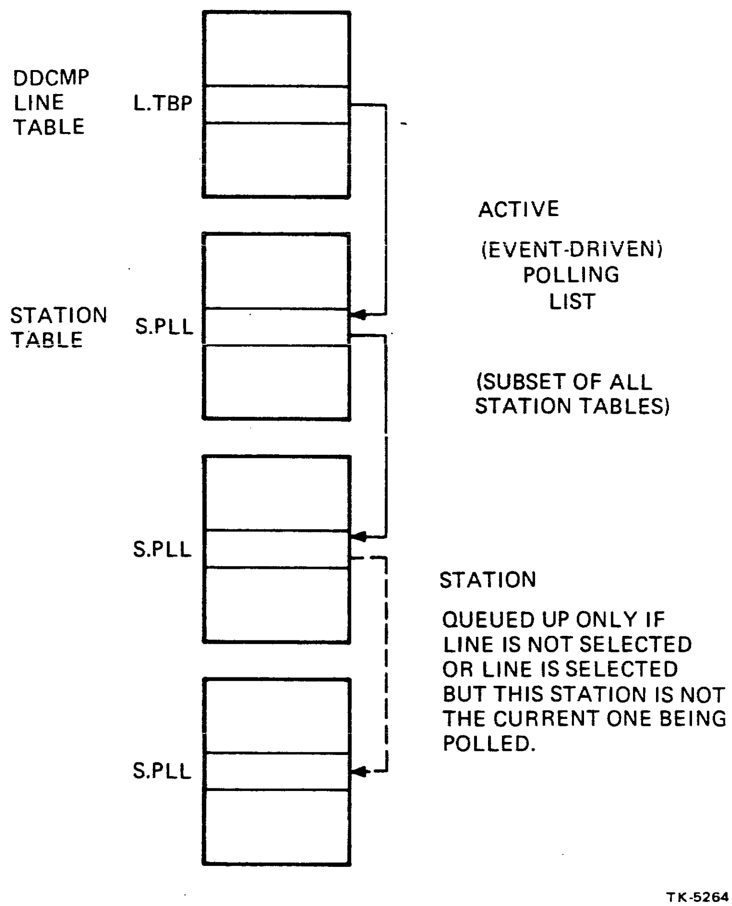


Figure 4 Active Polling List

Polling is Invoked

- If a select is received on a multipoint master.
- After a select timeout on a transmitted message to a multipoint slave.
- On startup for the first station on a multipoint line (already mentioned).
- After a one-second timer completion, if one had not been previously available to poll.

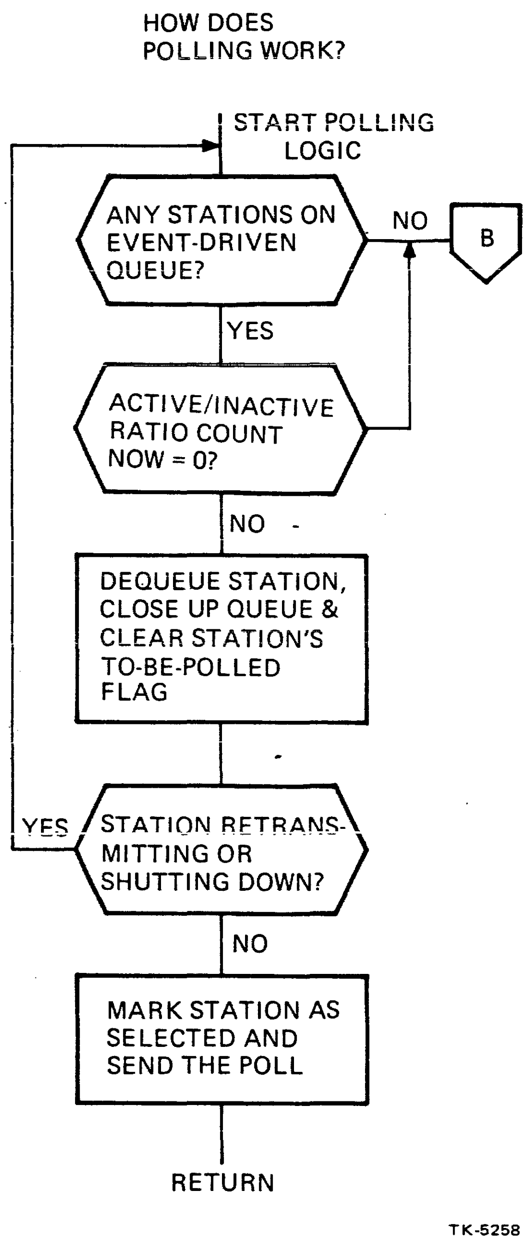


Figure 5 Polling Mechanism (Sheet 1 of 3)

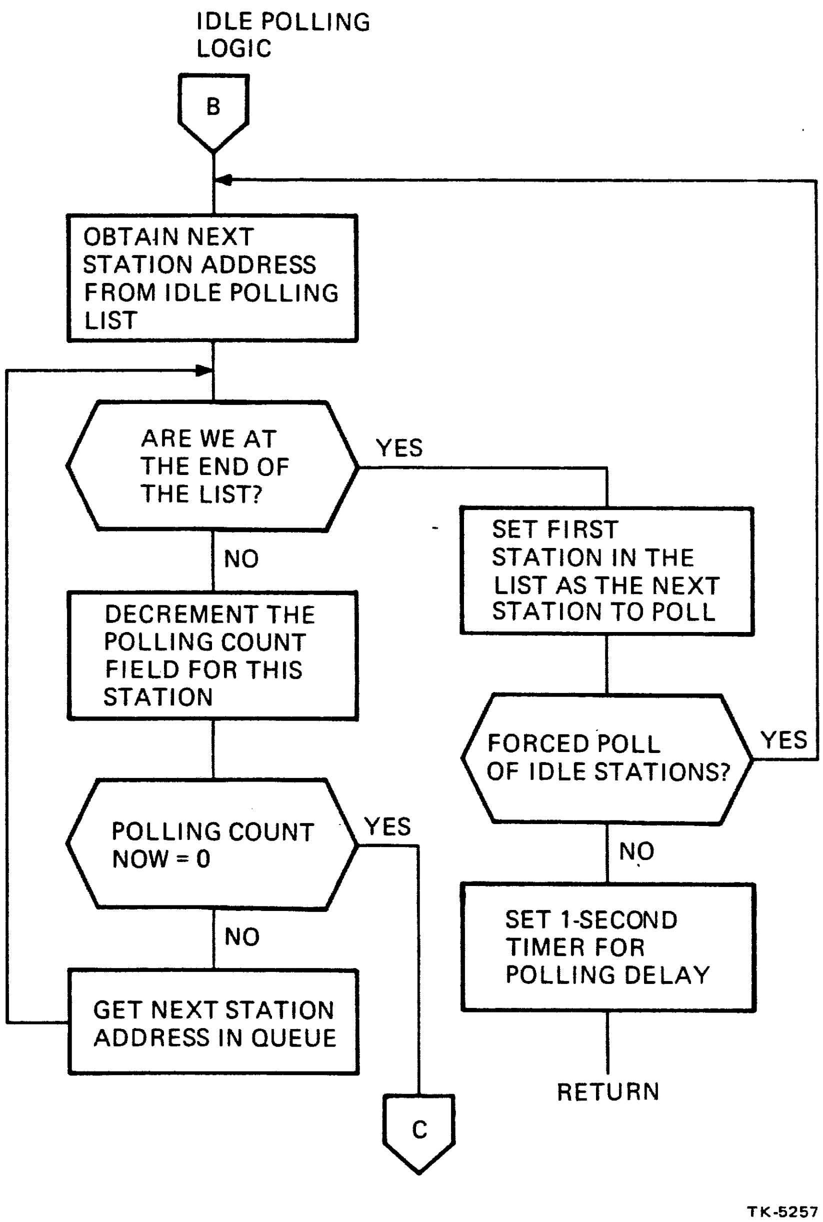


Figure 5 Polling Mechanism (Sheet 2 of 3)

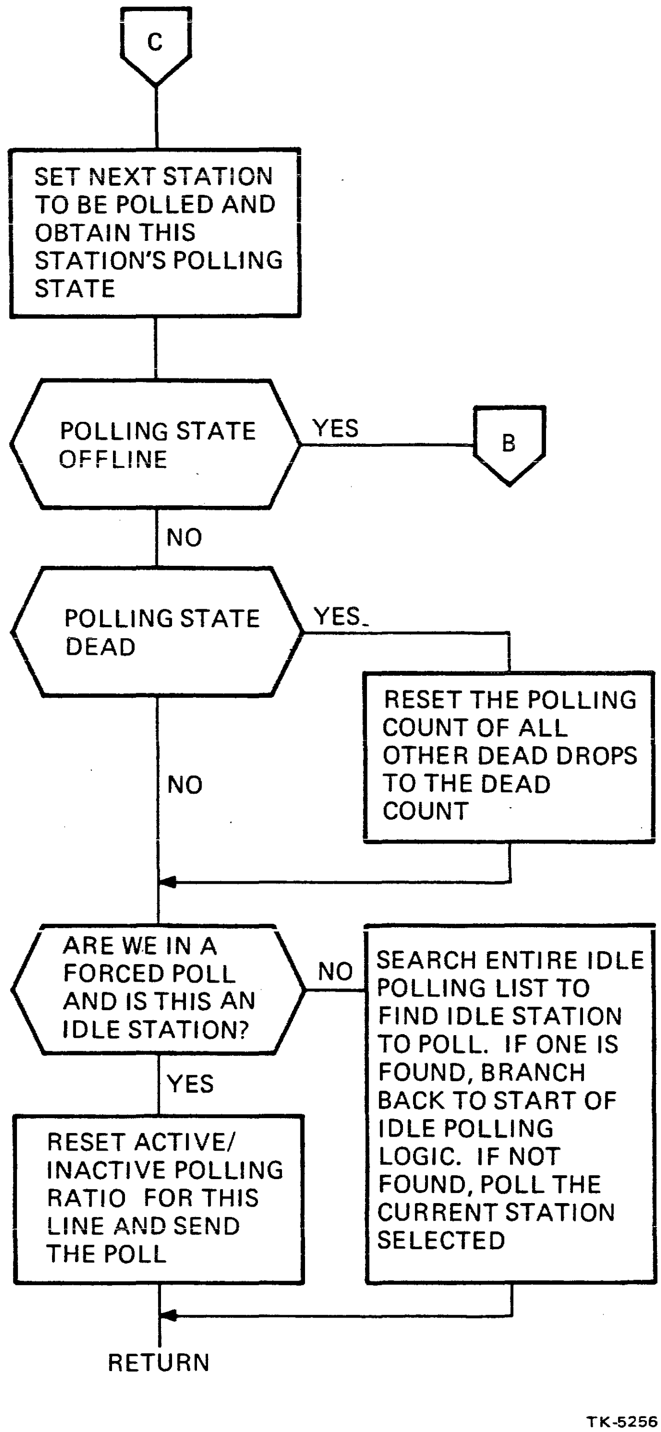


Figure 5 Polling Mechanism (Sheet 3 of 3)

Ethernet Basic Concepts

Ethernet is a high-speed, baseband local area network communications network. DIGITAL's Ethernet specification and its implementation conform to the IEEE 802 standard.

This specification is incorporated in two of the lower layers of the DIGITAL Network Architecture (DNA). These layers provide packet delivery service between nodes in the network, and interface with the higher DNA layers to provide capabilities for network management, error recovery, internetwork communication, and the user interface.

The technique for accessing the Ethernet channel is called Carrier Sense Multiple Access with Collision Detection (CSMA/CD). CSMA/CD can be defined as follows:

- Carrier Sense (CS) -- A device listens for a clear channel before transmitting. If the channel is in use (carrier detected), the device delays transmission.
- Multiple Access (MA) -- When the channel is clear, all users have equal access to it.
- Collision Detection (CD) -- Two or more users may sense a clear channel and simultaneously try to transmit. This results in collision of data. Ethernet senses the collision condition, stops the transmission, and automatically tries to retransmit the data after waiting a randomly selected amount of time.

Ethernet Advantages

- Simplified Network Design

  Ethernet's architecture and design rules are streamlined and uncomplicated. If the cable, transceivers, and controllers meet the Ethernet specifications, the design rules are as follows:

  - A single cable segment cannot exceed 500 meters in length.
  - No more than 100 transceivers can be connected per cable segment.
  - No more than two repeaters can be placed between any two nodes in the network.
  - A network cannot exceed 1023 nodes.

- Simplified Installation

  Ethernet can be brought up one node at a time. A newly installed node can communicate immediately with all other active nodes on the network.

- Reduction of Wiring

  Ethernet's single-network cable replaces the many interconnecting cables in traditional networks.

Ethernet Communications Controllers

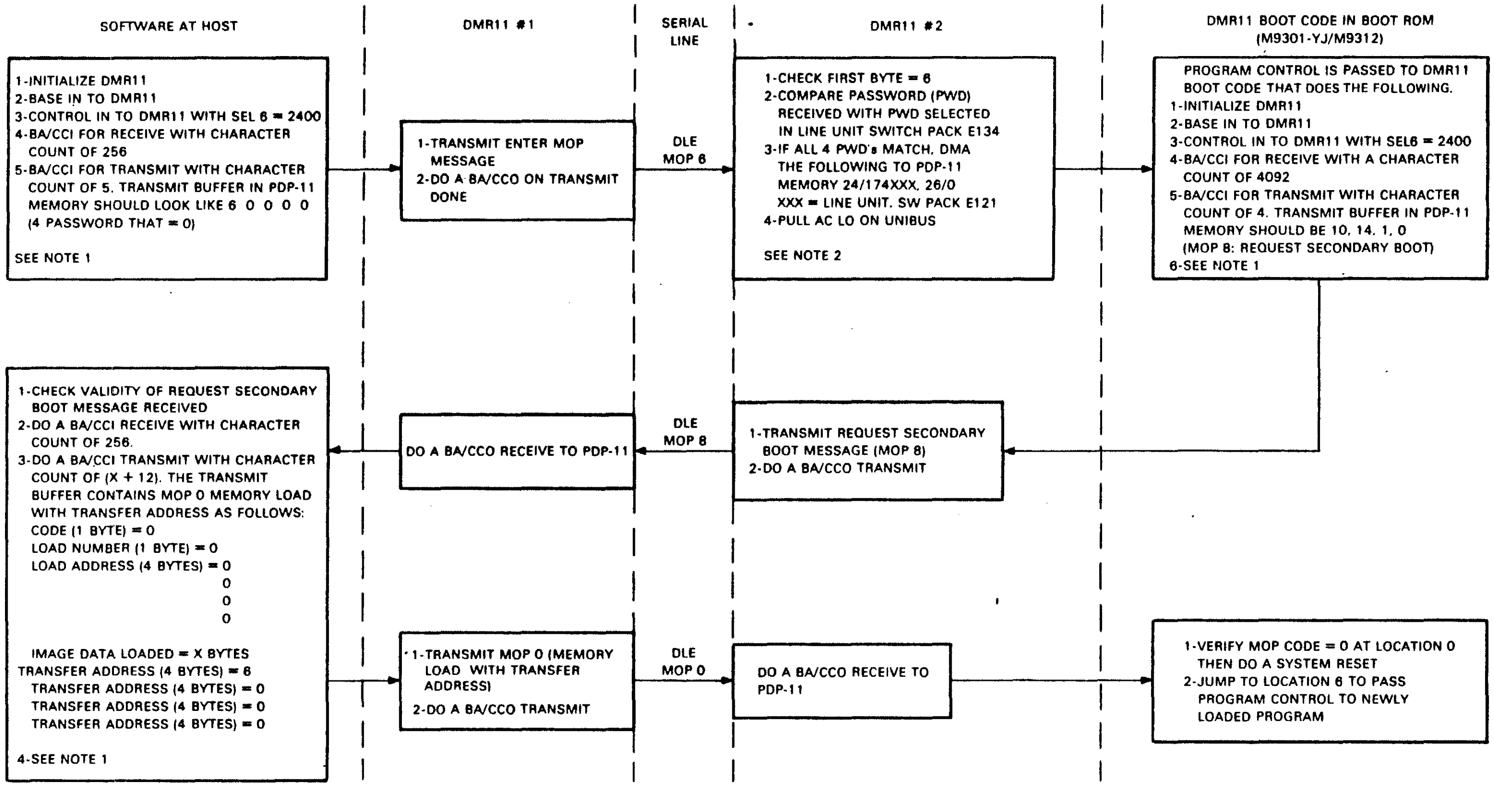
The Ethernet communications controllers connect the host system input/output bus to the transceiver cable and provide the hardware interface to the cable. DIGITAL has three types of Ethernet communications controllers:

- DEUNA (UNIBUS to Ethernet Communications Controller) -- Used for VAX-11 and PDP-11 UNIBUS host systems.
- DEQNA (Q-BUS to Ethernet Communications Controller) -- Used for LSI-11 Q-BUS host systems.
- DECNA (Professional 300 personal computer to Ethernet Communications Controller) -- Used for the Professional 300 Series personal computers.

See Appendix D for details.

APPENDIX A
Details of Down-Line System Load

DMR11 Down-Line Load



NOTES:

1) IN DDCMP MAINTENANCE, THERE IS NO RETRANSMISSION; BA/CCO FOR A TRANSMIT OPERATION DOES NOT ASSURE SUCCESSFUL RECEPTION AT THE REMOTE END. THE SOFTWARE MUST KEEP A REPLY TIMER OF 3 SECONDS TO RETRANSMIT THE MESSAGE WHEN NO REPLY IS RECEIVED WITHIN THAT TIME

2) XXX MUST BE THE OFFSET FROM 173000 OF THE DMR11 BOOT CODE IN M9301-YJ/M9312

MK-2246

Figure 6 Typical Down-Line Load Procedure (DMR11 to DMR11)

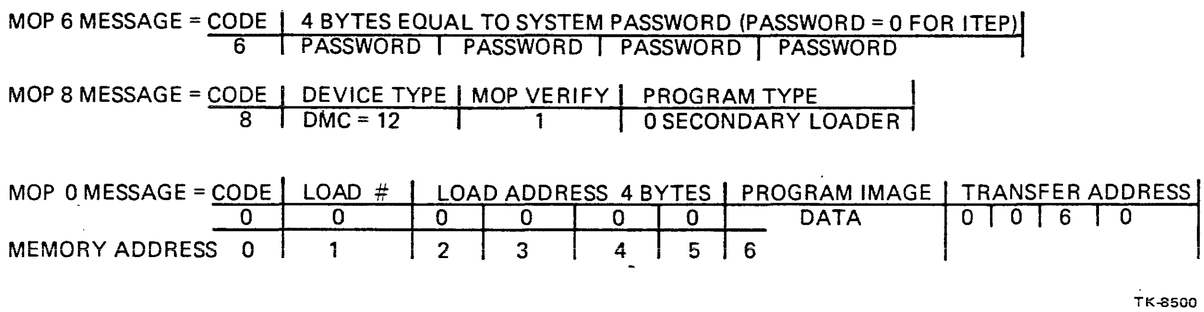


Figure 7 Down-Line Load Message Flow

DMC11 Line Unit

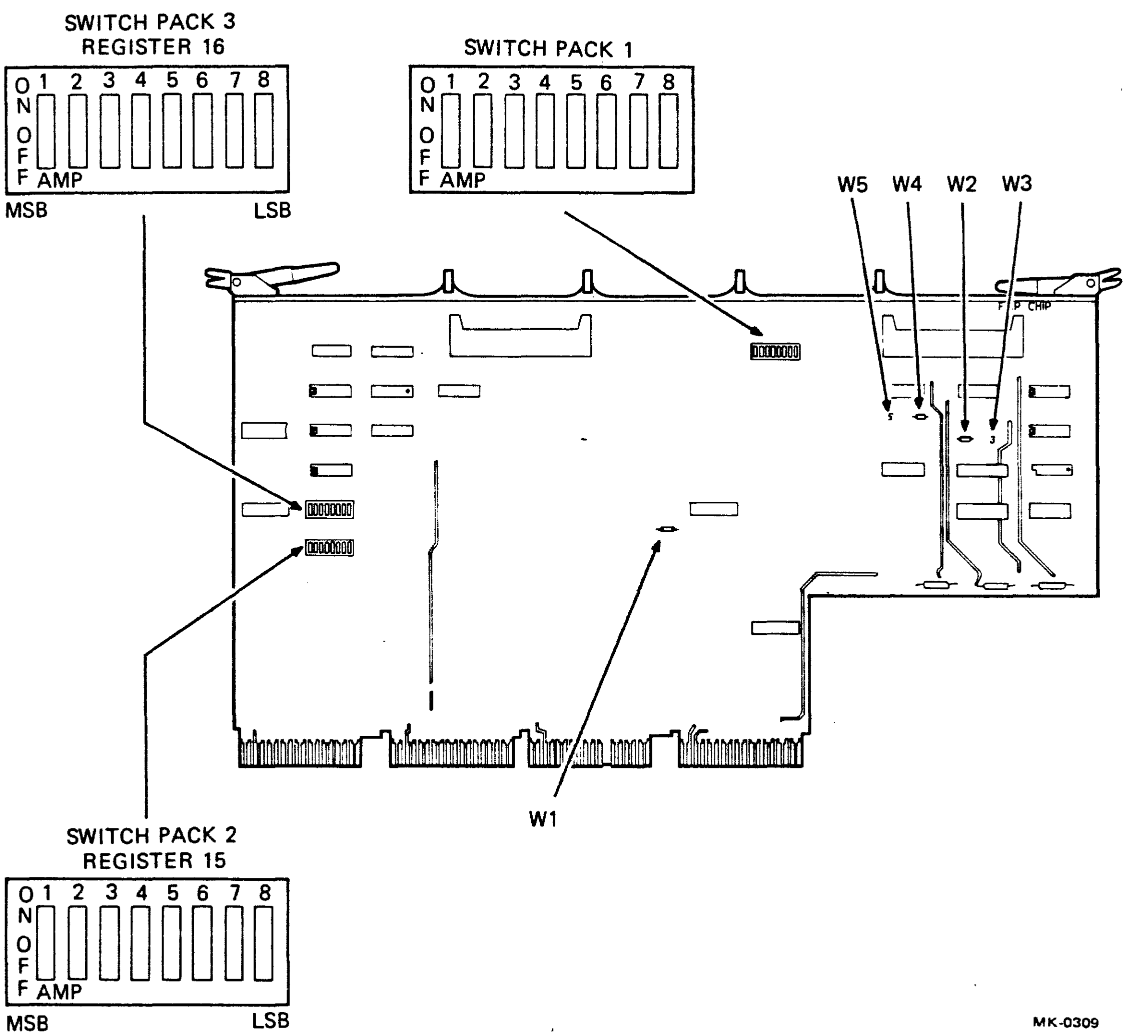


Figure 8 DMC11 Line Unit

Switch Pack 1 -- An eight-switch DIP installed in location E26 on the M8201 line unit, and in location E29 on the M8202-YA/YD line unit. These switches are used to configure the specific line unit type.

Switch Pack 2 -- An eight-switch DIP installed in location E87 of the M8201 line unit, and in location E90 on the M8202-YA/YD line unit. This switch pack is used only during down-line load procedures, and is configured for the required MOP password.

Switch Pack 3 -- An eight-switch DIP installed in location E88 of the M8201 line unit, and in location E91 on the M8202-YA/YD line unit. This switch pack is used only during down-line load procedures and is configured for the required boot offset address.

Configure switch pack 2 for selection of the special Remote Load Detect (RLD) bootstrap password. Table 5 represents a detailed description of the switch settings for switch pack 2 of the M8201 and M8202-YA/YD line unit. This switch pack is only used for down-line load, and is configured for the bootstrap password.

Table 5 M8201/M8202-YA/YD Switch Pack 2
Bootstrap Password

Switch	Function
	NOTE Switch OFF equals a logical one (1).
1 - 8	Bootstrap Password Selection -- These switches are physically connected to IBUS register 15, with switch 1 being the most significant bit and switch 8 the least significant bit. In the DMC11, this switch pack contains the bootstrap password if the bootstrap feature is being used. Otherwise it contains an octal 377 (switches 1 through 8 OFF (open)) that disables the RLD feature. Example of a password of octal 012:

	MSB							LSB
Switch	1	2	3	4	5	6	7	8
	ON	ON	ON	ON	OFF	ON	OFF	ON

Configure switch pack 3 for selection of the special RLD boot offset address. Table 6 represents a detailed description of the switch settings for switch pack 3 of the M8201 and M8202-YA/YD line unit. This switch pack is only used for down-line load, and is configured for the boot offset address.

Table 6 M8201/M8202-YA/YD Switch Pack 3
Boot Offset Address

Switch	Function
	NOTE Switch OFF equals a logical one (1).
1 - 8	Bootstrap Offset Address Selection -- These switches are physically connected to IBUS register 16, with switch 1 being the Most Significant Bit (MSB) and switch 8 the Least Significant Bit (LSB). When the RLD feature is used, switch pack 2 (switches 1-8) must contain the appropriate offset entry address in the bootstrap program. The address formed by the DMC11 is 173XXX, where XXX is the content of E88 or E91 switches 1-8. Variations in bootstrap ROMs may require different entry addresses to boot the DMC11/DMR11. If the RLD feature is not used, the offset must be set to octal 377, switches 1-8 all OFF (open). The following examples are for the M9301-YJ bootstrap module. Depending on the bootstrap module used, reference should be made to the appropriate manual for specific details. 1. M90301-YJ Bootstrap Technical Manual 2. M9312 Technical Manual

Table 6 M8201/M8202-YA/YD Switch Pack 3
Boot Offset Address (Cont)

Switch	Function
	To boot DMC11 unit 0 without CPU diagnostics, address 356 must be selected:

	MSB							LSB	
Switch	1	2	3	4	5	6	7	8	= 356
	OFF	OFF	OFF	ON	OFF	OFF	OFF	ON	

To boot DMC11 unit 1 without CPU diagnostics, address 374 must be selected:

	MSB							LSB	
Switch	1	2	3	4	5	6	7	8	= 374
	OFF	OFF	OFF	OFF	OFF	OFF	ON	ON	

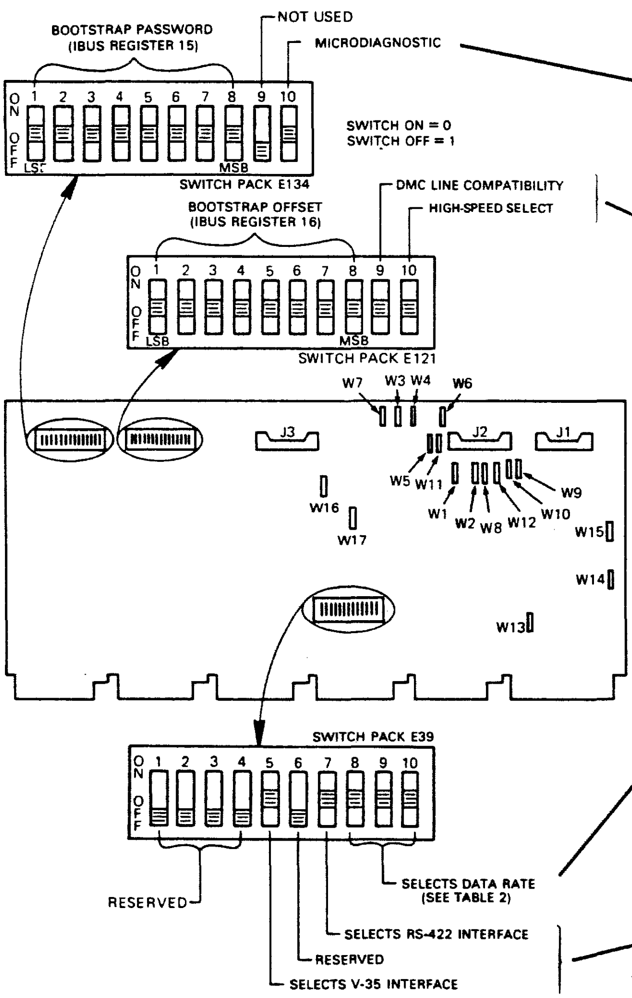
To boot DMC11 unit 0 with CPU diagnostics, address 354 must be selected:

	MSB							LSB	
Switch	1	2	3	4	5	6	7	8	= 354
	OFF	OFF	OFF	ON	OFF	OFF	ON	ON	

To boot DMC11 unit 1 with CPU diagnostics, address 372 must be selected:

	MSB							LSB	
Switch	1	2	3	4	5	6	7	8	= 372
	OFF	OFF	OFF	OFF	OFF	ON	OFF	ON	

DMR11 Line Unit



SEL 0 BIT 13	SW10 AT E134 ON M8203	EXECUTION OF MICRODIAGNOSTICS
CLEAR	ON	NO MICRODIAGNOSTICS IS RUN
CLEAR	OFF	RUN MICRODIAGNOSTICS
SET	ON	RUN MICRODIAGNOSTICS
SET	OFF	NO MICRODIAGNOSTICS IS RUN

SW 9	SW10	CONFIGURATION
ON	ON	DDCMP VERSION 4.0 (NOT CONNECTED TO DMC11) WITH LINE SPEED LESS THAN 1M B/S.
ON	OFF	DDCMP VERSION 4.0 (NOT CONNECTED TO DMC11) WITH LINE SPEED AT 1M B/S.
OFF	ON	CONNECTED TO LOW-SPEED DMC11-DA (M8200 YA)
OFF	OFF	CONNECTED TO HIGH-SPEED DMC11-AL (M8200 YB)

SPEED	SWITCH		
	8	9	10
1 MEG	ON	ON	ON
500K	OFF	ON	ON
250K	ON	OFF	ON
56K	OFF	OFF	ON
*19.2K	ON	ON	OFF
9.6K	OFF	ON	OFF
4.8K	ON	OFF	OFF
2.4K	OFF	OFF	OFF

* NORMAL SWITCH SETTING UNLESS THE INTEGRAL MODEM OR NULL MODEM CLOCK FEATURES ARE USED.

	SW5	SW6	SW7
RS-232-C OR RS-423-A OR INTEGRAL**	OFF	OFF	OFF
V.35	ON	OFF	OFF
RS-422	OFF	OFF	ON

** INTEGRAL MODEM IS SELECTED BY BC55R CABLE WHEN INSTALLED IN J1 OF THE M8203 LINE UNIT. MODULE CONNECTOR J2 MUST NOT HAVE ANY CABLE OR TEST CONNECTOR INSTALLED.

MK-2119

Figure 9 DMR11 Line Unit

Switch Pack E39 -- This switch pack enables the user to select the desired interface option and line speed.

Switch Pack E121 -- During down-line load, this switch pack is used to configure the required BOOT OFFSET address. The switch pack is also used normally to select device and speed compatibility.

Switch Pack E134 -- During down-line load, this switch pack is used to configure the device for the proper MOP password. This switch pack is also normally used for autoanswer and microdiagnostic enabling or disabling.

Table 7 Switch Pack E134 Description

<table>
<tr><th>Switch</th><th>Function</th></tr>
<tr><td></td><td>NOTE
Switch OFF equals a logical one (1).</td></tr>
<tr><td>1-8</td><td>Bootstrap Password Selection -- These switches are physically connected to IBUS register 15, with switch 1 being the least significant bit and switch 8 the most significant bit. In the DMR11, this switch contains the bootstrap password if the bootstrap feature is being used. Otherwise it contains an octal 377 (switches 1-8 OFF (open)). A password of 377 disables the RLD feature.

Example of a password of octal 012:

<table>
<tr><td></td><td>LSB</td><td></td><td></td><td></td><td></td><td></td><td></td><td>MSB</td></tr>
<tr><td>Switch</td><td>1</td><td>2</td><td>3</td><td>4</td><td>5</td><td>6</td><td>7</td><td>8</td></tr>
<tr><td></td><td>ON</td><td>OFF</td><td>ON</td><td>OFF</td><td>ON</td><td>ON</td><td>ON</td><td>ON</td></tr>
</table>
</td></tr>
<tr><td>9</td><td>Autoanswer Enable -- When this switch is in the ON position, autoanswer is disabled. Following a powerup or master clear, the DMR11 asserts DTR, allowing the DMR11 to answer an incoming call for remote load. The call terminates only when the user program issues a Halt Request or the remote end terminates the call.

When this switch is in the OFF position, autoanswer is enabled. This allows the DMR11 to monitor the Ring Indicator (RI) and Data Set Ready (DSR) to answer and control incoming calls. Control is established using a 20-second call set-up timer. Also, if the DMR11 is connected to a modem and the user program has not assigned Base In/Control In to the DMR11, when the DMR11 detects Ring Detect (RD) it drops DTR to disable answering the call. This switch is physically connected to IBUS register 11 bit <13>.</td></tr>
</table>

Table 8 Switch Pack E121 Description

Switch	Function
	NOTE Switch OFF equals a logical one (1).
1 - 8	Bootstrap Offset Address Selection -- These switches are physically connected to IBUS register 16, with switch 1 being the least significant bit and switch 8 the most significant bit. When the RLD feature is used, switch pack E121 (switches 1-8) must contain the appropriate offset entry address in the bootstrap program. The address formed by the DMR11 is 173XXX, where XXX is the content of E121, switches 1-8. Variations in bootstrap ROMs may require different entry addresses to boot the DMC11/DMR11. If the RLD feature is not used, the offset must be set to octal 000, switches 1-8 all ON (closed). NOTE When the RLD feature is used, the microdiagnostics in the bootstrap ROM module (M9301-YJ or M9312) must be disabled. The following examples are for the M9201-YJ bootstrap module. Depending on the bootstrap module used, reference should be made to the appropriate manual for specific details. 1. M9301-YJ Bootstrap Technical Manual 2. M9312 Technical Manual To boot DMR11 unit 0 without CPU diagnostics, address 356 must be selected:

	LSB							MSB	
Switch	1	2	3	4	5	6	7	8	= 356
	ON	OFF	OFF	OFF	ON	OFF	OFF	OFF	

Table 8 Switch Pack E121 Description (Cont)

Switch	Function

To boot DMR11 unit 1 without CPU diagnostics, address 374 must be selected:

	LSB							MSB	
Switch	1	2	3	4	5	6	7	8	= 374
	ON	ON	OFF	OFF	OFF	OFF	OFF	OFF	

To boot DMR11 unit 0 with CPU diagnostics, address 354 must be selected:

	LSB							MSB	
Switch	1	2	3	4	5	6	7	8	= 354
	ON	ON	OFF	OFF	ON	OFF	OFF	OFF	

To boot DMR11 unit 1 with CPU diagnostics, address 372 must be selected:

	LSB							MSB	
Switch	1	2	3	4	5	6	7	8	= 372
	ON	OFF	ON	OFF	OFF	OFF	OFF	OFF	

9 DMC Line Compatible -- The DMR11 microcode uses this switch to determine whether to implement the DIGITAL Data Communications Message Protocol (DDCMP), Version 4.0 or DMC Line Compatible Mode. The switch is physically connected to IBUS register 11 bit <2>.

- o OFF = DMC11 Line Compatible Mode
- o ON = DDCMP Version 4.0, DMR11 Operating Mode

10 High-Speed Select -- The DMR11 microcode uses this switch to determine if the device is running a 1M bps (because the microcode is modified at that speed). This switch is physically connected to IBUS register 11 bit <1>.

NOTE

The combination of switches 9 and 10 must be appropriately selected to satisfy the configuration requirements as listed in Table 9.

Table 9 Switch Pack E121 Description

Switch 9	Switch 10	Configuration
ON	ON	DDCMP Version 4.0 (not connected to DMC11), with line speed less than 1M bps.
ON	OFF	DDCMP Version 4.0 (not connected to DMC11), with line speed at 1M bps.
OFF	ON	Connected to low-speed DMC11-DA (M8200-YA).
OFF	OFF	Connected to high-speed DMC11-AL (M8200-YB).

DMP11 Line Unit

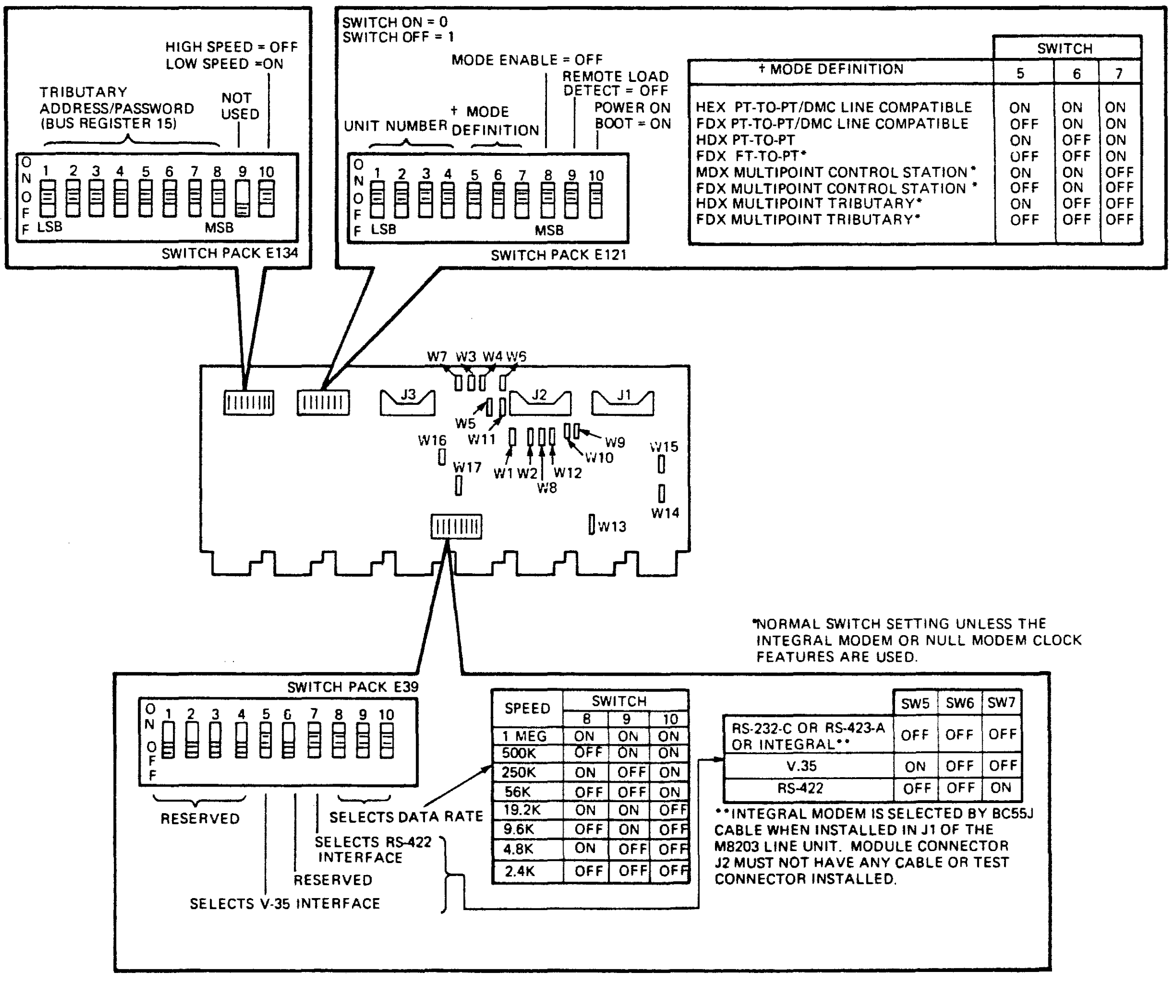


Figure 10 DMP11 Line Unit

Table 10 Switch Pack E121 Selections

Switch	Function
	NOTE Switch OFF equals a logical one (1).
1 - 4	Unit Number Selection -- These switches are physically connected to IBUS register 16 bits 0-3, with switch 1 being the least significant bit and switch 4 the most significant bit. The unit number is used to identify each unique DMP11 if several are installed on the same system. This is particularly important for systems using the RLD and boot features of the DMP11. The boot program must know which of the DMP11s in the floating address space is performing the down-line load.
5 - 7	Mode Definition Selection -- These switches are physically connected to IBUS register 16 bits 4-6, and are used in conjunction with switch 8 to define the mode of operation for the DMP11.

Mode	Switch 5	Switch 6	Switch 7
HDX PT-TO-PT/DMC Line Compatible	ON	OFF	ON
FDX PT-TO-PT/DMC Line Compatible	OFF	ON	ON
HDX PT-TO-PT*	ON	OFF	ON
FDX PT-TO-PT*	OFF	OFF	ON
HDX MULTIPOINT CONTROL STATION*	ON	ON	OFF
FDX MULTIPOINT CONTROL STATION*	OFF	ON	OFF
HDX MULTIPOINT TRIBUTARY*	ON	OFF	OFF
FDX MULTIPOINT TRIBUTARY*	OFF	OFF	OFF

*DDCMP Version 4.0

Switch	Function
8	Mode Enable -- This switch is physically connected to IBUS register 16 bit 7, and is used to indicate that the mode definition is actually selected in switches 5-7. Switch in OFF Position: Mode defined in switches Switch in ON Position: Mode not defined in switches defined by software (overridden by software)

Table 10 Switch Pack E121 Selections (Cont)

Switch	Function
9	Remote Load Detect -- This switch is physically connected to IBUS register 11 bit 2, and is used to enable the RLD feature of the DMP11. Switch in OFF Position: Enables Switch in ON Position: Disables
10	Power ON Boot -- This switch physically connected to IBUS register 11 bit 1, and is used to enable or disable the power ON boot feature of the DMP11. Switch in OFF Position: Disables Switch in ON Position: Enables

NOTE

If the RLD and power on boot features are used, then switches 1 through 10 must be appropriately set. RLD and power on boot features are only applicable to the PDP-11 processors and do not apply to VAX-11 processors.

Table 11 Switch Pack E134 Selections

Switch	Function
	NOTE Switch OFF equals a logical one (1).
1 - 8	Tributary Address/Password -- These switches are physically connected to IBUS register 15 bits 0-7, with switch 1 being the least significant bit and switch 8 being the most significant bit. The tributary address switches allow the user to define the address in switches to which the device responds, for message traffic over the communication link. Valid addresses are: 1 for point-to-point stations and 1 through 255 for multipoint tributary stations; address of zero is illegal. If the power ON boot and RLD features are used, these switches must be set to the tributary address. In addition, for RLD, this address also serves as a password for the maintenance message ENTER MOP MODE. (Tributary address and password must be equal.)
9	Not used in DMP11 -- This switch is physically connected to IBUS register 11 bit 3.
10	High Speed/Low Speed -- This switch is physically connected to IBUS register 11 bit 5. The setting of this switch indicates to the microcode whether the data rate over the communication line is greater or less than 250K bps. Switch in OFF Position: High speed (250K bps) Switch in ON Position: Low speed (<250K bps)
	NOTE If the boot feature is selected, it is mandatory that the switches in switch packs E121 and E134 be appropriately configured according to the following charts.

Table 11 Switch Pack E134 Selections (Cont)

Switch	Function										
		LSB			MSB						
	E121	1	2	3	4	5	6	7	8	9	10
		OFF	OFF	ON	ON	OFF	OFF	OFF	OFF	ON	ON
		Example: Unit Number 3, Mode = FDX TRIB									
		LSB							MSB		
	E134	1	2	3	4	5	6	7	8	9	10
		OFF	OFF	ON	OFF	OFF	OFF	OFF	ON	X	X
										High/Low Speed	
		X = Don't care									
		Example: Trib address = 123 (decimal); must be set in octal representation (123_{10} = 173_8)									
		LSB			MSB						
	E121	1	2	3	4	5	6	7	8	9	10
		ON	ON	OFF	ON	ON	OFF	OFF	OFF	OFF	OFF
		Example: Unit Number = 4, Mode = HDX TRIB									
		LSB							MSB		
	E134	1	2	3	4	5	6	7	8	9	10
		OFF	OFF	ON	OFF	OFF	OFF	OFF	ON	X	X
										High/Low Speed	
		X = Don't care									
		Example: Trib address and password = 123 (decimal); must be set in octal representation (123_{10} = 173_8)									

DMV11 Microprocessor and Line Unit

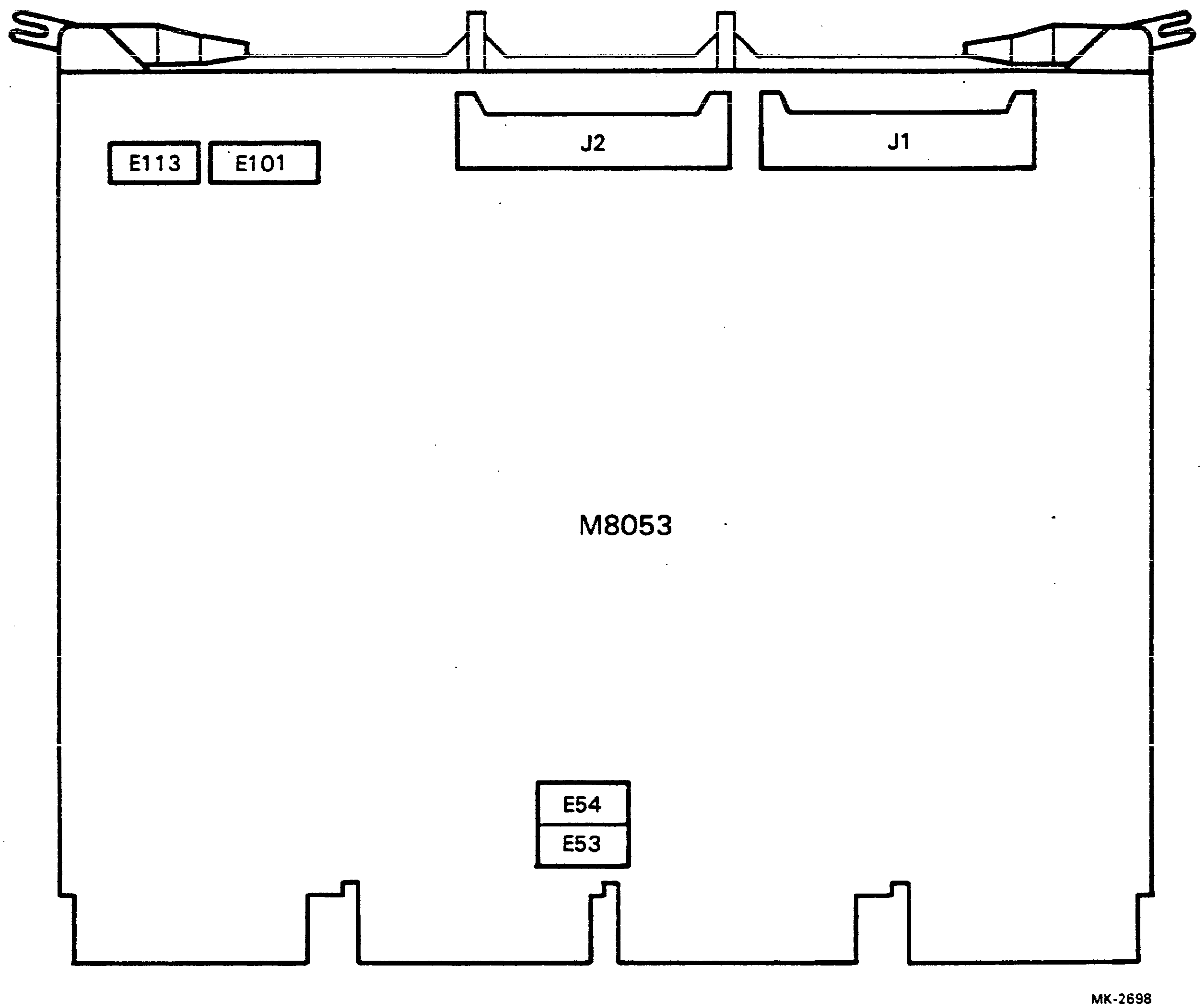
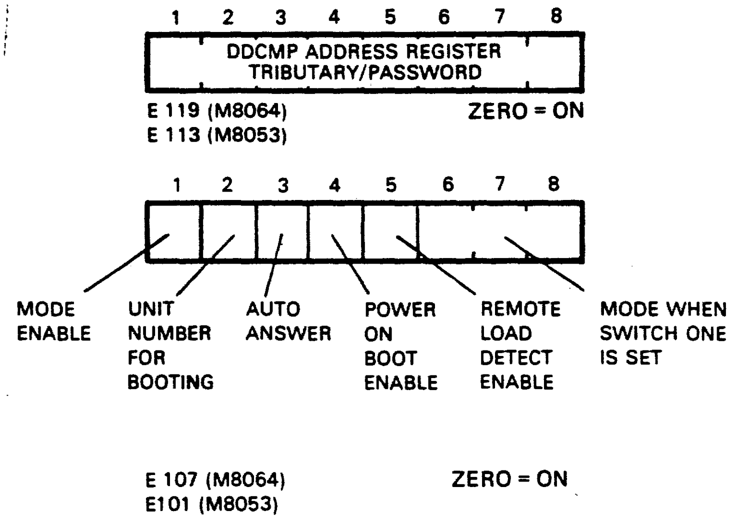


Figure 11 DMV11 Microprocessor and Line Unit



6	7	8	SWITCH SETTING FOR THE MODE OF OPERATION.
ON	ON	ON	HDX PT-TO-PT DMC COMPATIBLE
OFF	ON	ON	FDX PT-TO-PT DMC COMPATIBLE
ON	OFF	ON	HDX POINT-TO-POINT
OFF	OFF	ON	FDX POINT-TO-POINT
ON	ON	OFF	HDX CONTROL STATION
OFF	ON	OFF	FDX CONTROL STATION
ON	OFF	OFF	HDX TRIBUTARY STATION
OFF	OFF	OFF	FDX TRIBUTARY STATION

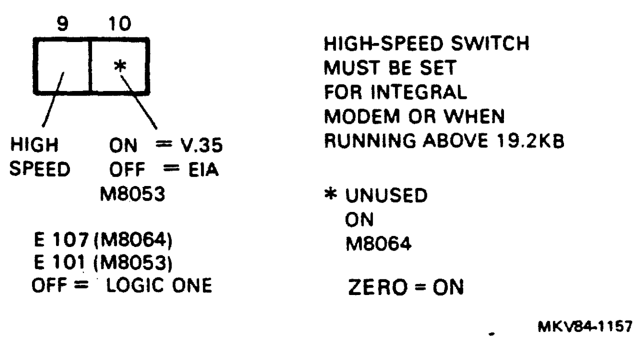


Figure 12 DMV11 Switch-Selectable Features

M9301-YJ Bootstrap Module

Special customer configurations for RLD may require a M9301-YJ or M9312 bootstrap module. The M9301-YJ and M9312 bootstrap modules (M9301-YJ, see Figure 13) provide DECnet bootstrapping capabilities for the DMC11/DMR11 on PDP-11 systems with or without the console switch register. Refer to Table 12 for a description of M9301-YJ switch settings.

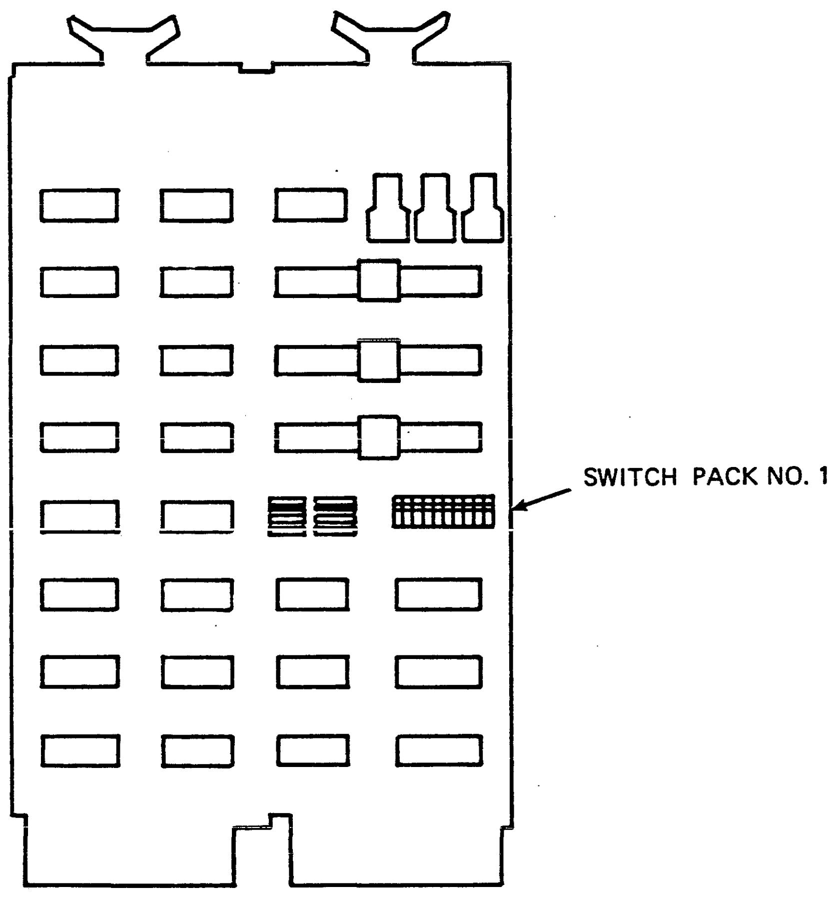


Figure 13 M9301-YJ Bootstrap Module

Table 12 M9301-YJ Switch Settings for DMC11 or DMR11 Bootstrapping from Console

Function	S1-1	2	3	4	5	6	7	8	9	10	Octal Code
Vector 24	-	OFF	-	-	-	-	-	-	-	-	-
Console Emulator with Diag.	ON	ON	ON	ON	ON	ON	ON	ON	ON	ON	000
Unit 0 with Diag.	ON	ON	ON	OFF	OFF	OFF	ON	OFF	OFF	ON	354
Boot DMC11/DMR11 Unit 0 without Diag.	-	ON	ON	OFF	OFF	OFF	ON	OFF	OFF	OFF	356
Boot DMC11/DMR11 Unit 1 with Diag.	ON	ON	ON	OFF	OFF	OFF	OFF	OFF	ON	OFF	372
Boot DMC11/DMR11 Unit 1 without Diag.	-	ON	ON	OFF	OFF	OFF	OFF	OFF	OFF	ON	374

Note that when using the console boot switch, the state of S1-2 has no relevance.

M9312 Bootstrap Module

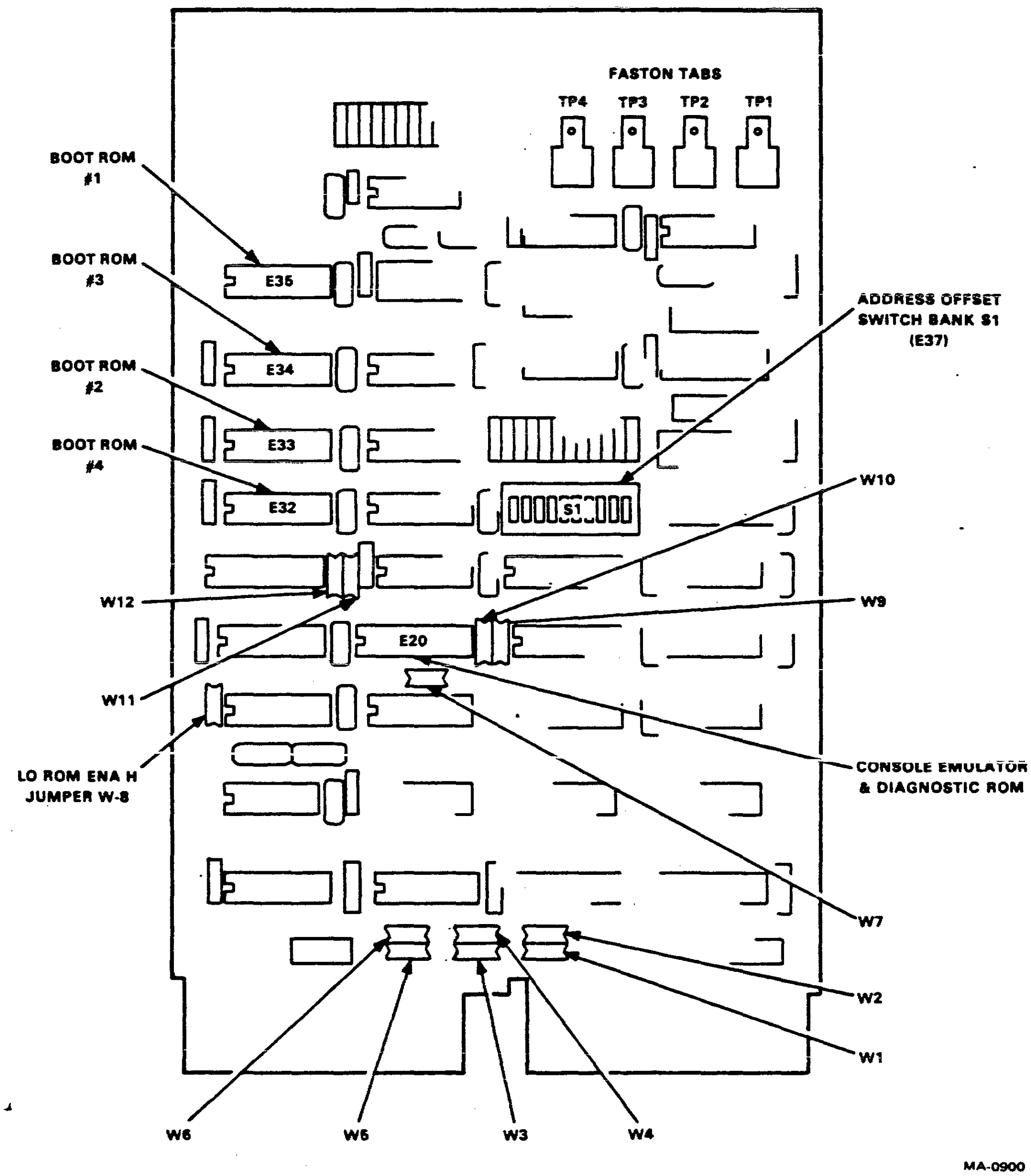


Figure 14 M9312 Bootstrap/Terminator Module

M9312 Setup for DMC11 Down-Line Load

The DMC11 ROM locations (MR11K-AA kit) and the M9312 switch settings required for DMR11 down-line load are as follows (see Figure 14).

Table 13 Switch Settings for DMR11 Down-Line Load

Device	ROM Location (MR11K-AA)	Unit	Start** Address	S-1 Settings									
				1	2	3	4	5	6	7	8	9	10
DMR11	ROM1 23-862A9	0	173004	OFF	*	OFF	OFF	OFF	OFF	OFF	OFF	ON	OFF
	ROM2 23-863A9												
	ROM3 23-864A9	1	173030	OFF	*	OFF	OFF	OFF	OFF	ON	ON	OFF	OFF
DMR11	ROM2 23-862A9	0	173204	OFF	*	OFF	ON	OFF	OFF	OFF	OFF	ON	OFF
	ROM3 23-863A9												
	ROM4 23-864A9	1	173230	OFF	*	OFF	ON	OFF	OFF	ON	ON	OFF	OFF

* When ON, S1-2 enables power boot. (Must be OFF for PDP11/60. See PDP11/60 documentation for power-up boot. On an 11/34, this is a "don't care" switch.) If in doubt, the suggested setting for S1-2 is OFF.

** The starting address for the console load address and start sequence.

CAUTION

The boot ROMs MUST absolutely and without exception be installed in sequence, that is, 23-862A9 through 23-864A9 in boot ROM positions 1 through 3, or in boot ROM positions 2 through 4. A halt at location 173124 or 173324 is the indication of ROM misplacement.

DEUNA Set Up

The remote boot and down-line load features implemented in the DEUNA are used to allow the PDP-11 system in which the DEUNA is installed to be booted and to load a system image into the processor. This function is useful with systems requiring remote booting and loading of their system images. Figure 15 shows a basic PDP-11 system with a DEUNA.

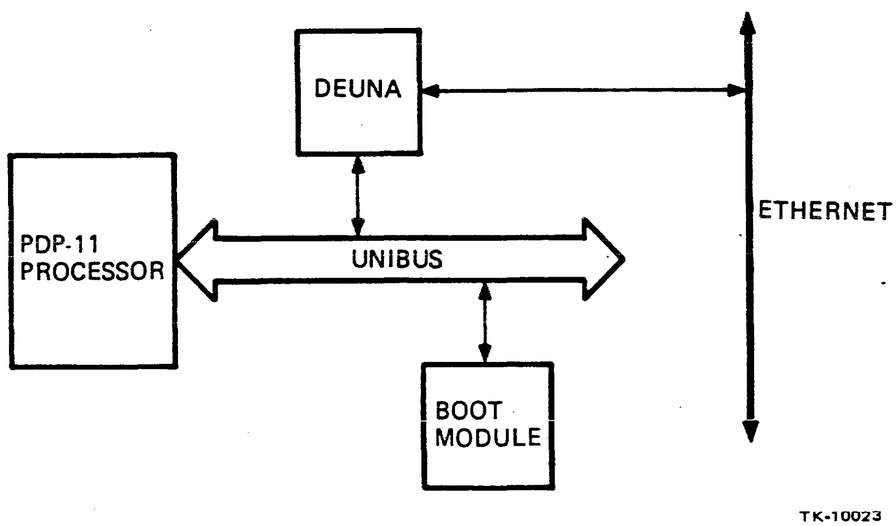


Figure 15 PDP-11 System

System Configuration Guidelines -- When configuring a system to be remote booted and/or down-line loaded use the following guidelines:

- System Processor
  - When ACLO is asserted on the UNIBUS, the processor must be set up to assert DCLO (powerfail sequence).
  - When DCLO is asserted, the processor is initialized and then HALTED. For a boot from ROM function, the processor should start to execute from the boot ROM on the system boot module.

- System Boot Module (except for boot from boot ROM)
  - Disable powerup boot
  - Disable system self-test

NOTE

When configuring a system to meet these guidelines, refer to the processor and boot module manuals for the system.

Table 14 summarizes the system configuration guidelines.

Table 14 Remote Down-Line Load Configuration Guidelines

DEUNA Boot Function	Boot Module	Processor
Boot with ROM	Configure for boot from ROM	ACLO --> DCLO Boot from ROM
Remote Boot	Disable boot on power up Disable system self-test	ACLO --> DCLO Initialize and HALT
Remote/Powerup Boot	Disable boot on power up Disable system self-test	ACLO --> DCLO Initialize and HALT

Remote Boot Disabled -- Remote boot is disabled when the BOOT SEL switches are configured as follows:

BOOT SEL 0 = ON
BOOT SEL 1 = ON

When remote boot is disabled, the system processor can only be booted by the DEUNA by using a BOOT port command. It cannot be booted by using a boot request from another node on the Ethernet.

Remote Boot with System Load -- Remote boot with system load is enabled when the BOOT SEL switches are configured as follows:

BOOT SEL 0 = ON
BOOT SEL 1 = OFF

When remote boot with system load is selected, the DEUNA accepts a boot message received on the Ethernet, boots the system processor and down-line loads the system image.

When a boot message for system boot is received from another station on the Ethernet (NI), the DEUNA performs the following (see Figure 16).

- Boot message is received by DEUNA.
- The DEUNA checks the verification code, message type, and so on.
- The DEUNA transfers a program from ROM by using DMA to system memory.
- The DEUNA asserts ACLO. This simulates a powerfail to the system.
- The DEUNA sends a program request message onto the NI and waits for a memory load with transfer address. The program request message is sent every five seconds for the first eight messages, then every 30 seconds until the memory load with transfer address is performed.
- The DEUNA checks the memory load message, transfers it to WCS, then executes the instructions starting at the transfer address.

  The program loaded into WCS is the secondary loader. This loader is used to bring a tertiary loader into system memory. The tertiary loader is used to load the system image.

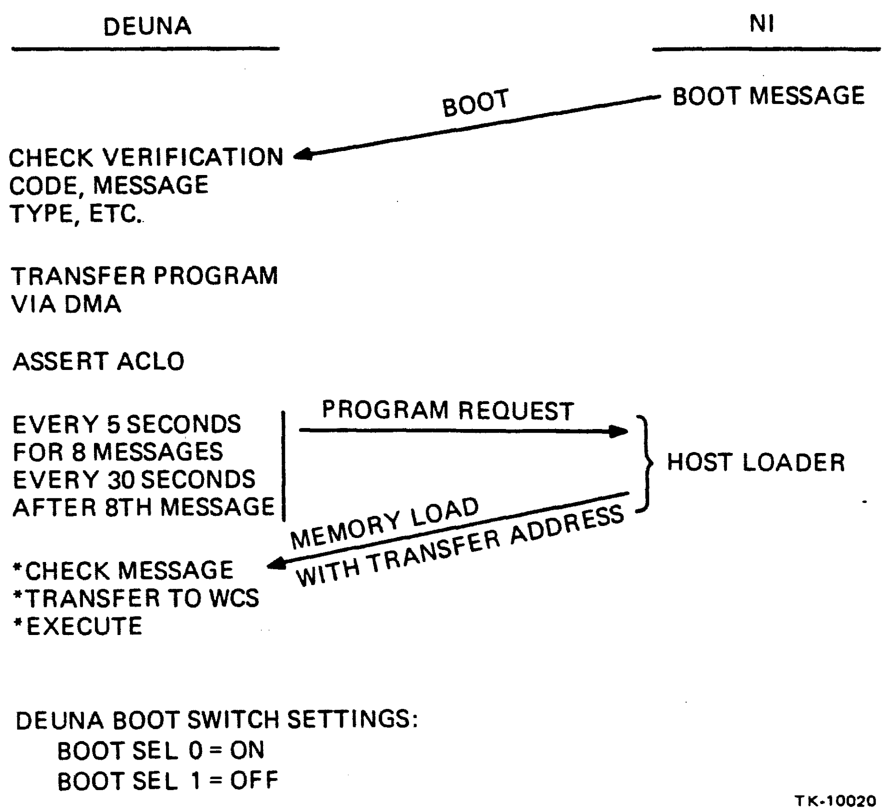


Figure 16 Remote Boot with System Load Functional Flow

Remote Boot with ROM -- When remote boot with ROM is selected, the DEUNA accepts a boot message received on the Ethernet, then boots the system by using ROM-based instructions contained on the system boot module.

Remote Boot with ROM is selected when the boot select switches are configured as follows:

BOOT SEL 0 = OFF
BOOT SEL 1 = ON

When a boot message for system boot is received from another station on the NI, the following sequence occurs (Figure 17):

- The DEUNA checks the verification code, message type, and so on.

- The DEUNA asserts ACLO; this simulates a power failure to the system.

- The system then performs a powerup boot using the ROM-based boot program.

- The boot program, in addition to booting the system, should:

  - Self-test the system
  - Issue a BOOT port command to the DEUNA

- The DEUNA then enters the primary load state.

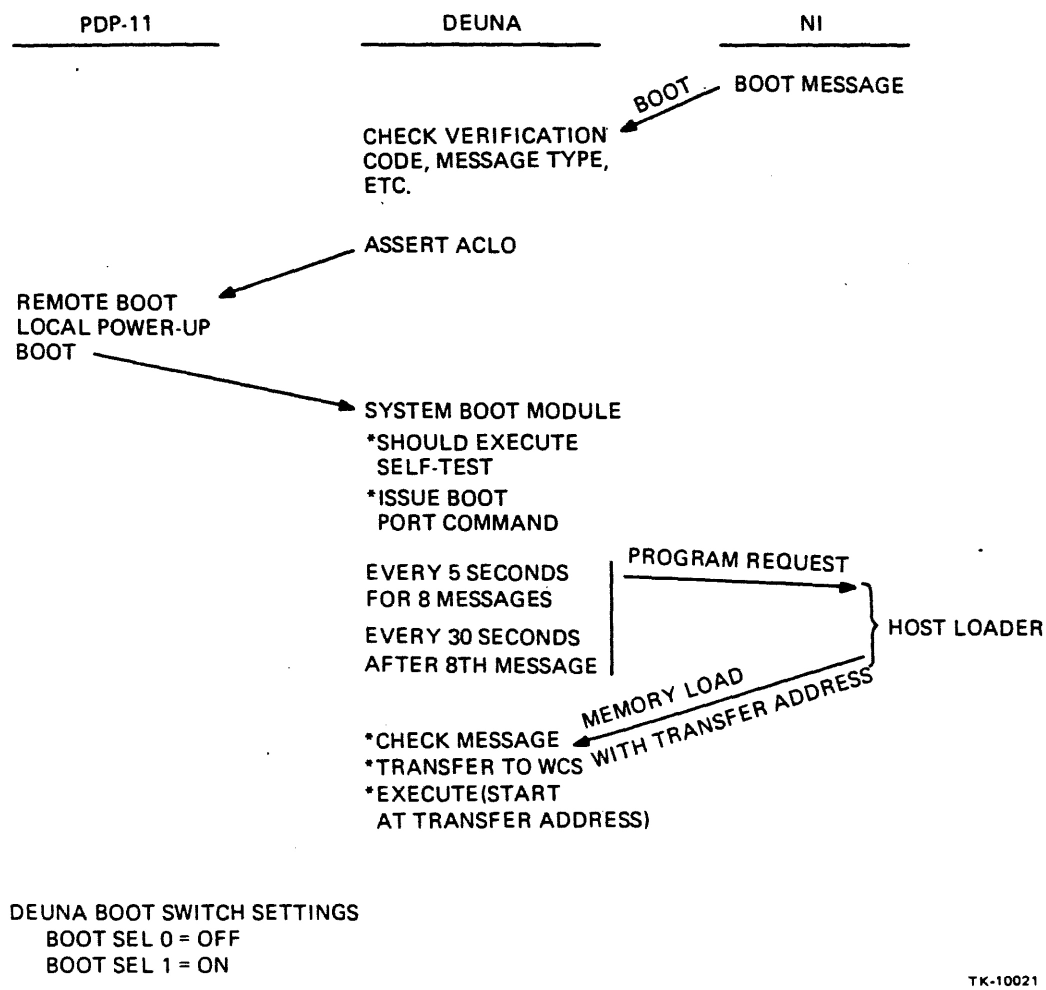


Figure 17 Remote Boot with ROM Functional Flow

Remote Boot/Powerup Boot with System Load -- When the DEUNA is configured for Remote Boot/Powerup Boot with System Load, the DEUNA can boot and perform a system load over the Ethernet in two ways:

- On system powerup
- On receipt of boot message over the Ethernet

The boot select switches on the port module of the DEUNA are configured as follows:

BOOT SEL 0 = OFF
BOOT SEL 1 = OFF

When the system is powered up, the DEUNA performs the following (Figure 18):

- Transfers a program from ROM by using DMA to system memory.
- Asserts ACLO; this simulates a power failure to the system.
- Sends a program request message onto the NI and waits for a memory load with the transfer address. The program request message is sent every five seconds for the first eight messages, then every 30 seconds until the memory load with transfer address is performed.
- Checks the memory load message, transfers it to WCS, then executes the instructions starting at the transfer address.

  The program loaded into WCS is the secondary loader. This loader is used to bring a tertiary loader into system memory. The tertiary loader is used to load the system image.

When the DEUNA receives a boot message from another station on the Ethernet, it functions in the same manner as a Remote Boot with System Load.

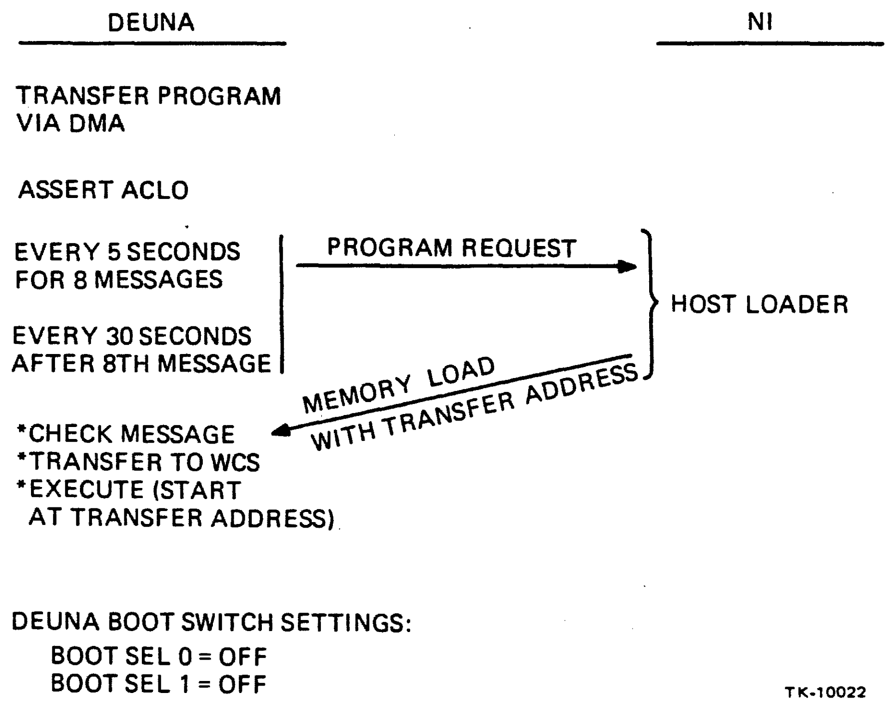


Figure 18 Power-Up Boot with System Load

APPENDIX B
PCL11-B Parallel Communications Link

Features

- DMA block transfer multi-CPU link
- Up to 16 PDP-11 or VAX-11 computers
- Hardware protocol with error checking: CRC and word parity
- PCL11 bus bandwidth is up to 1M bps
- Time division multiplex bus with adjustable time slicing
- RSX-11M and VAX/VMS device driver available
- Failure protected for high-availability systems
- Individual nodes can be powered-off
- PCL11 bus is T-connected; not daisy-chained
- Computers can be removed with PCL11 bus running
- Any node can be easily set up as bus timing master
- Automatic failure from primary to secondary PCL11 bus master
- Receiver can reject transmitter under program control

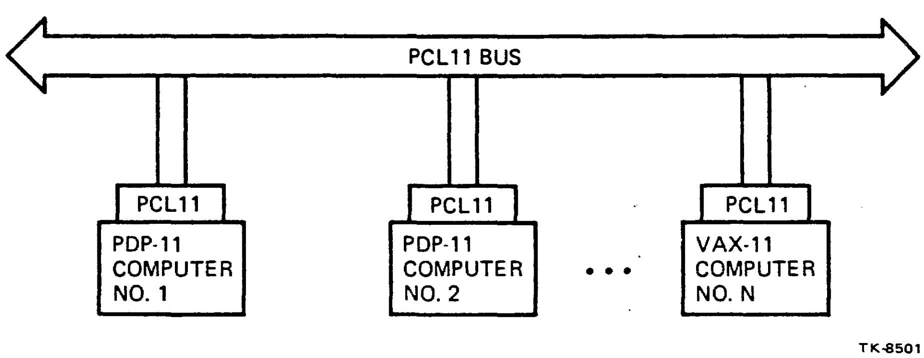


Figure 19 Multiple Processor System Using the PCL11-B

The PCL11

The Parallel Communications Link (PCL11) is a high-performance computer link used for interconnecting multiple PDP-11 and/or VAX-11 computers in a local distributed processing network. Up to 16 processors can be connected to the PCL11 network. Each computer can send or receive messages or data blocks to or from any other computer in the network. Communications occur in a DMA block transfer mode over a Time-Division-Multiplexed (TDM) 16-bit parallel bus. Because of the TDM nature of the PCL11 bus, up to 16 conversations may be conducted concurrently.

The power and features of the PCL11 system make it the appropriate multiprocessor communications link for applications of five or more processors in local networks. The PCL11 allows for communications between any computer pair in the network, a flexibility that would otherwise require a very large number of two-processor links.

Even if your network is starting at three or four processors with potential growth later, the PCL11 offers an advantage: the ability to add more processors to the network by adding additional PCL11 nodes. There is no need to disrupt or reconfigure the existing nodes in the network.

The PCL11 is used to build local networks in a variety of applications. These include distributed processing, distributed database management, industrial data collection and control systems, simulation systems, transaction processing, laboratory data collection and control networks.

Transparency of Operation

The user experiences transparency of operation. The PCL11 hardware manages the protocol and error checking for establishment of communication with a desired recipient computer and for successful transfer of the data or message. The user tells the PCL11 what data is to be sent to which recipient computer. The user is then notified upon successful completion of the transfer. If after multiple attempts the PCL11 is unsuccessful in transferring the data, it informs the user of this with an indication of the reason (receiver busy, no acknowledgment, unresolved errors, and so on).

Error Checking

The PCL11 hardware provides error checking in the hardware by use of work parity and block CRC-16. If an error is detected, the message is automatically retransmitted by the software driver.

High-Availability Systems

The PCL11 is designed for use in high-availability systems. The connection of the PCL11 interface on each computer to the PCL11 bus is such that a computer and interface can be powered on or off without disabling the bus or the rest of the network. The bus is connected to the computer and interface by a T connection, not a daisy-chain. A computer can even be physically unplugged from the PCL11 bus, moved away, and replaced without stopping the network. If a data error occurs, the error is detected by the hardware and the data is automatically retransmitted by the software driver.

Provision is made for backup of the PCL11 unit that is clocking the PCL11 bus. If the PCL11 interface on the network designated to clock and control the PCL11 bus fails, a designated secondary or backup PCL11 unit automatically assumes control. That computer is notified of the action and the user may, by using software, designate a new secondary or backup PCL11 unit. All PCL11 units are identical and any unit can be designated as master or secondary.

High Bandwidth and Flexibility

The PCL11 provides high-bandwidth data transfer rates plus flexibility in the allocation of that bandwidth among the various nodes in the network.

The maximum PCL11 bus bandwidth is 1M bps. There are two mechanisms for dividing the bandwidth among nodes. The default allocation simply divides the bus bandwidth equally among the PCL11 nodes on the bus. That is, the TDM time slices go round-robin among the nodes.

In addition, the allocation of the TDM time slices can be set and varied under software control. The user may load a register in the PCL11 with an explicit list of the time-slice allocations by transmitter number. This can be set to give whatever bandwidth allocation is desired to different PCL11 transmitters in the network to handle differing data rate requirements including a maximum of half of the bandwidth to one node and a minimum of none to another. For example, a large database management processor in a network might be given a greater proportion of the bandwidth since it needs to send large amounts of data to other processors in the network.

For total PCL11 bus lengths in excess of 50 feet, the bandwidth is reduced as a function of the bus length.

Multiple Units and Multiple Buses

Additional flexibility and power are provided by the ability to put multiple PCL11 units on one processor and the ability to implement multiple bus systems.

The use of two or three PCL11 buses to interconnect the processors can give increased reliability through redundancy as well as increased throughput. In the event of a failure disabling one PCL11 bus, the system would continue operation using the other bus. The use of dual bus systems is recommended for medium to large networks.

In addition, a dual or triple bus system gives greater flexibility. For example, one bus could be heavily loaded with long data transfers while another bus is kept lightly loaded and used for command and control messages where quick response is desired. A third bus could be kept in reserve for backup.

A processor can also have more than one PCL11 node on the same PCL11 bus. The use of two nodes from the same bus can be used for processors having large amounts of traffic. The two nodes operate independently and concurrently, thus allowing for greater throughput on that processor.

Building the System

A PCL11 network is a system constructed from several components that are combined to meet the particular application needs.

A PCL11-B is the UNIBUS interface that contains the PCL11 electronics and connects to the PCL11 bus cable. One PCL11-B forms one node on the network. The PCL11-B interfaces are interconnected by cables of appropriate length to suit the site geometry.

Sample Applications

The PCL11 is successfully utilized in many different applications. The following are a few examples.

Several customers implemented various forms of distributed database management and processing systems using the PCL11. Typical of these is one application consisting of several PDP-11/40s, each managing a particular database and performing given functions. User terminals are connected to some of the processors. The application software is written to interpret the inquiry of command from the terminal and, if it is not one performed in that processor, to send the inquiry or command by using the PCL11 to the appropriate processor for servicing. The response is generated and returned back to the originating processor and displayed on the user terminal. Thus, a distributed database is implemented by means of a virtual computer concept.

Although the initial implementation of that system required only three processors, the software was designed in a modular manner for the general case so that as the application grows, additional processors can be added and part of the work load moved to them. The PCL11 facilitates this expansion since all that is required to expand the network is to add another length of PCL11 bus cable and another PCL11 node. There is no need to rewrite the software or to rebuild the existing processors to allow links to the new processor.

In another application, a high-energy nuclear physics research laboratory, a PCL11 network is utilized to control large and complex instrumentation. Different processors are interfaced with certain portions of the facility and perform certain functions. The PCL11 network permits an experimental run to be set up in minutes rather than hours as would otherwise be required. Thus it becomes an important contributor in the research for new forms of energy.

In a very large transaction processing network, 500 on-line terminals are serviced by 16 PDP-11/34 and PDP-11/70 processors acting as communications processors, transaction processors, and database managers. The processors are all interconnected by a dual PCL11 network. The system is designed for high-availability and fail-soft operation. Each processor in the network backs up another processor in the network and receives duplicate data inputs. In the event of a processor failure, the data has been protected and the system automatically reconfigures itself for continued operation without interruption.

APPENDIX C
DEUNA/DMP11/DMV11 Network Examples

DMP11 Block Diagram

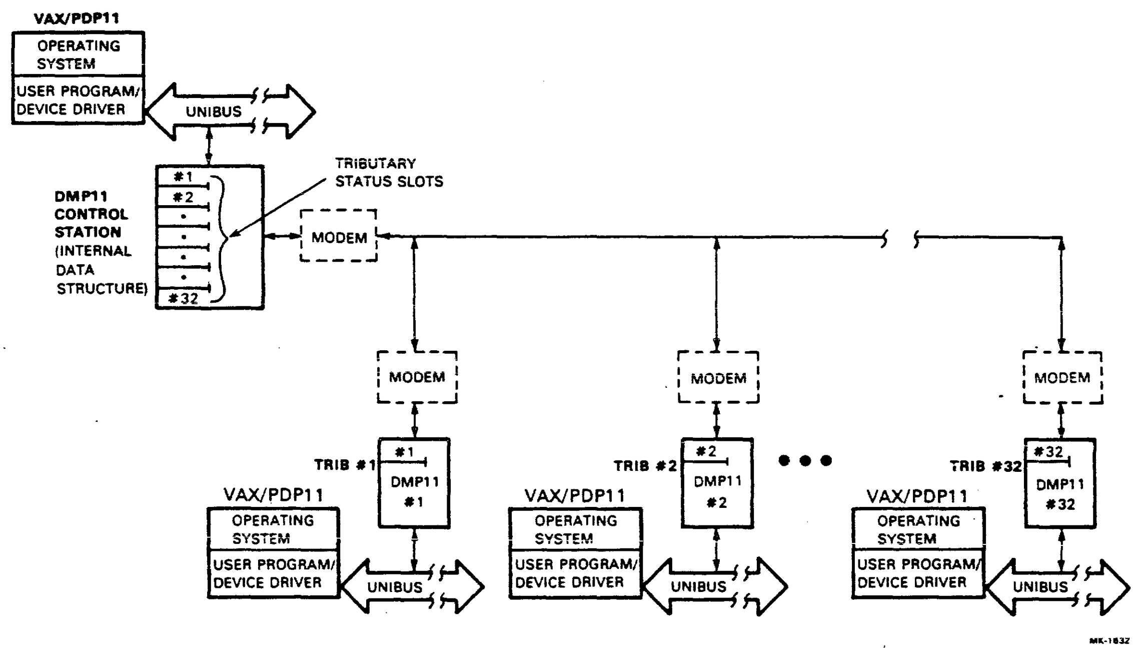


Figure 20 Block Diagram of a Multipoint Network with Single Assigned Addresses

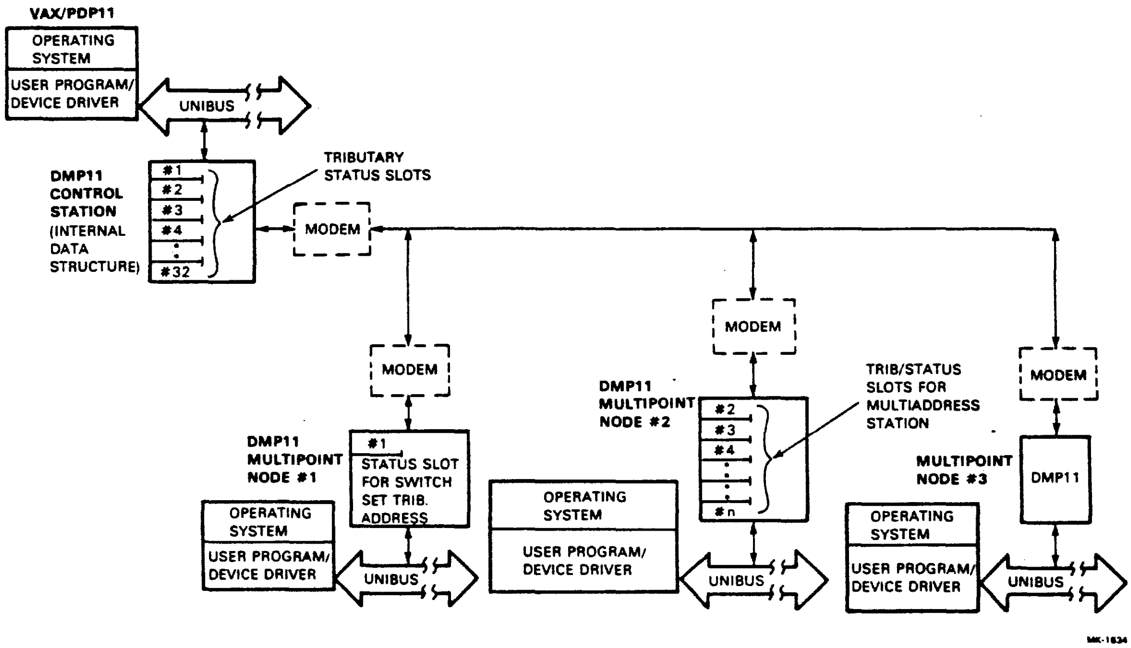


Figure 21 Block Diagram of a Multipoint Network with Multiple Tributary Addresses

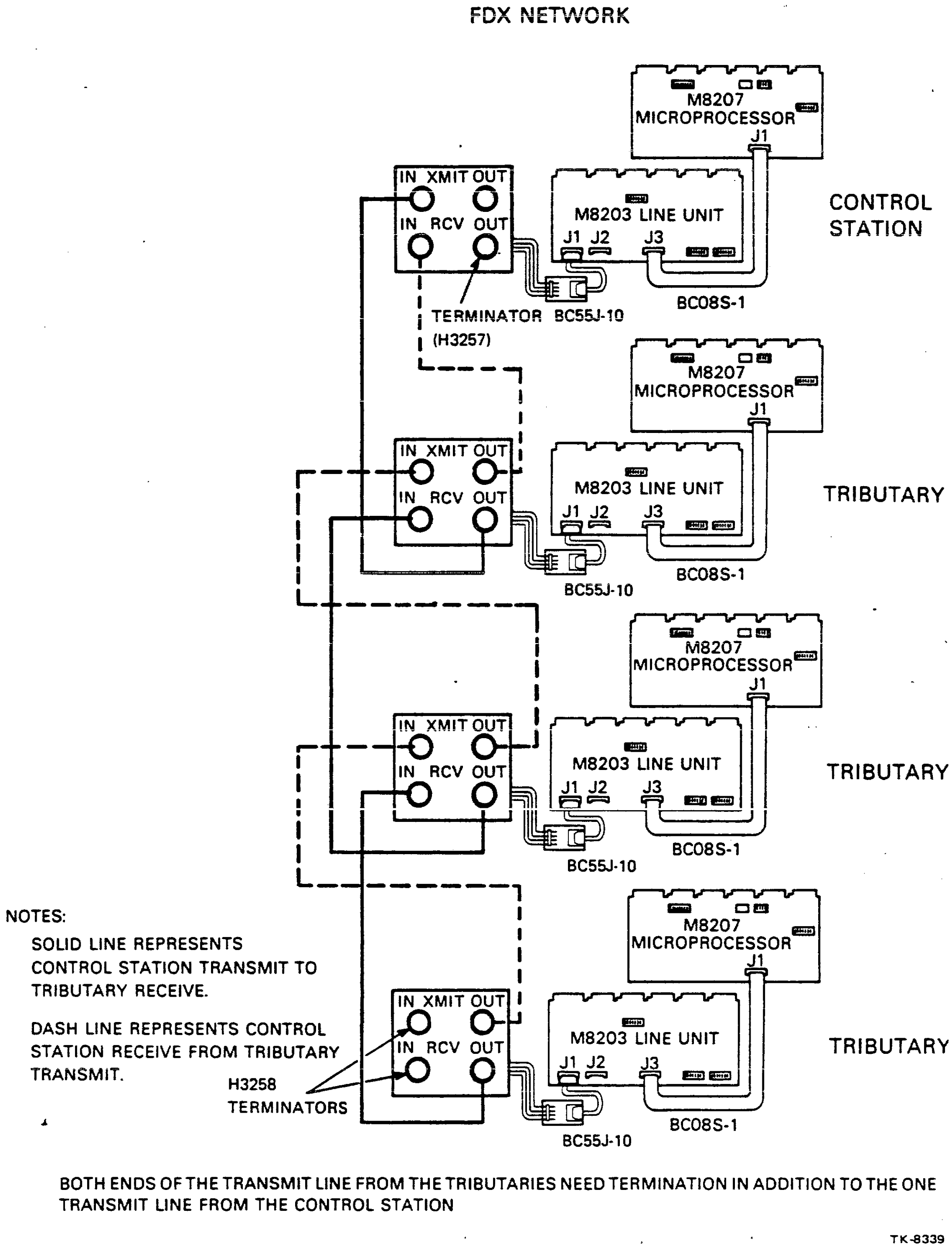


Figure 22 DMP11 Network Example

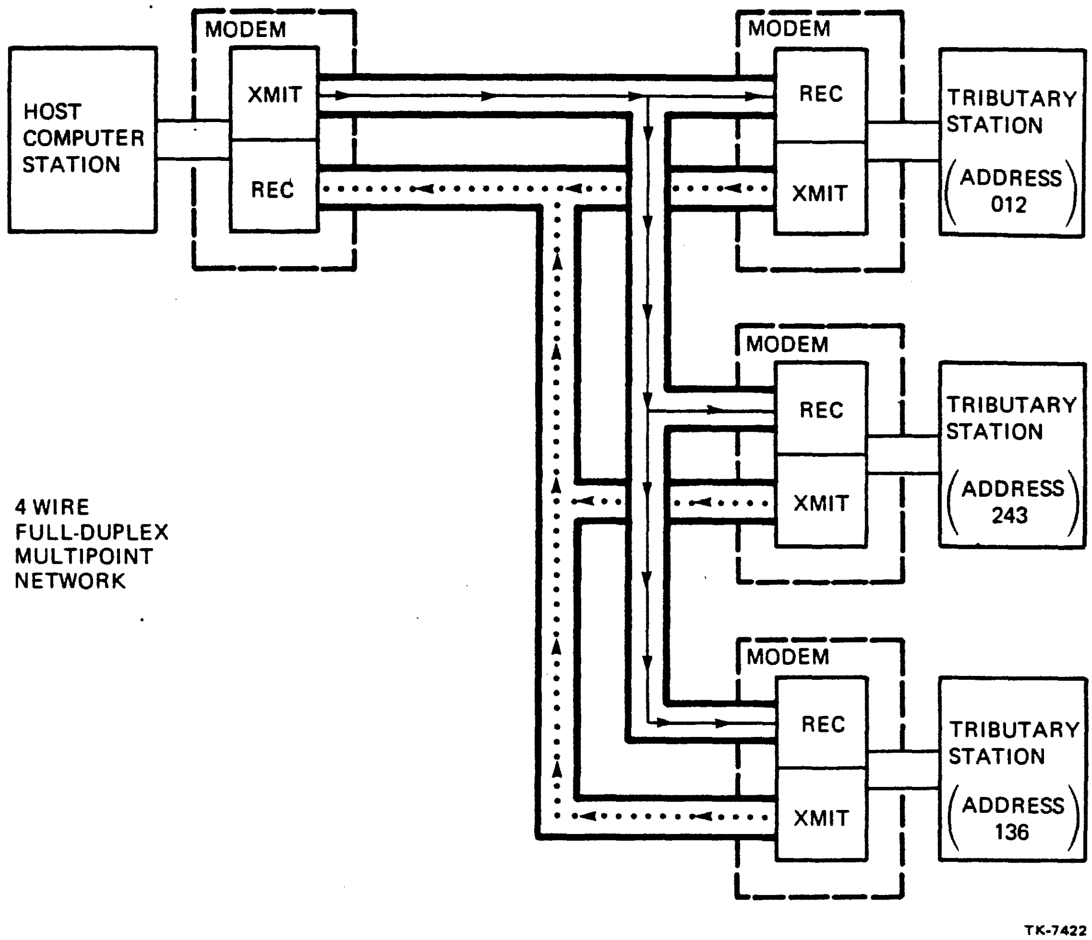


Figure 23 Typical Full-Duplex Multipoint Local Network

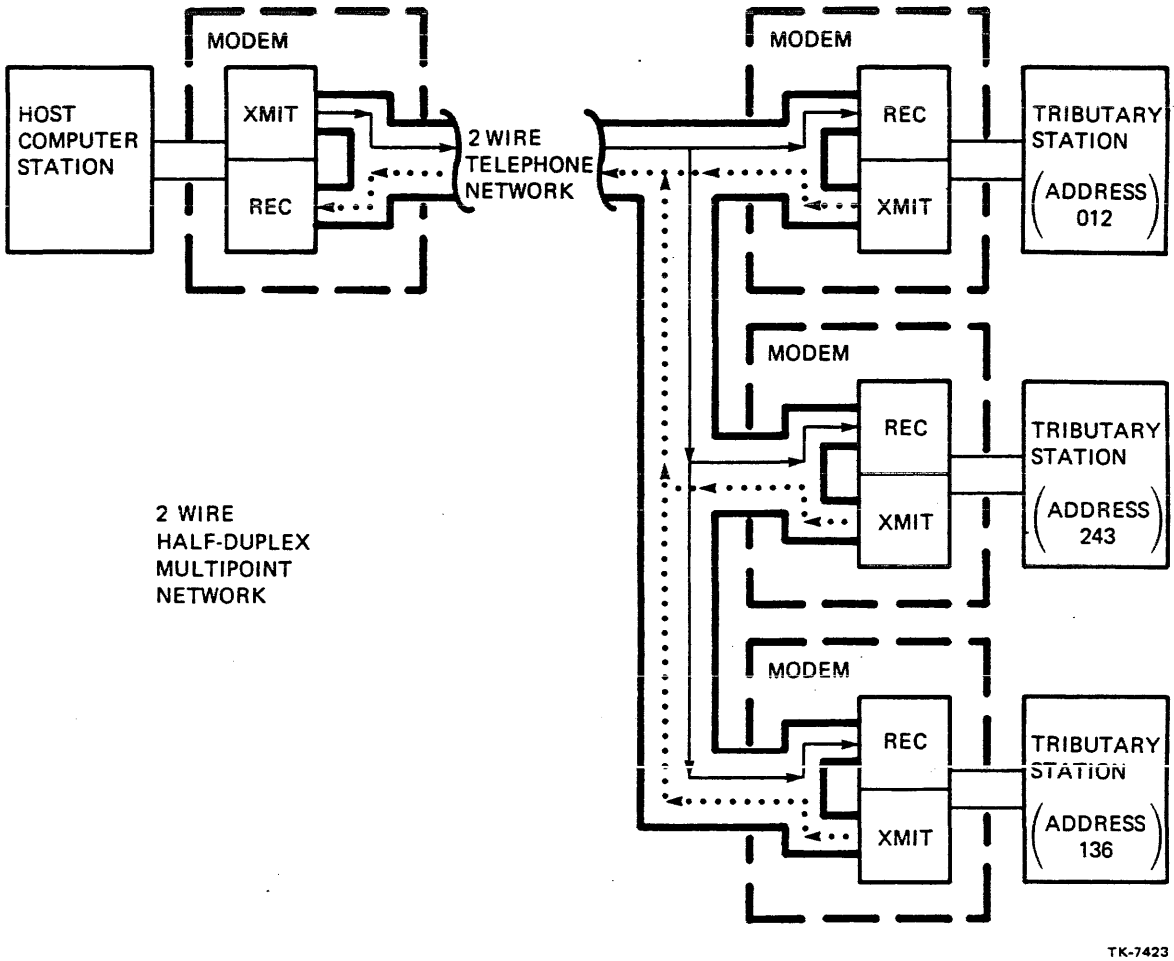


Figure 24 Typical Half-Duplex Multipoint Network

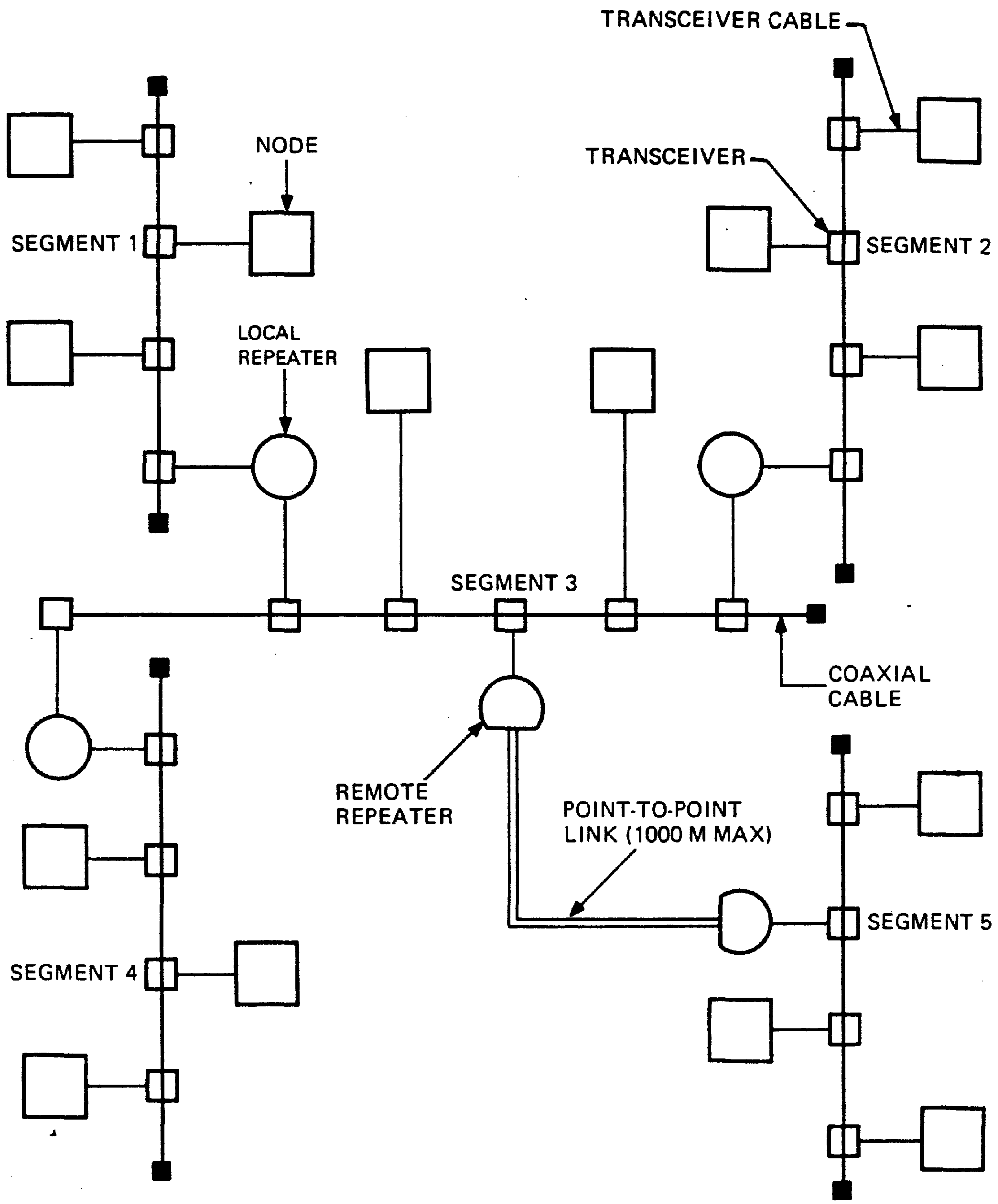


Figure 25 Typical Large-Scale Ethernet Configuration

APPENDIX D
DEUNA/DEQNA Ethernet Communications Controller

DEUNA

The DEUNA is a UNIBUS to Ethernet communications controller. It lets you connect VAX and UNIBUS-based PDP-11 computers to the Ethernet network.

The DEUNA consists of two hex cards, distribution panel, and internal cabling. The DEUNA fits into two option slots on the UNIBUS backplane and connects to the H4000 transceiver using a transceiver cable.

Table 15 DEUNA Features and Benefits

Features	Benefits
32K byte (16K word) buffers in Random Access Memory (RAM)	Provides data transmission and reception and handles maintenance-oriented datagrams
Independent Direct Memory Access (DMA) controller	High performance
32-bit CRC error detection	Ensures data accuracy
Down-line load and remote load detect	Enables start-up of unattended nodes or nodes without local storage
Filters multicast addresses	Reduce host processor load
Gathers network statistics	Aids in network maintenance
Internal microdiagnostics	Minimizes time to isolate and diagnose faults

Backplane Power Checks and Preparation

Perform the following operations on the backplane slots previously selected for DEUNA module installation.

- With system power OFF, conduct resistance checks on the backplane voltage sources to ground to be sure that no short circuit conditions exist.
- Turn system power ON. Verify the backplane voltages are within specified tolerances.
- If present, remove the grant continuity modules.
- If present, remove the Nonprocessor Grant (NPG) jumper wire that runs between backplane pins CA1 and CB1 on the slot selected for installation of the M7792 port module.

NOTE

If the M7792 port module is removed from the system, either replace the NPG jumper wire and install a G727 single-height grant module, or install a G7273 dual-height grant module.

Device Address Assignment

Assign the DEUNA a device address from the Input/Output (I/O) page of memory address space. The first DEUNA being installed in a system must be assigned the address 774510. For the second, and any subsequent DEUNA being installed in the same system, the device address must be selected from the floating address space of the I/O page. The device address is assigned by configuring switch pack E40 on the M7792 port module to the desired address.

First DEUNA Device Address (774510) -- Assign device address 775410 to the first DEUNA being installed in a system by configuring switch pack E40 on the M7792 port module as shown. Note that this address could overlap the twenty-third DP11 if present in the system. Refer to Figure 26 for the location of E40 on the M7792 module.

M7792 - E40

S1	S2	S3	S4	S5	S6	S7	S8	S9	S10
OFF	ON	ON	OFF	ON	OFF	ON	ON	OFF	OFF

NOTE

An OFF (open) switch responds to a logical one (1) on the bus.

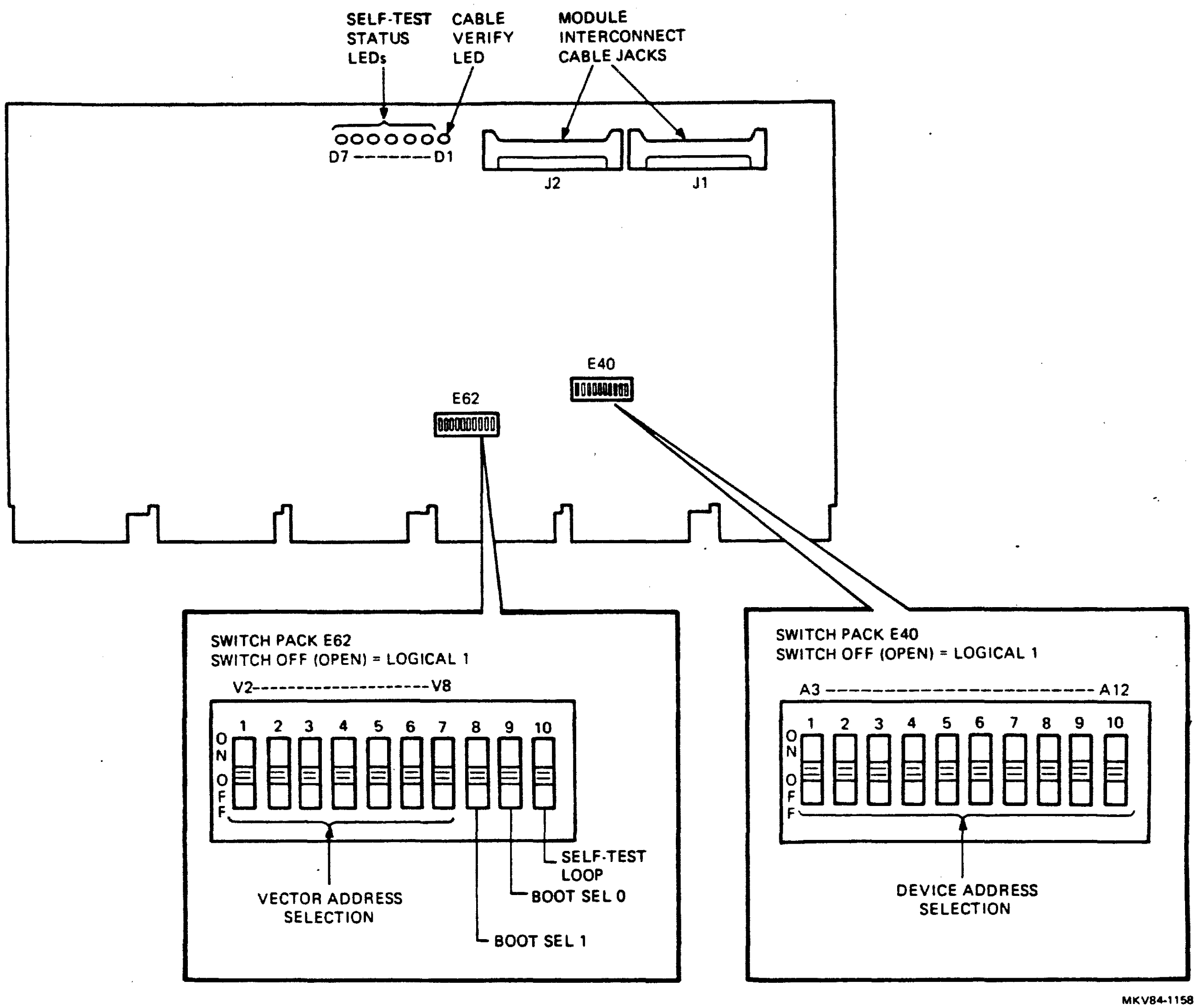


Figure 26 M7792 Port Module Physical Layout

Second DEUNA Device Address (Floating Address) -- Assign a device address to the second (or subsequent) DEUNA being installed in a system by configuring switch pack E40 on the M7792 port module to the desired address determined from the floating address allocation. Refer to Figure 27 for the correlation between switch number and address bit. The ranking device address assignment of the DEUNA is 25.

MSB — LSB

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
1	1	1	SWITCHPACK E40										0	0	0

SWITCH NUMBER	S10	S9	S8	S7	S6	S5	S4	S3	S2	S1	FLOATING ADDRESS
										OFF	760010
									OFF		760020
									OFF	OFF	760030
								OFF			760040
								OFF		OFF	760050
								OFF	OFF		760060
								OFF	OFF	OFF	760070
							OFF				760100

						OFF					760200

						OFF	OFF				760300

					OFF						760400

					OFF		OFF				760500

					OFF	OFF					760600

					OFF	OFF	OFF				760700

				OFF							761000

			OFF								762000

			OFF	OFF							763000

		OFF									764000

NOTE: SWITCH OFF (OPEN) RESPONDS TO LOGICAL ONE ON THE UNIBUS.

TK-9760

Figure 27 Floating Address Assignment

Vector Address Assignment

Assign the DEUNA a vector address from the reserved vector area of memory address space. The first DEUNA being installed in a system must be assigned the vector 120. The second (and any subsequent) DEUNA being installed in the same system must select the vector address from the floating vector area of reserved vector address space. The vector address is assigned by configuring switch pack E62 on the M7792 port module to the desired vector.

First DEUNA Vector Address (120) -- Assign vector address 120 to the first DEUNA in the system by configuring S1-S7 of switch pack E62 on the M7792 port module, as shown. Note that this vector is also used by the XY11. Refer to Figure 26 for the location of E62 on the M7792 module.

M7792 - E62

S1	S2	S3	S4	S5	S6	S7
ON	ON	OFF	ON	OFF	ON	ON

NOTE

An OFF (open) switch produces a logical one (1) on the bus.

Second DEUNA Vector Address (Floating Vector) -- To assign a vector address to the second (or subsequent) DEUNA, configure S1-S7 of switchpack E62 on the M7792 port module to the desired vector determined from the floating vector allocation. Refer to Table 17 for the correlation between switch number and address bit. The ranking vector address assignment of the DEUNA is 47.

15	14	13	12	11	10	09	08	07	06	05	04	03	02	01	00
0	0	0	0	0	0	0	SWITCH PACK E62							0	0

SWITCH NUMBER	S7	S6	S5	S4	S3	S2	S1	FLOATING VECTOR
		OFF	OFF					300
		OFF	OFF				OFF	304
		OFF	OFF			OFF		310
		OFF	OFF			OFF	OFF	314
		OFF	OFF		OFF			320
		OFF	OFF		OFF		OFF	324
		OFF	OFF		OFF	OFF		330
		OFF	OFF		OFF	OFF	OFF	334
		OFF	OFF	OFF				340
		OFF	OFF	OFF			OFF	344
		OFF	OFF	OFF		OFF		350
		OFF	OFF	OFF		OFF	OFF	354
		OFF	OFF	OFF	OFF			360
		OFF	OFF	OFF	OFF		OFF	364
		OFF	OFF	OFF	OFF	OFF		370
		OFF	OFF	OFF	OFF	OFF	OFF	374
	OFF							400

	OFF		OFF					500

	OFF	OFF						600

	OFF	OFF	OFF					700

TK-10019

Figure 28 Floating Vector Assignment

Boot Option Selection (PDP-11 Host Systems Only)

The DEUNA provides for remote booting and down-line loading of PDP-11 family host systems. These functions are switch-selectable by using two boot option select switches located on switch pack E62 on the location of E62 on the M7792 port module.

When installing a DEUNA in a PDP-11 family host system, configure switches S8 and S9 on the switch pack E62 (M7792 module) for the boot function desired. Table 16 lists the switch settings and corresponding boot option functions. Refer to Figure 26 for the location of E62 on the M7792 module.

When installing a DEUNA in a VAX-11 family host system, set both S8 and S9 on E62 (M7792 module) to the ON (disabled) position.

NOTE
An OFF (open) switch produces a logical one (1). This is the enabled state of the switch function.

Table 16 Boot Option Selection (M7792 E62 - S8 & S9)

BOOT SEL 1	BOOT SEL 0	Function
ON*	ON*	Remote boot disabled
OFF	ON	Remote boot with system load
ON	OFF	Remote boot with ROM
OFF	OFF	Remote boot with powerup boot and system load

* Switch settings for a DEUNA installed in a VAX-11 system.

Self-Test Loop (for Manufacturing Use)

The self-test loop is provided on the DEUNA for manufacturing testing. This is a switch-selectable feature that allows the on-board self-test diagnostic program, once it is initiated, to continuously loop on itself. This feature is controlled by S10 on switch pack E62 on the M7792 port module and should be disabled during installation.

When installing a DEUNA, disable the self-test loop feature by setting S10 on switch pack E62 (M7792 module) to the ON (closed) position. Refer to Figure 26 for the location of E62 on the M7792 module.

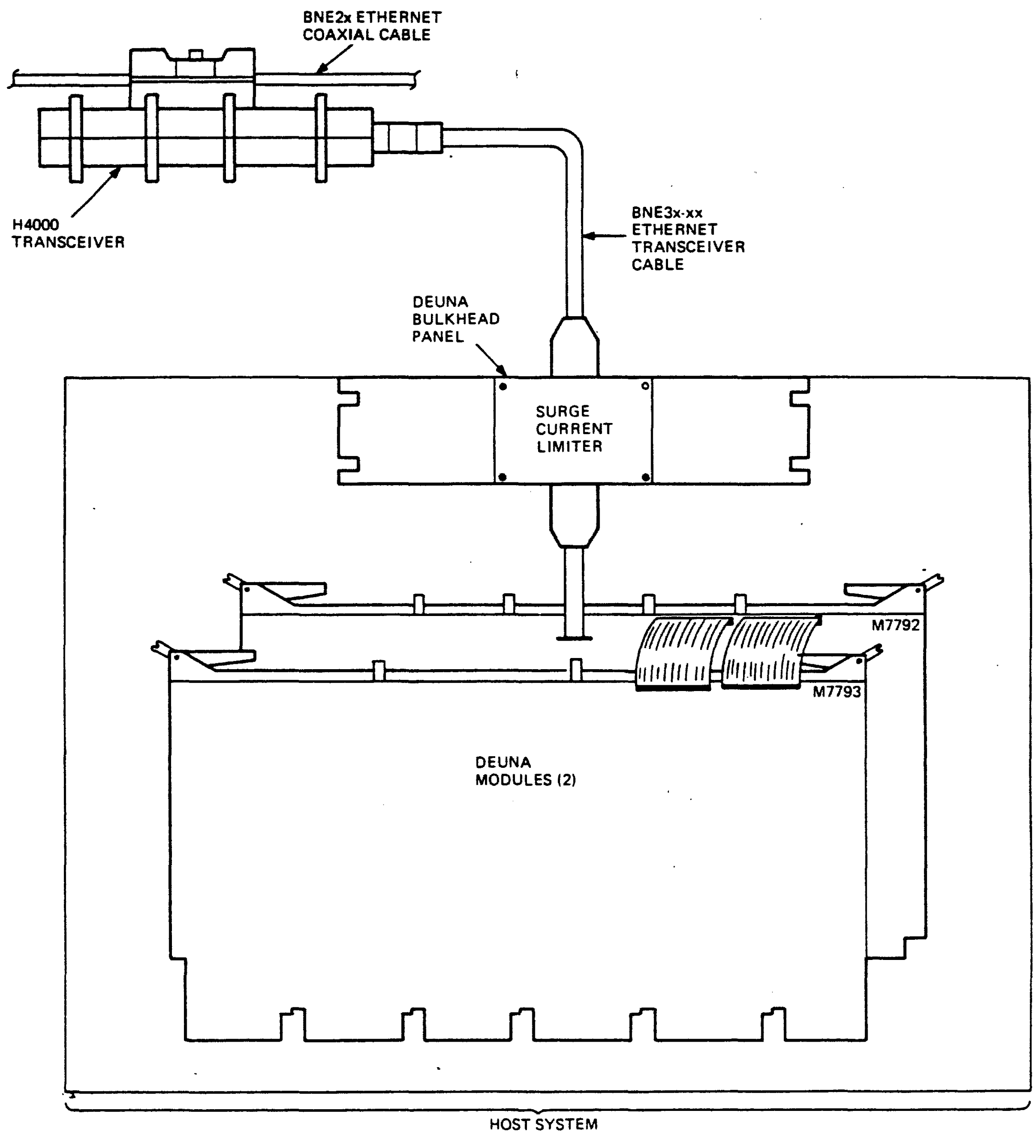


Figure 29 DEUNA to Ethernet Connection

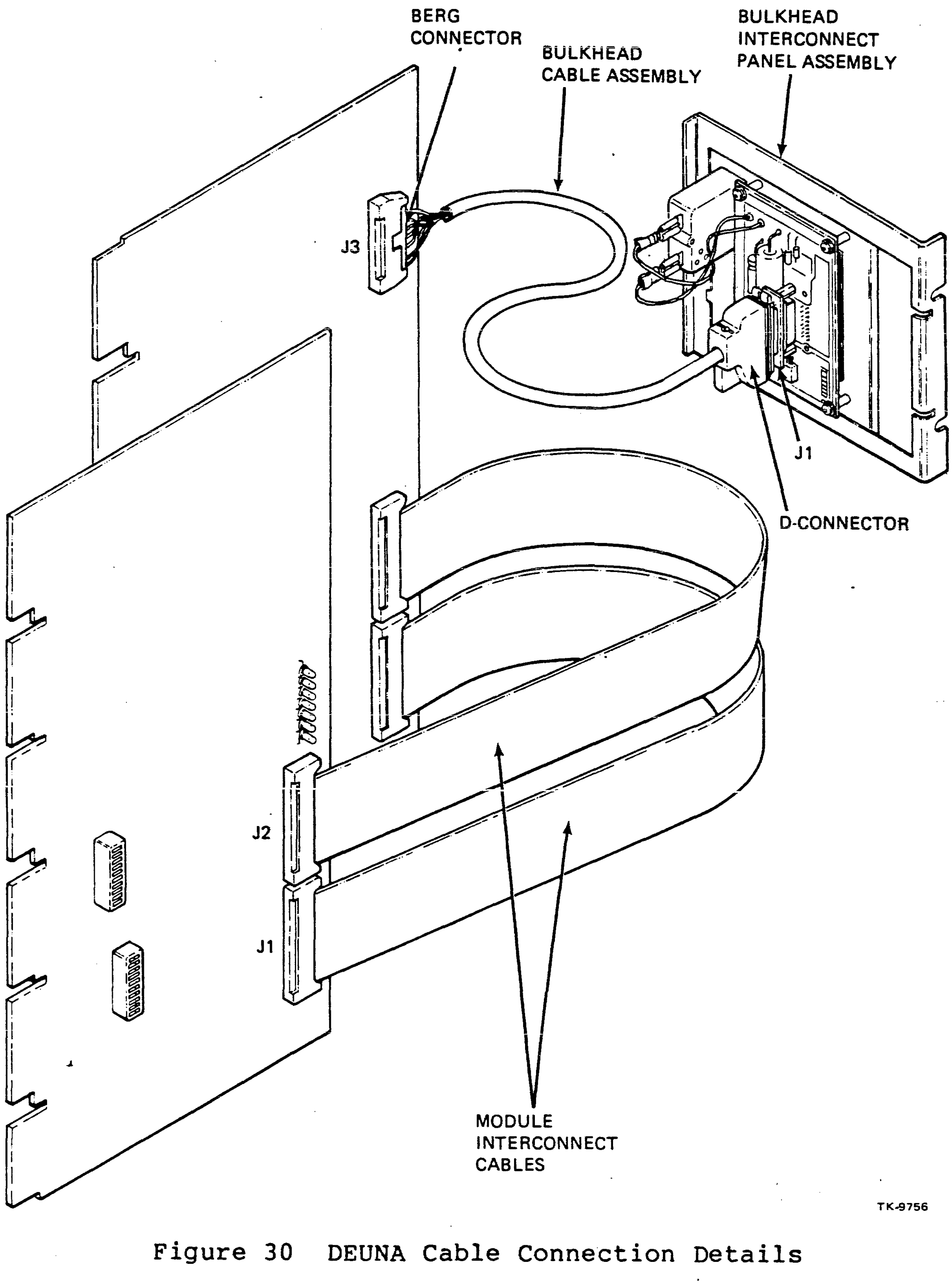


Figure 30 DEUNA Cable Connection Details

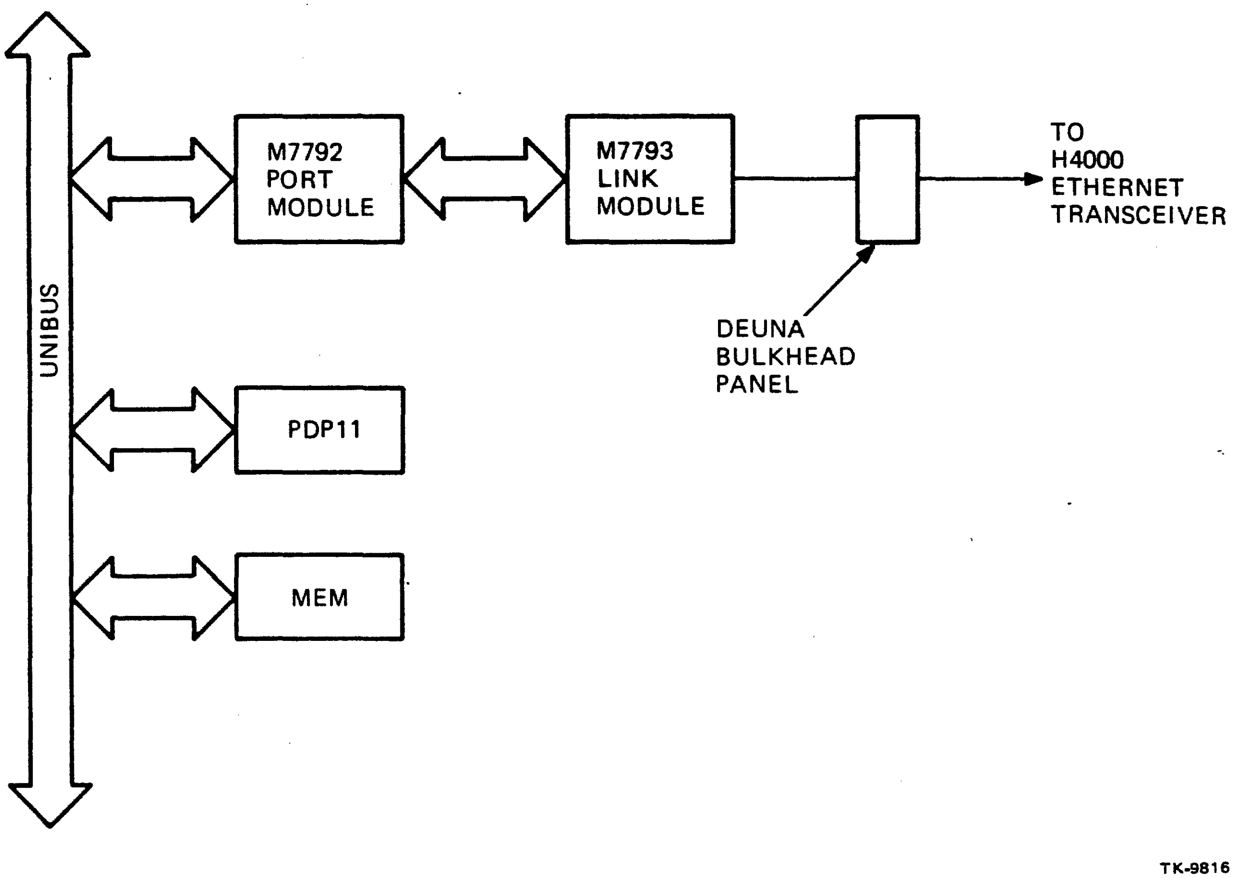


Figure 31 PDP-11 Host System Block Diagram

DEQNA

The DEQNA is a Q-BUS to Ethernet communications controller. It lets you connect Q-BUS systems, such as the LSI-11, to the Ethernet network. The DEQNA provides data link layer and some physical channel functions for connection to Ethernet.

The DEQNA consists of one dual module board plus the appropriate CPU kit that consists of a distribution panel, and internal cabling. The DEQNA controller board fits into an option slot on the Q-BUS backplane and connects to the H4000 Transceiver using a transceiver cable.

The DEQNA has all the features of the DEUNA device. It supports the LSI-11 or Q-BUS based CPU's: PDP-11/03, PDP-11/23A, PDP-11/23-PLUS, PDP-11/23S, and the Micro PDP-11 with DECnet Phase IV software.

EXERCISES

For each of the items find the appropriate definition. Each item may be used more than once or not at all.

1. CETAB.MAC
2. HLDTAB.MAC
3. SLD
4. HLD
5. Multipoint
6. Ethernet
7. Tributary
8. RSX-11S
9. NCP
10. CFE

a. ____ A network topology where there is one line and up to 12 nodes connected to that line.

b. ____ The task that acts as the LDR task on stand-alone DECnet nodes.

c. ____ A network topology where all nodes are considered adjacent.

d. ____ The task that maintains the volatile down-line system loading database.

e. ____ The server task for down-line task loading.

f. ____ The permanent database for task loading.

g. ____ The node that controls polling on a multipoint topology.

SOLUTIONS

For each of the items find the appropriate definition. Each item may be used more than once or not at all.

1. CETAB.MAC
2. HLDTAB.MAC
3. SLD
4. HLD
5. Multipoint
6. Ethernet
7. Tributary
8. RSX-11S
9. NCP
10. CFE

a. 5 A network topology where there is one line and up to 12 nodes connected to that line.

b. 3 The task that acts as the LDR task on stand-alone DECnet nodes.

c. 6 A network topology where all nodes are considered adjacent.

d. 9 The task that maintains the volatile down-line system loading database.

e. 4 The server task for down-line task loading.

f. 2 The permanent database for task loading.

g. none The node that controls polling on a multipoint topology.

LAB EXERCISES

1. Set up a node in your volatile database that can be down-line system loaded to a designated satellite computer with an available RSX-11S operating system.

2. Down-line system load the satellite using the database created in Lab Exercise 1.

SOLUTIONS TO LAB EXERCISES

1. See Example 1

2. NCP LOAD NODE REMSYS VIA UNA-0
   OR
   NCP TRIGGER NODE REMSYS

CONTENTS

FIGURES

TABLES

INTRODUCTION

DECnet-RSX is implemented as a layered product on top of the RSX-11 operating system. The Communication Executive (CEX) shields the DECnet software from the RSX-11 Executive. CEX manages communication processes, tasks, and buffers for DECnet. An end user accesses the communication software by issuing QIO calls to pseudodrivers (NS: for logical link, NW: for PSI, and X: for DLX).

Communication processes are divided into three categories:

- Logical Link Control (LLC)
- Data Link Control (DLC)
- Device Driver Module (DDM)

Processes communicate with each other using communication control blocks, which are small data structures allocated out of the Dynamic Storage Region (DSR). Communication control blocks contain information on the status of a current data packet that DECnet software processes; for example, received or transmitted, which process owns the packet (ECL, XPT, or DMC), whether the packet comes from LLC to DLC to DDM and so on.

Pseudodrivers (NS:, NW:, and NX:) are implemented as communication processes, and some of them, particularly NS: and NW:, have corresponding ACP tasks (NETACP for NS: and X25ACP for NW:). ACPs essentially process most complicated requests; for example, NETACP contains a session control mechanism used for creating the logical link; X25ACP processes the interrupt calls on a virtual circuit. One may view the NS:-NETACP (and NW:-X25ACP) pair as a team that processes fast network requests in the driver and slow network requests in the ACP. So SND/XMT and REC/RCV data over the existing logical links/virtual circuits are processed in the NS:/NW: because those requests are logically faster than CON/CAL, for example.

CEX contains a module, Communications Executive Common (CEXCM), that is essentially a set of global pointers to the DECnet software. CEXCM also points to the Configuration Database (CETAB) that reflects a particular node configuration (types of lines, processes generated by NETGEN and so on).

CEX, communication processes, buffers, and CETAB are loaded to the system by the Network Loader (NTL) task that reads the CETAB.MAC file. The standard RSX-11 loader is not used.

This module covers major components, data structures, and data flow for the DECnet-RSX software.

OBJECTIVES

Upon completion of this module, the specialist should be able to:

- Specify major components of the DECnet-RSX software.
- Identify the functions of the major components and data structures.
- Understand the general data flow in the DECnet-RSX software.

RESOURCES

1. DECnet-RSX Network Generation and Installation Guide
2. DECnet-RSX Overview
3. DECnet-RSX Release Notes

COMMUNICATIONS EXECUTIVE (CEX)

- Set of modules that help communication processes run.
- For /M-PLUS, it resides in the CEXBF module of the RSX executive (code starts at $CEX::). For /M/S, it resides in DSR (in CEXPAR by convention).
- It provides:
  - Interprocess communication routines -- To allow processes to communicate with one another
  - Process scheduling routines -- To allow processes to schedule other processes for execution
  - Buffer management routines -- To allocate/deallocate main memory buffers from available buffer pool
  - Timer support routines -- To provide processes with logical timers
  - Modem control routines -- To support modem handling for lower-level processes (resident in AUX process)
  - Set of common routines -- To be used by communications processes

COMMUNICATIONS PROCESSES

- Logical modules of the DNA
- Event-driven sequential processes (with the exception of handling interrupts from the hardware)
- Processes can be scheduled by the CEX and, if necessary, suspended with their context saved
- Processes are not reentrant and do not share databases
- Processes communicate with one another in the following way:
  - A sending process creates an interprocess communication message.
  - Such a message is placed by the CEX in the destination process' queue.
  - The destination process is scheduled for execution by the CEX.
- Processes are channel-oriented
  - A channel is an entity that a given process performs a specific function on (for example, for a device handler, a channel is generally one device, or possibly many identical devices; for ECL, a channel is a logical link).
- A process performs certain functions on a channel, and if there is more than one channel, the process handles each channel identically but independently from all other channels
- The implications of channel-oriented communications
  - Most processes have a separate database for each channel they handle. Each database is identical in format to the databases of all other channels handled by the same process.
  - When a process is scheduled, it is generally requested to perform a service for only one of the channels that it handles. That is, each element or interprocess communication message that drives a process generally requests a service on only one channel.

- Types of Processes

  - Logical Link Control (LLC)

    AUX - Auxiliary Process

    EVL - Event Logger

    ECL - End Communications

    XPT - Routing Process

    DLX - Direct Line Access

    NW - X.25 Access Process

    PLI - Packet Level Interface

  - Data Link Control (DLC)

    DCP - DDCMP Process ie. DUP

    LAB - LAB Protocol Process

    EPM - Ethernet Protocol Manager ie. UNA

  - Device Driver Module (DDM)

    DMC - DMC-11, DMR-11 Device Driver

    DMP - DMP-11 Device Driver

    DUP - DUP-11 Device Driver

    PCL - PCL-11 Device Driver

    DLM - Data Link Mapping Driver

- Each process is less than 4 Kw and is mapped to by KAPR5 when required.

- Specific processes can be included in or excluded from the running system if needed.

- On the PAR display or crash dump, the processes are shown as dynamic subpartitions in a form of NT.XXX, where XXX is a process name (for example, NT.AUX for AUX; NT.XPT for XPT).

LLC Processes

- Auxiliary Process (AUX)
  - Non-time-critical extension of the CEX
  - Functions

    Buffer allocation failure recovery

    Powerfail recovery

    Modem control

    Timer services

    Fork processing and process dispatch

    Allocation/deallocation of buffers in network pool (POOL..)

- Event Logging Process (EVL)
  - Creates records of network events
  - Dispatches the events to local or sink nodes

- End Communications Layer (ECL)
  - Accepts QIO requests from user tasks for logical links (NS: driver)
  - Processes these requests (SND$, REC$, XMI$) or gives them to NETACP for processing (all other calls)
  - Contains time-critical code for logical link processing
  - Implements the NSP protocol

- Routing Layer (XPT)
  - Contains time-critical code for routing processing (contains a copy of Reachability Vector and Adjacency Block and a copy of Minhop/Mincost vector) on routing type nodes.
  - Manages communication between DLCs and LLCs
  - Implements the routing protocol
- Direct Line Access Process (DLX)
  - Accepts and processes QIO requests from user tasks for DLX access (NX: driver)
- NW: Driver for X.25 Access Process (NW)
  - Accepts QIO requests from user tasks for X.25 access
  - Processes these requests (XMT$, RCV$) or gives them to X25ACP for processing (all other calls)
- Packet Level Interface (Level 3 protocol) Process (PLI)
  - Implements the X.25 Level 3 protocol
  - Processes requests from NW: and X25ACP originated by user QIOs
  - Provides for transmissions and reception of data packets and controls virtual circuit processing (multiplexing virtual circuits on a physical line, allocation/deallocation of logical channel numbers)

DLC Process

- Provides sequential, error-free data transmission and reception on the physical link
- Primary functions
  - Starts/stops the physical line
  - Provides the frame-level protocol for data to be transmitted and supervising the transmission of data over the line
  - Decodes the frame-level protocol in supervising the reception of data over the line
  - Examines/changes line characteristics (line services)
  - Provides special conditions notification to higher-level processes
- DDCMP Control Process (DCP)
  - Implements DDCMP for devices other than those that can provide microprocessor-based DDCMP support
  - Contains the polling algorithm for multipoint operations on devices other than the PCL11-B, DMP11 and DMV11.
- LAP-B Protocol Handler (LAB)
  - Implements the X.25 Level 2 protocol
  - Required for DUP11 and DPV11 if used for PSI environment
  - Transmits and receives data for PLI
  - Supports the trace function
- Ethernet Protocol Manager (EPM)
  - Implements the Ethernet protocol
  - Required for DEUNA or DEQNA devices

DDM Process (Except DLM)

- I/O handler for a communication device(s)
- Executes as a result of either a device interrupt or an I/O request from a DLC
- Primary functions
  - Starting and stopping the device
  - Giving the device buffers to receive and transmit
  - Retrieving/setting device characteristics
  - Terminating the current operation (transmit or receive) on the device
  - Providing special notifications to higher-level processes

Data Link Mapping Process (DLM)

- Pseudo DLC/DDM process that allows for DECnet data to be sent over the X.25 network
- Implementation as an interface between the Routing and PLI
- DECnet views the DLM as a process that drives a multipoint line (up to 64 stations per controller, up to 4 controllers)
- ECL adjusts its segment size to the PLI packet size, so that a long message may be sent as a number of PLI packets
- A CRC-16 field is added to ensure error-free transmission over X.25 circuits

MAJOR TASKS

Network Loader (NTL)

- Performs loading and unloading network components by reading the CETAB.MAC file
- Allocates/deallocates process-related data structures

KMC Microcode Loader (KMCL)

- Loads KMC microcode during system initialization for only 11-S systems or after powerfail (NTL loads microcode upon system initialization for M, M-PLUS)
- Used for KDP, KDZ, and KMX devices

General Microcode Loader (MDL)

- Loads Ethernet devices with proper microcode
- Required for DEUNA and DEQNA devices

Network Initializer (NTINIT)

- Performs initialization functions on the node when the network is turned on: SET SYS, SET EXEC STATE ON, and SET MOD X25-SERVER STATE ON

Event Collector

- Server task for the EVL process
- Dispatches event records to a file, terminal, or user-written monitor task
- Transmits the events to a specified sink node by using a logical link to Event Receiver Task (EVR)

DECnet Network ACP (NETACP)

- Works in conjunction with NS: (ECL)
- Processes user QIO requests from NS: for logical link (other than (SND$, REC$, XMI$) that are processed by the NS: itself)
- Accomplishes incoming calls by requesting tasks for them
- Notifies user tasks when network events occur
- Contains session control routines

Routing Control Processor (RCP)

- Performs routing protocol functions for the DECnet functions for the DECnet routing type node
- Maintains the routing database
- Monitors circuits for down-line load, up-line dump, and circuit loopback tests

X.25 Network ACP (X25ACP)

- Works in conjunction with NW: and PLI
- Accomplishes connect and disconnect calls as well as incoming calls for user tasks
- Notifies user tasks when network events occur

Network Control Program (NCP)

- User interface to the Network Management Layer
- Provides parsing functions for user QIO requests before giving them to the local NM: driver or remote NICE task (by using logical link)
- Contains dispatch tables for processing the commands, thus coordinating the commands' execution

Network Management Device Driver (NMDRV) (NM:)

- Provides initial checking for given NCP or NICE requests
- Gives valid requests to NMVACP for processing

Network Management Volatile ACP (NMVACP)

- Executes NCP or NICE commands per requests from NM:
- Makes changes to volatile database only

$ add 1 more $

Network Information and Control Executor (NICE) (NIC$$ task)

- Serves network management requests from remote nodes (Object 19)

Event Receiver (EVR)

- Receives events from remote nodes and dispatches them to a terminal, a file, or a user-written monitor task (MON... by default)

Other Tasks

- A set of privileged and nonprivileged tasks that are not as vital for the DECnet running as specified previously.
- They perform user interface functions for specific applications (for example, file transfer, down-line task loading, remote terminal facility).
- They use primarily the logical link facility so they can be grouped into pairs.

BUFFERS

- Communications Control Blocks (CCBs)
  - Obtained from the remaining space of CEXPAR, if available, or DSR
  - Used for interprocess communication
- Network Buffer Area (POOL..)
  - Contains various buffers and some of the data structures for DECnet and PSI
  - Allocated dynamically on network loading
- Small Data Buffers (SDBs)
  - Used for protocol or task control messages
  - Obtained from the network buffer area (POOL..)
- Large/Receive Data Buffers (LDBs/RDBs)
  - Obtained from POOL..
  - System Buffers for data communication over a network
  - Each LDB/RDB is pointed to by a corresponding CCB
  - Certain number of RDBs are reserved only for receiving data (receive threshold - congestion control)

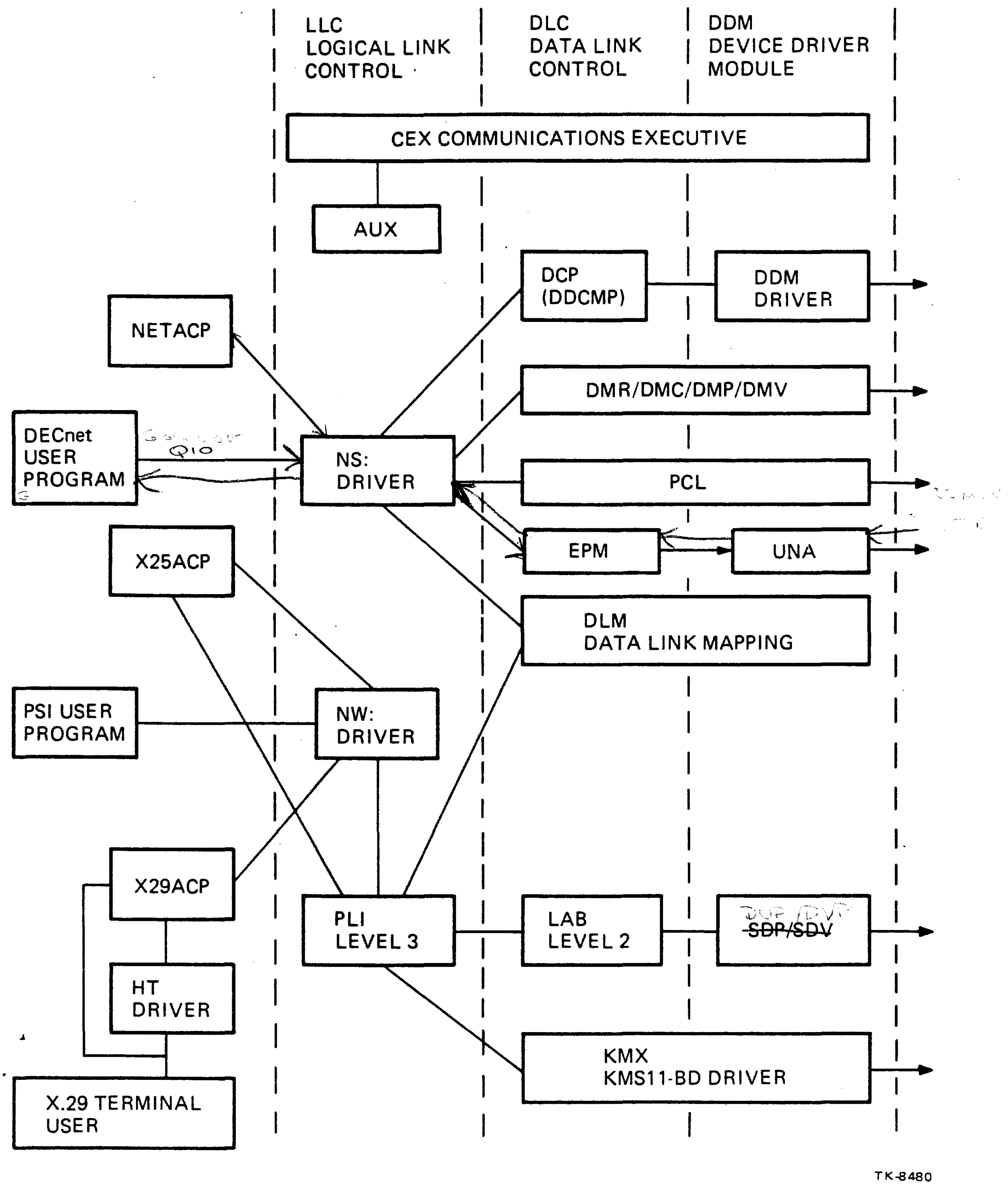


Figure 1 Integration of Network Components

RSX/NETWORK INTERFACE

- Network programming is implemented as a standard RSX QIO interface:
  - User task issues QIO to a device driver by specifying Logical Unit Number (LUN) for the driver in the I/O call
  - I/O packet created by DRQIO routine in RSX executive is given to the driver or ACP based on function mask in the driver's DCB
  - Control returns to the user tasks by using $DIRXT:: in SYSXT module of the RSX executive
- The following pseudodevice drivers support network I/O operations:
  - NS: driver -- Provides I/O interface for logical link facility; it is implemented in the ECL process
  - NW: driver -- Provides I/O interface for PSI facility; it is implemented in the NW process
  - NX: driver -- Provides I/O interface for direct line access facility; it is implemented in the DLX process
- These drivers contain standard RSX-like entry dispatch tables. They have associated with them DCBs, UCBs, SCBs, and code, but they contain no interrupt service routines because they do not drive any hardware. (Hardware interrupts for communication devices are handled by the corresponding DDM processes: DMC, DUP, DL, UNA.)
- Such a structure allows shielding of the network software from the RSX executive.
- DCBs, UCBs, and SCBs for NS:, NW:, and NX: are created by the NTL task on loading the network. The standard RSX LOAD command is not used.

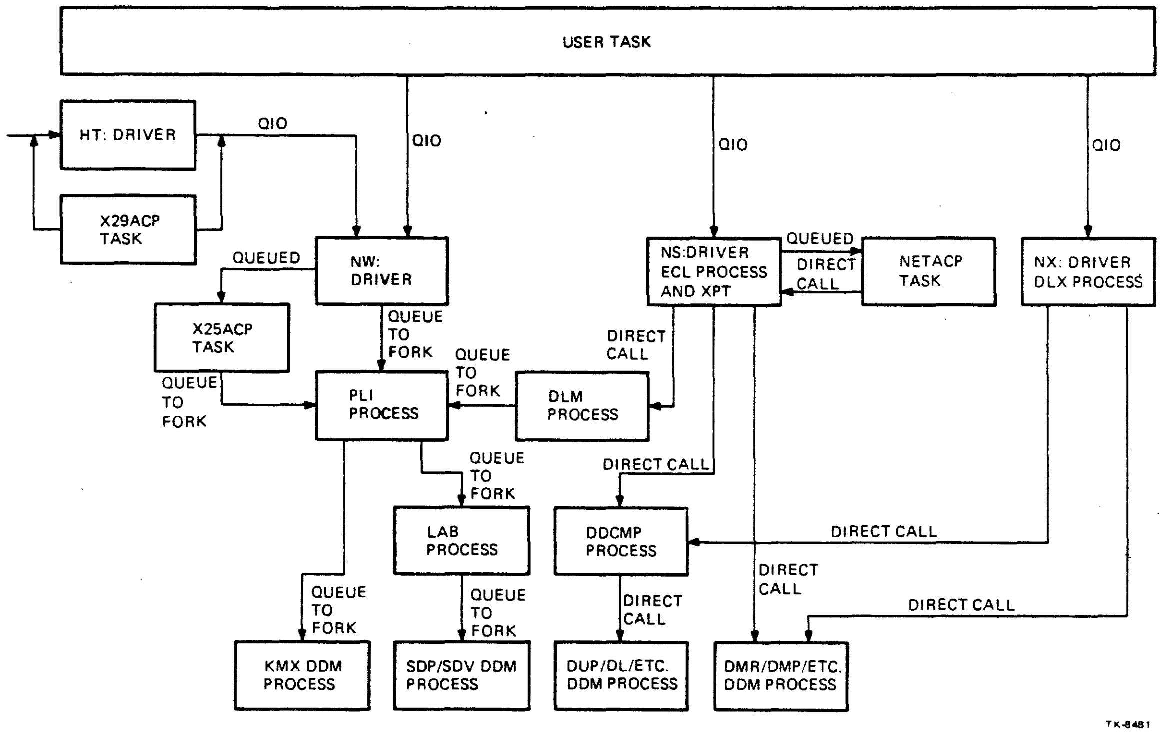


Figure 2 General Data Flow

Notes on Figure 2

1. Direct call means that the current process gives a CCB to the next process to be run for that CCB and schedules that process for execution through the CEX.

2. Queued means that the I/O packet is given to ACP by using the ACP's receive queue in the task's header.

3. Queued to fork means that a CCB is queued by the current process to the end of the network fork queue and it will be dispatched to the next process by the network fork processing routines (resident in the AUX process). Network fork processing routines are activated by the RSX executive (SYSXT) the same way as any standard fork processing routines.

4. All incoming messages are dispatched by the network fork processing routines (see item 3 for details).

DETAILS OF THE RSX-11/NETWORK INTERFACE

- The QIO directive is accomplished by:

  - $EMTRP in DRDSP
  - $DIRSV in SYSXT
  - $DRQIO in DRQIO

  The I/O packet is either passed to the ACP or passed to the driver's initialization routine. If passed to the ACP, the ACP's Stop bit is unset.

  Control is then returned to the user task by using $DIRXT in SYSXT.

- The determination of whether a function goes to the ACP or the driver is done in DRQIO by examining the function code mask of the driver in the corresponding DCB.

Device Control Block (DCB)

POINTER TO NEXT DCB		0	D.LINK
POINTER TO FIRST UCB OF THIS DEVICE TYPE		2	D.UCB
DEVICE NAME (IN ASCII)		4	D.NAM
HIGHEST UNIT NUMBER | LOWEST UNIT NUMBER		6	D.UNIT
LENGTH OF EACH UCB OF THIS DEVICE TYPE		10	D.UCBL
POINTER TO DRIVER DISPATCH TABLE		12	D.DSP
LEGAL FUNCTION MASK	FOR FUNCTIONS 0 – 17	14	D.MSK
CONTROL FUNCTION MASK		16	
NO-OP FUNCTION MASK		20	
ACP FUNCTION MASK		22	
	FOR FUNCTIONS 20 – 37	24	
		26	
		30	
		32	
PCB ADDRESS OF LOADABLE DRIVER		34	D.PCB

TK-5272

Figure 3 DCB Offset Definitions

Device Function Mask for NS:

	For Functions 20-37	For Functions 0-17
	(Obtained from system crash dump)	
Legal	037000	000001
Network control Disconnect from process Connect to process Receive message process Transmit message process		Cancel I/O request
Control	020000	000001
No-op	000000	000000
ACP	017000	000000
Network ACP functions	No Files-11 ACP functions	Non-ACP functions

Cancel I/O request required by the device for aborting the task.

Device Function Mask for NW:

	For Functions 20-37	For Functions 0-17
	(Obtained from system crash dump)	
Legal	037000	000001
Network control Disconnect from process Connect to process Receive message process Transmit message process		Cancel I/O request
Control	037000	000001
No-op	000000	000000
ACP	000000	000000
Network ACP functions	No Files-11 ACP functions	Non-ACP functions

Cancel I/O request required by the device for aborting the task.

Device Function Mask for NX:

	For Functions 20-37	For Functions 0-17
	(Obtained from system crash dump)	
Legal	077000	000200
Illegal		
Network control		Close out LUN
Disconnect from process		
Connect to process		
Receive message process		
Transmit message process		
Control	030000	000200
No-op	000000	000000
ACP	000000	000000
Network ACP functions	No Files-11 ACP functions	Non-ACP functions

- All the QIO calls are given initially to the drivers (NS:, NW:, and NX:) for initial checking. Then the driver may request ACP processing for an I/O packet.
  - QIO calls to NS: that do not require NETACP processing are SND$, REC$ and XMI$
  - QIO calls to NW: that do not require X25ACP processing are XMT$ and RCV$
  - QIO calls to NX: do not require ACP processing at all

- I/O function codes for the network pseudodrivers
  - NS:

SND/XMI	(FC=31)
REC	(FC=32)
CON/ACC	(FC=33)
DIS/ABT/REJ	(FC=34)
OPN/CLS/SPA/GND	(FC=35)

  - NW:

XMT/RES/INT/ICF	(FC=31)
RCV	(FC=32)
CAL/CAC/APV	(FC=33)
CRJ/CTR/CLR/DPV	(FC=34)
OPX/CLX/SPX/GNX/TDN	(FC=35)

  - NX:

IO.XOP	(FC=33)
IO.XCL	(FC=34)
IO.XIN	(FC=35)
IO.XTM	(FC=31)
IO.XRC	(FC=32)
IO.XHG	(FC=36)

- I/O flow

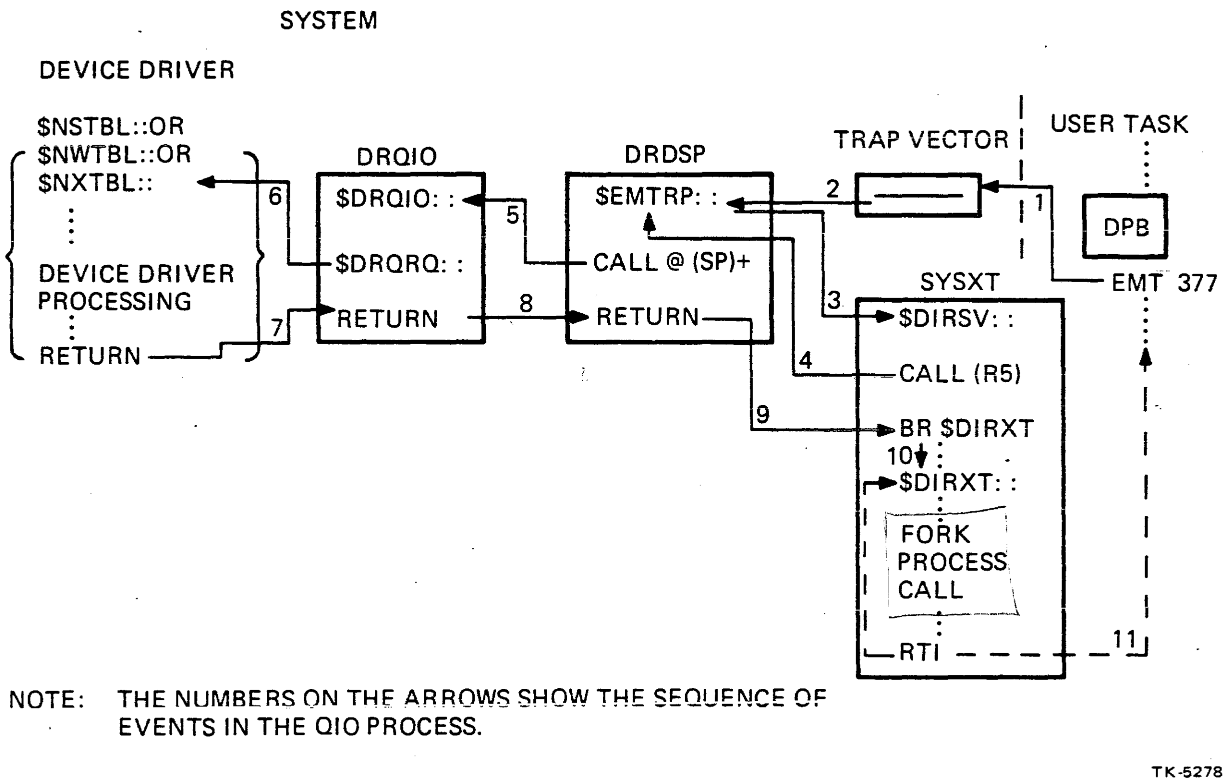


Figure 4 I/O Flow from the User Task to the Device Driver

NETWORK DATA STRUCTURES

- All the DECnet data structures are described in NETLIB.MLB and can be retrieved by using LBR utility:

```
LBR TI:=DM0:[130,10]NETLIB.MLB/EX:xxxxx
```

- For PSI data structures, use PSI.MLB:

```
LBR TI:=DM0:[1,1]PSI.MLB/EX:xxxxx
```

where

DM0: is a device for the network distribution

xxxxx is the name of the macro for a specific structure

- There are several databases in the system that point to all other data structures
  - SYSCM -- RSX executive database (module in RSX executive)
  - CEXCM -- Communications Executive database (module in CEX). For DECnet-11M-PLUS, the module's name is NETCM
  - DDB -- ECL database (allocated from DSR on loading the network)
  - SESDAT -- NETACP database (module SESDAT in NETACP)
  - PHB -- PSI home block (allocated from DSR on loading the network)
  - PLI database (allocated from DSR on loading the network)
  - X25ACP database (module in ACPROD for X25ACP)
- The following databases can be found from NETLIB.MLB or PSI.MLB
  - DDB -- NETLIB.MLB, macro ECDDB$
  - PHB -- PSI.MLB, macro PHBDF$
  - PLI database -- PSI.MLB, macro PLIDF$
- The remaining databases can be retrieved from the corresponding source listings

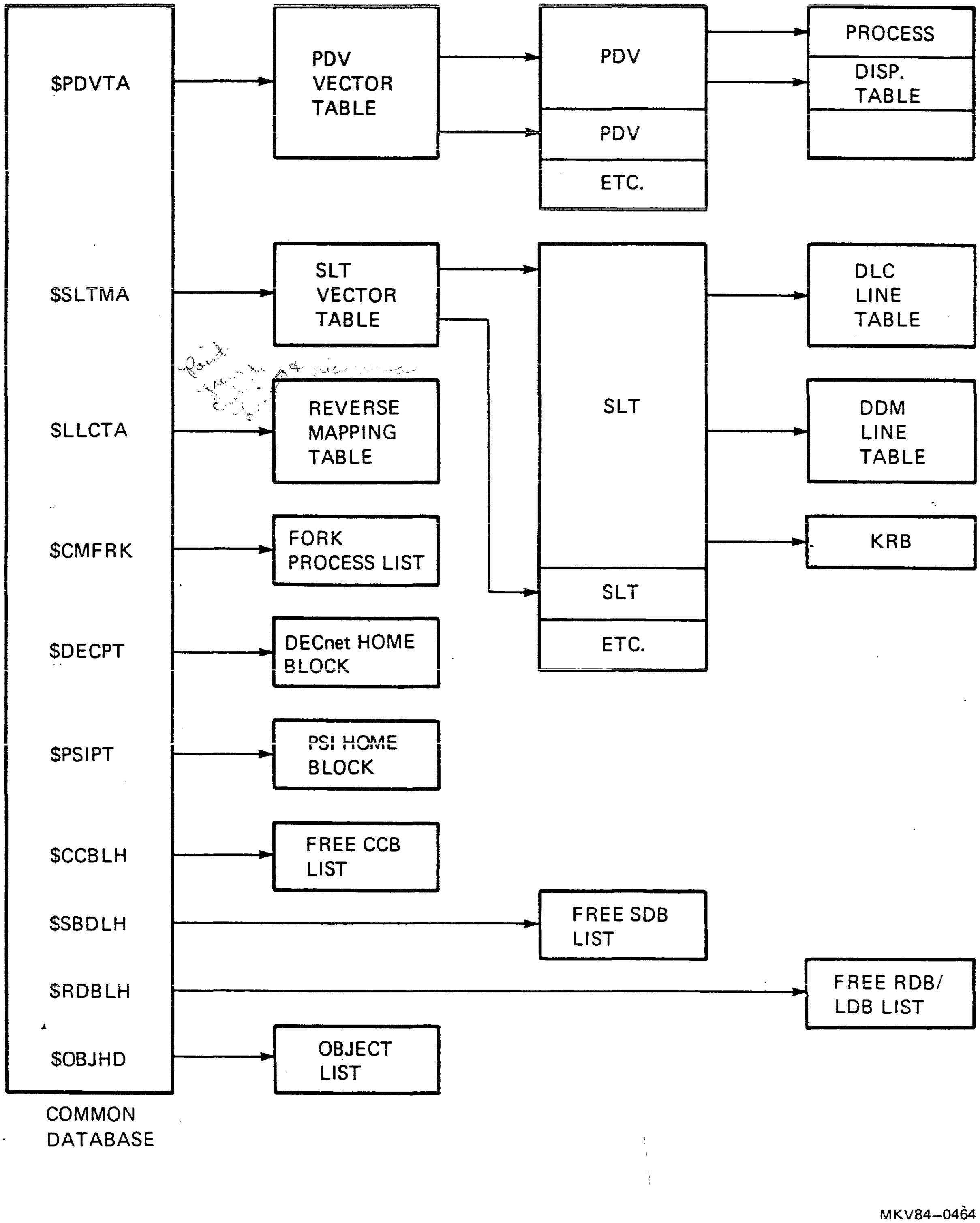


Figure 5 CEX Database (CEXCM) - CETAB

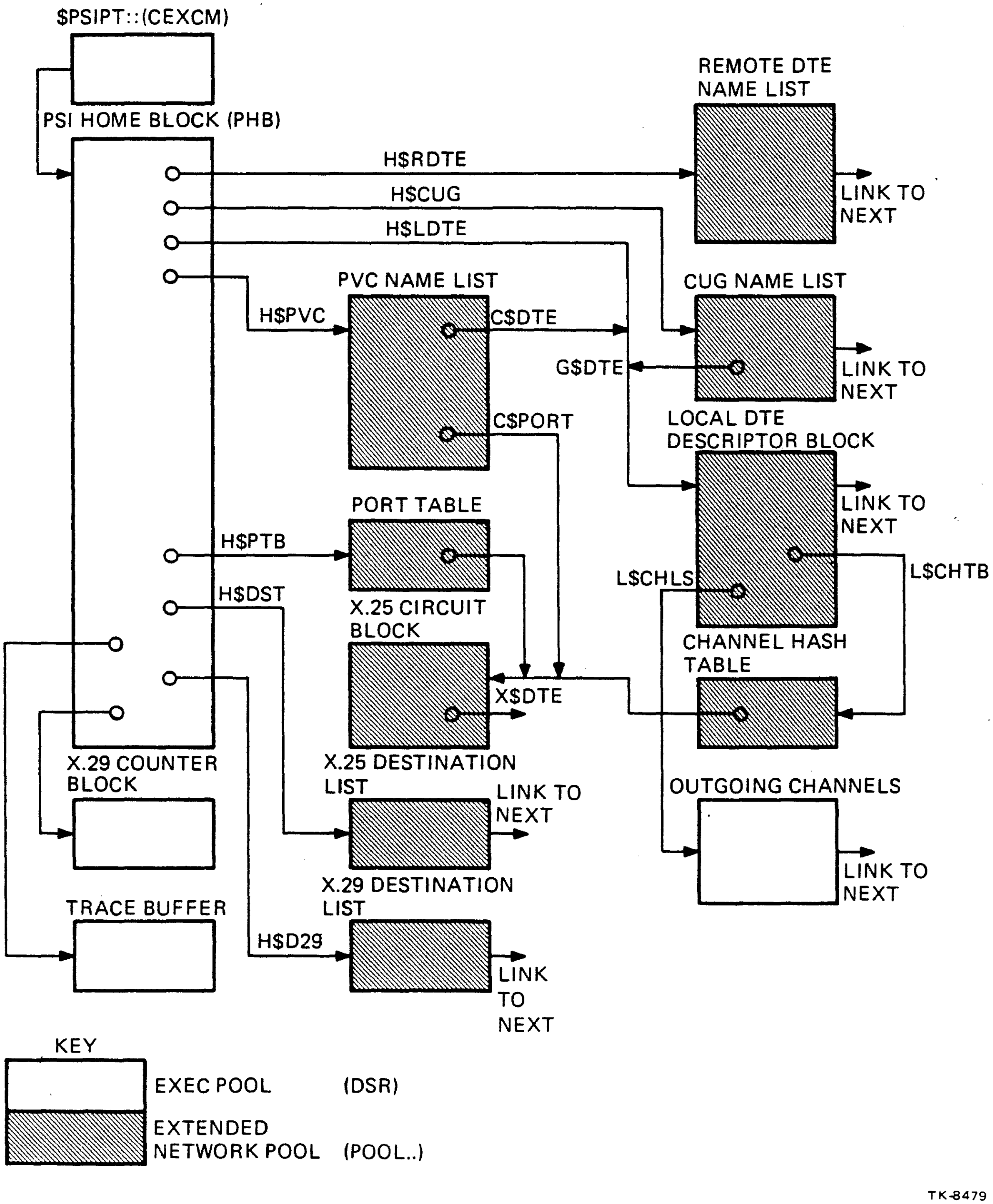


Figure 6 Major RSX-11 PSI Data Structures

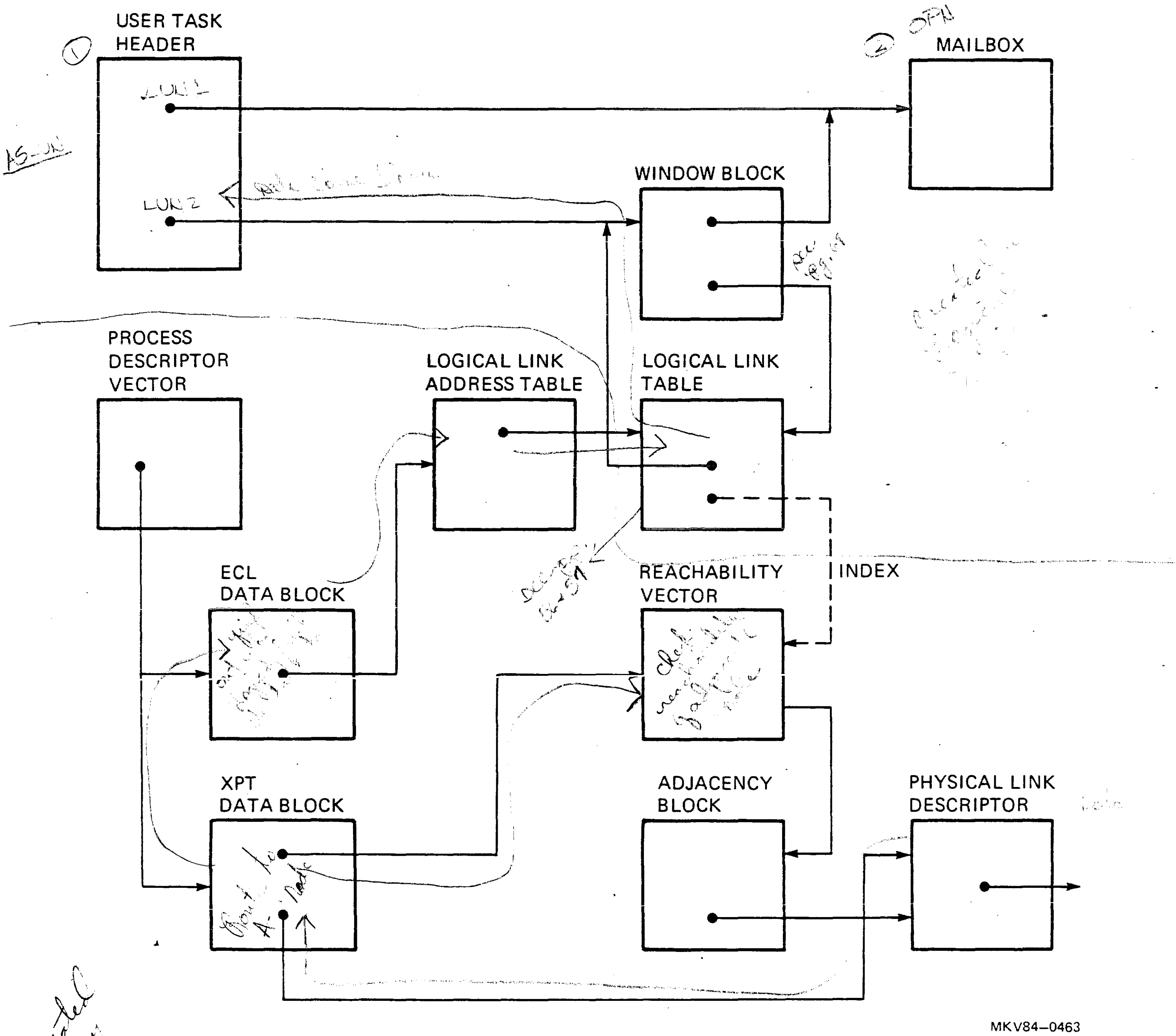


Figure 7 General Picture of Logical Link Operations

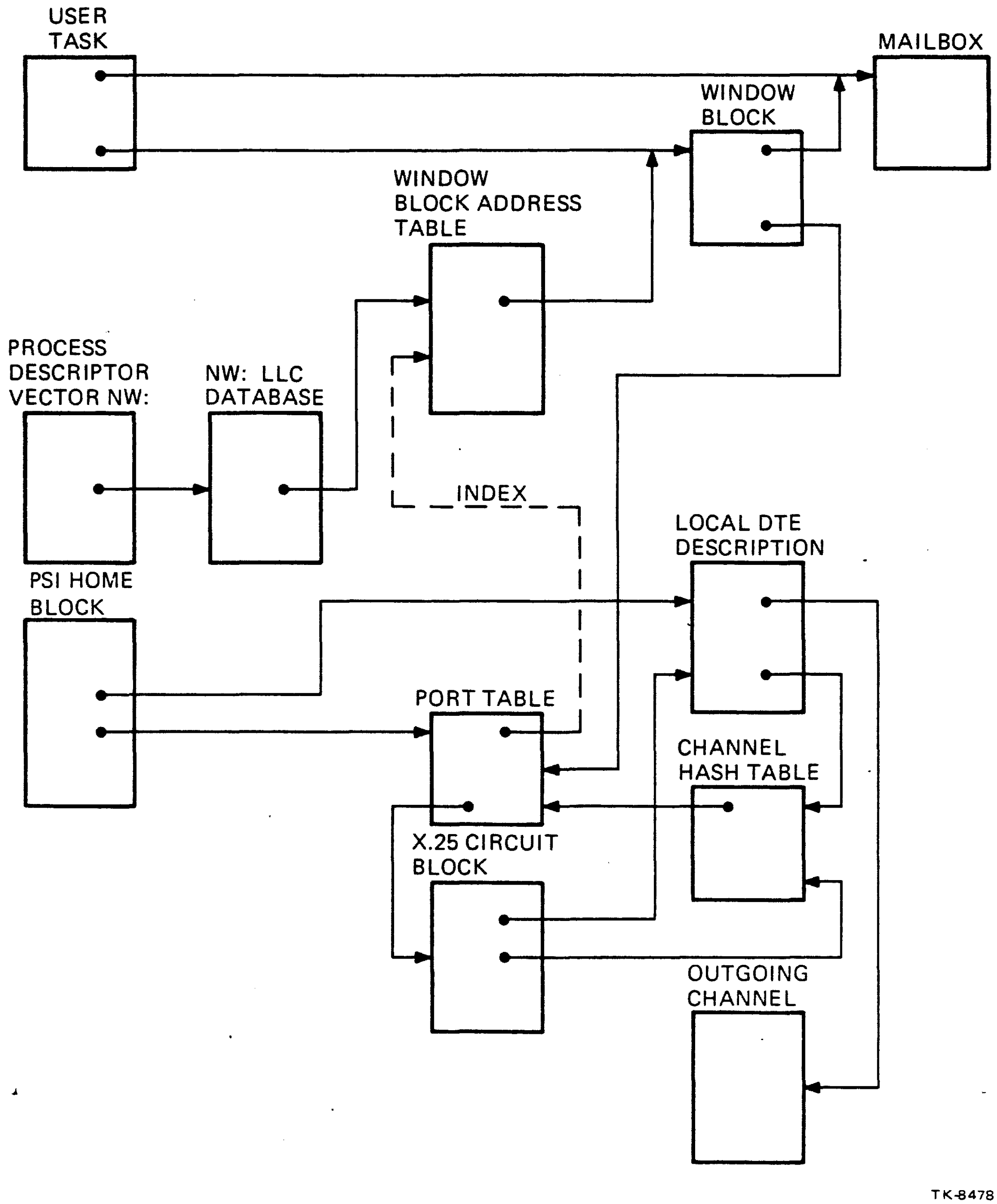


Figure 8 General Picture of Virtual Circuit Operations

Appendix A contains details of connections between various data structures.

Appendix B contains a summary of the data structures for DECnet and PSI as well as offset definitions.

DETAILED SYSTEM FLOW

Interprocess Communication

In general, one communication process communicates with another by sending it a CCB. The CEX contains routines that place the address of a specified CCB in the queue that drives a particular communication process. These routines may also dispatch the destination process immediately or queue it for subsequent execution.

In addition to the use of CCBs for interprocess communication, the CEX contains routines that allow one process to send the contents of a single register to another process for selected special functions.

Process Scheduling

A major function of any executive, including the CEX, is to provide mechanisms for scheduling the orderly execution of processes. Communications processes running in the environment provided by the CEX and the RSX-11 operating system can schedule other such processes for execution using the following mechanisms:

- At the operating system's fork level through the fork-level or clock queues

- Immediately, through a process dispatch operation

Fork-Level Process Scheduling

The CEX provides routines to allow communication processes to append a CCB (chain) to a CEX-internal fork process CCB list and then indirectly dispatch to a target process by placing the CEX process queue server routine in the operating system's fork queue (beginning of the list). The CEX process queue server then sequentially dispatches those processes that are to receive CCBs in the process queue when the operating system next decides to process its fork queue. That decision usually occurs between the time RSX-11 completes interrupt processing and the time it switches the processor back to the user state.

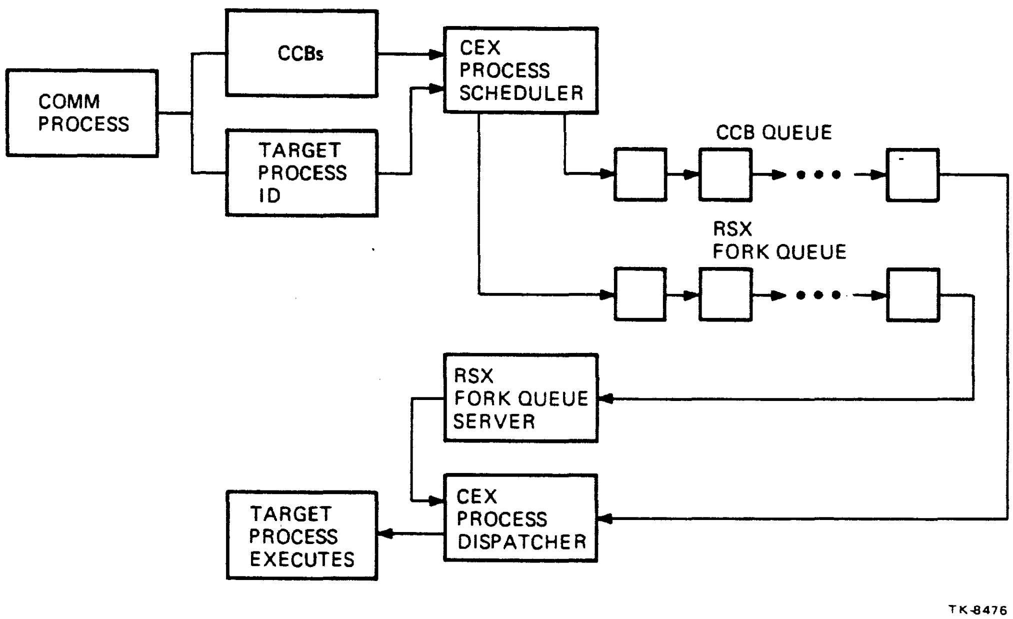


Figure 9 Fork-Level Process Scheduling

TIMER SUPPORT

Timer Process Scheduling

The CEX provides timer support functions for any processes that require them. There are two separate types of logical timers managed by the CEX:

- 100-millisecond resolution (one-shot) timers -- Used by processes to cause single events to occur or to dispatch to other processes, one time only, at the end of a relatively short time period.

- One-second resolution (repetitive) timer -- Used by processes to schedule events and dispatch to other processes on a periodic (one second) basis once a relatively long time period has elapsed.

100-Millisecond Timers

Processes requesting a 100-millisecond timer calls a CEX routine to place a logical timer call in a list kept by the CEX. This four-word cell (see CEX routine $STSTM) contains the number of 100-millisecond units to initialize the timer and the process to execute when the timer runs out. The CEX posts a 100-millisecond timer request to the operating system so that the CEX timer processor is queued for fork-level execution once the 100-millisecond period has ended. When executed, this timer-processing routine decrements each timer in the CEX queue and executes the process specified in the timer queue entry when the timer has expired. The timer entry is then deleted from the list.

One-Second Timers

The CEX cooperates with the operating system in another way to provide processes with one-second timers. When the CEX is loaded and initialized, the Network Loader (NTL) posts a one-second timer request to the operating system with the CEX one-second timer processor as the timeout routine (which is executed at fork level). This CEX timer processor always posts another timer request to the operating system and is therefore self-propagating.

Although one-second timers are requested by processes differently than 100-millisecond timers, they are handled by the CEX in a similar fashion. Each time it is executed, the one-second timer processor scans the system line table for all active lines that require timer support and services them one at a time. If a line qualifies, the timer processor executes the line's DDM and DLC (sequentially if separate processes) if their timers have expired on the current scan; otherwise the timers are decremented by one second. Note that expired system line timers do not cause the repetitive scheduling of their DDM/DLC processes.

After servicing the system lines, the timer processor scans the Process Descriptor Vector (PDV) table for processes that require timer support and execute those that request it. This type of support involves no timer cells and is repetitive as long as the process timer support flag bit is set.

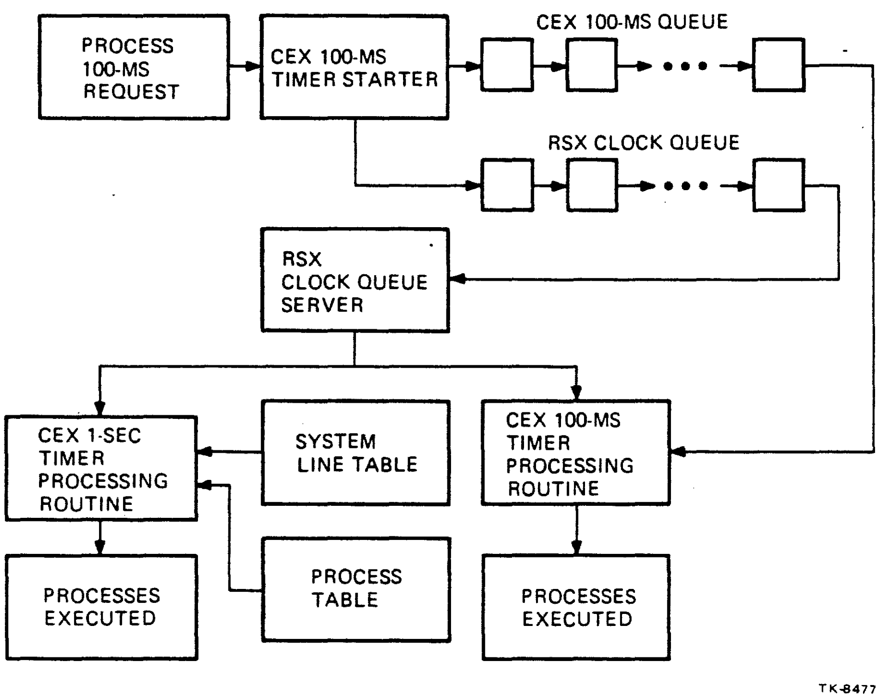


Figure 10 Time-Dependent Process Scheduling

Modem Control Services

Another facility that the CEX provides is a general modem control service for both synchronous and asynchronous lines. This service is transparent to communications processes and removes the burden of modem control and modem/line state handling from them. Hence processes need only be concerned with the following line-related control functions:

- Wait for ring
- Enable line
- Disable line

When a process sends one of these requests to another process (DLC to DDM, for example) by utilizing the CEX process dispatching routines, the CEX delays the dispatch long enough to perform some modem control bookkeeping.

In addition, the CEX monitors all active lines at one-second intervals by dispatching to their DDM processes' "modem sense" entry points. Each of those DDMs returns modem status information to the CEX, which uses that data to update its own modem control table. If the status (state) of the modem changes since the last monitoring interval ended, the CEX executes an internal modem control routine to service the new modem state.

This service usually involves the notification of the line's DLC process in situations such as ring detects, line failures, and the disabling of inactive lines.

Mapped System Support

When the CEX starts a process in a mapped system, it guarantees that memory is mapped as in Figure 11.

This mapping structure imposes a limitation on the size of an individual communication process to 4K words. For those situations where a process needs more address space, CEX routines exist to provide a simple overlay capability to the process. With this capability, the extra address space can reside in auxiliary 4K processes that can be overlaid on the original process' virtual address space. The overlaid process context (mapping) is saved and restored after the auxiliary process exits. Since the system stack is used to save the context, the overlay process may be nested several times.

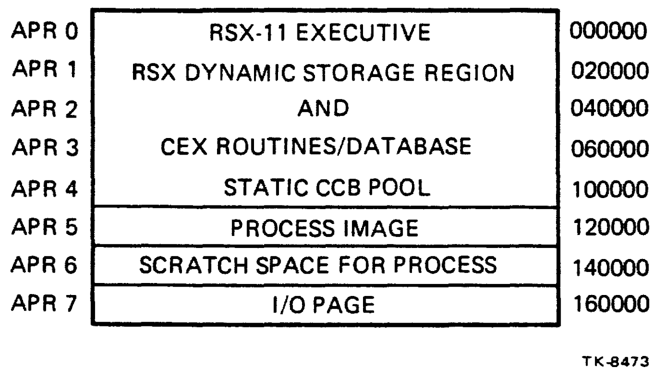


Figure 11 Process Address Space Mapping

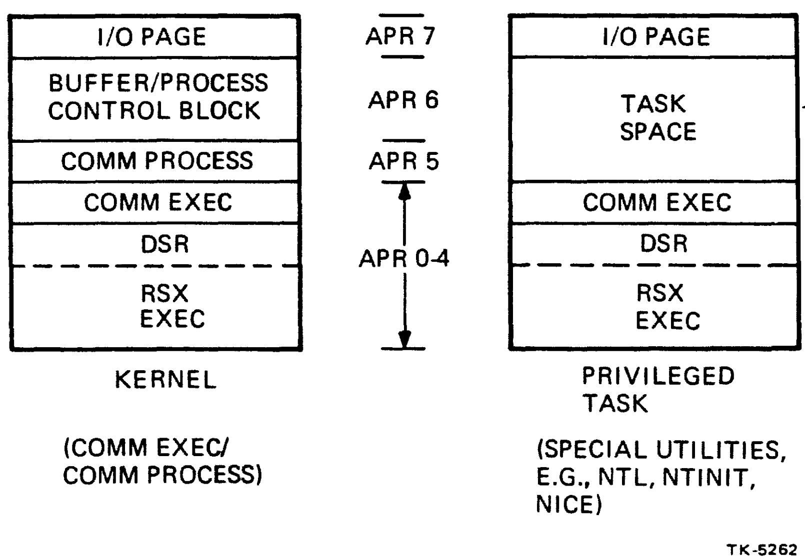


Figure 12 RSX-11 Mapping for DECnet

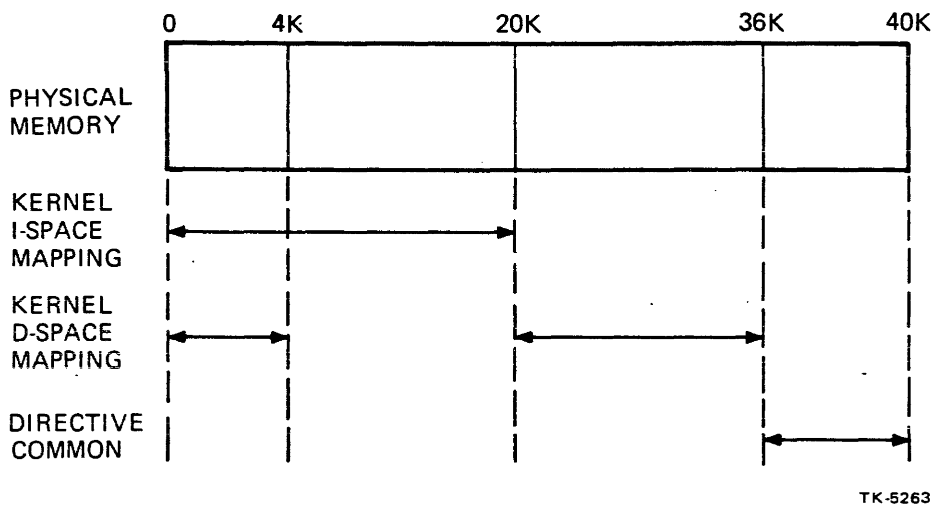


Figure 13 RSX-11M-PLUS EXEC Layout

Event Logging Services

The CEX provides an event-logging interface routine for communication processes (see CEX utility routine $CELOG). The requesting process passes to the CEX routine the event control information and an event description so that the CEX can dispatch the LLC process that performs all network event logging. When the event has been recorded, the control is returned to the requesting process.

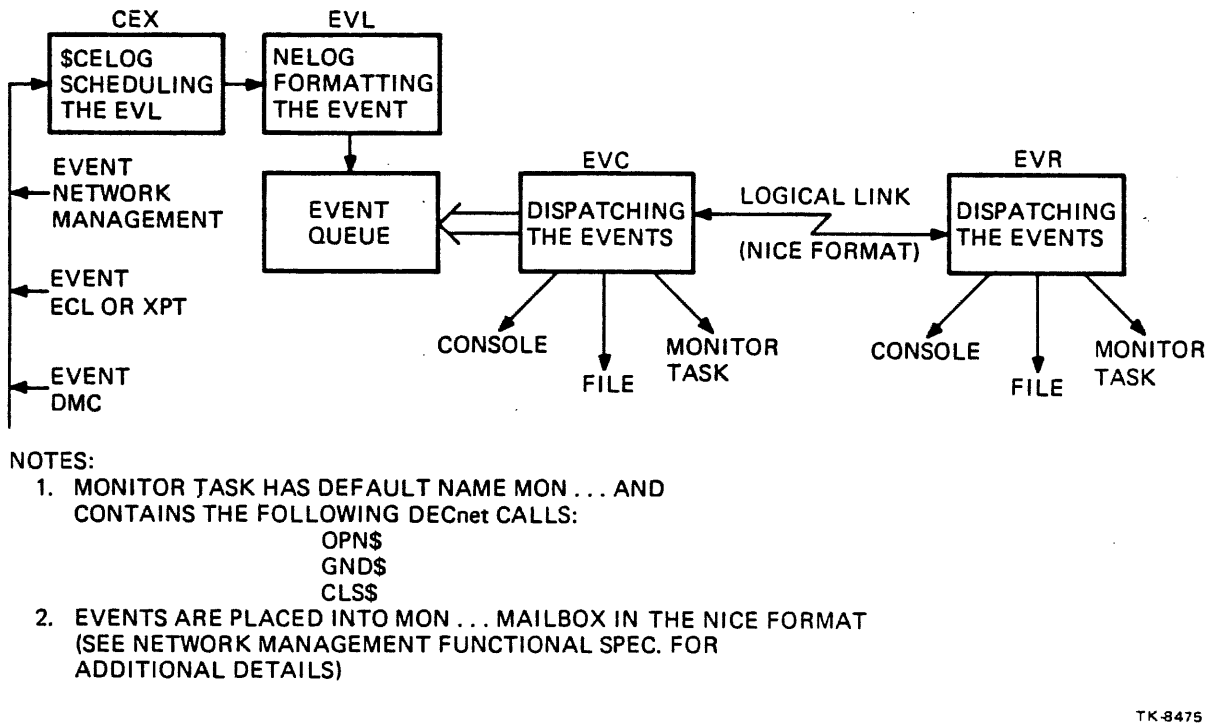


Figure 14 Details of Event Logging Services

Network Management Processing

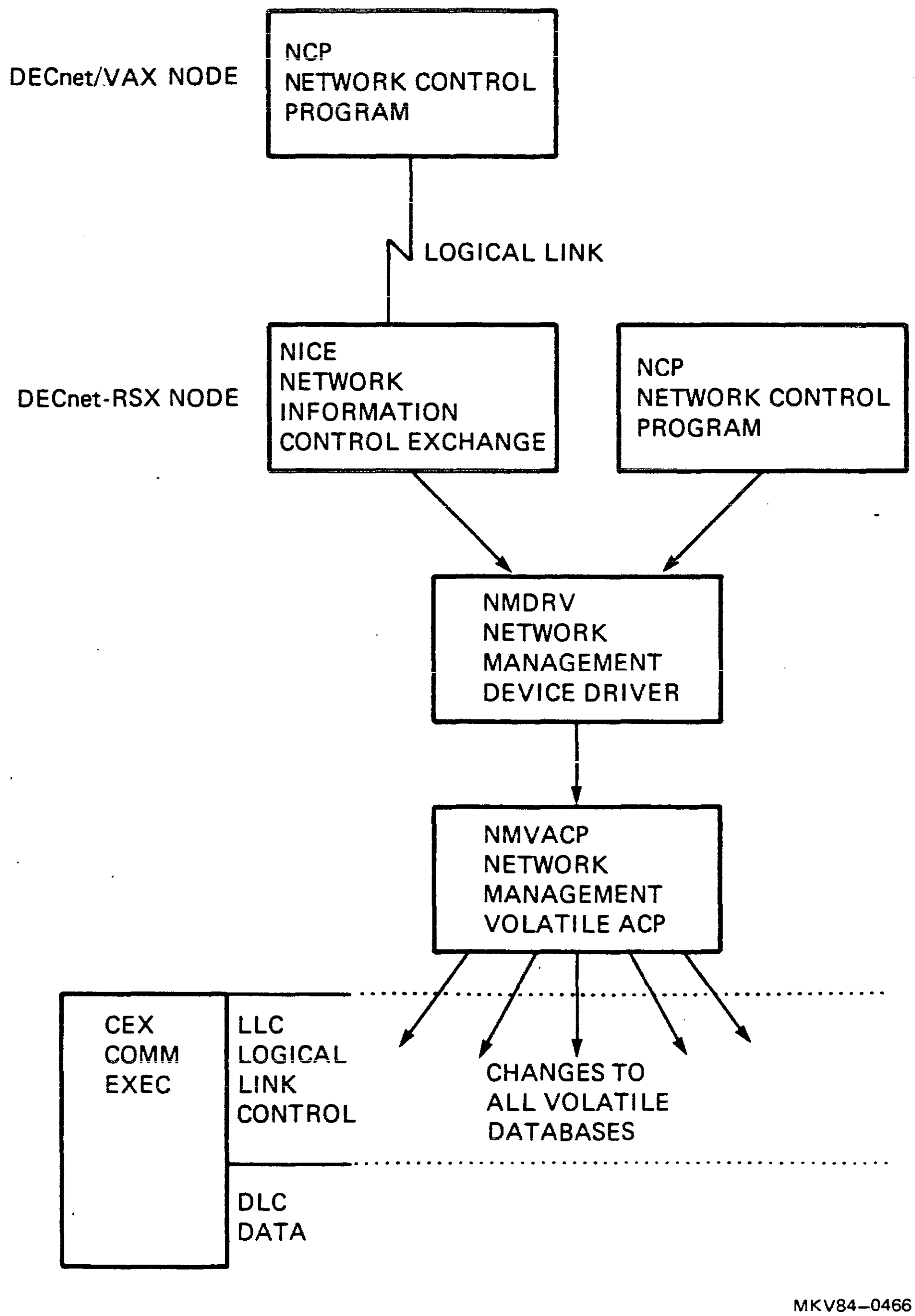


Figure 15 NCP-NICE Processing

ACP PROCESSING

NETACP Processing

Initialization is done upon:

NCP> SET EXEC STATE ON

or

VNP> SET EXEC STATE ON

NETACP is in the main loop state for CCBs and I/O packet processing. CCBs are high priority. When NETACP is idle, it sets the Stop bit. Whenever NETACP processing is required by ECL, the Stop bit is unset and NETACP is ready to process the request.

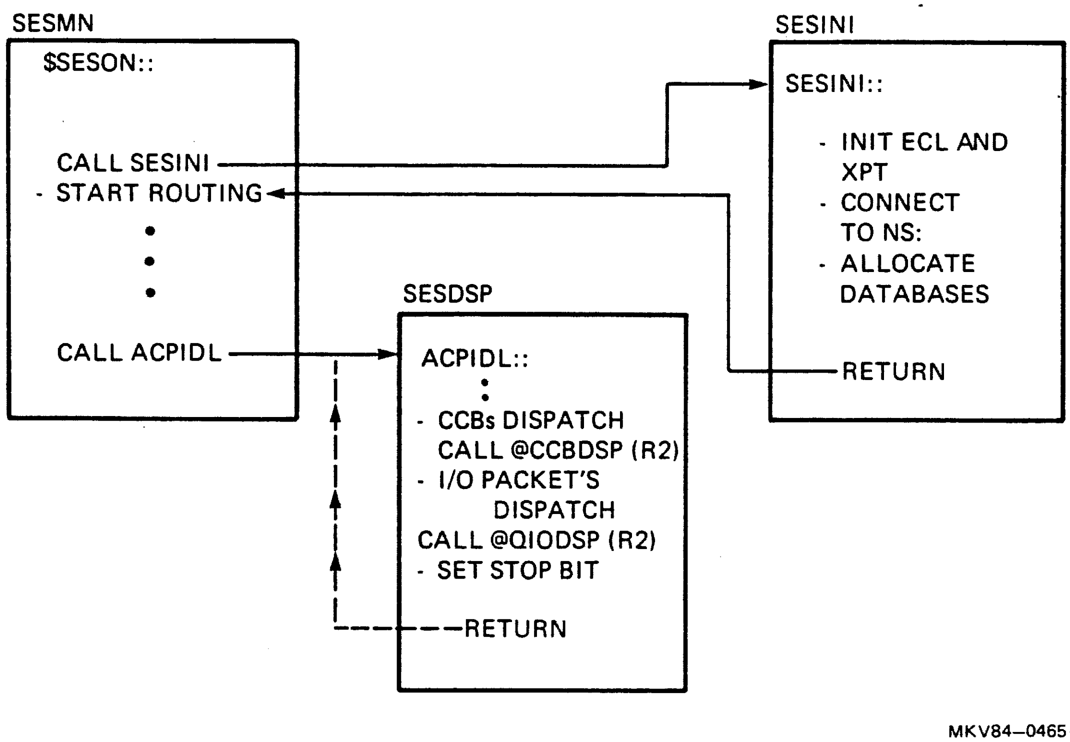


Figure 16 NETACP Initialization and Main Loop

Flow Control Processing

Processes described in this section are:

- Route-through
- Packets coming from ECL to the network
- Connect initiate processing
- Data subchannel and other data subchannel
- Timers and acknowledgments on a logical link

Route-Through Traffic

When a CCB is given to XPT by the lower level processing (for example, DCP or DMC by using CEXCM scheduling and mapping done by CEX), it is time to check if RDB (which is associated with this CCB) contains the local node number as a destination address. If it does, this packet is passed to ECL. If not, the CCB is mapped by the Reachability Vector through the Adjacency Block. The Adjacency block contains information about each adjacent node to the PLD that has the lowest circuit cost to the destination and queued there in P$FWD (the store and forward queue).

If the destination node is unreachable and the flag Return to Sender is set in the packet's routing header, this packet will be sent back to the sender; if the flag is unset, the packet is discarded.

XPT checks if the limited visits threshold is exceeded and, if it is, XPT discards the route-through packet. XPT discards the route-through packet if the square root limit that regulates channel workload and buffer availability is exceeded.

$$\text{square root limit} = \frac{\text{Number of Buffers}}{\sqrt{\text{Number of Active Circuits}}}$$

End Traffic Handling

PLD has an Input Packet Limiter (PC$IPL) that regulates the number of packets that XPT can queue to a channel at any given time (default is 6). This mechanism, together with the square root limiter, optimizes network performance; route-through packets receive higher priority to use the local node's resources compared with ECL packets.

When you issue SND$, the I/O packet is stored in the window block reserved for the LUN that you specified in the CALL. ECL then allocates an LDB, loads it with data from the user task, and gives it to XPT. XPT checks if the ECL Input Packet Limit (PC$IPL) is not exceeded and, if not, places the corresponding CCB to the end of the PLD's store and forward queue. There is no checking of this type for REC$ and XMI$ macros because they do not allocate LDBs, but rather use SDBs and, as a result, do not share resources with route-through traffic. If ECL cannot allocate system resources for the I/O packet (for example, no LDB or SDB available), it tries to do that again a few times (5 is default).

Two mechanisms are involved with this type of retrying: the one-second timer that requests CEX to reschedule ECL, and completion of processing a request by ECL.

Connect Initiate Processing

The Connect Initiate request is always sent with a Return to Sender flag which means if the destination node is unreachable for some reason the packet will be returned to the original node. The Connect Initiate timer is set by default to 32. seconds. If ECL did not get the CIACK by that time, the CI message is retransmitted. The retransmission threshold for Connect Initiate is equal to 5 (default). When it is exceeded, the connection is rejected by the ECL (completion code IE.NRJ in CON$).

Data Subchannel and Other Data Subchannel

Normal and interrupt data are transmitted by using the same logical link, but there are actually two separate subchannels: data subchannel and other data subchannel.

On the receiving side, CCBs associated with RDBs in which incoming messages have been stored are placed into the fork queue processed by AUX in FIFO order.

CCBs with interrupt data are placed into the NETACP queue for processing. CCBs with normal data or link service are processed by ECL that reloads data from the RDBs to the corresponding user task's buffers.

When NETACP processes the interrupt data CCB (which always happens before control is returned to the user task) based on the function code in the CCB (which was previously loaded into the CCB by ECL), it queues this CCB to the user task's mailbox queue. NETACP then executes a special routine, ADDMAI.

ADDMAI:: in SESSUB is used for any event that requires queuing the CCB to the task's mailbox data queue. This mechanism forces the user's task to execute an AST service routine that contains the request Get Network Data (GND) from the mailbox queue before continuing.

The user task always starts checking the mailbox to see if anything was stored there by NETACP.

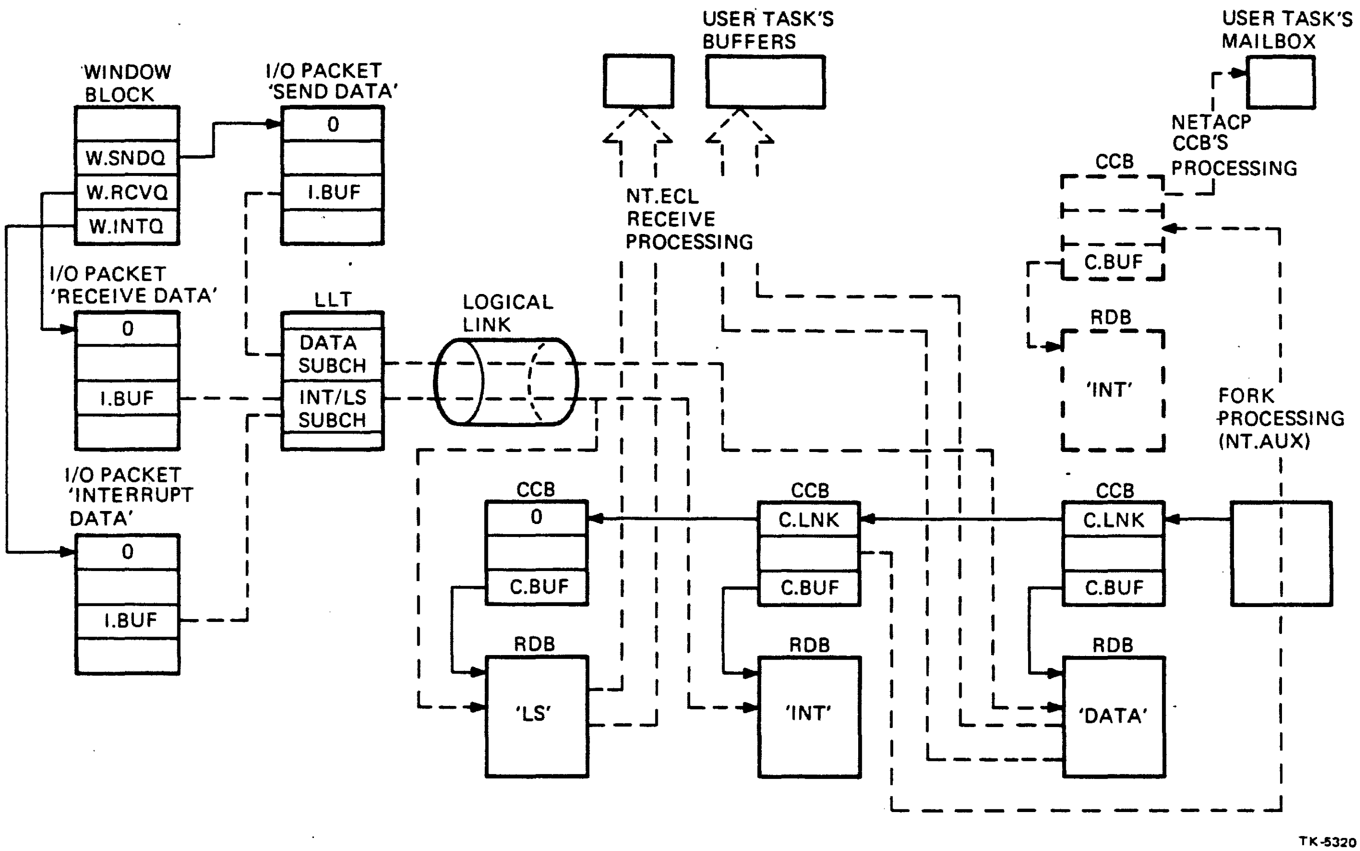


Figure 17 Data and Other Data Subchannel~~s~~ 6

Timers and Acknowledgements on a Logical Link

Whenever ECL sends data from the node, it starts the round-trip timer. The threshold for this timer for Connect Initiate (CI) messages is defaulted to 10 seconds. The time from sending the CI message until processing CIACK by ECL on the local node is considered to be the current round-trip timer threshold over the particular logical link. If this threshold is exceeded, ECL retransmits the packet. The number of times for retransmission is defaulted to 5.

When the logical link is established, it is ready to send and receive messages by the tasks on both sides. The message flow control mechanism is used by both ECLs. The algorithm of control is as follows:

- The receiving side issues the Link Service message that contains the request: "I have a buffer for you, send me a message."

- The sending side divides its message into segments (size of the segment is defined by the CI-CC process over the logical link) and begins to send it segment by segment.

- The sending side expects to receive a Data ACK for every message sent out. The logical link table keeps track of time and the message number.

- Each message has a number associated with it; that is, the receiving side sends Data ACK for message 1, message 2, and so on. If the message arriving at the receiving side contains the message number out of sequence, the receiver drops the message and sends Data NAK for the last correct message's number.

- On the sending side, the timer is reset when Data ACK/NAK is received for the message number that was sent with starting of the timer.

- If the timer expires, the sender retransmits the message.

Fork Processing

The entry $CMFRK:: in CEXCM describes the format of the fork block for DECnet. It is different than normal RSX-11M fork block format:

```
$CMFRK::      .WORD 0         ; Link
              .WORD 0         ; Fork processes routine
              .WORD 0         ; List head of CCBs
```

This fork block is linked to the chain of the fork blocks for RSX-11M $FRKMD:: and is called by $DIRXT:: (the SYSXT module in RSX-11M) on any directive exit.

The prestored fork process routine address is the $SQSRV:: in the AXSCM module of AUX.

If a CCB has a mask (for example, from DLC to LLC process), then according to that mask, AUX dispatches the CCB to the LLC process by CALL $PDDSP.

Once AXSCH takes control, it dispatches all CCBs in the fork queue before exiting.

APPENDIX A
Correspondence Between DECnet Data Structures

These structures are in addition to those discussed in the module.

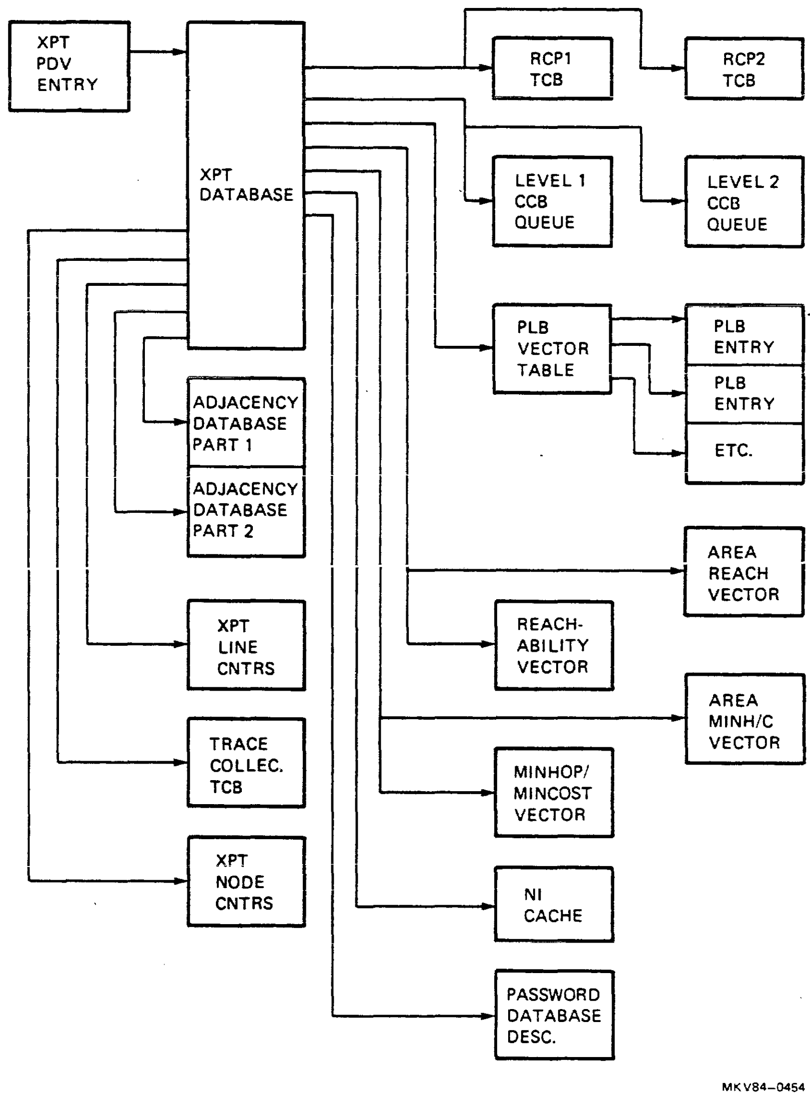


Figure 18 XPT Databases

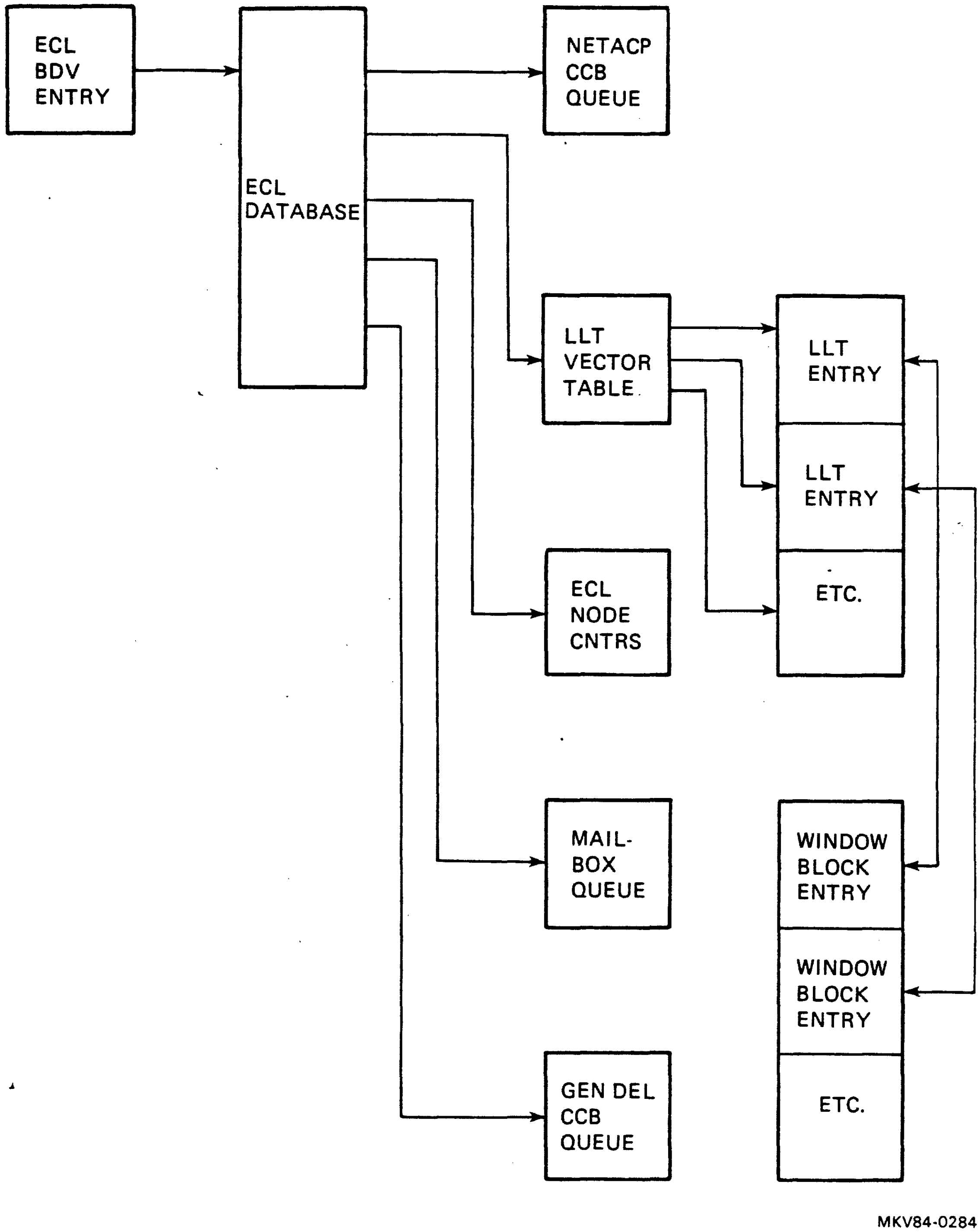


Figure 19 ECL Databases

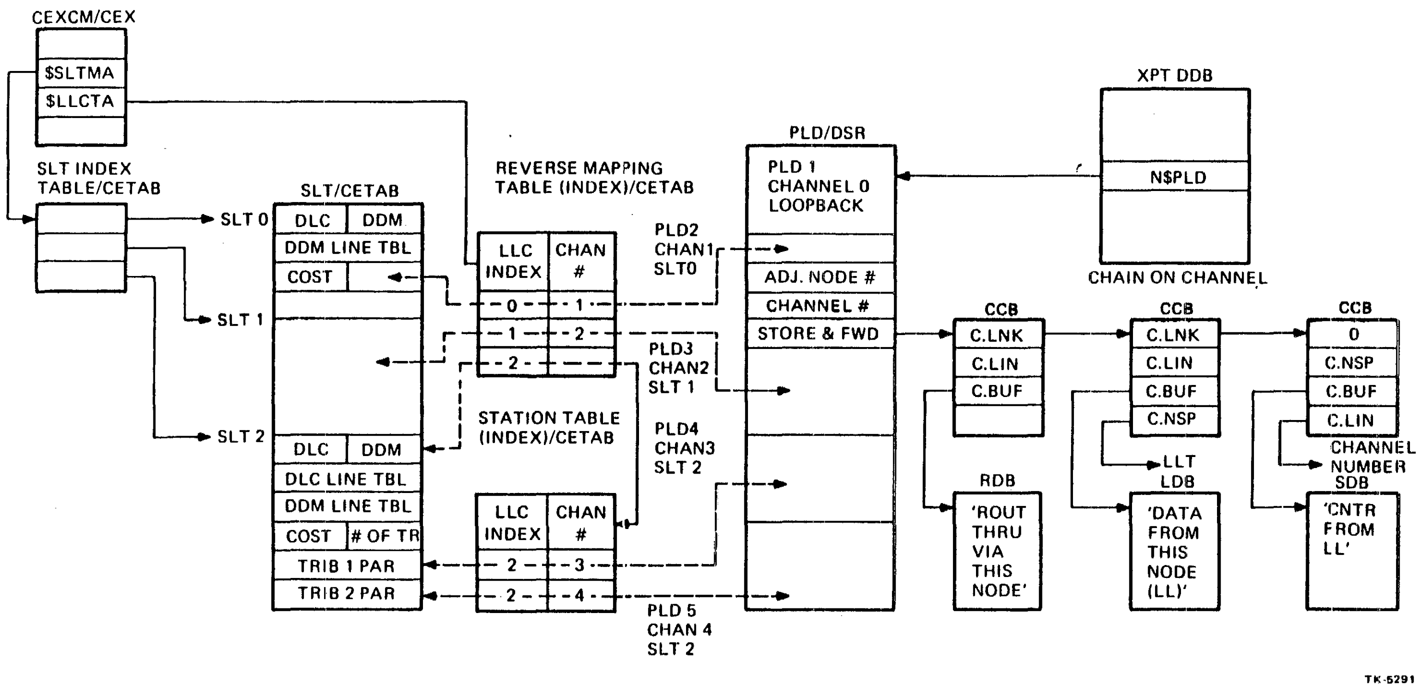


Figure 20 PLD and SLT

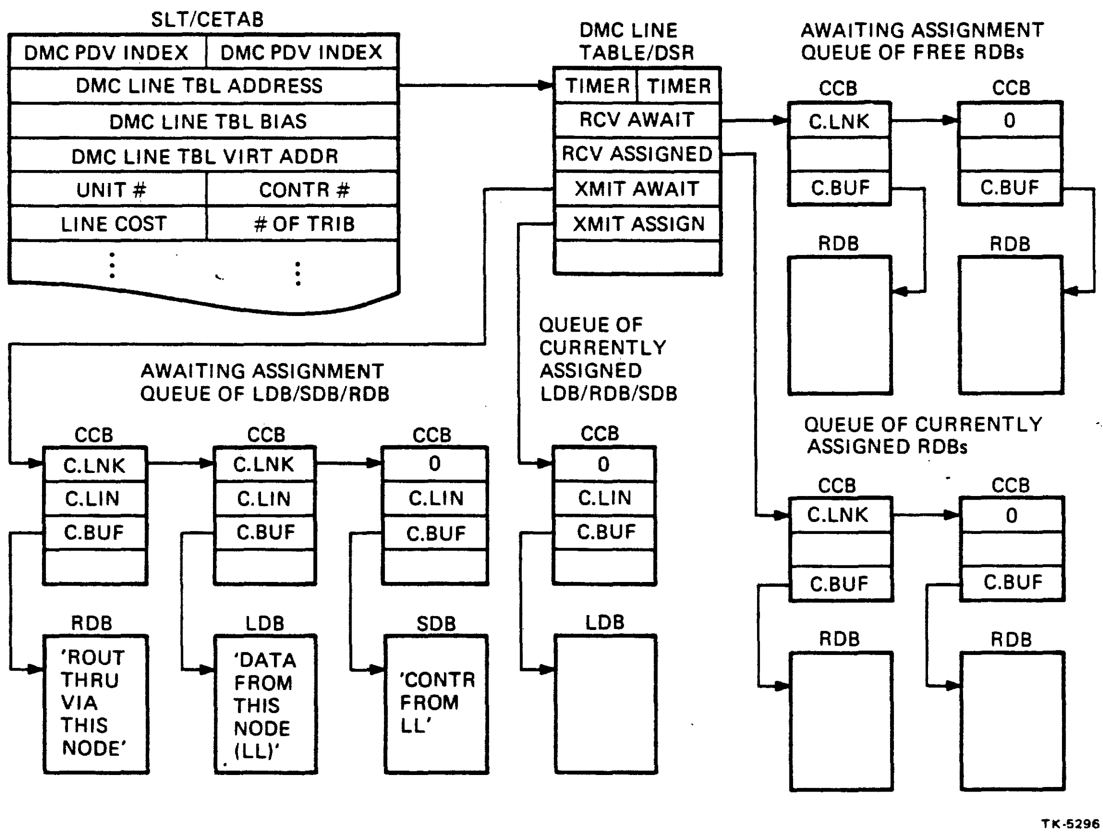


Figure 21 SLT and DDM Line Table (DMC)

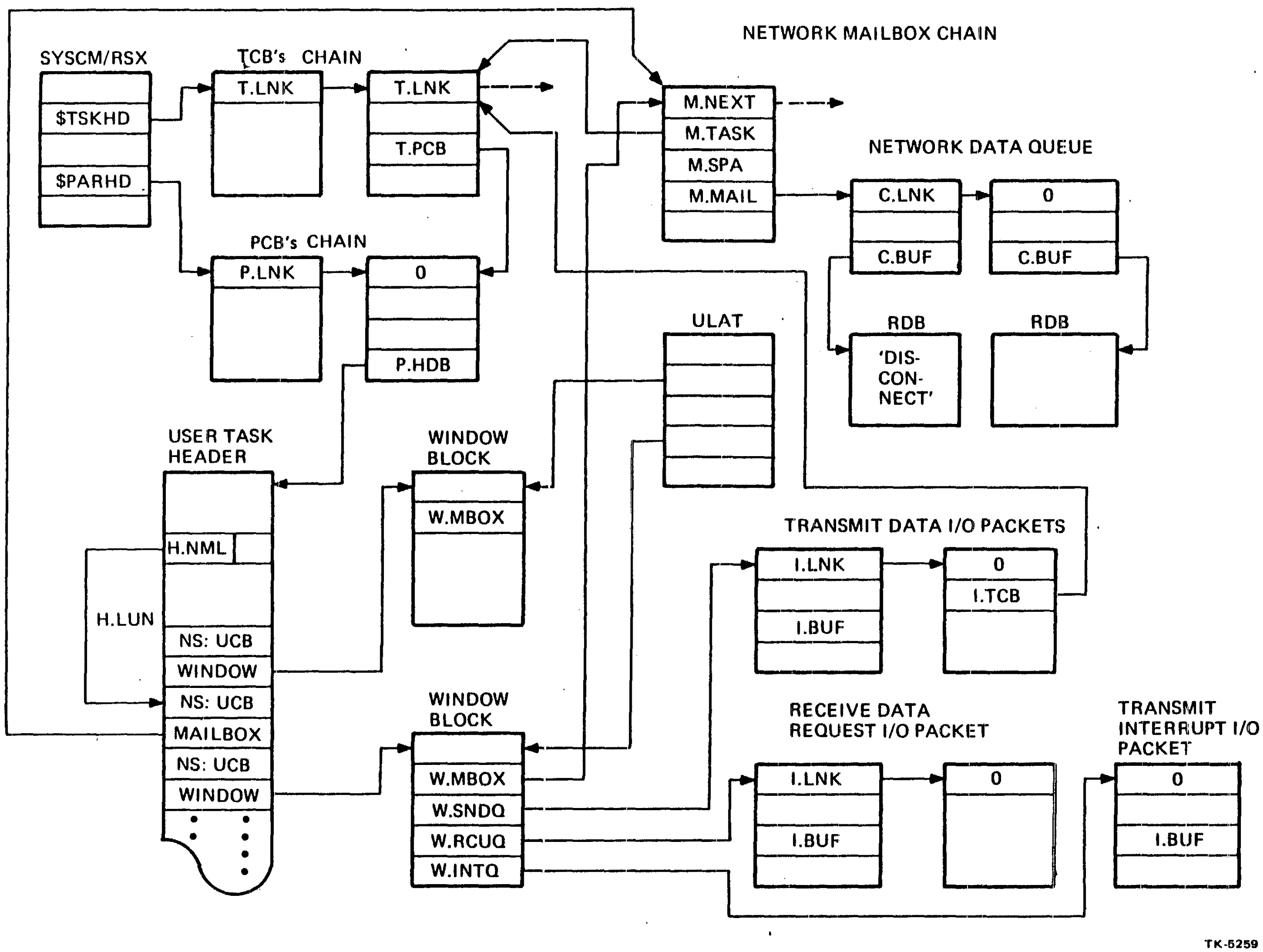


Figure 22 User Task Header, Network Data Queue, and Window Blocks

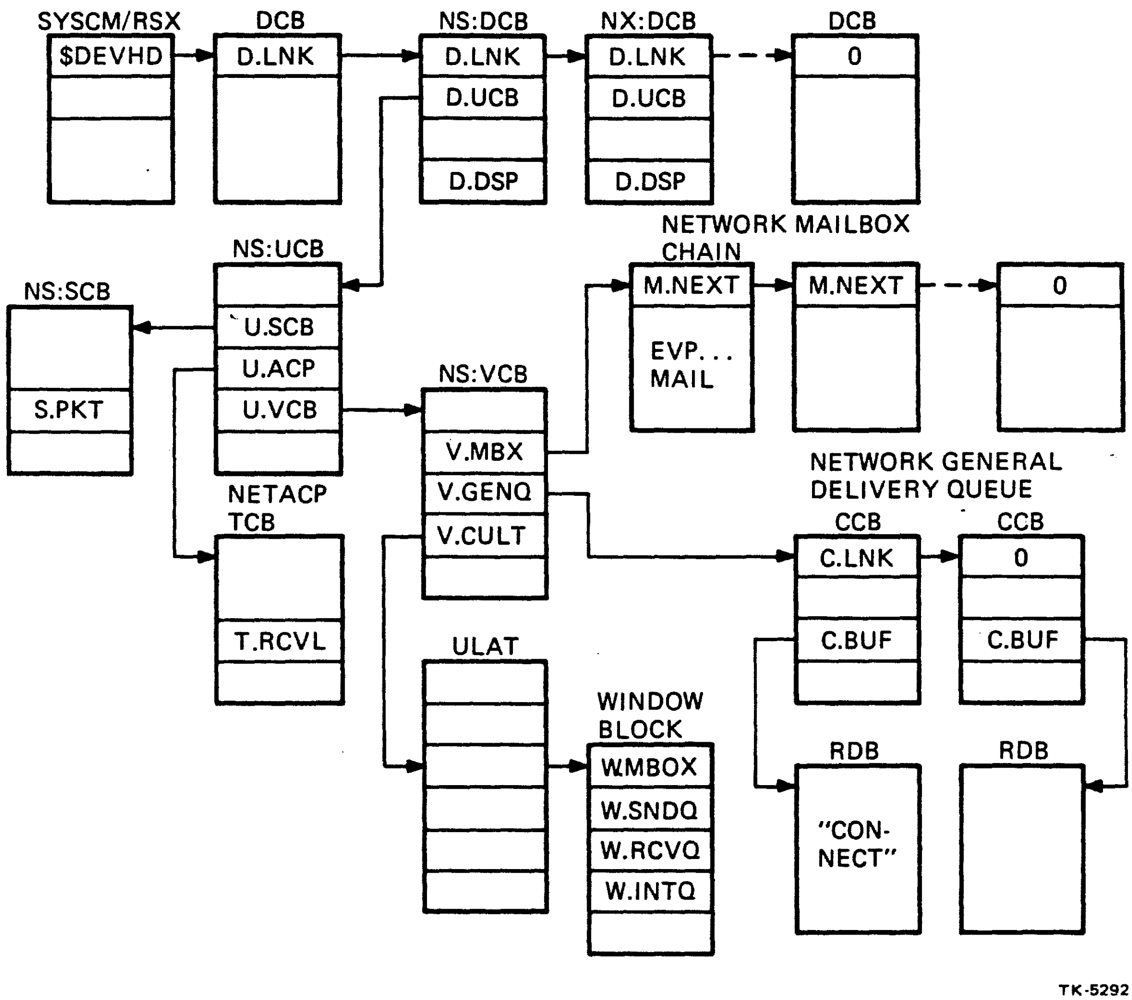


Figure 23 NS: UCB and VCB

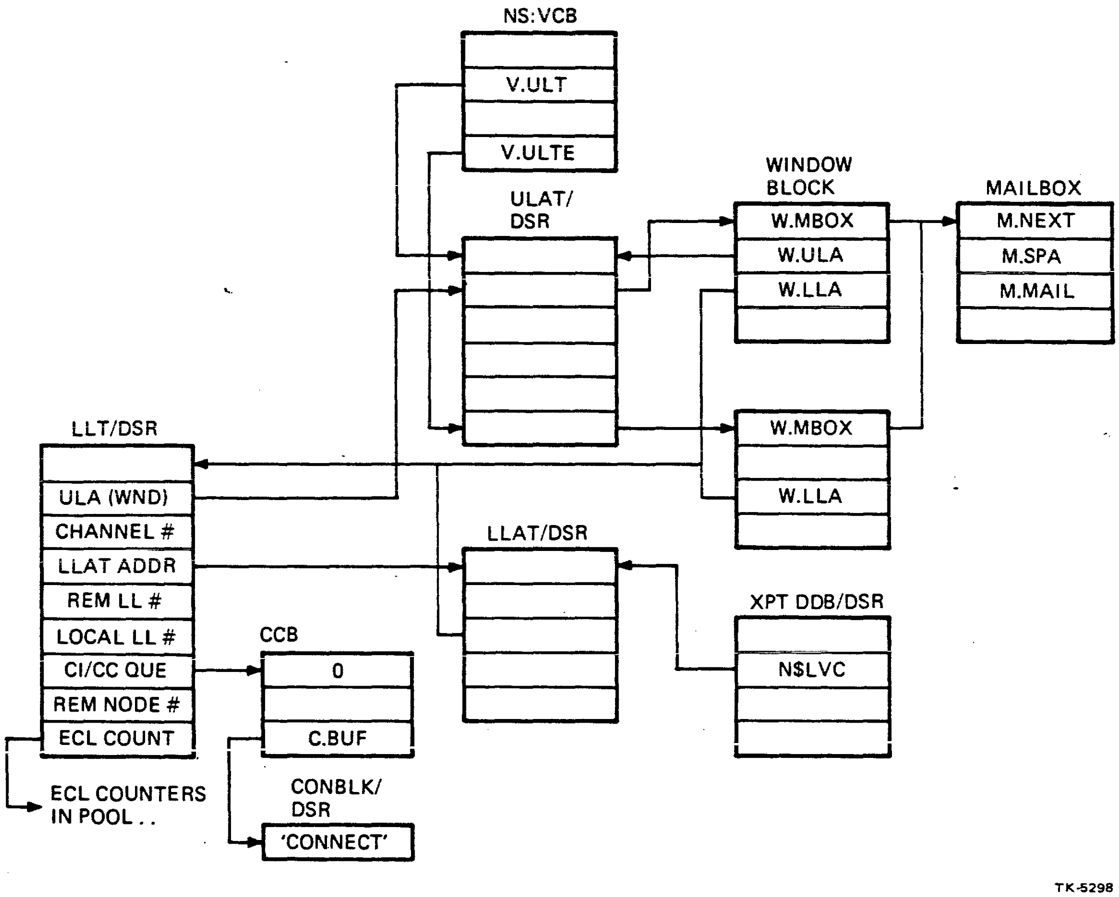


Figure 24 LLT, Window Block, and Mailbox

APPENDIX B
Data Structure Summary and Offset Definitions

Data in Comm/Exec/RSX Pool

Within the Comm/Exec partition is the Comm/Exec pool. The size of the Comm/Exec pool is determined at Network Generation time. The following data structures are located in either the Comm/Exec pool or the system pool (DSR). Both are mapped in the Executive virtual address space.

NOTE

The Comm/Exec pool is not available for data structures in RSX-11M-PLUS systems with I/D space. The Comm/Exec resides in I space provided by the RSX executive.

DHBDF$ - DECnet Home Block Format

The DECnet Home Block contains DECnet specific information that is accessed by multiple DECnet processes and/or tasks. The DECnet Home Block is allocated from RSX system pool (DSR) and is pointed to by using a $DECPT in the Comm/Exec Common. The DECnet Home Block can be examined in a running system by using the CEDump /DH switch.

ALIAS NODE NAME LISTHEAD	D$ANN
REMOTE NODE NAME LISTHEAD	D$RNN
POINTER TO END	
LOCAL NODE NAME (6 BYTES)	D$LNAM
LOCAL NODE NUMBER	D$LNUM
IDENT STRING LENGTH	D$LID
LOCAL NODE IDENTIFICATION (32 BYTES)	
HOST NODE ADDRESS	D$HOST
HIGH-ORDER FOUR BYTES OF ETHERNET ADDRESS	D$HIOR
NUMBER OF ROUTING CHANNELS	D$NLN
DLL SERVICE DATABASE ADDRESS	D$SER
DLL DATABASE FNB ADDRESS	D$FNB
ECL SEGMENT SIZE	D$SEG

MKV84-0276

Figure 25 DECnet Home Block (Sheet 1 of 2)

The following Home Block cells are present for routing nodes only:

NUMBER OF BROADCAST ROUTER ADJACENCIES	D$NBRA
NUMBER OF BROADCAST END NODE ADJACEN'S	D+NBEA
NUMBER OF NODES IN NETWORK	D$NN
NUMBER OF AREAS IN NETWORK	D$NA
MAXIMUM COST	D$MAXC
MAXIMUM HOPS	D$MAXH
MAXIMUM VISITATION COUNT	D$MAXV
AREA MAXIMUM COST	D$AMXC
AREA MAXIMUM HOPS	D$AMXH
SQUARE ROOT LIMITING FACTOR	D$SQRL
ROUTING TIMER (SECONDS)	D$RTMR
BROADCAST ROUTING TIMER (SECONDS)	D$BRTM
BROADCAST ROUTER PRIORITY	D$BRPR

MKV84-0277

Figure 25 DECnet Home Block (Sheet 2 of 2)

ADJDF$ - Adjacency Database Part 1

The Adjacency Database contains information about each adjacent node (point-to-point neighbors and NI neighbors). Due to the size and attendant mapping problems, the database is split into two sections (ADJ1 and ADJ2). Adjacency Database Part 1 can be mapped by the bias contained in N$ADJ1 of the XPT database. The Adjacency Database is not present for nonrouting (end) nodes.

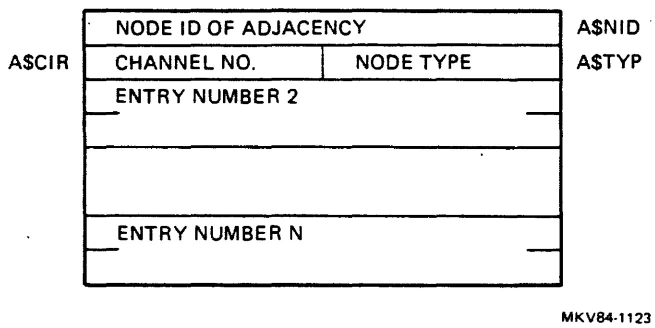


Figure 26 Adjacency Database Part 1

A$TYP Node Type Flag Definitions

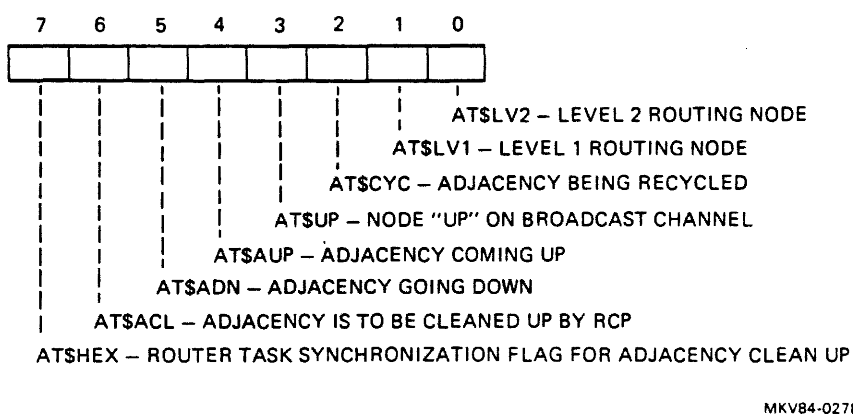


Figure 27 A$TYP Node Type Flag Definitions

ADJDF$ - Adjacency Database Part 2

The Adjacency Database contains information about each adjacent node (point-to-point neighbors and NI neighbors). Due to the size and attendant mapping problems, the database is split into two sections (ADJ1 and ADJ2). Adjacency Database Part 2 can be mapped by the bias contained in N$ADJ2 of the XPT database. The Adjacency Database is not present for nonrouting (end) nodes.

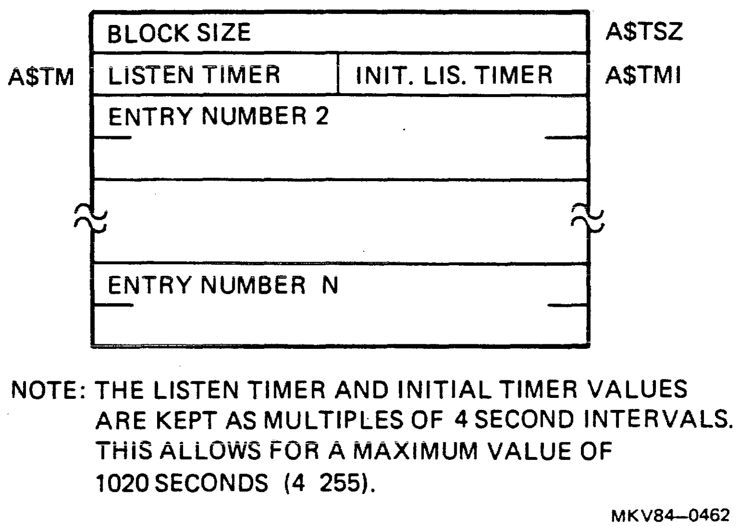


Figure 28 Adjacency Database Part 2

LLTDF$ - Logical Link Table Entry

Each LLT entry contains the current information for a single logical link. the LLT is used by the ECL process and the NETACP task to service logical links. the LLT entries of a running DECnet system can be dumped using the CEDump /LL switch.

L.TYP	LINK TYPE	LINK STATE	L.STA
	LOCAL LOGICAL LINK ADDRESS		L.LLA
	REMOTE LOGICAL LINK ADDRESS		L.RLA
	REMOTE NODE 16 BIT ADDRESS		L.REM
L.TIPD	# DATA XMTS IN PROG.	# I/LS XMTS IN PROG.	L.TIPI
L.VER	REMOTE NSP VERSION	LINK FLAGS	L.FLAG
	NEXT DATA SEGMENT NUMBER TO BE ASSIGNED		L.NXN
	NEXT I/LS SEGMENT NUMBER TO BE ASSIGNED		L.NIN
	NEXT DATA SEGMENT NUMBER TO BE RECEIVED		L.RNO
	HIGHEST ACK # FROM USER ON DATA CHAN		L.USA
	NEXT I/LS SEGMENT NUMBER TO BE RECEIVED		L.LNO
	HIGHEST I/LS ACK # FROM USER		L.LSA
	LAST DATA SEGMENT NUMBER ACK'D		L.LDA
	LAST INT/LS SEGMENT NUMBER ACK'D		L.LIA
L.CSTA	NET DISC SUBSTATE	USER DISC SUBSTATE	L.USTA
	POINTER TO USER'S WINDOW BLOCK		L.WIND
L.TIC	INTERRUPT COUNT	TRANSMIT COUNT	L.TC
	FLOW CONTROL REQUEST COUNT (I/LS)		L.LSFI
	FLOW CONTROL REQUEST COUNT (DATA)		L.LSFD
	REMOTE FLOW CONTROL COUNT ESTIMATE		L.RCF
	I/LS PENDING ACK QUEUE		L.ILSQ
L.RTYD	RETRY CELL (DATA)	TIMER CELL (DATA)	L.TMRD
L.RTYI	RETRY CELL (I/LS)	TIMER CELL (I/LS)	L.TMRI
	MESSAGE AWAITING RETRANSMISSION		L.RTQ
	LONG TERM TIMER		L.LIT

MKV84-0460

Figure 29 Logical Link Table Entry (Sheet 1 of 2)

L.LPT	PERIODIC TIMER	TIMER (SECONDS)	L.SEC
	INITIAL LONG TERM TIMER		L.ILTT
	MESSAGE REASSEMBLY QUEUE		L.MASQ
L.MAST	REASS. QUEUE TMR.	SIZE OF REASS. QUEUE	L.MASZ
	POINTER TO ECL COUNTER BLOCK		L.CTR
	SEGMENT SIZE FOR THIS LINK		L.SEGZ
	DISCONNECT REASON CODE		L.DCR
L.OPD	OPTIONAL DATA	OPTIONAL DATA LEN.	L.OPDL
	(16 BYTES)		
	(RESERVED)		

The following additional fields are only included if System Level Interface (SLI) is present.

L.ULA	USER LINK ADDRESS	SOURCE CHANNEL #	L.CHN
L.PDVD	DATA PROCESS PDV	CONTROL PROCESS PDV	L.PDVC
	TRANSMIT MESSAGE QUEUE DOUBLEWORD LISTHEAD		L.XMTQ
	CURRENT TRANSMIT CCB ADDRESS		L.CXMT
	INTERRUPT MESSAGE TRANSMIT QUEUE DOUBLEWORD LISTHEAD		L.INTQ
	CURRENT INTERRUPT CCB ADDRESS		L.CINT
	PENDING CONTROL CCB ADDRESS		L.PCTL
	PENDING ACCEPT CCB ADDRESS		L.ACC

MKV84-0456

Figure 29 Logical Link Table Entry (Sheet 2 of 2)

PLBDF$ - Physical Link Block (PLB) <DSR>

The Physical Link Block is used by XPT to maintain the current status and control/monitor each physical connection to an adjacent node. Pointers to each PLB entry reside in the Physical Link Block Vector Table.

P$TYP	ADJ. NODE TYPE	LINK STATE	P$LST
P$RTIM	RECOVERY TIMER	RECOVERY FLAGS	P$LCD
P$CNT	# MESG'S QUEUED	XPT CHANNEL #	P$CHN
	PENDING CONTROL FUNCTION QUEUE		P$PFQ
	GENERAL PROTOCOL TIMER		P$TIM
	FLAGS		P$FLG
	(RESERVED)	INP. PACKET LIN	P$IPL
	MAX DELAY FOR ROUT. MSG. LEVEL 1		P$RMX1
	MAX DELAY FOR ROUT. MSG. LEVEL 2		P$RMX2
	LEVEL 2 STATE	LEVEL 1 STATE	P$STA 1
	TRANSPORT BLOCK SIZE		P$TSIZ
	COUNT OF ADJ. NODES OF LOW TSIZ		P$TSCT
	STORE AND FORWARD QUEUE DOUBLEWORD LISTHEAD		P$FWD
	TRANSPORT COUNTER BLOCK ADDRESS		P$CTR
	16-BIT ADDRESS OF DES. ROUTER		P$DRTR
	NUMBER OF ROUTERS ON NI		P$NRNI
	CIRCUIT ROUTING PRIORITY		P$RPRI

The following cells are only used for DLM circuits:

DATA LINK MAPPING PACKET SIZE	P$PKSZ
INPUT REASSEMBLY CCB	P$ICCB
OUTPUT SEGMENTATION QUEUE	P$OCCB
HOLDING AREA FOR INIT. SEED (4 WORDS)	P$SEED

MKV84-0455

Figure 30 Physical Link Block (Sheet 1 of 2)

XPDDB$ - XPT Database

The data shared between the XPT process and the Routing tasks (RCP1/RCP2) is contained in this structure. The pointer to the data is contained in the Process Descriptor LLC database address cell (Z.DAT) for the XPT process.

	TCB ADDRESS OF LEVEL 1 ROUTER		N$RT1
	LEVEL 1 ROUTING MESSAGE		N$LV1
	DOUBLEWORD LISTHEAD		
	TCB ADDRESS OF LEVEL 2 ROUTER		N$RT2
	LEVEL 2 ROUTING MESSAGE		N$LV2
	DOUBLEWORD LISTHEAD		
N$RTM2	LEVEL 2 RT. TIMER	LEVEL 1 RT. TIMER	N$RTM1
	PHYSICAL LINK BLOCK VECTOR SIZE		N$PLD
	PHYSICAL LINK BLOCK VECTOR ADDRESS		
	REACHABILITY VECTOR SIZE		N$ROA1
	REACHABILITY VECTOR BIAS		
	REACHABILITY VECTOR ADDRESS		
	AREA REACHABILITY VECTOR SIZE		N$ROA2
	AREA REACHABILITY VECTOR BIAS		
	AREA REACHABILITY VECTOR ADDRESS		
	MINHOP/MINCOST VECTOR SIZE		N$MHC1
	MINHOP/MINCOST VECTOR BIAS		
	MINHOP/MINCOST VECTOR ADDRESS		
	AREA MINHOP/MINCOST VECTOR SIZE		N$MCH2
	AREA MINHOP/MINCOST VECTOR BIAS		
	AREA MINHOP/MINCOST VECTOR ADDRESS		
	NI CACHE SIZE		N$CACH
	NI CACHE ADDRESS		
	NUMBER OF PASSWORD DATABASE ENTRIES		N$VER
	PASSWORD DATABASE ADDRESS		

MKV84-0459

Figure 31 XPT Database (Sheet 1 of 2)

BIAS OF ADJACENCY DATABASE PART 1	N$ADJ1
BIAS OF ADJACENCY DATABASE PART 2	N$ADJ2
BROADCAST ROUTER PRIORITY TABLE ADDR.	N$PRI
NUMBER OF TRANSPORT LINE CNTR BLOCKS	N$TLC
TRANSPORT LINE COUNTER BIAS	
TRANSPORT LINE COUNTERS ADDRESS	
TRACE CONTROL WORD	N$TRC
TRACE COLLECTOR TCB ADDRESS	N$TTCB
TRACE CONTROL BLOCK ADDRESS	N$TCTL
TRANSPORT NODE COUNTER SIZE	N$TNC
TRANSPORT NODE COUNTERS ADDRESS	

THE FOLLOWING ADDITIONAL CELL IS PRESENT ONLY ON DLM NODES:

STCRC LIBRARY POINTER	N$CRC

MKV84-0457

Figure 31 XPT Database (Sheet 2 of 2)

ECDDB$ - ECL Database <DSR>

The data shared between the ECL process and the NETACP task is contained in this structure. The pointer to the data is contained in the Process Descriptor LLC database address cell (Z.DAT) for the ECL process.

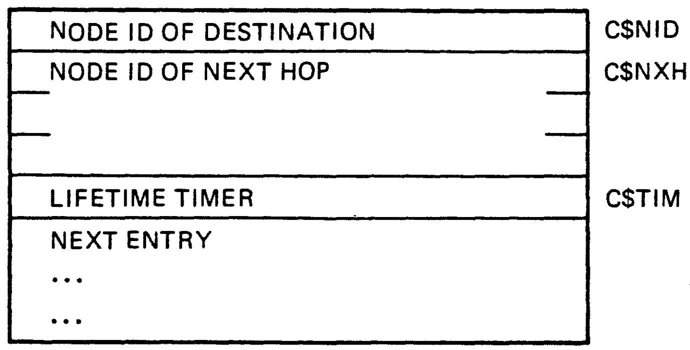
	NETWORK ACP CCB QUEUE DOUBLEWORD LISTHEAD		N$ACQ
N$FLG	FLAGS BYTE	TIMER COUNT	N$TIM
	FUNCTION CODE		N$FNC
	DUMMY VCB		N$VCB
	SOURCE NODE ADDRESS		N$SNOD
	(RESERVED)	ROUND TRIP DELAY	N$DLY
	SOURCE LINK ADDRESS		N$SLA
	DESTINATION LINK ADDRESS		N$DLA
	ERROR CODE		N$ERRC
	MAPPING OF CURRENT LLT		N$LLTM
	CURRENT LLT VIRTUAL ADDRESS		N$LLT
	CURRENT LLT PHYSICAL ADDRESS		N$PLLT
N$HIGH	MAX ACT LOG LINKS	CUR ACT LOG LINKS	N$ACTL
	COUNT OF CI'S IGNORED DUE TO RESOURCES		N$CIR
	LOGICAL LINK TABLE LENGTH LOGICAL LINK TABLE ADDRESS		N$VLC
	ECL NODE COUNTERS DOUBLEWORD LISTHEAD ECL MODE COUNTERS APR BIAS		N$ENC
	MAILBOX QUEUE LISTHEAD		N$MBXQ
	GENERAL DELIVERY CCB QUEUE LISTHEAD		N$GENQ
N$GTI	GEN DEL INI TIMER	GEN DEL CUR TIMER	N$GTM

MKV84-0458

Figure 32 ECL Databases

CACDF$ - NI Cache <DSR>

The NI Cache is only present for nonrouting (end) Ethernet nodes. It contains information about the next hop to get to certain nodes. Entries are made in the cache when a logical link is set up to a node and removed when a time interval has elapsed without any communications with that node. The pointer to the NI Cache is contained in the XPT database.

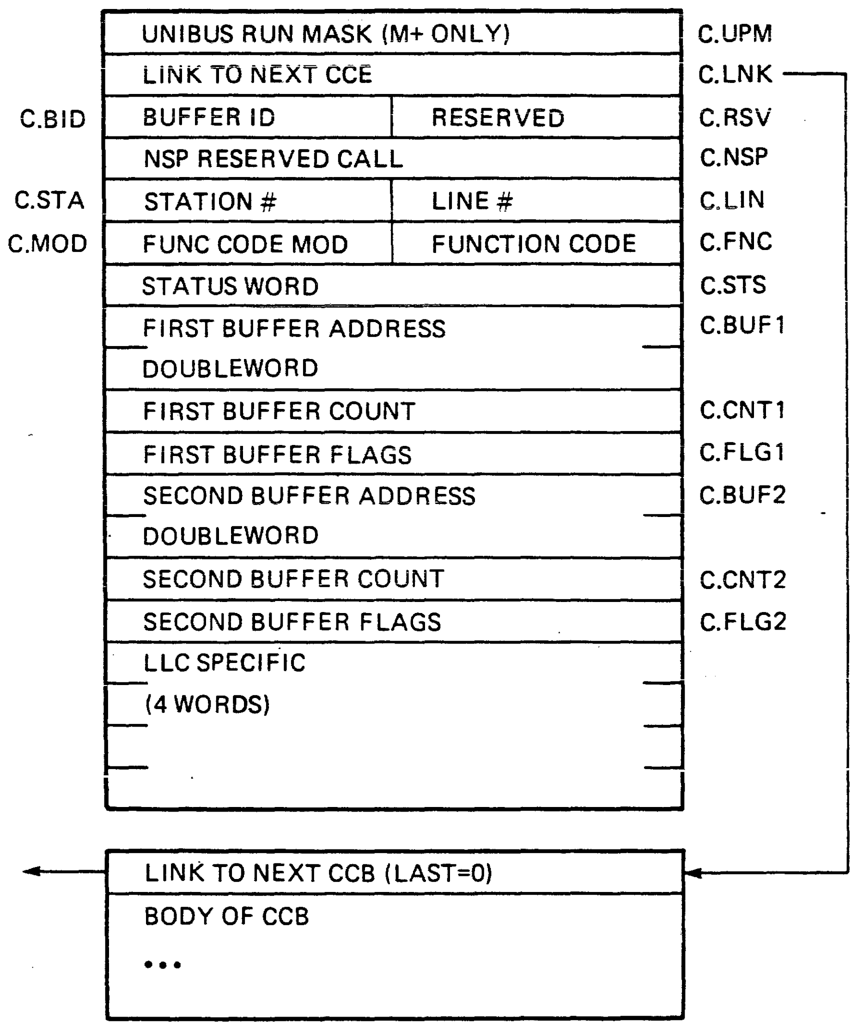


NOTE: THE NODE ID OF DESTINATION (C$NID) IS A 16 BIT NODE ADDRESS AND THE NODE ID OF NEXT HOP (C$NXH) IS A 48 BIT NODE ADDRESS.

MKV84-0279

Figure 33 NI Cache

Free Communication Control Block (CCB) List <DSR>



MKV84-0467

Figure 34 Free Communication Control Block List

LLC Specific Data for Ethernet

48-BIT ETHERNET ADDRESS	C.ADD
ETHERNET PROTOCOL TYPE	C.PRO

MKV84-0578

Figure 35 LLC Specific Data for Ethernet

LLC Specific Data for PSI

C.XID	CALLING DTE	LVL. 3 PORT NO.	C.XPT
C.XTC	PACKET TYPE	PLI USER BYTE	C.X25
	SCRATCH WORD FOR PLI/LLCS		C.XPLI
	(RESERVED)		

MKV84-0280

Figure 36 LLC Specific Data for PSI

Process Descriptor Vector (PDV) Table <CES Pool>

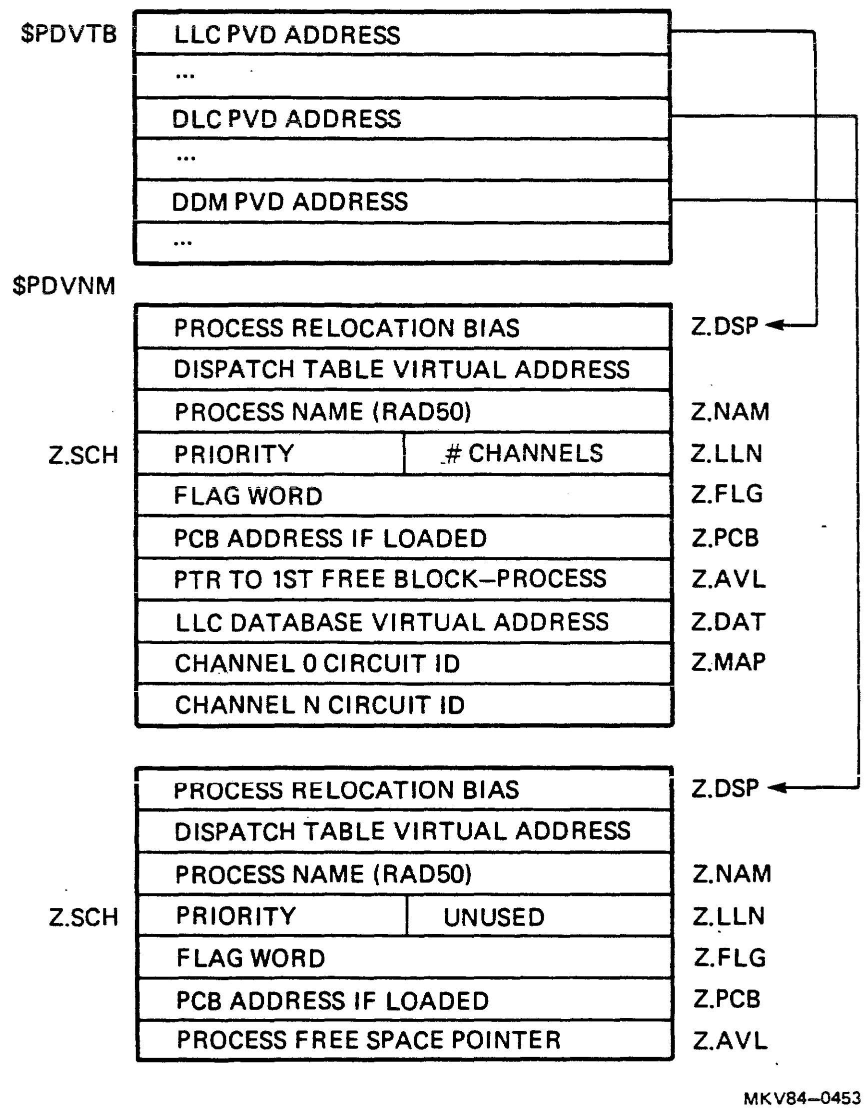
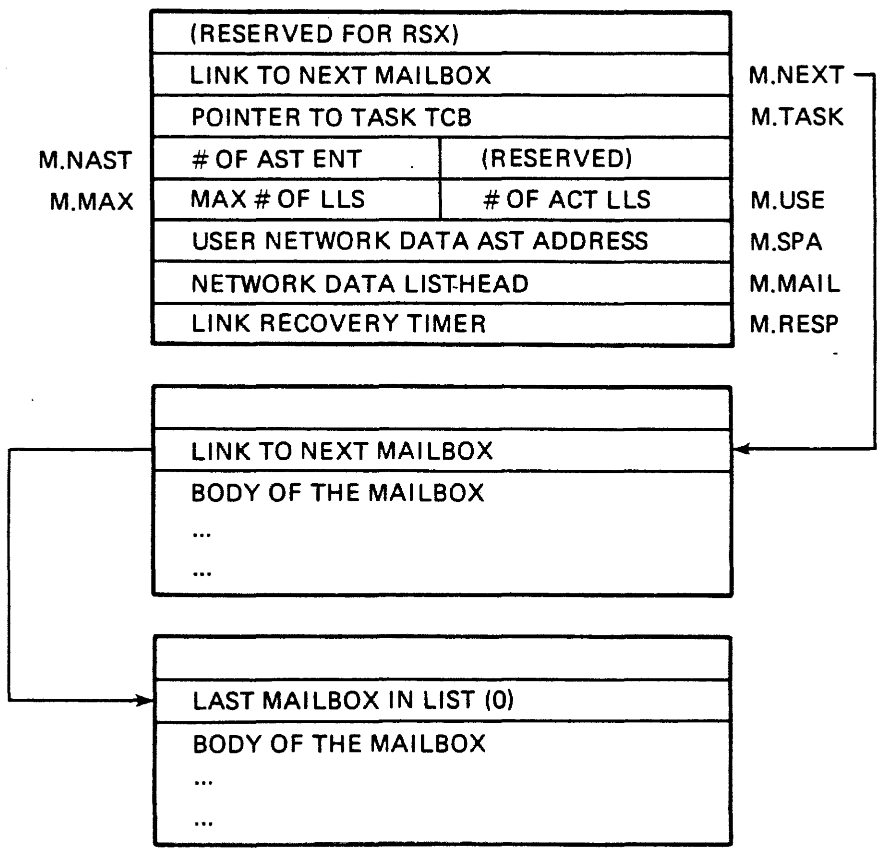


Figure 37 Process Descriptor Vector Table

MBXDF$ - Mailbox Queue <CRS Pool>

A mailbox is created when a user task indicates that it is an active network user by issuing an OPN$ function. The mailbox holds common information about the user task for use by the network software.



MKV84-0281

Figure 38 Mailbox Queue

System Line Table (SLT) Vector - Indexed by SLM <CEX Pool>

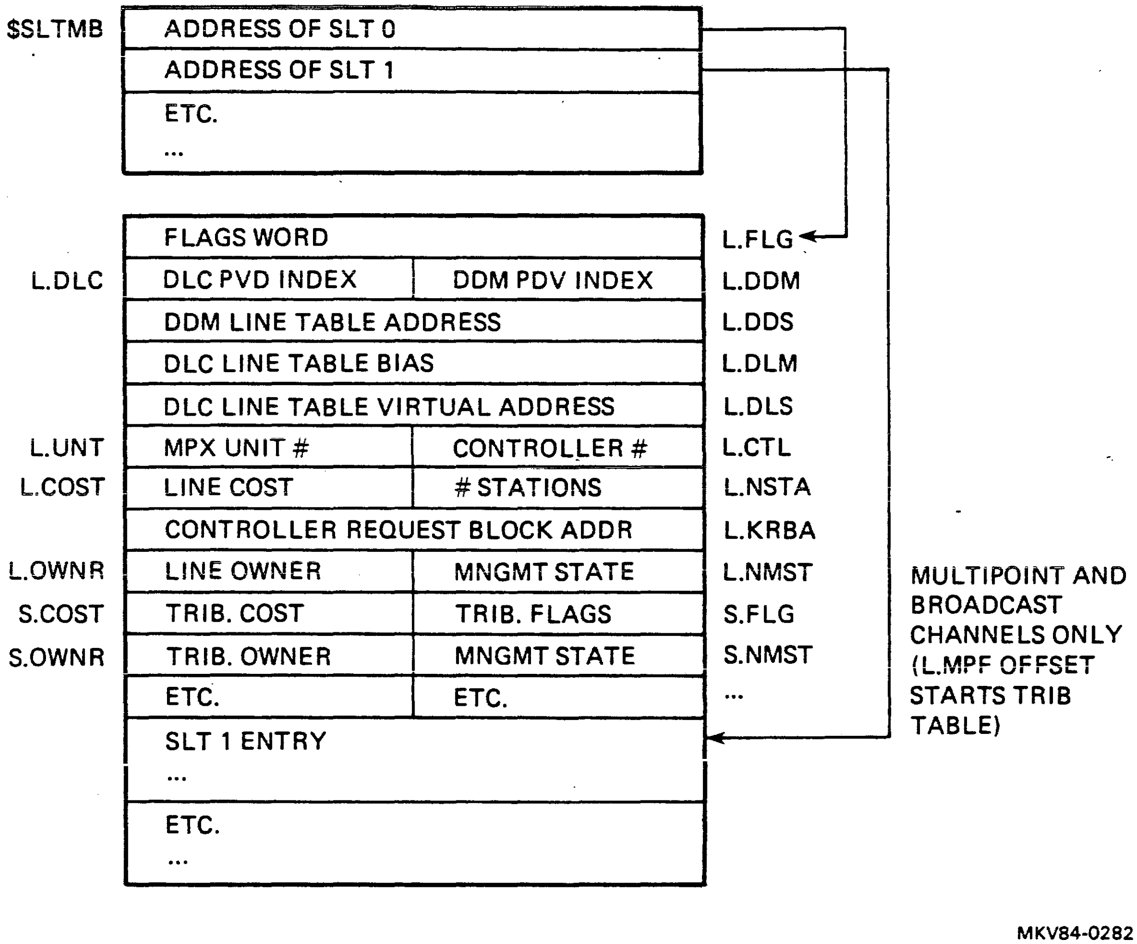


Figure 39 System Line Table Vector - Indexed by SLN

LLWDF$ - Window Block <CEX Pool>

The window block holds the current state of a logical link with respect to the QIOs issued by the user task. It is pointed to by the second LUN word in the task header.

	(RESERVED FOR RSX)		W.CTL
W.LUN	TASK LUN	LINK STATUS	W.STAT
	POINTER TO ASSOCIATED LLT		W.LLT
	SEGMENT SIZE		W.SEGZ
	TEMPORARY WORKSPACE		W.TMP
	POINTER TO MAILBOX		W.MBOX
	(RESERVED)	KN1 AST STATUS	W.KAST
	TRANSMIT QUEUE LISTHEAD		W.SNDC
	POINTER TO CURRENT XMT I/O PKT		W.CSND
	POINTER TO CURRENT INT I/O PKT		W.CINT
	RECEIVE QUEUE LISTHEAD		W.RCVC

MKV84-0283

Figure 40 Window Block

PSI Data Structure Summary

Table 1 Important PSI Data Structures

Name	Size (Bytes)	Allocated From:	Usage	Related PSI Component	Library Macro
CCB - Communication Control Block	28. (30. for multiprocessing) +6. for PSI	CEX remaining space and DSR	Interprocess communication	CEX NT.XXX X25ACP	PSI.MLB: CCBDF$
PDV - Process Descriptor Vector	14.+2 for PLIDB 2. per channel	CETAB - CEX remaining space	Communication process description	XXX	PSI.MLB: PDVDF$
SLT - System Line Table	18.+4 per tributary	CETAB - CEX remaining space	Interprocess interface for the communication channel (PLI)	XXX	PSI.MLB: SLTDF$
DTEDB - Local DTE Descriptor Block	72.	POOL..	Describes local DTE's static and dynamic parameters	PLI	PSI.MLB: DTEDF$
EVL - PLI Event Block	28.+ additional data	DSR	Formats event for EVC	EVC	PSI.MLB: EVLDF$
DDM - Line Table	Variable	XXX process space*	Communication device-specific data, interrupt lineage code, etc.	XXX	PSI.MLB: DLMDF$, KMXDF$, LABDF$,
DST - X.25/X.29 Destination Descriptor Block	24.	POOL..	Describes X.25 and X.29 destinations	NW, PLI	PSI.MLB: DSTDF$

*Note that XXX refers to network processes.

Table 1 Important PSI Data Structures (Cont)

Name	Size (Bytes)	Allocated From:	Usage	Related PSI Component	Library Macro
Task's Mailbox	20.	DSR	Controls the task's overall network access; has listhead to network mail (CCBs) and AST routine address	PLI	PSI.MLB: MBDF$
NUB - Network Virtual Terminal Control Block	26.	HTDRV	Controls PAD access	X29ACP, HT:	PSI.MLB: NVBDF$
NWDB - NW.LLC Database	22.	DSR	Contains pointers to various important NW:-related data structures, window blocks, etc.	NW, PLI	PSI.MLB: NWDF$
Object Descriptor Block	10.	POOL..	Describes a Network Object	PLI	PSI.MLB: OBJDF$
PHB - PSI Home Block	34.	DSR	PSI database, contains pointers to various data structures	PLI and others	PSI.MLB: PHBDF$
PLIDB - PLI LLC Database	22.	DSR	Contains pointers to various important PLI-related data Structures	PLI	PSI.MLB: PLIDF$

Table 1 Important PSI Data Structures (Cont)

Name	Size (Bytes)	Allocated From:	Usage	Related PSI Component	Library Macro
PVC - Name Block	14.	POOL..	Describes a PVC	NW, PLI	PSI.MLB: PVCDF$
Remote DTE Descriptor Block	20.	POOL..	Describes a remote DTE	NW, PLI	PSI.MLB: RDTDF$
Trace Block	12.	DSR	Control block for trace facility	PLI TRADF	PSI.MLB:
VCB - Volume Control Block for NW:	30.	DSR	Listhead for general delivery queue and mailboxes		PSI.MLB: VCBDF$
Window Block	34.	POOL..	Allocated per VC and contains current information on that circuit from the user task	NW, PLI	PSI.MLB: WBDF$
X.25 Circuit Block	70.	POOL..	Circuit management-related information used by PLI	PLI	PSI.MLB: XCBDF$
X.29 Database	14.	DSR	X29 counter block	X29ACP	PSI.MLB: X29DF$

PSI Data Structure Offsets

PSI Home Block

PSI structure: PHB
Size (bytes): 34.
Total number: One
Allocated from: DSR
Allocated when: On loading the network by NTL
Pointed to by: $PSIPT:: in CEXCM
Macro in PSI.MLB: PHBDF$

GENERAL FLAGS	00	H$FLG
LOCAL DTE DESCRIPTOR LIST	02	H$LDTE
REMOTE DTE DESCRIPTOR LIST	04	H$RDTE
PVC NAME LIST	06	H$PVC
CUG NAME LIST	10	H$CUG
X.25 DESTINATION BLOCK LIST	12	H$DST
X.29 DESTINATION BLOCK LIST	14	H$D29
TRACE BLOCK	16	H$TRB
PORT TABLE POINTER	20	H$PTB
NUMBER OF PORTS	22	H$NPT
NETWORK NAME (ASCII)	24	H$NETW
	26	
	30	
LOW TRANSPORT SUBADDRESS	32	H$LOTS
HIGH TRANSPORT SUBADDRESS	34	H$HITS
SVC BLOCKS (FOR DLM)	36	H$SVC
POINTER TO X.29 DATABLOCK	40	H$X29C

TK-8471

Figure 41 PSI Home Block (PHB)
Offset Definition

NW:LLC Database

PSI structure: NWDB
Size (bytes): 22.
Total number: One
Allocated from: DSR
Allocated when: By NTL on loading the network
Pointed to by: Z.DAT in PDV for NW:
Macro in PSI.MLB: NWDF$

	POINTER TO WINDOW BLOCK ADDRESS TABLE		00 02	Z$WBAT
	POINTER TO END OF WB TABLE		04	Z$WEND
	QUEUE LISTHEAD OF CCBS FOR X25ACP		06 10	Z$ACPQ
	POINTER TO UCB FOR NW:		12	Z$UCB
	COUNTER TIMER PERIOD		14	Z$CTIM
	COUNT OF CURRENTLY ACTIVE CIRCUITS		16	Z$ACTC
	TIME WHEN COUNTERS LAST ZEROED		20	Z$TCLZ
	MAX CIRCUITS ACTIVE		22	Z$MXAC
Z$LLRE	UNUSED	INCOMING CALLS RESOURCE ERRORS	24	Z$ICRE

TK-8472

Figure 42 NW: LLC Database Offset Definition

PLI Data Descriptor Block (PLI LLC Database)

PSI structure: PLIDB
Size (bytes): 22.
Total number: One
Allocated from: DSR
Allocated when: On loading the network by NTL
Pointed to by: Z.DAT in PDV for PLI
Macro in PSI.MLB: PLIDF$

	FLAG WORD		00	Y$FLG
	DEFAULT PACKET SIZE		02	Y$DPSZ
	MAXIMUM PACKET SIZE		04 06	Y$MPSZ
Y$MWSZ	MAX WINDOW SIZE	DEFAULT WINDOW SIZE	10	Y$DWSZ
Y$TCAL	CALL REQUEST TIMER	RESTART TIMER	12	Y$TRST
Y$TCLR	CLEAR TIMER	RESET TIMER	14	Y$TRES
Y$MRES	MAX RESET RETRIES	MAX RESTART RETRIES	16	Y$MRST
Y$LMPK	LOG2 MAX PACKET SIZE	MAX CLEAR RETRIES	20	Y$MCLR
	DTE ADDRESS LENGTH		22	Y$DTAL
	DTE SUBADDRESS LENGTH		24	Y$DSAL

TK-8469

Figure 43 PLI LLC Database Offset Definition

Window Block for X.25

PSI structure: W
Size (bytes): 34.
Total number: One per active circuit
Allocated from: POOL..
Allocated when: CAL$ or CAC$ calls
Pointed to by: .WBATB:: listhead in ACPROO module of X25ACP;
Z$WBAT in NW:LLC database
Macro in PSI.MLB: WBDF$

	FLAG WORD		00	W$FLAG
W$LUN	LUN # FOR CIRCUIT	CIRCUIT TYPE	02	W$TYP
	SPARE	CIRCUIT PORT #	04	W$PORT
	PACKET SIZE		06	W$PKSZ
	MAILBOX ADDRESS		10	W$MBOX
	PENDING XMT$ I/O PACKETS LISTHEAD		12 14	W$SNDQ
	PENDING RCV$ I/O PACKETS LISTHEAD		16 20	W$RCVQ
	LDBS PENDING TRANSMISSION PACKET LISTHEAD		22 24	W$TXCQ
	RDBS PENDING RECEPTION PACKET LISTHEAD		26 30	W$RXCQ
	IN-PROGRESS I/O PACKETS LISTHEAD (FOR X25ACP)		32 34	W$PIO
W$DAST	OUTSTANDING ASTS	CCB IN USE FOR XMT$	36	W$CCBU
W$FLUX	HIGH BYTE OF FLAGS (XMT$)	SPARE	40	

TK-8470

Figure 44 Window Block for X.25 Offset Definition

X.25 Circuit Block

PSI structure: XCB
Size (bytes): 70.
Total number: One per active circuit
Allocated from: POOL..
Allocated when: CAL$ or CAC$, preallocated for PVCs
Pointed to by: Port table entry (through H$PTB in PSI home block)
Macro in PSI.MLB: XCBDF$

X$TIMC	CALL/CLEAR TIMER	RESET TIMER	00	X$TIMR
X$DIAG	LAST DIAGNOSTIC USED	RETRY COUNT	02	X$RTRY
X$ST	CIRCUIT STATE	CIRCUIT SUBSTATE	04	X$SS
X$NPR	# TO ABORT IMMEDIATELY	# OF PACKETS IN LOWER LEVEL	06	X$NPL
X$PR	P(R) FOR NEXT TX	P(S) FOR NEXT TX	10	X$PS
X$RPR	LAST RECEIVED P(R)	NEXT EXPECTED RX P(S)	12	X$RPS
X$TYP	TYPE BYTE	FLAG BYTE	14	X$FLG
	PACKET SIZE (TX)		16	X$PKSZ
X$PRT	PORT NUMBER	WINDOW SIZE	20	X$WSZ
X$MOWN	OWNER	HIGHER LEVEL PDV/SLN	22	X$USR
	POINTER TO LOCAL DTE		24	X$DTE
	LOGICAL CHANNEL NUMBER		26	X$LCN
	TIME COUNTERS LAST ZEROED		30	X$TCLZ
	BYTES RECEIVED		32	X$NRBY
			34	
	BYTES TRANSMITTED		36	X$NTBY
			40	
	DATA PACKETS RECEIVED		42	X$NRPK
			44	
	DATA PACKETS TRANSMITTED		46	X$NTPK
			50	
X$NRRE	REMOTE RESETS	LOCAL RESETS	52	X$NLRE
X$ALF	ALLOCATION FAILURE FLAGS	NETWORK GENERATED RESETS	54	X$NNRE
	AUTOMATIC TIMER COUNTDOWN		56	X$AUC
	TIMER COUNTER VALUE		60	X$CTIM
	WAITING ACKNOWLEDGEMENT		62	X$WAQ
			64	
	AWAITING TRANSMISSION		66	X$TXQ
			70	
	RECEIVER HOLD QUEUE		72	X$RXQ
			74	
	LIST OF CCBS TO ABORT		76	X$ABQ
			100	
	TRANSMIT INTERRUPT		102	X$TXI
	RECEIVE INTERRUPT		104	X$RXI

TK-8468

Figure 45 X.25 Circuit Block Offset Definition

APPENDIX C
DECnet-RSX Listing Trace

Listing Trace Technique

- The technique of multiple passes is recommended.
- Preparation work:
  - Have an idea of the major components and their interaction.
  - Have copies of the map and listings available either on paper or on microfiche.
  - Learn how to use the map and listings effectively, in particular:
    1. Be able to identify the starting address of the major component, if any.
    2. Be able to find out how many modules make up a major component.
    3. If given a global symbol, be able to identify which module and which line of code defines the global symbol.
- Pass 1
  - Treat the listing like a road map. Make sure you don't get lost.
  - Only focus on instructions that pass control to another module or subroutine. Do not try to follow every instruction in the first pass.
  - If you have paper listings, use a highlight pen to trace the major paths. If you use microfiche, chart out a simple major path.

- Pass 2

  - Assume an error free path. Use the map or highlighted listings to extract interesting information.

  - Under DECnet, the interesting paths may be to trace through:

    1. Issuing network open
    2. Sending connect initiate
    3. Sending connect confirm
    4. Sending/receiving normal data
    5. Sending/receiving interrupt data
    6. Disconnecting logical link
    7. Specifying network AST
    8. Getting network data
    9. Disconnecting logical link
    10. Issuing network close

- Pass 3

  - Relate the data structure to the code.

  - Put in sufficient comments to get a simple tour guide through the listings.

- Subsequent passes are to obtain specified information.

- An example of this technique is shown on tracing sending and receiving of a connect initiate message.

Sending of Connect Initiate Message

Table 2 Tracing Connect Initiate

Path	Ref	Comments
1	RSX Internals Course	The Connect Initiate request is translated to a QIO or QIOW request.
		(a) The Directive Parameter Block (DPB) is set up and control is passed to the RSX-11M executive by using the EMT377 instruction.
		(b) The EMT trap vector (in module LOWCR) passes control to the Directive Dispatcher Module (DRDSP).
		(c) General purpose registers are saved by using a coroutine call to the system entrance and exit module (SYSXT) and control is passed back to DRDSP.
		(d) The parameters in the DPB are checked and the correct directive handling module is called. In this case, the directive handling module is DRQIO.
		(e) The I/O request packet is built and the RSX executive assists the device driver by looking at the function masks of the driver. For functions to be handled by the device driver, the RSX executive calls the driver at the initialization routine.
		(f) For connect requests, the NS driver is used. The I/O request is passed to it at NSCHK in the ECL communication process.
2	ECLQIO	(a) Check whether OPN$ has been issued. If so, check the buffers containing the connect request block.
		(b) Allocate space and set up certain fields for the window block.
		(c) Call NETACP.

Table 2 Tracing Connect Initiate (Cont)

Path	Ref	Comments
3	SESDSP	(a) The NETACP is characterized by its dispatch or stop main loop. When there is nothing to do, the NETACP sets its own stop bit. When activated by the executive (unsetting its Stop bit), it looks for work in one of the two queues - a Communication Control Block (CCB) queue and the I/O request packet queue.
		(b) If there is something in these queues, the NETACP dispatches to one of its processing modules and keeps on looping until all entries are processed.
4	SESCON	For the Connect Initiate request, an internal connect block is set up to prepare for sending of the Connect Initiate message.
5	SESUSR	(a) The node name is converted to node address.
		(b) Data structures related to logical link are created.
		(c) The session control portion of the message is built, and control is passed back to ECL by using CEX.
6	ECLSUB ECLXMT	(a) The data subchannel is used to transmit the connect initiate message.
		(b) The routing module (XPT) is called by using CEX.
7 8	XPTRCV XPTXMT	(a) The process dispatch table in XPTRCV is entered. The entry in this table is transmit enabled.
		(b) The appropriate headers are built and control is passed to the device driver module by using CEX - another mapped subroutine call.

Table 2 Tracing Connect Initiate (Cont)

Path	Ref	Comments
9 10		(a) The connect initiate message is sent out. (b) The data link header is built by the DMC hardware.
11 12		(a) The connect initiate message is received by the data link driver and passed to the XPT process by using fork processing. (b) The routing module determines whether the packet is for the local node. If so, it passes it to the ECL process also by using the fork processing.
13	ECLRCV	The process dispatch table in ECL is entered and because it is a control function, NETACP is called. The CCB queue is used to pass information to NETACP.
14 15	SESDSP	(a) The CI message is checked and the logical link address table is checked to see if another one can be created. (b) Tell user about received CI and send the Connect Acknowledge back.

Connect Initiate (Part 1)

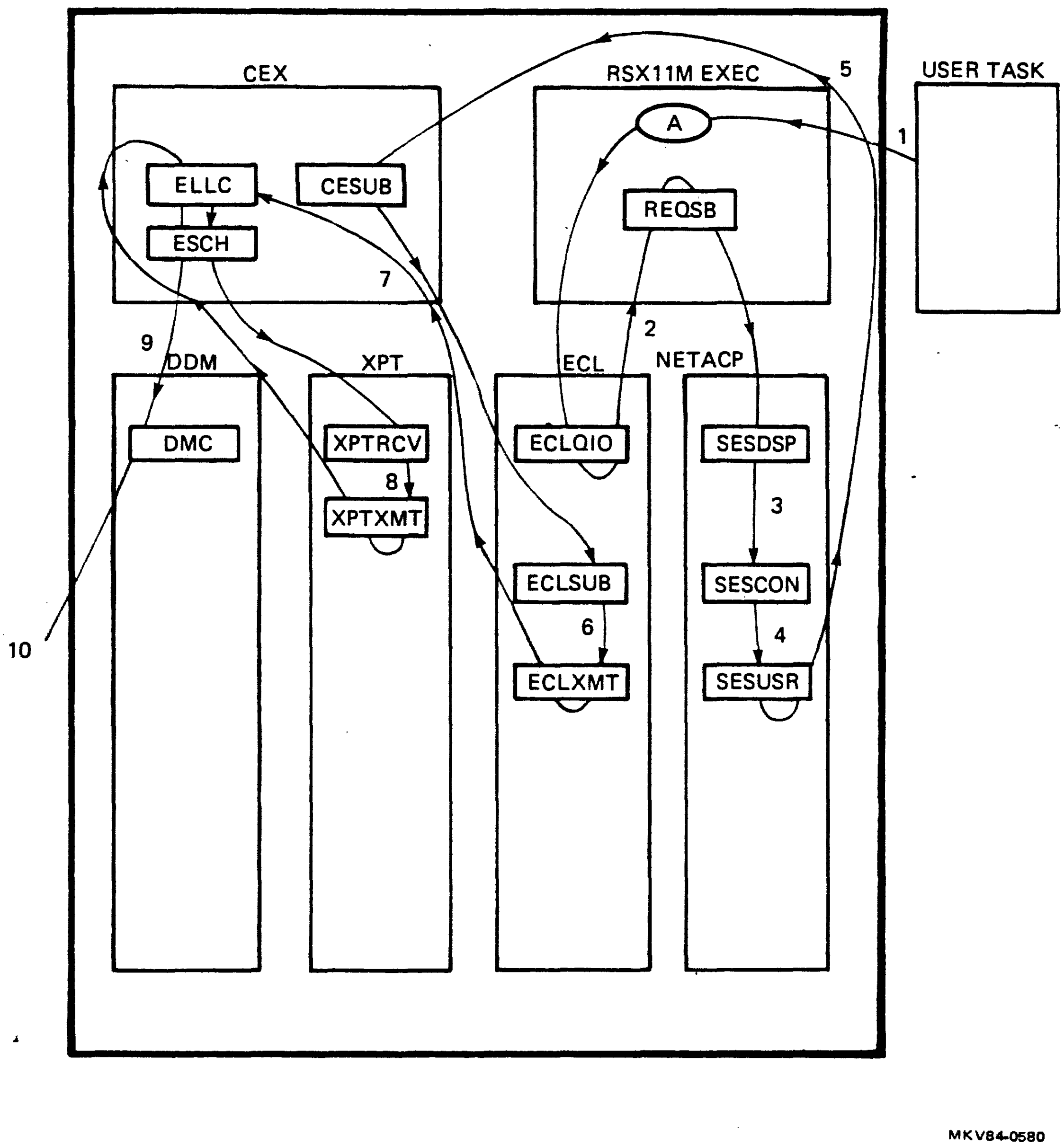


Figure 46 Sending Connect Initiate

Connect Initiate (Part 2)

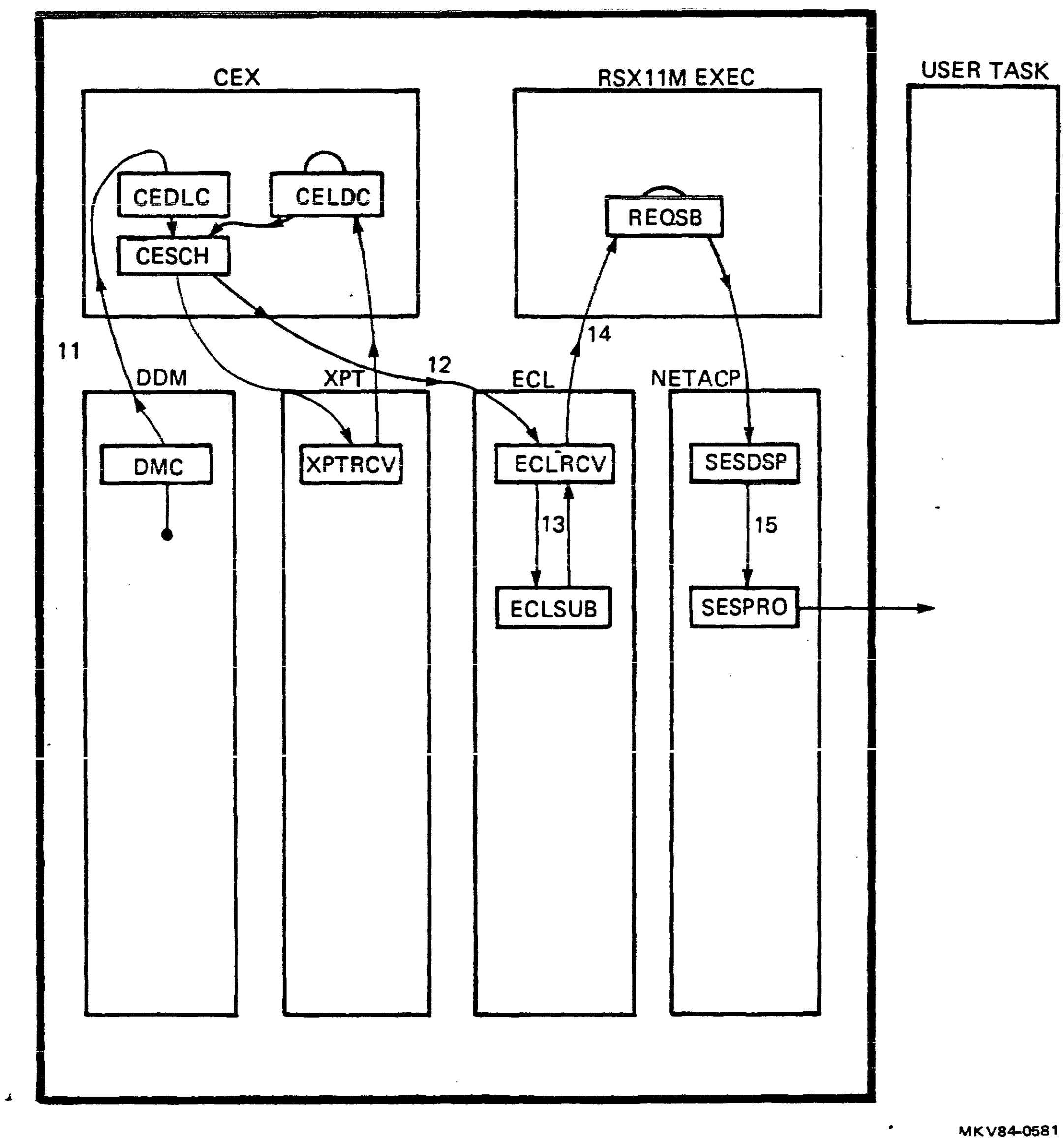


Figure 47 Receiving Connect Initiate

EXERCISES

Select the best answers for each of the following statements.

1. Under RSX-11, DECnet is implemented as a:

   a. Privileged task

   b. Core-resident process

   c. Layered product

   d. b and c only

2. The NETACP task is used to support:

   a. The N: driver

   b. The NS: driver

   c. The NW: driver

   d. The NS: and N: drivers

3. The Network Data Queue is allocated when the:

   a. Connect Initiate is issued

   b. Open the Network call is executed

   c. Connect Accept is issued

   d. None of the above

4. The user task retrieves data messages:

   a. From the Network Data Queue

   b. Directly from the NS: driver

   c. From the NETACP task

   d. b and c only

5. Communications Control Blocks are allocated from:

   a. Dynamic Storage Region

   b. Network Buffer Pool

   c. The GEN partition

   d. The remaining space in CEXPAR

6. All DECnet tasks on RSX-11 are usually mapped by:

   a. Kernel APR3

   b. Kernel APR5

   c. RSX-11 Executive

   d. Program Counter

7. All incoming messages are initially dispatched by:

   a. The RSX-11 Fork Queue dispatcher

   b. The Fork processing routine in the AUX process

   c. The DDM process to the CEX

   d. None of the above

SOLUTIONS

Select the best answers for each of the following statements.

1. Under RSX-11, DECnet is implemented as a:

   a. Privileged task

   b. Core-resident process

   c. Layered product

   d. b and c only

2. The NETACP task is used to support:

   a. The N: driver

   b. The NS: driver

   c. The NW: driver

   d. The NS: and N: drivers

3. The Network Data Queue is allocated when the:

   a. Connect Initiate is issued

   b. Open the Network call is executed

   c. Connect Accept is issued

   d. None of the above

4. The user task retrieves data messages:

   a. From the Network Data Queue

   b. Directly from the NS: driver

   c. From the NETACP task

   d. b and c only

5. Communications Control Blocks are allocated from:

   a. Dynamic Storage Region

   b. Network Buffer Pool

   c. The GEN partition

   d. The remaining space in CEXPAR

6. All DECnet tasks on RSX-11 are usually mapped by:

   a. Kernel APR3

   b. Kernel APR5

   c. RSX-11 Executive

   d. Program Counter

7. All incoming messages are initially dispatched by:

   a. The RSX-11 Fork Queue dispatcher

   b. The Fork processing routine in the AUX process

   c. The DDM process to the CEX

   d. None of the above

CONTENTS

EXAMPLES

INTRODUCTION

Simple node troubleshooting techniques are available for dealing with network problems on a node.

The problems with the network node may be grouped into the following categories:

- Memory-related -- Not enough physical memory to load the network components, not enough DSR space to run the RSX-11 with the DECnet-RSX software.

- Network configuration-related -- Communication hardware is not installed properly or not operational, or a communication line is noisy.

- Network management-related -- Two or more nodes in the network have the same node numbers; verification for the network objects is not set properly; running out of LDB/RDBs or logical links.

- Node load-related -- The RSX-11 local activity during peak periods slows down the DECnet-RSX operations and introduces undesirable effects in the network.

A set of tools is available for detecting and fixing simple network node problems. When a problem is more complicated (for example, a network software problem), more sophisticated tools may be used such as a CRT data analyzer, network trace programs, crash/network dump analysis. Source listings may be used for retrieving specific information that is not documented by the standard reference materials.

This module presents some common tools available for troubleshooting the DECnet-RSX node. Standard network management techniques (loopback tests, evaluating error counters, DTS/DTR tests, NTD displays) are presented in the Network Management module, and therefore are not discussed here. The topics covered in this module are:

- Problems with insufficient memory

- Insufficient number of network components

- Tuning and modifying the global symbols

- Crash dump and related problems

OBJECTIVES

Upon completion of this module, the specialist should be able to:

- Recognize and understand typical network problems on the DECnet-RSX node.
- Troubleshoot a given node problem using available software tools.

RESOURCES

1. DECnet-RSX Network Generation and Installation Guide
2. DECnet-RSX Release Notes
3. DECnet-RSX Overview

LOADING THE NETWORK

One way of loading the network is to use the NCP command:

>NCP SET SYSTEM

This command initiates a series of actions. The executive first loads NCP in GEN. NCP then makes an I/O request to the Network Management Driver (NMDRV). The driver calls the Network Management Volatile ACP (NMVACP). NMVACP in turn calls the Network Initialization Task (NTINIT). After performing initialization, NTINIT calls the Network Loader (NTL), to actually load the network. Thus the entire sequence is as follows:

NCP → NMDRV → NMVACP → NTINIT → NTL → Network paste up draw arrows

In performing this sequence, each task is loaded into GEN in turn. Each task can checkpoint the task that precedes it in the calling sequence, if it needs memory to run. However, because NCP has an I/O request outstanding, it cannot be checkpointed by other tasks; it remains in memory until the entire cycle is completed. Likewise, when NTL loads the network, it cannot checkpoint other tasks to make room for the network processes. This is due to certain technical features of NTL.

Generally, the loading procedure described proceeds satisfactorily. However, consider the following case. Suppose the loading procedure occurs and NCP, NMDRV, NMVACP, NTINIT, and NTL are all in memory at the same time. No checkpointing occurs because there is enough memory for all; there is even 10K of memory left over. NTL then tries to load the network. The load fails because the network takes a minimum of 16K to load and there is only 10K available. NTL tries to load in the existing space, but since it is incapable of checkpointing other tasks, it fails and prints an error message.

This problem can only occur when you attempt to load the network and, depending upon the size of your system, can be prevented in a variety of ways. These include removing extraneous application tasks and loading the network when booting the system, thus ensuring that the maximum amount of memory is available.

INSUFFICIENT MEMORY

```
>
>
>@LB:[300,1]NETINS
>* Do you want to install and load the CEX system? [Y/N]: Y
>* Do you want to install and start DECnet? [Y/N]: Y
>* On what device are the network tasks [D=DL0:] [S]:
>INS XX:[300,54]NTINIT
>INS XX:[300,54]NTL
>INS XX:[300,54]EVC
>INS XX:[300,54]NCP
>; INS XX:[300,54]CFE
>INS XX:[300,54]NMVACP
>SET /SYSUIC=[300,54]
>ASN XX:=LB:
>LOA NM:/HIGH
>ASN =LB:
>SET /SYSUIC=[1,54]
>INS XX:[300,54]NETACP
>INS XX:[300,54]NICE
>INS XX:[300,54]EVR
>INS XX:[300,54]NTD
>INS XX:[300,54]NTDEMO
>INS XX:[300,54]LIN
>INS XX:[300,54]LOO
>INS XX:[300,54]MIR
>INS XX:[300,54]NVP
>INS XX:[300,54]NFT
>INS XX:[300,54]FAL
>INS XX:[300,54]MCM
>INS XX:[300,54]RMT
>INS XX:[300,54]RMTACP/CLI=YES
>CLI /INIT=RMTACP/NULL/RESTRICT/CPR='<15><12>/RMT>/'
>SET /SYSUIC=[300,54]
>ASN XX:=LB:
>LOA HT:/HIGH
>SET /SYSUIC=[1,54]
>ASN =LB:
>INS XX:[300,54]RMHACP
>INS XX:[300,54]TLK
>INS XX:[300,54]LSN
>INS XX:[300,54]TCL
>ASN =XX:
>SET /UIC=[5,1]
>NCP SET SYS
NTL -- DLX Process File -- Partition GEN Too Fragmented
NTINIT -- Failed To Load Process DLX
NTINIT -- Clearing System
NCP -- Set failed, operation failure
       Network Initializer function failed
>NCP SET EXE STA ON
NCP -- Set failed, component in wrong state, System
>@ <EOF>
>
```

Example 1 Bringing Up DECnet-RSX
(Sheet 1 of 2)

```
 RUN $RMD

RSX-11M V4.1 BL35       (ZIRCON)  124K                         15-DEC-83 15:56:12
TASK=  *IDLE*                 FREE=   SY0:4118.
                                      DL1:DMO                           PARS
POOL=6898.:7146.:18.
     6898.:7146.:18.                                                   CEXPAR:C
                                                                       EXCOM1:C
IN:                  C E  E      T     M     F     F  P   .            EXCOM2:C
7                    E X  X      T     C     C     1  M   .            LDRPAR:T
16K                  X C  C      :     R     S     1  T   .            TTPAR :T
OUT:                 P O  O            .     R     A  .   R            DRVPAR:D
0                    A M  M            .     E     C  .   M            SYSPAR:T
OK                   R 1  2           ..     S     P  .   D            FCSRES:C
                    !!--!--!---------)->------!---->->----->           FCPPAR:D
0*******7*******15******23******31******38*******46******54******      GEN   :D
E------P-----------CC--C--T---------T-C------D----D--------------
-----------------------------------------------------------------
62******69******77******85******93******100*****108*****116*****
    <----->    <----><--->        !!-!--!--!!!-------------------!
       T          E     N         NN  N  N NN           P              ERRSEQ
       T          V     E         TT  T  T TT           O              0.
       O          C     T         ..  .  . ..           O
                  .     A         DD  X  E EA           L
                  .     C         ML  P  C VU           .
                  .     P         CX  T  L LX           .
>
```

Example 1 Bringing Up DECnet-RSX
(Sheet 2 of 2)

LOW DSR POOL

```
>
>NCP SHO EXE CHA

Node characteristics as of 9-APR-82 10:00

Executor node = 3 (BU3)

   Identification = DECNETPSIBL12TRY1, Management version = 3.0.0
   Host = 3 (BU3), NSP version = 3.2.0
   Maximum links = 15
   Routing version = 1.3.0
   Type = Routing
   Maximum address = 32
   Maximum circuits = 6, Maximum cost = 1022
   Maximum hops = 6, Maximum visits = 12
   Verification state = On

>
>
>
>RMT BU3

Connected to 'BU3    ', System type = RSX-11M
System ID: DECNETPSIBL12TRY1


>
>
>HELLO
Account or name: YANUSHPOLSKY
Password:

RSX-11M BL32   [1,54] System     BU3
10:01:26  Login  user YANUSHPOLSKY  [7,5] HT0:
09-APR-82 10:01 Logged on Terminal HT0:

Good Morning

>
>
>NCP
NCP>LOOP EXE COU 1000 WITH MIX
```

Cause [illegible]

Example 2 Low DSR Pool
(Sheet 1 of 3)

```
>
>
>ACT /ALL
LDR...
NETACP
EVP...
EVC...
MCR...
...MCR
F11ACP
PMT...
RMHACP
RMTACP
...NCP
LOO.0
MIR.0
>
>
>
>NTD

9-APR-82 10:01:57 -- Warning -- Pool is critically low
Total free pool = 600. bytes
Largest fragment = 236. bytes

NTDEMO -- Unable to connect to remote server.


9-APR-82 10:01:57 -- Low pool - Please exit active tasks - BU3


9-APR-82 10:01:57 -- Low pool - Please exit active tasks - BU3


>
NCP>^Z
>NTD
9-APR-82  10:02:17     Node: BU3    , DECNETPSIBL12TRY1
                  Mapped, RSX-11M, Full Routing Node          Allocation
  LNKS   4 / 15 XXXXXXXXXX!----------------------------      Failures:
  CCB    0 / 16 ----------------------------------------        0
  SDB    0 / 31 ----------------------------------------        0
  LDB    0 / 53 ----------------------------------------        0 :    0

           Remote Nodes:                              Network Tasks:
  Remote      Circuit   Hop/Cst Lnk/Dly  Task      TI   Links   Mbox  XMT  RCV

                                         EVP...   COO:   0       0     0    0
                                         RMTACP   COO:   1       0     0    1
                                         RMHACP   COO:   1       0     0    1
                                         ...NTD   HTO:   1       0     0    1
                                         NTD...   COO:   1       0     0    0
```

Example 2 Low DSR Pool
(Sheet 2 of 3)

```
>
>
>
>NCP SHO EXE COU

Node counters as of 9-APR-82 10:03

Executor node = 3 (BU3)

        122  Seconds since last zeroed
     262125  Bytes received
     262125  Bytes sent
       8672  Messages received
       8577  Messages sent
          4  Connects received
          4  Connects sent
          0  Response timeouts
          1  Received connect resource errors
          6  Total maximum logical links active
          0  Aged packet loss
          0  Node unreachable packet loss
          0  Node out-of-range packet loss
          0  Oversized packet loss
          0  Packet format error
          0  Partial routing update loss
          0  Verification reject
          6  Node maximum logical links active
          0  Total received connect resource errors

>
>
>
>
>
>EXIT RMT

RMT -- Control returned to node BU3::
10:04:20  Logout user [7,5] HT0:

>
>
```

Example 2 Low DSR Pool
(Sheet 3 of 3)

INSUFFICIENT NUMBER OF NETWORK COMPONENTS

```
>
>
>
>NCP SHOEXEC CHAR
NCP -- Command not accepted, unrecognized function or option
>NCP SHO EXEC CHAR

Node characteristics as of 19-DEC-83 15:42:40

Executor node =   231 (ZIRCON)

   Identification = ZIRCON_ENCRUSTED_TWEEZERS, Management version = 4.0.0
   Host =   231 (ZIRCON), Loop count = 1, Loop length = 40
   Loop with = Mixed, NSP version = 4.0.0
   Maximum links = 2, Routing version = 2.0.0, Type = Nonrouting IV
   Maximum circuits = 1
   Segment buffer size = 576, Verification state = On

>
>
>
>RMT ZIRCON

Connected to 'ZIRCON', System type = RSX-11M
System ID: ZIRCON_ENCRUSTED_TWEEZERS


>HEL RYAN
Password:

RSX-11M BL35   [1,54] System     ZIRCON
19-DEC-83 15:42 Logged on Terminal HT0:

Good Afternoon

15:42:54  Login  user RYAN          [5,1] HT0:

        ********************************************
        *                                          *
        *                                          *
        *       WELCOME ABOARD ZIRCON              *
        *                                          *
        *            Any Problems?                 *
        *                                          *
        *               See                        *
        *                                          *
        *            Pete Della Pelle              *
        *                                          *
        *            Dan Ryan                      *
        *                                          *
        ********************************************


>
>
>
>
>
>NTD
NTDEMO -- Unable to connect to remote server.

>
>
```

Example 3 Insufficient Number of Network Components
(Sheet 1 of 4)

```
>
>NTD
NTDEMO -- Unable to connect to remote server.

>
>
>
>NCP SHO EXEC COU

Node counters as of 19-DEC-83 15:43:28

Executor node =   231 (ZIRCON)

         40  Seconds since last zeroed
       1995  Bytes received
       1995  Bytes sent
        261  Messages received
        261  Messages sent
          1  Connects received
          1  Connects sent
          0  Response timeouts
          0  Received connect resource errors
          2  Total maximum logical links active
          0  Aged packet loss
          0  Node unreachable packet loss
          0  Node out-of-range packet loss
          0  Oversized packet loss
          0  Packet format error
          0  Partial routing update loss
          0  Verification reject
          2  Node maximum logical links active
          2  Total received connect resource errors

>
>
>NCP LOOP EXEC
NCP -- Loop failed, Mirror connect failed, network resources
       Unlooped count = 1
>
>ACT /ALL
LDR...
RMDEMO
NETACP
EVC...
MCR...
...MCR
F11ACP
FMT...
RMHACP
RMTACP
>
>
>EXIT RMT

RMT -- Control returned to node ZIRCON::

>
 15:44:19  Logout user [5,1] HT0:

>
>
>NCP
NCP>LOOP EXEC
NCP>
NCP>^Z
>
```

Example 3 Insufficient Number of Network Components
(Sheet 2 of 4)

```
>INS [300,54]CFE
>E^U
>CFE
Enter filename: [300,54]CETAB.MAC
CFE>LIST EXEC

Executor characteristics as of 19-DEC-83 15:45:02

  Identification = ZIRCON_ENCRUSTED_TWEEZERS
  Name = ZIRCON, Address =   231, Host =   231
  Maximum links = 2,  Maximum node counters = 4
  Type = Nonrouting IV, Verification state = On
  Segment buffer size = 576

CFE>DEFINE EXEC MAX LINKS 10
CFE>^Z
>NCP SET EXEC STATE OFF
>

 Event type 68.14, Normal usage terminated
Occurred 19-DEC-83 15:45:53 on node   231 (ZIRCON)
Circuit DMC-0


 Event type 2.0, Local node state change
Occurred 19-DEC-83 15:45:56 on node   231 (ZIRCON)
Reason for state change: Operator command, Old node state = On
New node state = Off


>
>NCP CLE SYS ALL
>
>
>NCP SET SYS ALL
>
>
>NCP SET EXEC STATE ON

 Event type 2.0, Local node state change
Occurred 19-DEC-83 15:48:13 on node   231 (ZIRCON)
Reason for state change: Operator command, Old node state = Off
New node state = On

>

 Event type 4.10, Circuit up
Occurred 19-DEC-83 15:48:16 on node   231 (ZIRCON)
Circuit DMC-0
Node address =   65 (RSXHUB)


>
>
>RMT ZIRCON

Connected to 'ZIRCON', System type = RSX-11M
System ID: ZIRCON_ENCRUSTED_TWEEZERS


>HEL RYAN
Password:

RSX-11M BL35   [1,54] System     ZIRCON
19-DEC-83 15:48 Logged on Terminal HT0:
```

Example 3 Insufficient Number of Network Components
(Sheet 3 of 4)

```
Good Afternoon

15:48:34  Login  user RYAN          [5,1] HT0:

        ********************************************
        *                                          *
        *                                          *
        *        WELCOME ABOARD ZIRCON             *
        *                                          *
        *             Any Problems?                *
        *                                          *
        *                See                       *
        *                                          *
        *            Pete Della Pelle              *
        *                                          *
        *            Dan Ryan                      *
        *                                          *
        ********************************************


>
>
>
>
>NTD
19-DEC-83 15:48:46   Node: ZIRCON(  231) ZIRCON_ENCRUSTED_TWEEZERS
                    RSX-11M, Non-Routing Node, DECnet V4.0
  LNKS    4 / 10 XXXXXXXXXXXXXXXX!----------------------   Alloc Fails:
  CCB     4 /  9 XXXXXXXXXXXXXXXXX!---------------------      0
  SDB     0 / 11 ---------------------------------------      0
  LDB     4 /  9 XXXXXXXXXXXXXXXXX!---------------------      0 :    0

         Known Circuits:                         Network Tasks:
Circuit  Cost Size  Adj Node/Status         Task    TI   Links MBX XMT RCV

DMC-0       5   576  RSXHUB(    65)         NTD...  CO0:     1   0   0   0
                                            ...NTD  HT0:     1   0   0   1
                                            RMHACP  CO0:     1   0   0   1
                                            RMTACP  CO0:     1   0   0   1




>
>
>
>EXIT RMT

RMT -- Control returned to node ZIRCON::

>
 15:48:58  Logout user [5,1] HT0:

>
>
>

 BYE
Have a Good Afternoon
19-DEC-83 15:49 TT0:  logged off ZIRCON
>
```

Example 3 Insufficient Number of Network Components
(Sheet 4 of 4)

TUNING BY MODIFYING THE GLOBAL SYMBOLS

Usually command files for the network components (NSP, NETACP, and so on) contain global symbols. In some cases, by changing the values of these symbols, you can improve the overall node performance for a specific set of applications. For example, a command file for building the NETACP contains symbol $V2TIM that corresponds to the general delivery queue timeout. The default value for this symbol is 17 (seconds). So, if the receiver task could not open a mailbox within 3 seconds since the connect initiate message was placed by the NETACP into the general delivery queue, the message would be dropped by the NETACP and deleted from the queue. Three seconds is usually enough time to activate the receiver task and open its mailbox, but sometimes (for example, during peak load on the receiver node) it may take longer and connect would not occur. By modifying the value of $V2TIM and rebuilding the NETACP, this problem can be corrected.

Do not change any of the global symbols unless you completely understand their functions. The original command file has to be saved, so restoration would be possible.

The command files for building the DECnet-RSX major components are in the Appendix ~~A~~ 6.

CRASH DUMP AND RELATED PROBLEMS

The common reasons for DECnet crashes are the same as for system crashes:

- A privileged task with an addressing mode error could corrupt part of the executive
- Wrong format in the data structure for user-written device drivers
- Coding error in user-written device drivers
- Performance degeneration due to lack of dynamic storage region
- Unintended loop that uses up the dynamic storage region, thus locking out all other tasks including MCR
- Corrupted task images due to erroneous QIO programs to disks
- Sporadic hardware interrupts such as a long, hanging lead to DL11
- Environmental problems such as electrostatics and temperatures that affect hardware performance
- System software bug (you must fill in SPR)
- Hardware problem
- Running a task whose load will fail (for example, task image deleted, disk down, and so on)

To obtain necessary information for successful crash analysis, the following utilities are available:

- Crash Dump Analyzer (CDA)
- Network Dump Analyzer (NDA)
- Communication Executive Dump (CEDUMP) (unsupported)

NDA as well as CDA operates in the following manner:

- It reads the contents of the memory dump created by the executive crash dump routine.
- It then analyzes the dump in accordance with information contained in a combined executive/network symbol table file.
- Finally, it formats and outputs the dump to a line printer or a listing file for your evaluation.

CEDUMP, on the other hand, formats and outputs the CEX dump on the running system.

only 1 appendix. don't need the "A"

APPENDIX ~~A~~
Important Files for DECnet-RSX

Command files for building major DECnet-RSX components

```
;
; CEX (Communications Executive) task build command file
;
OU:[300,54]CEX.TSK/-MM/-HD,MP:[300,34]CEX/-SP,OU:[300,54]CEX=
IN:[130,24]CEX/LB:CEBUF1:CESCH:CETIM:CELOG1
IN:[130,24]CEX/LB:CEDDMNDM:CEDLC:CELLC:CESUB1:CEXCM
OU:[300,054]RSX11M.STB
LB:[1,1]EXELIB/LB/SS
/
STACK=0
PAR=CEXPAR:112000:6000
GBLDEF=DLC$OV:0
GBLDEF=CRC$OV:0
GBLDEF=PD$AUX:0
GBLDEF=$CXFLG:14
GBLDEF=$DDMAN:0
/
;
; AUX (Communications Executive Auxiliary routine LLC) task build command
;
OU:[300,54]AUX.TSK/-MM/-HD,MP:[300,34]AUX/-SP,OU:[300,54]AUX=
IN:[131,24]AUX/LB:AXDSPP:AXBFR:AXSCH:AXSUB:AXTIM
OU:[300,54]CEX.STB/SS
OU:[300,054]RSX11M.STB/SS
/
STACK=0
PAR=GEN:120000:20000
//
;
; CETAB task build command file
;
OU:[300,54]CETAB.TXK/-MM/-HD,MP:[300,34]CETAB/-SP,OU:[300,54]CETAB=
OU:[300,24]CETAB
/
STACK=0
PAR=CTBPAR:0:10000
//
;
; DMC DDM task build command file
;
OU:[300,54]DMC.TSK/-MM/-HD,MP:[300,34]DMC/-SP,OU:[300,54]DMC=
IN:[131,24]DDM/LB:DMC
OU:[300,54]AUX.STB/SS
OU:[300,54]CEX.STB/SS
/
STACK=0
PAR=GEN:120000:20000
//
```

```
;
; NTINIT (network initializer) task build command file
;
OU:[300,54]NTINIT.TSK/MM/CP/PR/-FP,MP:[300,34]NTINIT/-SP,OU:[300,54]NTINIT
OU:[300,24]NTINITBLD/MP
PRI=55
PAR=GEN:0:0
STACK=50
UNITS=2
ASG=TI:1
GBLDEF=FE.CEX:20000
//
;
; NTL (network loader) task build command file
;
OU:[300,54]NTL.TSK/MM/PR:5/SL/-FP,MP:[300,34]NTL/-SP=
OU:[300,24]NTLBLD/MP
/
PAR=GEN:0:0
GBLDEF=.CBIAS:0
GBLDEF=FE.CEX:20000
GBLDEF=$NTCFS:0
GBLDEF=$XSNA:0
GBLDEF=$ASUMR:0
GBLDEF=$DEUMR:0
TASK=NTL...
PRI=60
STACK=144
UNITS=3
ASG=SY:1
ASG=TI:2
//
;
; EVL (event logger) task build command file
;
OU:[300,54]EVL.TSK/-MM/-HD,MP:[300,34]EVL/-SP,OU:[300,54]EVL=
OU:[131,24]EVL/LB:EVL
OU:[300,24]EVLDAT
OU:[300,54]CEX.STB/SS
LB:[1,1]EXELIB/LB/SS
/
STACK=0
PAR=GEN:120000:20000
//
```

```
;
; EVC (event collector) task build command file, overlaid
;
OU:[300,54]EVC.TSK/PR/CP,MP:[300,34]EVC/-SP=OU:[300,24]EVCBLD/MP
/
PAR=GEN:0:0
TASK=EVC...
PRI=198
//
;
; NCP (network control program) task build command file for
; RSX-11M/M-PLUS
;
OU:[300,54]NCP.TSK/PR:0/-FP/CP/MM,MP:[300,34]NCP/-SP=
OU:[300,24]NCPBLD/MP
/
TASK=...NCP
UNITS=8
ASG=TI:1:2:3
PAR=GEN:0:0
STACK=200
//
```

```
;
; NMVACP, NMDRV (network management ACP and driver) task build command file
;
; NMVACP
;
OU:[300,54]NMVACP.TSK/MM/PR/-SE/-FP/CP,MP:[300,34]NMVACP/-SP,OU:[300,54]NM
OU:[300,24]NMVACPBLD.ODL/MP
/
TASK=NMVACP
PAR=GEN:0:0
PRI=100
STACK=100
UNITS=2
GBLDEF=$TMLUN:1
GBLDEF=$MBLUN:2
GBLDEF=$TMEFN:4
;
; The psect "$$$XXX" contains the context area buffer pool.  Each simult
; user requires a 506 byte context block. Space for additional users can
; be obtained by changing the EXTSCT below and rebuilding, by installing
; the with "/INC=n", or by installing the task as checkpointable, so it
; automatically extent itself via the EXTK$ directive at run time.
;
; Space is originally allocated for one user.
;
EXTSCT=$$$XXX:506
/
;
; NMDRV (NM:)
;
OU:[300,54]NMDRV.TSK/-MM/-HD,MP:[300,34]NMDRV/-SP,OU:[300,54]NMDRV=
OU:[300,24]NMDRV,NMTAB
OU:[300,54]CEX.STB/SS
LB:[1,1]EXELIB/LB/SS
/
STACK=0
PAR=GEN:120000:4000
//
```

```
;
; XPT (transport process LLC process) task build command file (end-node)
;
OU:[300,54]XPT.TSK/-MM/-HD,MP:[300,34]XPT/-SP,OU:[300,54]XPT=
IN:[131,24]XPTE.OLB/LB:XPEDAT:XPTEVT:XPELIN:XPERCV:XPESUB
IN:[131,24]XPTE.OLB/LB:XPETIM:XPTRC:XPEXMT
OU:[300,54]CEX.STB/SS
OU:[300,54]AUX.STB/SS
LB:[1,1]EXELIB/LB
/
STACK=0
PAR=GEN:120000:20000
;
; Define Routing parameters
;
GBLDEF=PC$TH2:4                  :DLC queue length
GBLDEF=PC$IPL:6                  :Input packet limiter
//
;
; ECL (network services LLC process/driver) task build command file
;
OU:[300,54]ECL.TSK/-MM/-HD,MP:[300,34]ECL/-SP,OU:[300,54]ECL=
IN:[131,24]ECLSLI/LB:ECLACK:ECLBUF:ECLCTR:ECLDAT:ECLINI:ECLPRO
IN:[131,24]ECLSLI/LB:ECLQIO:ECLRCV:ECLSLI:ECLSUB:ECLTIM:ECLUSR
IN:[131,24]ECLSLI/LB:ECLXMT
OU:[300,54]CEX.STB/SS
LB:[1,1]EXELIB/LB
/
STACK=0
PAR=GEN:120000:20000
;
; Define ECL parameters
;
GBLDEF=LR$CNT:5                       ;ECL retry count
GBLDEF=LR$MAS:3                       ;Maximum message reassembly queue
                                      ;length
GBLDEF=LR$LST:2                       ;Link service threshold
GBLDEF=LT$INF:12                      ;Outgoing connect timer
GBLDEF=LT:LPT:36                      ;ECL idle timer
/
;
; ECL (network services LLC process/driver tables) task build command file
;
OU:[300,54]ECLTAB.TSK/-MM/-HD,,OU:[300,54]ECLTAB=
OU:[300,24]ECLTAB
OU:[300,54]ECL.STB/SS
/
STACK=0
PAR=GEN:0:1000
//
```

```
;
; NETACP (network services ACP) task build command file
;
OU:[300,54]NETACP.TSK/AC/-FP/AL/MM,MP:[300,34]NETACP/-SP,OU:[300,54]NETACP
IN:[131,24]NETACPSLI/LB:SESCON:SESCTL:SESCTR:SESDAT:SESDIS
IN:[131,24]NETACPSLI/LB:SESDMO:SESDSP:SESINI:SESMN:SESQIO:SESPRO
IN:[131,24]NETACPSLI/LB:SESSLI:SESSUB:SESTCB:SESTIM:SESUSR
LB:[1,1]EXELIB/LB/SS
OU:[300,54]CEX.STB/SS
OU:[300,54]AUX.STB/SS
OU:[300,54]ECL.STB/SS,XPT.STB/SS
/
TASK=NETACP
PAR=GEN:0:0
PRI=200
STACK=32
UNITS=0
GBLDEF=$V2TIM:17
GBLDEF=N$$SMC:10
//
;
;  NICE (network information/control exchange server) task build command file
;  RSX-11M and RSX-11M-PLUS
;
OU:[300,54]NICE.TSK/MM/PR:5/-FP/CP,MP:[300,34]NICE/-SP=
OU:[300,24]NICEBLD/MP
/
PAR=GEN:0:0
TASK=NIC$$$
UNITS=4
STACK=100
//
;
; EVR (event receiver) task build command file
;
OU:[300,54]EVR.TSK/PR/CP,MP:[300,34]EVR/-SP=
OU:[300,24]EVRBLD/MP
/
PAR=GEN:0:0
PRI=197
TASK=EVR$$$
//
```

```
;
; NTD (node state display user task) task build command file
;
OU:[300,54]NTD.TSK/MM/PR/-FP/CP,MP:[300,34]NTD/-SP=
;
; To allow file operations in NTD, place a semicolon before the first line
; and remove the semicolon from before the second line.
;
IN:[135,24]NTD/LB:NTD:DISPLY
IN:[135,24]NTD/LB:NTD:DISPLW
;
OU:[300,54]RSX11M.STB/SS
/
PAR=GEN:0:0
TASK=...NTD
STACK=100
;
; To allow file operations in NTD, place a semicolon before this line.
;
GBLDEF=$DSWRT:0
//
;
; NTDEMO (Node state display server) task build command file
;
OU:[300,54]NTDEMO.TSK/MM/PR/-FP/CP,MP:[300,34]NTDEMO/-SP=
IN:[135,24]NTD/LB:NTDEMO
LB:[1,1]EXELIB/LB/SS
OU:[300,54]CEX.STB/SS
OU:[300,54]RSX11M.STB/SS
/
PAR=GEN:0:0
TASK=NTD...
STACK=100
UNITS=5
EXTTSK=2700
//
```

```
;
; LIN (link watcher) task build command file
;
OU:[300,54]LIN.TSK/MM/PR/-FP/-SG/CP,MP:[300,34]LIN/-SP=
OU:[300,24]LINBLD/MP
/
PAR=GEN:0:0
TASK=LIN$$$
STACK=100
PRI=100
UNITS=4
;
; $LTXTM - number of seconds before DLX receive times out
; $LTXRR - number of times to retry receive
;
GBLDEF=$LTXTM:22
GBLDEF=$LTXRR:5
EXTSCT=..POOL:2734
//
;
; LOO (loop tester) task build command file
;
OU:[300,54]LOO.TSK/MM/PR/-FP/CP,MP:[300,34]LOO/-SP,OU:[300,54]LOO=
IN:[135,24]LOOP/LB:LOOPER:LOOPAR:LOOPNI
IN:[130,24]NETLIB/LB:XXBUF
IN:[135,24]DECMAN/LB:MAPADD:MAPNAM:TRNLOG
IN:[130,24]NETLIB/LB:NPARS
OU:[300,54]CEX.STB/SS
/
PAR=GEN:0:0
TASK=LOO$$$
PRI=100
STACK=100
;
; LBIT = BITS/BYTE including protocol overhead
; LBAU = BITS/SEC actual line speed (16 bit max)
; LLDY = Modem turnaround delay
;
GBLPAT=LOOPER:LBIT:13
GBLPAT=LOOPER:LBAU:454
GBLPAT=LOOPER:LLDY:16
EXTSCT=..POOL:1000
//
```

```
;
; MIR (loopback mirror) task build command file
;
OU:[300,54]MIR.TSK/MM/-FP/-SG/CP,MP:[300,34]MIR/-SP,OU:[300,54]MIR=
IN:[135,24]MIR/LB:MIRROR
IN:[130,24]NETLIB/LB:XXBUF
/
PAR=GEN:0:0
TASK=MIR$$$
STACK=36
//
;
; NVP (network verification) task build command file
;
OU:[300,54]NVP.TSK/MM/PR/CP/-FP,MP:[300,34]NVP/-SP=
OU:[300,24]NVPBLD/MP
/
PAR=GEN:0:0
TASK=NVP...
STACK=60
PRI=150
UIC=[1,1]
UNITS=3
GBLDEF=ENCRPT:0         ;PASSWORD ENCRYPTION ROUTINE (0 = NOT USED)
//
;
; NFT (network file access utility) task build command file, overlaid
;
OU:[300,54]NFT.TSK/MM/CP/-FP,MP:[300,34]NFT/-SP=
OU:[300,24]NFTBLD/MP
/
PAR=GEN:0:0
TASK=...NFT
UNITS=6
ASG=TI:3
ASG=TI:4
STACK=100
;
; Define NFAR parameters
GBLDEF=$NFRSZ:1004      ;Record buffer size
GBLDEF=$NFNMB:2         ;Buffering level
EXTSCT=$$FSR1:7354      ;FSR extension
GBLDEF=$TRLUN:6         ;Trace LUN
;
; Default directory listing width
GBLDEF=LI$WID:110
;
; Define logical link timeout interval
GBLDEF=$LLTIM:5
//
```

```
;
; FAL (Network FCS file access listener) task build command file, non-overlaid
;
OU:[300,54]FAL.TSK/PR:0/MM/-FP/CP,MP:[300,34]FAL/-SP=
IN:[133,24]FALFCS/LB:FALFCS:FALDEF:FALFLB:FALSTB:FALSUB
IN:[133,24]FALFCS/LB:FALACC:FALATT:FALCMP:FALCNF:FALCTL
IN:[133,24]FALFCS/LB:FALDAT:FALDEL:FALDIR:FALDTM:FALERR
IN:[133,24]FALFCS/LB:FALEXE:FALGET:FALGND:FALINI:FALNAM
IN:[133,24]FALFCS/LB:FALOPN:FALPRO:FALPCK
IN:[133,24]FALFCS/LB:WLDPRS:WLDSUB:.CSI3
IN:[130,24]NETLIB/LB:DIRW
;
; These modules are assembled during DECnet generation:
;  FALBUF - Contains the buffers and per link data bases.
;  FALPRM - Contains access parameters if verification is not
;  supported.
;
OU:[300,24]FALBUF
OU:[300,24]FALPRM
;
```

```
; Determine whether writing the log file "LB:[1,4]FAL.LOG" is supported:
;  FALLOG - Included only if log file is supported.
;  FALDLG - Included only if no log file support.
;
IN:[133,24]FALFCS/LB:FALLOG
;IN:[133,24]FALFCS/LB:FALDLG
;
; Determine whether tracing FAL messages to "FT:[1,4]FAL.TRC" is supported:
;  FALTRC - Included only if message tracing is supported.
;  FALDTR - Included only if no message trace support.
;
;IN:[133,24]FALFCS/LB:FALTRC
IN:[133,24]FALFCS/LB:FALDTR
/
PAR=GEN:0:0
TASK=FAL...
STACK=100
UIC=[1,1]
PRI=100
GBLDEF=..D2CT:0
;
; Allocate number of units equal to:
;  2*<number links>+4
;
; LUN 1 = Mailbox,    LUN 2 = Trace file
; LUN 3 = Log file,   LUN 4 = Scratch
;
UNITS=24
;
; Allocate file storage region for each link.  Add one if message tracing
; supported.
;
ACTFIL=10
;
; Define "Guest" UIC.
;
GBLDEF=$UICG:200
GBLDEF=$UICU:200
;
//
```

```
;
; MDM (command file submission task) task build command file
;
OU:[300,54]MCM.TSK/MM/PR/-FP/SL/CP,MP:[300,34]MCM/-SP=
IN:[133,24]FALFCS/LB:MCM
OU:[300,54]RSX11M.STB
/
TASK=.CMTS.
PAR=GEN:0:0
STACK=30
UNITS=2
;
; The following assignment controls the device from which the command file
; will be submitted to Indirect. It may be changed after the task has been
; installed using the MCR command.
;       >REA .CMTS. 1 xxn:
; where xxn: is the new device for command submission
;
ASG=TT0:1
//
```

```
;
; RMT, RMTACP (remote network terminal user task) task build command file
;
; RMT
;
OU:[300,54]RMT.TSK/MM/PR/CP/-FP/-SE,MP:[300,34]RMT/-SP=
IN:[140,24]RMT/LB:RMT
OU:[300,54]RSX11M.STB/SS
LB:[1,1]EXELIB/LB/SS
/
PAR=GEN:0:0
TASK=...RMT
STACK=30
UNITS=1
GBLDEF=RM$HST:0          ;1 = allow recursive RMT usage, 0 = disallow
/
;
; RMTACP
;
OU:[300,54]RMTACP.TSK/CP/MM/PR/-SE/-FP/CP,MP:[300,34]RMTACP/-SP=
OU:[300,24]RMTACP
IN:[140,24]RMT/LB:FMTPCD:RMTAST:RMTCCP:RMTCMD
IN:[140,24]RMT/LB:RMTGND:RMTICP:RMTINI:RMTKCP
IN:[140,24]RMT/LB:RMTMCL:RMTNIO:RMTOCP:RMTRCP
IN:[140,24]RMT/LB:RMTRSC:RMTSRV:RMTTCP:RMTTIO
OU:[300,54]RSX11M.STB/SS
LB:[1,1]EXELIB/LB/SS
/
TASK=RMTACP
UNITS=11
PAR=GEN:0:0
PRI=99
GBLDEF=RM$CLI:1         ;Define RMT as an alternate CLI
//
```

```
;
; RMHACP, HTDRV (network command terminal ACP/driver) task build command file
;
; RMHACP task build command file
;
OU:[300,54]RMHACP.TSK/MM/PR/-SE/-FP/CP,MP:[300,34]RMHACP/-SP=
OU:[300,24]RMHACP
IN:[140,24]RMHACP/LB:RMHPCD:RMHAST:RMHCAC:RMHDIS:RMHGND:RMHINI:RMHSRV
IN:[140,24]RMHACP/LB:RMHMCL:RMHNIO:RMHRCP:RMHSNK:RMHSNP:RMHTCP:RMHTMO
OU:[300,54]CEX.STB/SS
/
PAR=GEN:0:0
TASK=RMHACP
PRI=100
UNITS=6
;
; Define RM$PRV as U2.PRV if RMHACP is to make the terminal non-privileged
; when a new user logs on, usually defined as zero.
;
GBLDEF=RM$PRV:0
/
OU:[300,54]HIDRV.TSK/-MM/-HD,MP:[300,34]HTDRV/-SP,OU:[300,54]HTDRV=
IN:[140,24]RMHACP/LB:HTDRV
OU:[300,24]HTTAB
OU:[300,54]RSX11M.STB/SS
LB:[1,1]EXELIB/LB/SS
/
STACK=0
PAR=GEN:120000:4000
//
```

```
;
; TLK (network talk utility) task build command file
;
OU:[300,54]TLK.TSK/MM/CP/PR:0/-FP,MP:[300,34]TLK/-SP=
IN:[133,24]TLK/LB:TLK:TLKIMP:TLKNET:TLKPRS
IN:[133,24]TLK/LB:TLKCRT
IN:[133,24]TLK/LB:TLKDIA
IN:[133,24]TLK/LB:TLKFSR
IN:[135,24]NTD/LB:DISPLY
LB:[1,1]EXELIB/LB
IN:[130,24]NETLIB/LB.GCL:QFILE
/
TASK=...TLK
STACK=100
PAR=GEN:0:0
PRI=50
UNITS=4
ASG=TI:1:2
;
; TLK Dialogue Support: If you want support for this feature, you must have
; both Read-After-Prompt and Breakthrough-Write as Terminal Driver features.
; This mode will not work without both of these.
;
; To enable Dialogue support, place a semicolon before the GBLDEF and include
; the "TLK/LB:TLKDIA" module. To disable this support, remove the semicolon
; and do not include the module.
;
; GBLDEF=$DIALG:0
;
; TLK Video Dialogue Mode Support: If you want support for this feature, you
; must have Read-After-Prompt, Breakthrough-Write, and Attach-For-AST as
; Terminal Driver features. This mode will not work without all of these.
;
; To enable Video Dialogue support, place a semicolon before the GBLDEF and
; include the "TLK/LB:TLKCRT" module. To disable this support, remove the
; semicolon and do not include the module.
;
; GBLDEF=TLKCRT:0
;
; To enable Indirect Command File support, place a semicolon before the
; GBLDEF (RSX-11M/M-PLUS only) and include the "TLK/LB:TLKFSR" module.
; To disable, remove the semicolon and do not include the module.
;
;GBLDEF=$TLKFSR:0
;
; If you have an FCS Resident Library, you may wish to link to it.
;
//
```

```
;
; LSN (network talk server) task build command file, multi-copy
;
OU:[300,54]LSN.TSK/MM/PR:0/-FP/CP,MP:[300,34]LSN/-SP=
IN:[133,24]LSN2/LB:LSN:LSNSNG:LSNDLG,LB:[1,1]EXELIB/LB
IN:[133,24]TLK/LB:TLKCRT,[135,24]NTD/LB:DISPLY
/
TASK=LSN$$$
PAR=GEN:0:0
PRI=100
UNITS=3
GBLPAT=LSN:$ASK:1
STACK=100
//
;
; TCL (remote task control server) task build command file
;
OU:[300,54]TCL.TSK/MM/PR/CP,MP:[300,34]TCL/-SP=
IN:[134,24]NETFOR/LB:TCL
OU:[300,24]TCLPWD
OU:[300,54]RSX11M.STB/SS
/
PAR=GEN:0:0
TASK=TCL...
STACK=100
UIC=[1,1]
UNITS=1
//
```

```
;
; DLX (direct line access, LLC/driver) task build command file
;
OU:[300,54]DLX.TSK/-MM/-HD,MP:[300,34]DLX/-SP,OU:[300,54]DLX=
IN:[131,24]DLX/LB:DLXCEX:DLXCHR:DLXCTL:DLXDAT:DLXLIN
IN:[131,24]DLX/LB:DLXQIO:DLXSUB
OU:[300,54]CEX.STB/SS
LB:[1,1]EXELIB/LB/SS
/
STACK=0
PAR=GEN:120000:20000
GBLDEF=$$BUF:1170
EXTSCT=$$$DLX:54
/
;
; DLX (direct line access, LLC/driver) tables module build command file
;
OU:[300,54]DLXTAB.TSK/-MM/-HD,,OU:[300,54]DLXTAB=
OU:[300,24]DLXTAB
OU:[300,54]DLX.STB/SS
/
STACK=0
PAR=GEN:0:1000
//
;
; HLDTAB (HLD data table) task build command file
;
[300,54]HLDTAB/-HD/-MM=[300,24]HLDTAB
/
STACK=0
PAR=XXXX:0:10000
//
```

CONTENTS

FIGURES

INTRODUCTION

Understanding various performance parameters for the nodes and network helps the specialist to control and manage the network. This module summarizes performance considerations for the DECnet network.

OBJECTIVES

To tune and troubleshoot the network, the specialist must be able to:

- Interpret the following performance parameters:
  - CPU utilization
  - Line utilization
  - System and user buffer availability
  - Size/number considerations
  - Throughput, response time and message rate
- Practice the general procedures for upgrading network applications.
- Use available performance information for evaluating network activity.

RESOURCES

1. Martin, James. Computer Networks and Distributed Processing, Part 1, Chapter 3 and 8. Englewood Cliffs, NJ: Prentice Hall, 1981

2. Martin, James. Design and Strategy for Distributed Data Processing, Part 5, Chapter 35. Englewood Cliffs, NJ: Prentice Hall, 1981

VARIETY OF NETWORK REQUIREMENTS

The type of network used depends on a variety of parameters related to:

- Type of application
  - Airline reservation system
  - Bank loan service
  - Corporate data processing
  - Common carrier service
  - Office automation
- Cost versus response time, reliability, data integrity, and security
  - Private or public network
  - Vertical or horizontal topology
  - Local area network
  - Switching technique
  - Multiple access to communications media
  - Bandwidth
  - Transport functions
  - Node coupling
- Network growth
  - New communications media
  - Geographical distribution
  - New applications

FACTORS AFFECTING PERFORMANCE

Factors that affect performance are grouped into the following categories:

- Configuration related
  - CPU capacity
  - Communication line characteristics
  - System and user buffering levels
  - Routing
  - Multipoint
  - Ethernet
- Application related
  - Message size and variations
  - Message rate
  - Peak loading design
  - Application mix

CPU Capacity

The capacity of the processor is important in determining whether a node can accommodate the desired load. The load must be able to accommodate:

- Local processes
- Operating system overhead
- Communications software overhead
- Communications interrupt processing

Figure 1 presents a plot of CPU utilization for two different types of processors using the same user message size over a single communications line. Note the following:

- It takes less load for a more powerful CPU to be able to accommodate a given network workload.
- Neither one of the two CPU's is loaded to 100%, so any congestion that occurs in the system is due to the communications line.
- If more than one communications line is active on a processor, CPU utilization increases linearly until total CPU utilization reaches 70%. After that, CPU utilization becomes more difficult to predict.
- CPU utilization of over 70% for both network and local processing should be considered as a potential performance problem.

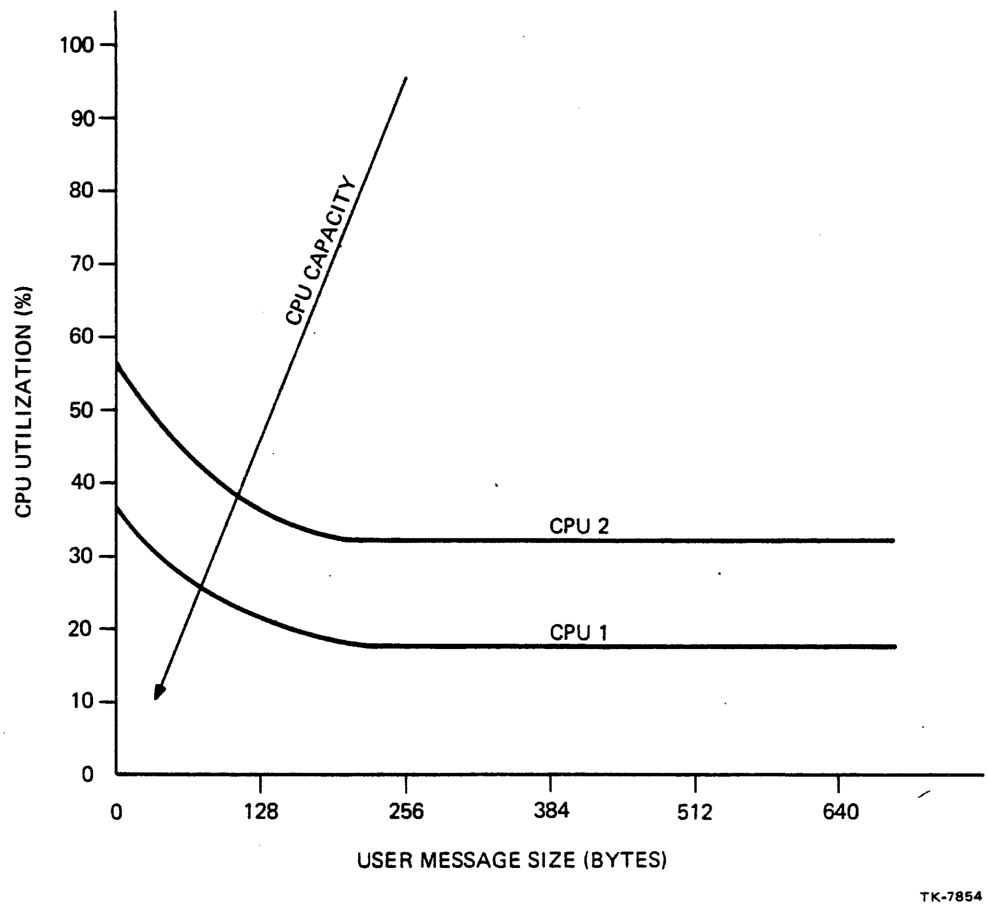


Figure 1 CPU Utilization versus User Message Size for Different Types of Processors

If two processors are connected over a communications line by using character-interrupt devices (DUP-11), it requires more CPU activity to support communications for each processor than for processors connected by a microprocessor-based device (DMR-11). Figure 2 illustrates this by using the two types of devices on the same CPU.

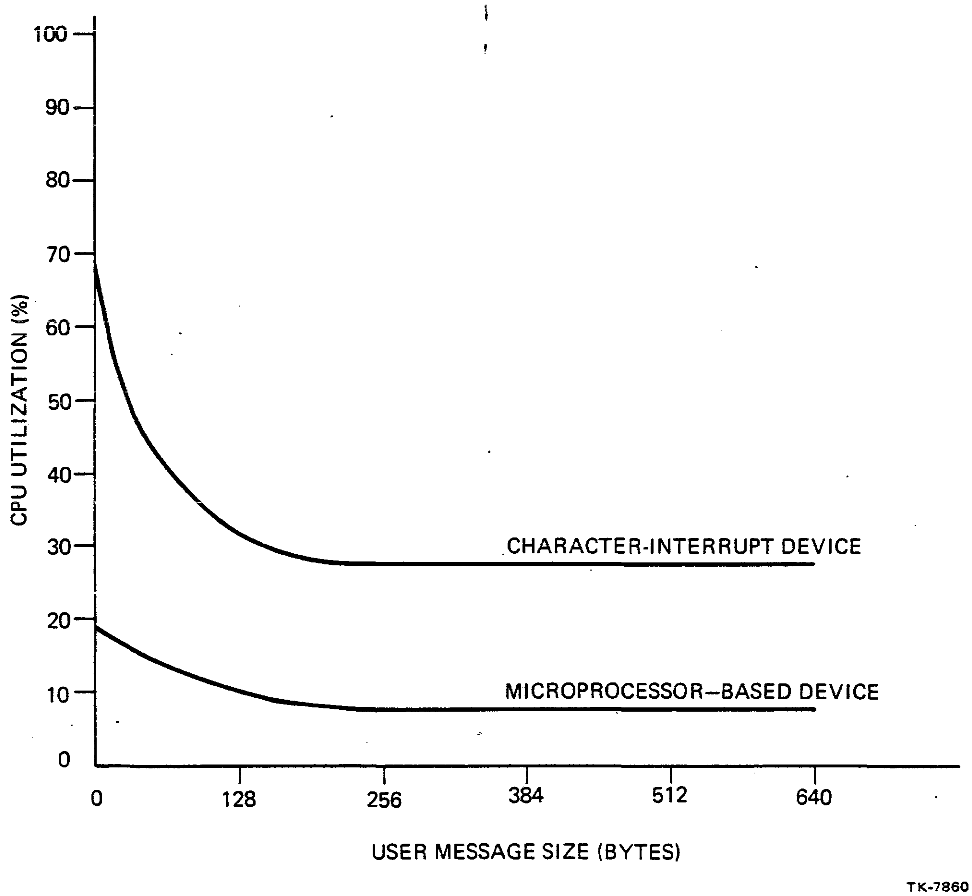


Figure 2 CPU Utilization versus User Message Size for Different Types of Communications Lines

Communications Line Characteristics

The bandwidth of the communications line is usually less than that of the CPU, therefore any congestion that occurs is due to the communications line. The throughput, for the same communications line, tends to increase with the line speed up to the point where the line becomes saturated. Figure 3 shows the throughput for the same line at various line speeds.

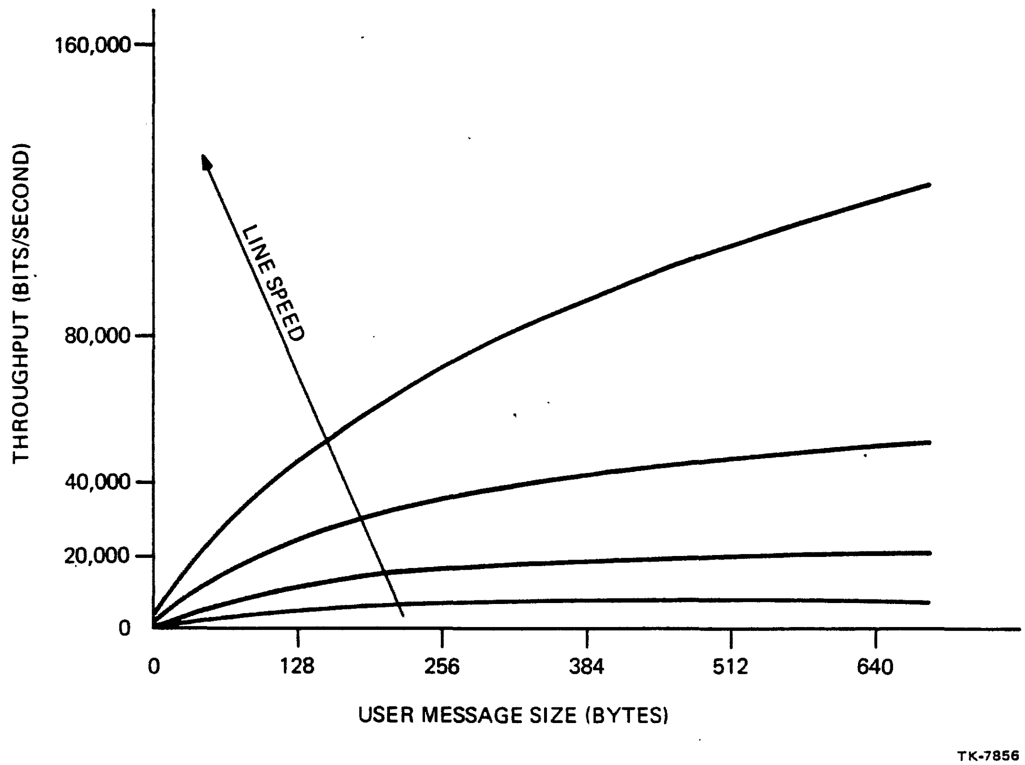


Figure 3 Throughput versus User Message Size for the Same Type of Communications Line

The net utilization of a line (actual useful data minus protocol overhead) is defined as:

$$\frac{\text{Actual User Data Rate (bits/second)}}{\text{Line Speed (bits/second)}}$$

Figure 4 presents an example of net line utilization versus the user message size for a communications device driven at different speeds.

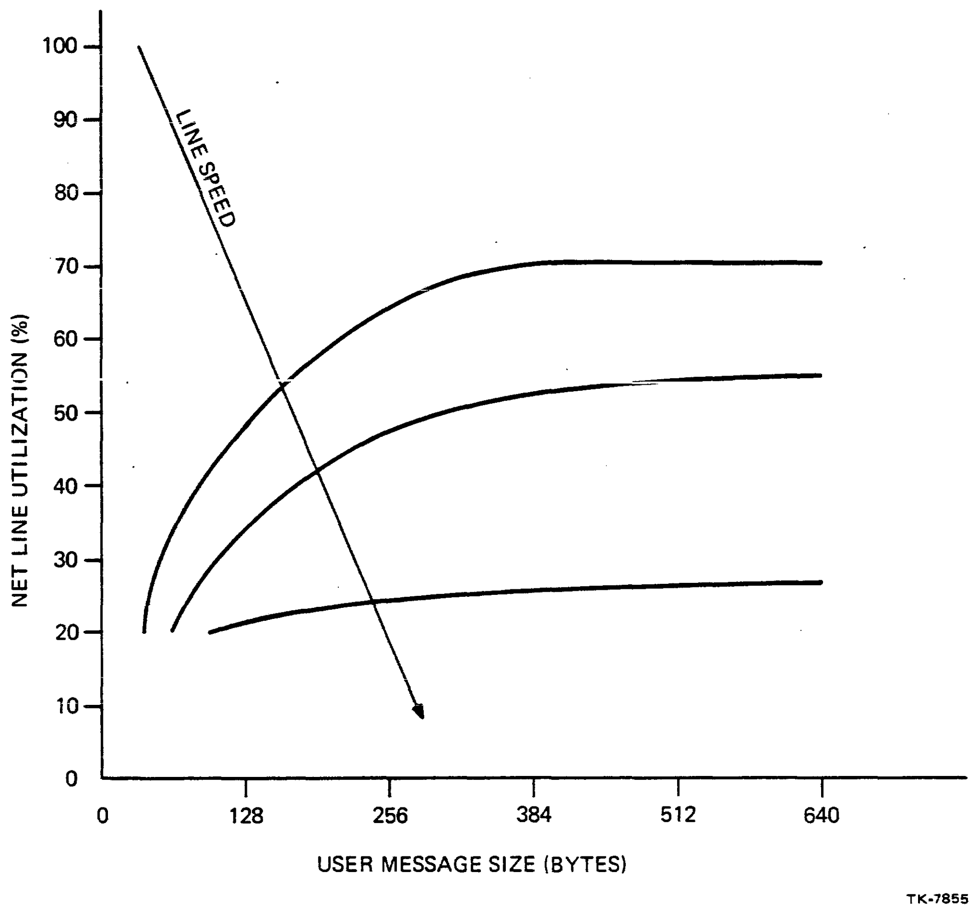


Figure 4 Net Line Utilization versus User Message Size

The throughput on a line can be affected by such things as line noise, since noisier lines require more retransmissions of the same data plus the same protocol overhead.

Throughput can also be affected by half-duplex lines which are less than half as efficient as full-duplex due to the nonzero turnaround time. Relative efficiency is further reduced on a noisy half-duplex line because the transmitter never detects errors until the line is turned around.

System and User Buffering Levels

There are two types of buffers that need to be considered: system and user. System buffers are used by the operating system and communications software to interface with the communications device(s). User buffers are allocated by the application program to hold data for interfacing with the operating system and communications software.

System Buffers

- The number of buffers is generally selected by a system (network) manager depending upon the traffic, line speed, and network topology.

NOTE

For VAX, the system (network) manager sets the maximum number of system buffers. The system returns unused buffers back to nonpaged pool until they are needed by the network.

- In general, the desired buffer level increases with heavier traffic, lower line speeds, and longer path length.
- If there are not enough buffers, resource errors result; if there are too many, space is wasted.

User Buffers

- The number is specified by the user application program.
- The desired buffer level increases with heavier traffic.
- If memory space is tight, the buffer level can be decreased at the expense of increased delay.

To achieve good throughput over the network, follow these rules:

- It is desirable to have the same system buffer size across the network or at least on the routing nodes (see Figure 5 for details).
- An average user buffer size must correspond to the system buffer size considering the protocol overhead in the system buffer.

If an application on node A wants to communicate with an application on node E, the End Communication Layer (ECL) buffer size is 600 bytes for that logical link. Every packet with a size more than 500 bytes over that logical link is discarded by the network as an oversized packet.

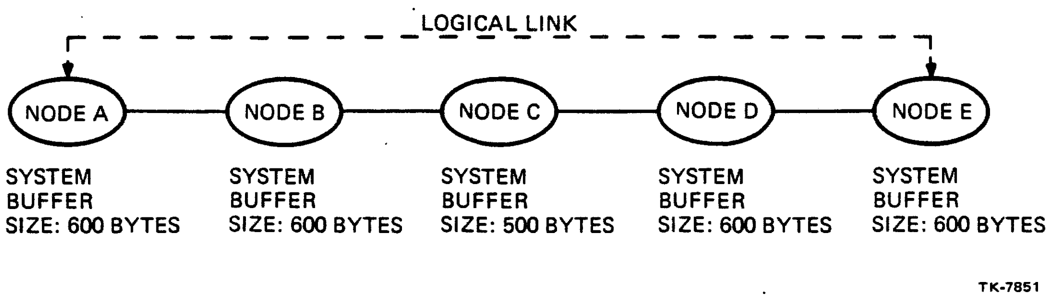


Figure 5 System Buffer Size in the Network

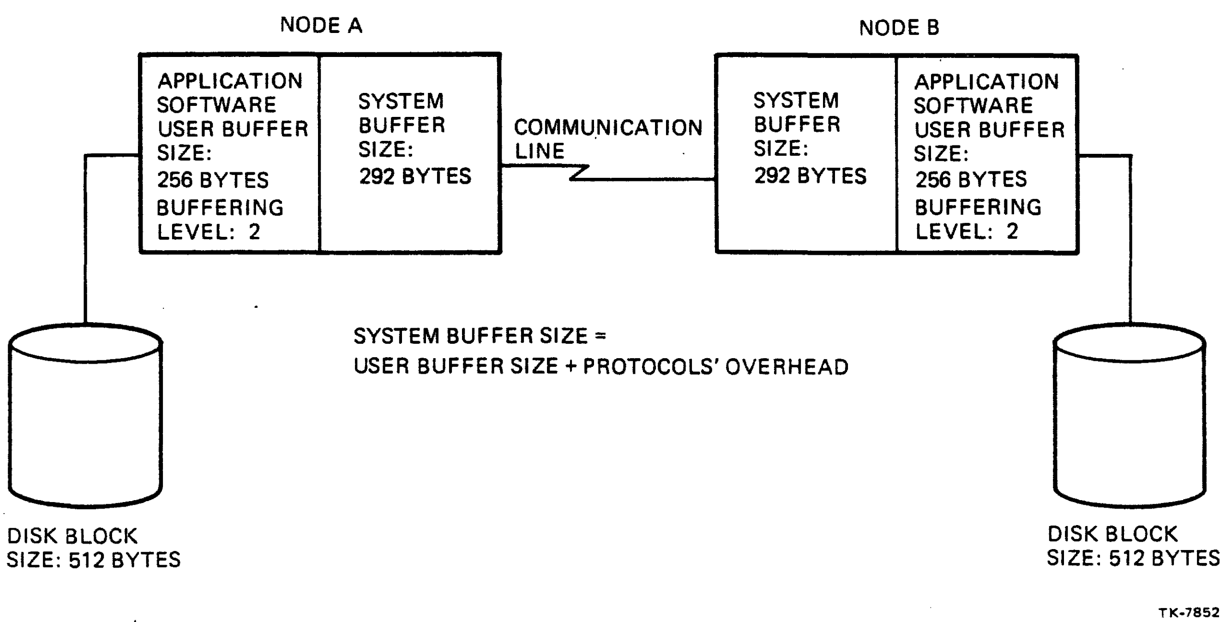


Figure 6 Correspondence of User to System Buffer Size

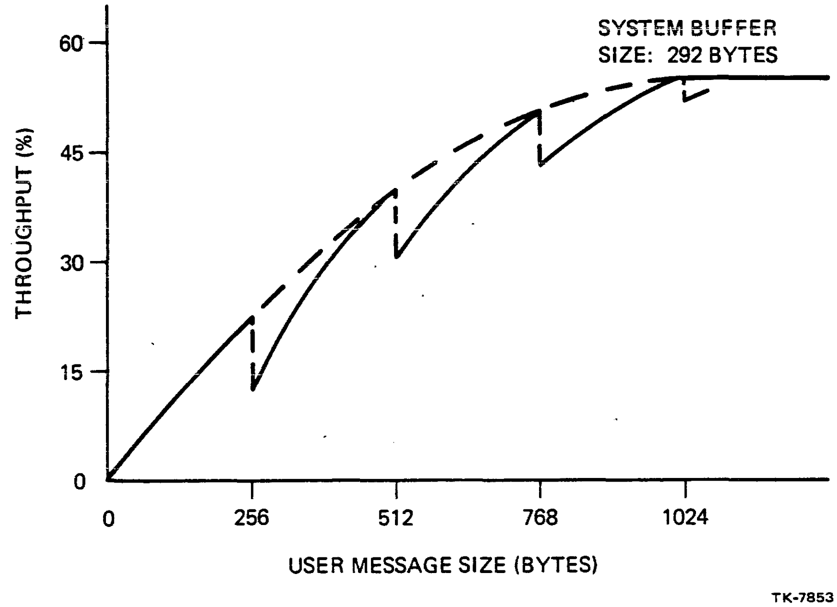


Figure 7 Throughput versus User Message Size for a Specified System Buffer Size

Routing

Routing loads communications lines and nodes with transport messages. Routing also requires a certain amount of CPU activity for routing a packet within a node from one physical line to another. Depending on the implementation of the communications software, routing may require more or less CPU activity on a node in end-to-end traffic. Figures 8 and 9 illustrate the required CPU activity to support DECnet routing on RSX and VMS operating systems, respectively.

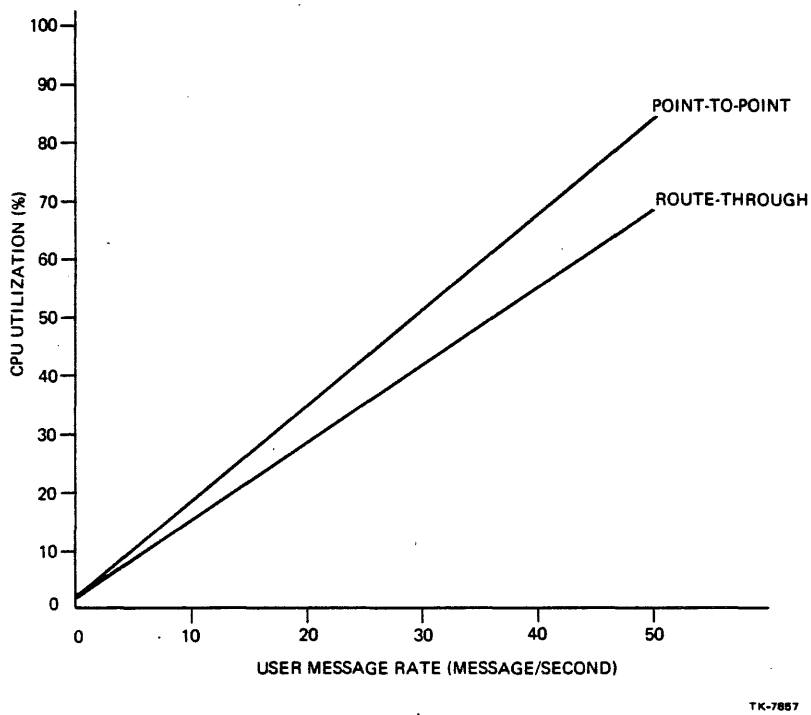


Figure 8 CPU Utilization versus User Message Rate for DECnet-RSX

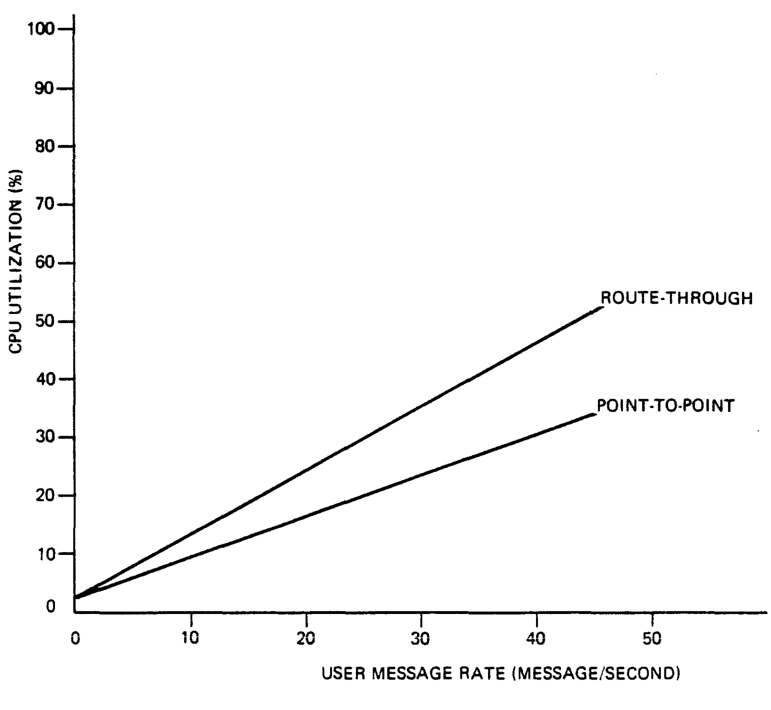


Figure 9 CPU Utilization versus User Message Rate for DECnet/VAX

Multipoint

Multipoint performance is complex and is dependent on a number of interrelated parameters and effects:

- For the software multipoint configuration, the CPU utilization of a multipoint tributary node is a function not only of messages per second that it exchanges with the control node, but also of any simultaneous conversations between the control node and other tributaries.
- For multipoint configurations with low-speed lines and significant traffic rates to and from several slave nodes, lines routinely approach saturation.
- Multipoint traffic does not make as efficient use of the line as point-to-point traffic, due to the increased polling overhead.

Ethernet

A major performance enhancement to DECnet is the support of Ethernet. Ethernet:

- Provides a data link that can transfer data faster and with less host system load than a 1 megabit DMR-11.
- Requires a smaller number of data messages than a DMC-11 because when two phase IV nodes communicate on the Ethernet the executor buffer size automatically increments to 1498 bytes (with a DMC the executor has a buffer size of 576 bytes).
- Less NSP layer messages sent back and forth between the two communicating nodes (due to the smaller number of data packets).

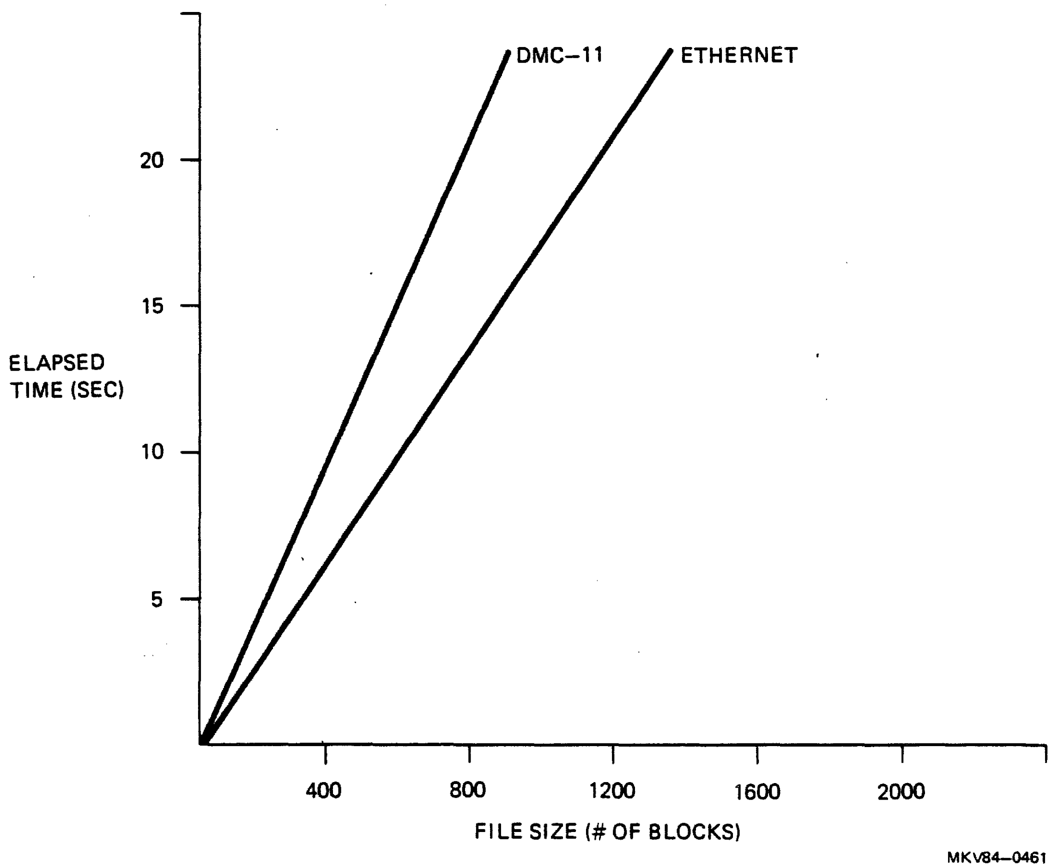


Figure 10 Comparison of File Transfer on Ethernet and DMC-11

MESSAGE SIZE AND VARIATIONS

The smaller the average message size, the lower the potential throughput. Since the protocol overhead does not diminish along with the message size, the percentage of processor or line bandwidth devoted to protocol increases.

The greater the variation in the message size, the more degraded the performance. When message lengths vary over a long range, the performance tends toward the shorter messages.

Message Rate

Processor utilization is directly proportional to message rate. The question is whether the CPU or the line becomes congested first. Figure 11 shows a typical plot for CPU utilization versus message rate for various line speeds.

NOTE

As the line approaches saturation, queue length increases, and the piggybacking feature minimizes overhead and thereby diminishes the processor overhead required to transmit or receive a message.

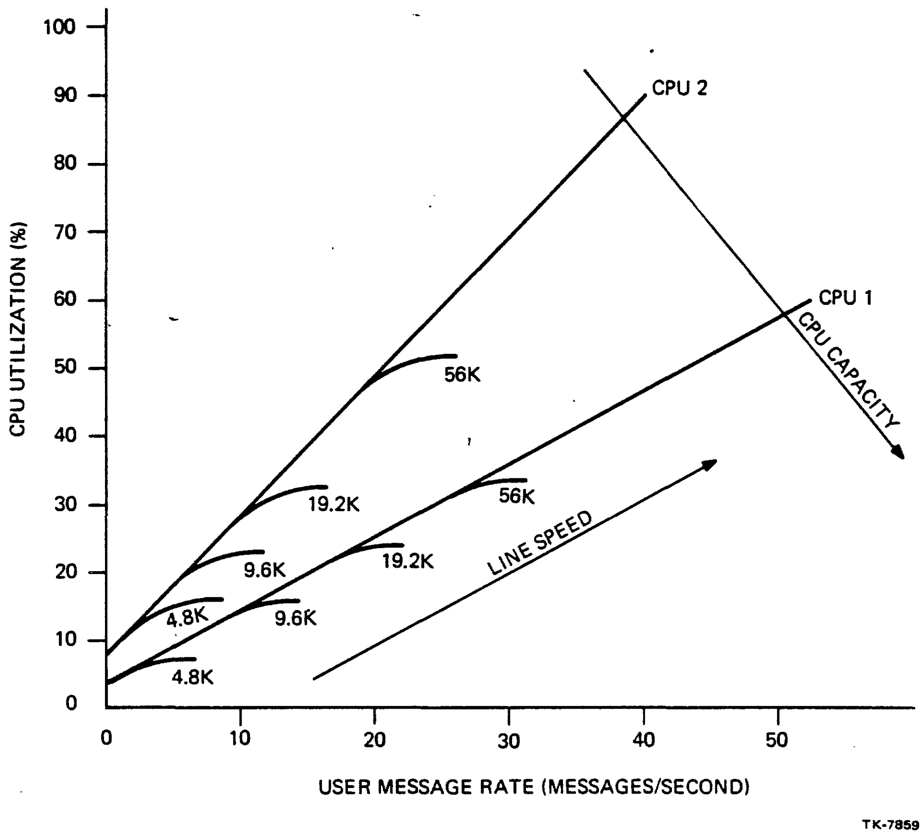


Figure 11 CPU Utilization versus User Message Rate for Various Line Speeds

Peak Load Design

A network should be designed for peak periods of the applications during the day, not for the total number of messages averaged over the number of hours the system is used.

Applications Mix

The mixture of applications running concurrently with the communications software influences network performance, particularly as the load increases and approaches resource saturation. Any code executed at elevated priority levels should be examined for its potential impact on the communications software. Any user code with heavy real-time requirements should also be checked.

MAJOR PERFORMANCE PARAMETERS

Although a variety of factors affecting network performance combined with a variety of network types creates a complicated picture for a specific network and application, in general, the performance behavior can be described with just two parameters: net line utilization and CPU utilization.

Net Line Utilization

- Varies primarily with message size, flow control, full- or half-duplex, and noise.
- Asynchronous communications devices are significantly less efficient than synchronous devices, because they require start and stop bits.
- At low speeds, lines can become saturated. Piggybacking becomes important at this point.
- It is measured in percent versus user message size.

CPU Utilization

- Both local and network processing should be considered.
- A combined CPU utilization of greater than 70% is a clue to potential performance problems.
- Varies primarily with the CPU, communications device, flow control, and the type of operating system.
- Character-interrupt devices use a lot more CPU than microprocessor-based devices.
- It is measured in percent versus user message rate (messages/second).

CONTENTS

TABLES

INTRODUCTION

New features of the DECnet-RSX software must be presented to customers in accordance with DIGITAL's marketing and support strategies. This module, utilizing articles from internal DIGITAL publications, provides summary information on the new features the product has, the capabilities of the product, and the pricing of the product along with the optional software/hardware that is available. Appendix A contains a copy of the SPD for each of the DECnet-RSX products. Information on performance of the product versus the previous version, and against itself is provided in Appendix B.

OBJECTIVES

Upon completion of this module the specialist should:

- Recognize the product's limitations.
- List the product's new features.
- Use the available performance graphs for evaluating the DECnet load on a node.

RESOURCES

1. DECnet-RSX Phase IV Update Seminar Videotape Presentation
2. Sales Update, October 3, 1983
3. DECnet-11S, SPD 10.74.11
4. DECnet-11M, SPD 10.75.11
5. DECnet-11M-PLUS, SPD 10.66.04
6. DECnet-RSX Version 4.0 Performance Report

BASIC PRODUCT DESCRIPTION

The DECnet-RSX Phase IV products are:

DECnet-11M, Version 4.0

DECnet-11S, Version 4.0

DECnet-11M-PLUS, Version 2.0

These products run on RSX11M, Version 4.1; RSX11S, Version 4.1; or RSX11M-PLUS, Version 2.0 respectively.

NEW FEATURES

- Large network support that increases the number of supported nodes from 255 to 1023.
- DEUNA and DEQNA support that allows the UNIBUS and Q-BUS based PDP-11 systems to connect to Ethernet.
- Extended data link services that includes the Ethernet protocol in addition to the existing DDCMP and X.25 protocols.
- DLX QIO calls using the Ethernet that allows the user to bypass some of the overhead associated with DECnet.
- Enhanced network management that can support Ethernet and communication server products including down-line loading and up-line dumping over the Ethernet.
- Phase III compatibility that allows upward migration to Phase IV. However, note that Phase II nodes are no longer supported.

PACKAGING CHANGES

- Full function kit that can generate either a routing node or an end node.
- End node kit that is moderately priced and has less overhead than the standard routing node.

STREAMLINED NETWORK GENERATION PROCEDURE

- PREGEN is no longer required for some users (those with RLØ2, RKØ6, RKØ7).
- NETGEN asks fewer questions (more defaults).
- NETGEN supports a standard function network (full component with predefined values for optional parameters).
- NETGEN supports extended network services for layered software that may be added later (heterogeneous command terminal support, SNA gateway access, X.25 gateway access).

NEW COMPONENTS

- The Ethernet Protocol Manager (EPM) that controls access to the Ethernet. DEUNA and DEQNA are devices that support access to the Ethernet and therefore require EPM.
- The End Communication Process and Driver (ECL) that manages all the time-critical network functions and accomplishes all QIO calls for user tasks. In the previous release this component was called NSP.
- The Routing Process (XPT) that controls sending and receiving of messages from and to the local node and forwarding route-through packets.
- The Routing Control Processor (RCP) that controls and updates the routing database for all nodes that support routing.
- Direct Line Access Controller (DLX) that buffers messages in network buffers and supports two new Ethernet devices, DEUNA and DEQNA.
- The Console Carrier Requester (CCR) task that communicates with Console Carrier Server (CCS) on the remote server node. CCR is used only for communicating with communication server systems.
- The General Microcode Loader (MLD) that loads devices (other than KMC or KMS) with the proper microcode. Currently, UNA is the only device that requires MLD.

COMMUNICATIONS SERVERS

Phase IV provides the following support for communications server systems from the host node:

- Support for down-line loading server systems -- The communication server system has no disk peripherals and the pregenerated system is down-line loaded from a host system over the Ethernet.

- Support for receiving up-line dumps from server systems -- On system crash the server system dumps its memory image to:

  - The host that downline loaded it.

  - A list of predefined hosts.

  - Any possible hosts.

- Support for the console carrier requester -- As the server system has no physical console terminal, commands normally issued from the console now come from a host node. The MOP protocol is enhanced to support this feature.

The Phase IV server products include:

- DECnet router server

- Terminal server

- DECnet Router/X.25 gateway

PROGRAMMING AND UTILITIES ENHANCEMENTS

- Network File Access Routines (NFARs)
  - Record access -- Support for accessing records randomly.
  - Extended attributes -- Support for resultant file names, full file attributes, dates and times, and protection.
  - Access options -- Support for various access control options.
- Network Display Utility (NTD)
  - Resource Display Page -- New format display shows circuits instead of remote nodes.
  - Remote Node display -- New display allows users to see up to 84 reachable nodes at one time.
- RMS FAL
  - Support for RMS-11, Version 2.Ø.
  - Support for multicopy and multiuser FAL on RSX-11M-PLUS systems.
- Network File Transfer (NFT)
  - Attribute directory switch (/AT) -- Displays a new directory listing format showing the file attributes.
  - Filespec logging switch (/LO) -- Displays file specifications while NFT does the operation.
  - Automatic block mode transfer (/AX,/BK,/RC) -- No switch needed. NFT automatically defaults to block mode transfer if the remote node system can support it.
  - Version number conversion -- Built-in feature converts file version numbers from octal to decimal or decimal to octal.
  - Improved wildcard handling -- Wildcards can be specified in the output file specification.

DECnet PHASE IV FOR PDP-11s

The Phase IV DECnet-RSX products are ready to book now. The DECnet-RSX products offer the major networking enhancements of the Ethernet and DNA Phase IV programs announced in May 1982, plus new pricing and new packaging to meet the needs of your customers.

PRODUCT DESCRIPTION

The DECnet-RSX Phase IV products are:

DECnet-11M, Version 4.Ø
DECnet-11S, Version 4.Ø
DECnet-11M-PLUS, Version 2.Ø

These software products offer large network and Ethernet support for the RSX-11M, RSX-11S and RSX-11M-PLUS operating systems.

The DECnet-RSX products support the existing DECnet Phase III features such as homogeneous network command terminals, file transfer, network management, file access, record access, multipoint support, X.25 support, down-line loading and up-line dumping, and task-to-task communications.

In addition, the DECnet-RSX Phase IV products offer the following new features as previously announced:

- DEUNA and DEQNA support -- The DEUNA and DEQNA (Ethernet-to-UNIBUS and Q-Bus communications controllers) allow PDP-11 systems to connect to Ethernet.

- Extended data link services -- Data link service now includes the Ethernet protocol in addition to existing DDCMP and X.25 protocols.

- Routing for large networks -- The routing algorithms are enhanced to increase the number of supported nodes in the network from 255 to 1Ø23.

- Enhanced network management -- Network management now offers support for Ethernet and communications server products. This support includes loading remote systems, dumping memory from remote systems, initiation and termination of network functions, and examination and modification of remote unattended systems.

- Phase III compatibility -- DECnet-RSX Phase IV offers Phase III features plus compatibility. This compatibility provides upward migration to Phase IV.

Network command terminals are supported for DECnet-RSX host to DECnet-RSX host only (homogeneous support). With this feature, a user at a terminal on a DECnet-RSX host appears directly connected to another DECnet-RSX host. In the next release of DECnet-RSX, this feature will be extended to other hosts supporting heterogeneous network virtual terminals.

SELLING STRATEGY

Customers can invest in the future now. The DECnet-RSX products offer flexibility for evolution to large networks and migration to Ethernet with the same software. Configuration flexibility allows the user to reconfigure as the network expands.

The DECnet-RSX Phase III compatibility feature provides insurance for existing customers by offering:

- Migration for applications developed on Phase III nodes
- Connectivity of Phase III nodes to Ethernet by using Phase IV nodes and to the Phase IV network
- Phased migration to Phase IV networks and Ethernet

PACKAGING

The new packaging for DECnet-RSX products offers more options to meet the needs of your customer. The new packaging includes the new End-Node Kit and the separation of installation service.

Kits

Two types of nodes may be generated using the DECnet-RSX products. The two types are routing nodes and end nodes. The routing node can initiate a message, be a destination for a message or be a transit point for messages passing through to the destination node (local or remote). The end node can initiate a message or be a destination for a message. The end node cannot be a transit point for messages passing through to the destination node.

Two types of DECnet-RSX kits are now available: the Full Function (Routing Node) Kit and the new End-Node Kit.

The Full Function Kit generates either a routing node or an end node. The Routing Node provides cost-efficient flexibility for the customer who purchases one total kit and multiple license-only kits.

The End-Node Kit is competitively priced and has less overhead than the standard routing node. The End-Node Kit meets the needs of the customer who requires the cost-effective solution for low-end application processing. For example, in an Ethernet LAN the lower priced end node is available for application processing. This offers the customer a cost-effective LAN solution. If routing is required, the routing node handles any routing overhead.

SERVICE OFFERINGS

DIGITAL-supported prices are reduced. Software installation is unpackaged from software warranty. The DECnet-RSX supported license customer receives a software license, distribution media, and 90 days of warranty support. Software warranty begins after software product installation and includes: telephone support, software product updates, software performance report responses, and critical on-site remedial support for identified program error after all remote services are exhausted.

Installation service is recommended for new customers and existing customers who are unfamiliar with DECnet installations. Installation services are delivered by your local office specialist and include: preinstallation preparation assistance, site and software verification, installation verification and check-out, review of the product operation, and a documentation set.

In addition to installation services, software product services are currently available (Self-Maintenance, Basic and DECsupport). Customers currently in warranty or on either of Software Product Services receives DECnet-11M, Version 4.0; DECnet-11S, Version 4.0; or DECnet-11M-PLUS, Version 2.0 at no additional charge.

Customers who are not in warranty and do have a service contract may purchase the appropriate H-Kit.

NOTE

In-warranty and Software Product Service customers receive the full-function software distribution.

PREREQUISITE SOFTWARE

RSX-11M, Version 4.1; RSX-11S, Version 4.1; or RSX-11M-PLUS, Version 2.1 is a prerequisite for DECnet-11M, Version 4.Ø; DECnet-11S, Version 4.Ø; or DECnet-11M-PLUS, Version 2.Ø, respectively.

PRICING AND LITERATURE INFORMATION

To provide a single reference source, pricing and literature information for most of the products announced in this issue follows.

If there are any questions concerning this information, contact the individual Product Manager or Distributed Systems Marketing.

Table 1 Product Information

Product Order Number	Description	MLP[1]	BMC	Discount E/U	Discount OEM	Product[2] Availability
Ethernet Hardware						
DEUNA-AA	Ethernet to UNIBUS Comm Controller	$3,500	Call F.S.	Y	1	NOW
DEQNA-KP	Integrated Q-Bus to Ethernet Controller	1,150	Call F.S.	Y	1	Q3/FY84
DEQNA-M	Base Module M7504 Q-Bus to Ethernet	1,000				Q3/FY84
CK-DEQNA-KA	Cab Kit for DEQMA-M in 11/23-S	150				Q3/FY84
CK-DEQNA-KB	Cab Kit for DEQNA-M in Micro/PDP-11	150				
CK-DEQNA-KC	Cab Kit for DEQNA-M in M/23-PLUS	150				
H4000	Ethernet Transceiver Qty 1-99	300	DECmailer only			NOW
	Qty 100-499	250				
	Qty 500 +	200				
DELNI-AA	Ethernet Local Network Interconnect	985	Call F.S.	Y	1	NOW
DELNI-AB	Ethernet LNI - non-U.S.	985				OCTOBER
DELNK-A*	DELNI Country Kit	10				
DEREP-AA,AB	Ethernet Local Repeater	1,500	Call F.S.			MARCH 1984
DEREL-A*	Country Kit for Ethernet Local Repeater	12				
DEREP-RA,RB	Ethernet Remote Repeater	4,400	Call F.S.			MARCH 1984
DEREK-R*	Country Kit for Ethernet Remote Repeater	12				
	*Several Country Kit Variations are available.					
Ethernet Communications Servers						
	16-Line Terminal Server	14,000(T)	135(T)	Y	2	Next major release of VMS
	32-Line Terminal Server	20,000(T)	225(T)	Y	2	
	Terminal Server S/W	1,000(T)		Y	2	
(The Terminal Server information is included for planning purposes only. Specific price, availability, and ordering information will be supplied upon announcement.)						
	Comm Server Upgrade Kit	4,750(T)	90(T)	Y	2	
DECSA-EA	One-Line Router or X.25 Gateway	13,500	183	Y	2	Q3/FY84
Q*725-Ax	DECnet Router Server S/W	3,500		Y	2	Q3/FY84
Q!727-Ax	DECnet Router/X.25 Gateway S/W	3,500		Y	2	Q3/FY84
Q!728-Ax	X.25/X.29 Extension Package	2,200		Y	2	Q3/FY84 (VAX)
Q!728-DZ	X.25/X.29 Ext Package License Only	1,500		Y	1	

* = C-11/730, D-11/750, E-11/780, P-RSX
! = C-11/730, D-11/750, E-11/780
x = media
T = Targer

1. These prices are for the U.S. Area only. Pricing in other areas will be determined by the area managers.
2. Dates listed here are worldwide product availability dates.

Table 1 Product Information (Cont)

Product Order Number	Description	MLP[1]	BMC	Discount E/U	Discount OEM	Product[2] Availability
Ethernet Communications Servers (continued)						
QLA01-DZ	Generic Server S/W License Only	$ 100		Y	1	
DECSA-AH	SNA Gateway, S/W on RL02 (VMS)	26,995	135	Y	2	Q2/FY84
DECSA-AM	SNA Gateway, S/W on 1600bpi MTa (VMS)	26,995	135	Y	2	Q2/FY84
DECSA-DZ	SNA Gateway w/License-to-Copy S/W	25,795	135	Y	1	Q2/FY84
DCSAX-LA	One Line Sync Card 19.2K bps	415	Call F.S.	Y	2	
DCSAX-LB	One Line Sync Card to 500K bps	650		Y	2	
DCSAX-LC	Two Line Async Card to 19.2K bps	375(T)		Y	2	
DECnet-RSX Full-Function Node						
QJ76*-Ax	Supported license, media and 90-day warranty (least cost media)	4,150	N/A	Y	2	Q3/FY84
QJ76*-DZ	License-to-Copy "A" Kit	2,950		Y	1	
* = 2-11S, 4-11M, 6-11M-PLUS						
End Node						
QJ76*-Ax	Supported license, media and 90-day warranty (least cost media)	1,950	N/A	Y	2	Q3/FY84
QJ76*-DZ	License-to-Copy "A" Kit	800		Y	1	
* = 3-11S, 5-11M, 7-11M-PLUS						
DECnet/VAX Full-Function Node						
Q*D05-Ax	Supported license, media and 90-warranty (least cost media)	3,950	N/A	Y	2	NOW
Q*D05-DZ	License-to-Copy "A" Kit	2,950		Y	1	
* = C-11/730, D-11/780, E-11/780, 11/782						
End Node						
Q*D04-Ax	Supported license, media and 90-day warranty (least cost media)	1,450	N/A	Y	2	NOW
Q*D04-DZ	License-to-Copy "A" Kit	950		Y	1	
* = C-11/730, D-11/750, E-11/780, 11/782						
OBSERVER V1.1						
QSP40-AD	OBSERVER-PLUS Supported	15,000	N/A	N	N	Q3FY84
QSP20-AD	OBSERVER Supported	12,000		N	N	Q3/FY84
QSP20-DZ	OBSERVER License Only	9,000		N	N	Q3/FY84

1. These prices are for the U.S. Area only. Pricing in other areas will be determined by the area managers.
2. Dates listed here are worldwide product availability dates.

Table 1 Product Information (Cont)

Product Order Number	Description	MLP	DPMC	BSMC	SMMC	A/O Change
Q*725-AM	DECnet Router Server S/W		$ 120	$ 70	$ 40	-
Q*725-AH	DECnet Router Server		130	80	50	
Q*725-HZ	DECnet Router RTC Upd	$ 180			15	
	Terminal Server S/W		120(T)	70(T)	40(T)	-
	Terminal Server S/W		130(T)	80(T)	50(T)	
	Terminal Server RTC Upd	180(T)			15(T)	
Q!727-AM	DECnet Router/X.25 Gateway S/W		120	70	40	-
Q!727-AH			130	80	50	
Q!727-HH	DECnet RTR/X.25 Gateway Upd RL2	1,080				
Q!727-HM	DECnet RT/X.25 Gateway Upd Magtape	960				
Q!727-HZ	DECnet Router X.25 RTC Upd	180			15	
Q!728-AG	X.25/X.29 Extension Package (VMS)		130	80	50	
Q!728-HG	X.25/X.29 Extension Package Upd TU58	960				
Q!728-AY			120	70	40	
Q!728-HY	X.25/X.29 Extension Package Upd RX01	960				
Q!728-HZ	X.25/X.29 Extension Package RTC Upd	180			15	
DECSA-AH	SNA Gateway, S/W on RL02		150	100	80	
QSF01-SZ	SNA Gateway Installation Service	4,000				
QJ732-HX	DECnet/SNA Gateway Upd N/S	1,920	150	100	80	$1,980
QJ732-HZ	DEcnet/SNA Gateway RTC Upd	360			30	1,980
QE452-ZY	DECnet/SNA Gateway Magtape		130	80	60	660
Q!452-AG	DECnet/SNA Gateway Magtape		135	85	65	660
Q!452-HG	DECnet/SNA Gateway Magtape N/S Upd	1,560				660
Q!452-HY	DECnet/SNA Gateway Magtape N/S Upd	1,440				660
Q!452-HZ	DECnet/SNA Gateway Magtape RTC Upd	240			20	660
QE453-AY	DECnet/SNA RJE		130	80	60	660
Q!453-AX	DECnet/SNA RJE		135	85	65	660
Q!453-HG	DECnet/SNA Upd N/S	1,560				660
Q!453-HZ	DECnet/SNA RJE RTC Upd	240			20	660
Q!453-HY	DECnet/SNA RJE Upd N/S	1,440				660
QE454-AY	DECnet/SNA VMS 3270TE	5,000	130	80	65	660
Q*454-AG	DECnet/SNA VMS 3270TE		135	85	65	660
QE454-HY	DECnet/SNA VMS 3270TE Upd	1,440				660
Q*454-HG	DECnet/SNA VMS 3270 Upd	1,560				660
Q*454-HZ	DECnet/SNA VMS 3270 RTC Upd	240			20	660
QE455-AY	DECnet/SNA VMS Appl Int.	8,500	130	80	60	660
Q*455-AG	DEcnet/SNA VMS Appl Int.	8,500	135	85	65	660
QE455-HY	DECnet/SNA VMS API Upd	1,440				660
Q*455-HG	DECnet/SNA VMS API Upd	1,560				660
Q*455-HZ	DECnet/SNA VMS API RTC Upd	240			20	660
QJ76*-AX			240	140	70	-
QJ76*-AX			175	100	50	-
QJ76*-AX	-		175	100	50	
QSF02-SZ	DECnet-RSX-11M, M+, 11S Instal. Serv.	2,800				
Q*D04-AX			240	140	70	
Q*D05-AX			240	140	70	

NOTE

Prices are for low cost media (floppy). the above prices are for U.S. Area only. Prices in other areas will be provided by the local Software Services Manager.

A 5% discount is available to customers who prepay for a year's service. Prerequisites for SPS are a license for the appropriate Communication Server and/or software product license for the appropriate host operating system, an SPS contract for the host operating system and SPS contract for the appropriate Communication Server and/or software product.

G = TU58, Y = RX01, X - RX02, H = RL02, M = 9-track 1600 magtape

! = C-11/730, D-11/750, E-11/780
* = C-11/730, D-11/750, E-11/780, P-RSX

APPENDIX A
Performance Report

INTRODUCTION

Configurations

RSX-DECnet 4.Ø performance measurements were obtained using various configurations. The following chart illustrates the devices and processors used to gather data for this report:

Table 2 Configurations

Processor	UNA	Device 1 Meg DMC*	56 K DMC
11/24 --> 11/24	X	X	X
11/44 --> 11/44	X	X	X
11/24 --> 11/75Ø	X	X	X
11/44 --> 11/75Ø	X	X	X

*The DMC and DMR communications devices perform the same. The terms will be used interchangeably in this report.

All configurations consisted of an end node communicating with a routing node, except the 11/44 --> 11/44 combination. Both end node --> end node and routing node --> routing node tests were run with this configuration.

Two RLØ2s were used for all tests except those to VMS. One RLØ2 contained the RSX Version 4.Ø operating system, and the other contained DECnet and the performance tools. The VAX did not have RLØ2 drives. An RKØ7 was used for both the operating system and DECnet.

Methodology

Two types of data were collected for this report. Task-to-task communication was measured by using 3 minute DTS/DTR echo tests. File transfer times were obtained by using NFT commands, surrounded by "show time" commands. A background drone task, which incremented registers at a known rate, was installed at a lower priority than both NFT and DTS/DTR. By running the drone and the tests simultaneously, one could calculate CPU utilization, CPU time/message and CPU time/block.

DTS/DTR gives the user statistics on the tests performed. The baud rate mentioned in this report refer to these numbers. DTS/DTR does not include overhead in its statistics. Therefore, the term "baud rate" refers to the number of user bits/seconds transmitted and received. The term "line utilization" describes the percentage of the line filled with user bits. It is defined as the quotient of the number of user bits/seconds traveling through the line divided by the line speed (times 100%). The "true" baud rate and line utilizations were actually higher than those quoted here. Please read about DTS/DTR graphs in Section 1.4.

Measurements were taken using NFT transfer commands. ASCII files of known lengths were transferred from a local node to a remote node ("pushed") and then from the remote node to the local node ("pulled"). Had the files transferred been image files, and not ASCII files, less CPU time would have been needed to copy them because image files do not need to be formatted, whereas ASCII files do. Each series of NFT tests was done twice: once with RMS FAL installed on the remote node, and once with FCS FAL installed on the remote node.

CPU Power and Baud Rates

Throughout this report, performance will be described in terms of CPU utilization, CPU cost/message and baud rate. CPU cost/message is defined as the amount of CPU time (in milliseconds) divided by the sum of the number of messages transmitted and received. The other two concepts are defined above.

The amount of CPU time spent by a system is directly proportional to its CPU utilization. In other words, holding the CPU utilization of a system constant is equivalent to holding the CPU time spent by a system constant. If the CPU is the bottleneck in a system, a task with a low CPU cost/message will perform better (or in the case of DTS/DTR tests, have a higher baud rate) than a task with a high CPU cost/message. This is true because CPU time is the limiting factor with respect to performance. The more "CPU efficient task" will have better performance than the less "CPU efficient task".

The same relationships hold true when comparing configurations. If two configurations have the same CPU types and they are both saturated, one can explain why one system has a higher baud rate than the other by knowing their CPU costs/message. The baud rate of the configuration with the cheaper CPU cost/message will be higher than the baud rate of the system with the more expensive CPU cost/message. In this study, the communications line often accounts for a difference in CPU cost/message between configurations.

The configuration with the lower CPU cost/message does not always have the higher throughput. Processor types have different instruction speeds. An 11/24 with a 50% CPU utilization cannot do as much as an 11/750 with a 50% CPU utilization. Even if the RSX system had a cheaper CPU cost/message than the VAX, it would probably be outperformed by the faster processor. If two configurations have identical processors, and one uses more of its CPU power on a task than the other, it may compensate for its higher CPU cost/message. By using more CPU time for the task than the other system, the less "CPU efficient" configuration may perform better.

System Buffer Sizes

One should take into account the relationship between the user message size and the large system buffer size when tuning a system. There are (at worst) 18 bytes of overhead in a DECnet message. Therefore, the sum of the user user message size +18 should be as close to the system buffer size as possible. This will avoid the fragmentation of messages. If "n" is a positive integer, and "buffsize" is the decimal number of bytes in the large system buffer, then the following formula describes the critical user message size:

user message size = [n X {buffsize-18)]

If the user message size is even one byte larger than this critical value, an extra DECnet message will be needed for each user message transmitted. There will be a sharp degradation in performance at these points, as measured by user throughput and CPU utilization. This phenomenon creates the staircase effect seen in Graphs 1 and 2.

All the DTS/DTR graphs in this report should follow these generic shapes. As one approaches a critical user message size, performance improves. Once one passes it, there is a sharp decline followed by a steady increase in performance until the next critical user message size is exceeded. For simplicity's sake, the graphs have been drawn as straight lines.

All system buffer sizes were set to 576 bytes. Preliminary tests were also run with buffer sizes of 518 and 1484 bytes. The results did not warrant the inclusion of the data in this report. Except for the expected differences in performance at the user message size of 512 bytes, due to the staircase effect mentioned earlier, the results were similar with all three system buffer sizes.

DECnet 4.0

Point-to-Point Results

DTR/DTS Results -- As expected, the 11/44 --> 11/75Ø configuration provided the highest throughput of all the processor combinations, followed by the 11/44 --> 11/44, the 11/24 --> 11/75Ø and the 11/24 --> 11/24 (see Chart 1 and Graph 3).

It is interesting to note that the 1 Meg DMC often performs better than the UNA. In order to understand this, one must realize that the CPU cost/message of the UNA is usually higher than the CPU cost/message of a DMC (see Graph 4). When CPU time is the limiting factor in performance, the device with the cheapest CPU cost/message will provide the highest throughput. This explains why the 1 Meg DMC gives a higher baud rate than the UNA in the 11/44 --> 11/75Ø and 11/44 --> 11/44 configurations. It also explains why the UNA gives better performance than the 1 Meg DMC in the 11/24 --> 11/75Ø configuration (the UNA's CPU cost/message iss lower than the DMC's cost/message in this case.) The 11/24 --> 11/24 configuration seems to be the only exception to this rule. Upon closer inspection, one sees that the 11/24 --> 11/24 CPU utilization drops off after the 512 byte user message size when a 1 Meg DMC is used (see Graph 5). This drop in CPU utilization does not occur with a UNA (see Graph 6). Note that the 1 Meg DMC is "outperformed" (in terms of throughput) exactly at this point (see Graph 3). Therefore, even though the CPU cost/message of the UNA is higher than that of the 1 Meg DMC, the higher CPU utilization of the processor with the UNA more than compensates for this fact.

Graphs 7 and 8 plot baud rate as a function of user message size. each graph depicts data for a specific processor. 11/24s can transmit 1Ø5K bits/second through an Ethernet while 11/44s can transmit more than 16ØK bits/second through the line.

Graph 9 plots line utilization as a function of user message size. All configurations using an Ethernet had line utilizations well under two percent and therefore were not included in this graph. As one would expect, the 56K DMC lines were the most heavily utilized. The configurations obviously have line utilizations proportional to their baud rates. The limiting factor of all devices, except the 56K DMC, does not appear to be line speed. The limiting factor is CPU power.

**This excludes the 56K DMC cases, which are limited by line speed, not processor speed. The 11/24 --> 11/75Ø Ethernet configuration is an exception which will be explained later.

No significant differences in baud rate and CPU utilization were found between the end node --> routing node configuration and the routing node --> routing node configuration. The data from these two experiments have therefore been combined in this report.

NFT Results -- Chart 2 shows the time needed to transfer files between two nodes for various processors and devices. It is no surprise that the fastest configurations include the 11/750, and the slowest file transfers took place on the 11/24s. The processor speeds account for this, along with the differences in VMS FALs and RSX FALs.

RMS FAL versus FCS FAL -- All NFT tests were run to both an RMS FAL and an FCS FAL, except those which transferred files to and from the 11/750. VMS does not support FCS FAL. In every configuration, except one, file transfers to and from a system with the FCS FAL installed were faster than those to and from a system with RMS FAL installed (see Graph 10 and Chart 2). This is because the RSX FCS FAL is simpler and smaller than the RSX RMS FAL. The RMS FAL supports more functionality than the FCS FAL, and is therefore bigger and slower. The VMS RMS FAL is very fast, however, because it uses a lot of memory for buffering, and it queues its writes. Graphs 11 and 12 plot the CPU utilizations of various processors against file size.

Usually the bottlenecks in the file transfers are the CPUs. This is obviously not the case all the time. The 56K DMC lines were the limiting factor when they were used. If two systems are finely tuned, the disk speed becomes a factor in performance. In these tests, the RL02s tended to fill up, and time was wasted looking for free blocks to use.

Push versus Pull -- In all configurations involving an RSX system and a VMS system, "pushes" (local --> remote file transfers) were faster than "pulls" (remote --> local file transfers). This is due to the fact that RMS on VAX uses more buffering than RMS on RSX, when writing to its files. It is faster to write to a VMS system than an RSX system because the VMS "writes to disk" are queued (see Chart 2).

In most RSX --> RSX configurations "pushes" are slower than "pulls", at least for file sizes under 512 blocks. The only two exceptions to this are 1) the 11/24 --> 11/24 communicating over the Ethernet with RMS FAL installed and 2) configurations using a 56K DMC. The 56K line gets saturated. Neither CPU time nor buffering levels are the bottlenecks in this case. "Pushes" and "pulls" take virtually the same amount of time and CPU power.

Routine Node versus End Node -- When two routing nodes communicate with each other, NFT file transfers take longer than if an end node talks to a routing node. Only 11/44 --> 11/44 configurations over a 1 Meg DMC and the Ethernet were tested (see Graphs 13 and 14).

Route-Through Results

Route-through tests were done with the following configurations:

1) 11/44 $\xrightarrow{\text{1 Meg DMC}}$ 11/44 $\xrightarrow{\text{UNA}}$ 11/750

2) 11/44 $\xrightarrow{\text{1 Meg DMC}}$ 11/24 $\xrightarrow{\text{UNA}}$ 11/750

DTS/DTR Results -- It is clear that the bottleneck in the second configuration is the 11/24's CPU. The CPU utilization reaches 90%, in the case of a 1 byte user message size (see Graph 15).

It is interesting to note that the 11/24 routing node consistently uses between 44 and 47 percent more CPU time/message than the 11/44 routing node (see Graph 16). Similarly, the 11/24 uses between 23 and 32 percent more of its CPU than the 11/44. The 11/44 routing node can transmit 25 percent more user bits/second than the 11/24 routing node.

the baud rate of the route-through 11/44 --> 11/44 --> 11/750 configuration is approximately 1/3 slower than that of an 11/44 --> 11/750 configuration (over a 1 Meg DMC). The baud rate of the 11/44 --> 11/24 --> 11/750 configuration is slightly faster than that of the 11/24 --> 11/750 configuration (over a UNA), suggesting that the 11/24 can route messages quite a bit faster than it can originate the transmission of messages.

As expected, a large message size yields a higher user baud rate than a small message size, because overhead is a constant regardless of message size until the fragmentation of messages occurs (see Graph 17).

NFT Results -- As in the case with task-to-task communication, the 11/24 routing node uses substantially more CPU time/message than the 11/44 when transferring files (see Graph 16). The 11/24 routing node uses fifty percent more CPU time than the 11/44 routing node to perform its routing function, when "pushing" files from a local to a remote node. In the case of a transfer from a remote node to a local node, the 11/24 uses between 1/3 and 2/3 more CPU than the 11/44 (see Graphs 18 and 19).

Three Node Ethernet Configuration

Three nodes were connected to a private Ethernet. The 11/44 sent messages to the 11/24, the 11/24 communicated with the 11/750 and the 11/750 talked to the 11/44. In other words, the three nodes talked in a circle to each other. The bottleneck in the DTS/DTR experiment was not the line, but the 11/24 CPU. It always had a CPU utilization of over 99% (see Graphs 20 and 21).

DECnet 4.0 VERSUS DECnet 3.1

DTS/DTR Results

DECnet 3.1 yields a higher user baud rate than DECnet 4.0 in every configuration tested except the 11/44 --> 11/750 (see Graphs 22 and 23). The difference in user baud rates becomes more pronounced as the user message size increases.

When looking at DTS/DTR performance, three main factors must be considered: CPU speed, CPU utilization and line speed. In a stand-alone environment, one would always prefer the highest baud rate possible, regardless of the CPU cost/message. In a multiuser situation, the CPU cost might be critical and one might compromise speed for CPU time. The following paragraphs discuss these factors as they relate to the configurations studied.

Two 11/24s communicating over a 56K DMC line perform almost exactly the same under DECnet 3.1 as under DECnet 4.0. The higher CPU cost/message of DECnet 3.1, compared to DECnet 4.0, is counterbalanced by its higher CPU utilization (see Graph 24).

Echo tests between two 11/24s over a 1 Meg DMC line show that a higher throughput can be obtained under DECnet 3.1 than under DECnet 4.0. Although the CPU time/message of DECnet 4.0 is lower than that of DECnet 3.1, the older version uses a much higher percentage of its CPU power to overcome this difference. The baud rates on the 1 Meg line exceeded three times the baud rates on the 56K line, for large user message sizes. The CPU utilizations were between two and three times higher on the 1 Meg line than on the 56K line.

Two 11/44s connected by a 1 Meg DMC have similar user baud rates for user message sizes under 512 bytes. Version 4.0 then has significantly higher user baud rates than Version 3.1. The CPU cost/message for user message sizes larger than 256 bytes is cheaper for the Phase IV DECnet than for the old DECnet (see Graph 25). Therefore, because the CPU utilizations are fairly close in value, DECnet 4.0 performs better than DECnet 3.1 for large user message sizes, and worse than DECnet 3.1 for small user message sizes.

The significant difference between 11/24 configurations and all the other configurations tested was the incredibly high CPU utilization (98+%) of the processor.

DECnet 3.1 outperforms DECnet 4.Ø for all message sizes, but the difference in performance becomes more significant when user message sizes are 512 bytes or larger. Here, the CPU cost/message of DECnet 3.1 is substantially lower than that of DECnet 4.Ø.

The 11/44 --> 11/75Ø configuration using a 1 Meg line is worth noting for several reasons. It yields the highest user baud rate of all the configurations tested (approaching 221K user bits/second). Also, it is the only configuration where DECnet 4.Ø clearly is faster than DECnet 3.1, although the CPU cost/message is higher. For a change, DECnet 4.Ø uses more of its CPU power than DECnet 3.1, and therefore has a higher user baud rate.

NFT Results

DECnet 4.Ø is slightly faster than DECnet 3.1 when doing local to remote file transfers between two 11/24s over a 56K DMC line (see Graph 26). Although the newer DECnet version has a higher CPU cost/block than DECnet 3.1, its higher CPU utilization compensates for this (see Graph 27). As user files get larger than 512 blocks, the line speed becomes the limiting factor, not the CPU speed. Notice how the CPU utilization slows its rate of increase once the line becomes heavily used.

In the 11/24 --> 11/24 configuration, using a 1 Meg DMC line, DECnet 4.Ø is slower than DECnet 3.1. This is not due to a difference in CPU cost/block transferred. It is caused by the higher CPU utilization of DECnet 3.1 compared to DECnet 4.Ø (see Chart 3). Therefore, even when the CPU cost/block is higher using DECnet 3.1 than 4.Ø, its higher processor use overcomes this (except in 1 block file transfers).

In some cases, such as communication between two 11/44s and an 11/44 and an 11/75Ø, over a 1 Meg line, the CPU cost/block is cheaper for one DECnet version than another. Because there is no significant difference in CPU utilization between the two DECnets, the version with the cheaper CPU time/block is faster (see Graph 28). Therefore, when two 11/44s are doing file transfers, DECnet 4.Ø is slower than DECnet 3.1. If an 11/44 is transferring files with an 11/75Ø, DECnet 3.1 is faster than DECnet 4.Ø.

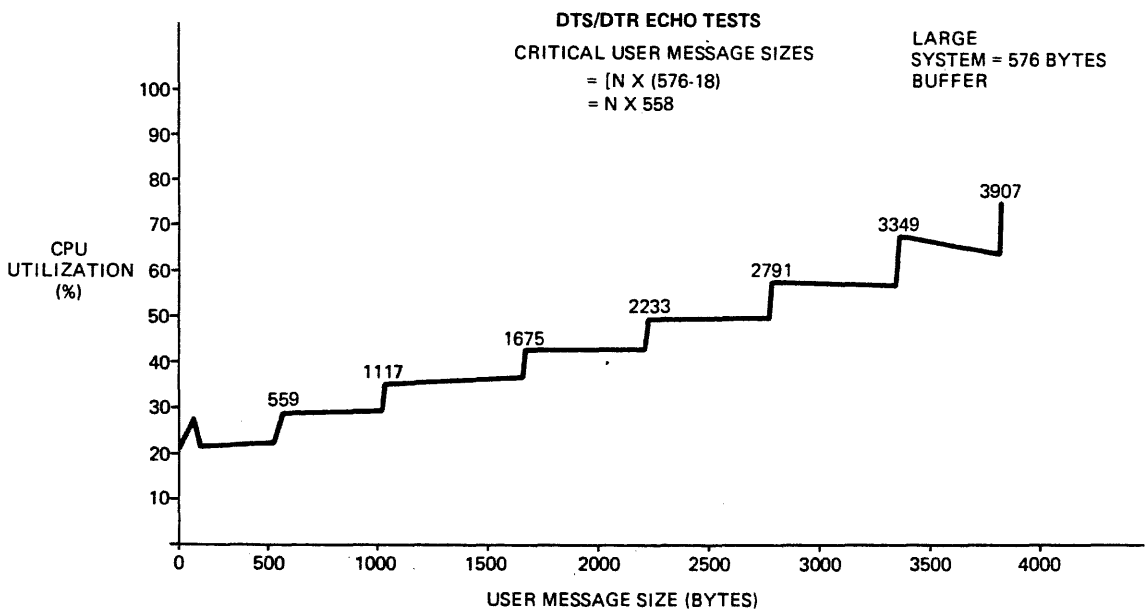
DECnet 4.0 VERSUS DECnet 2.1

DTS/DTR Results

DTS/DTR tests confirm that DECnet 4.0 and DECnet 2.1 have similar performance capabilities in task-to-task communication. DECnet M and DECnet M+ user baud rates and CPU times/message are within 10% of each other. DECnet 4.0 usually has a slightly higher user baud rate and uses slightly less CPU time/message than DECnet 2.1 (see Graph 29).

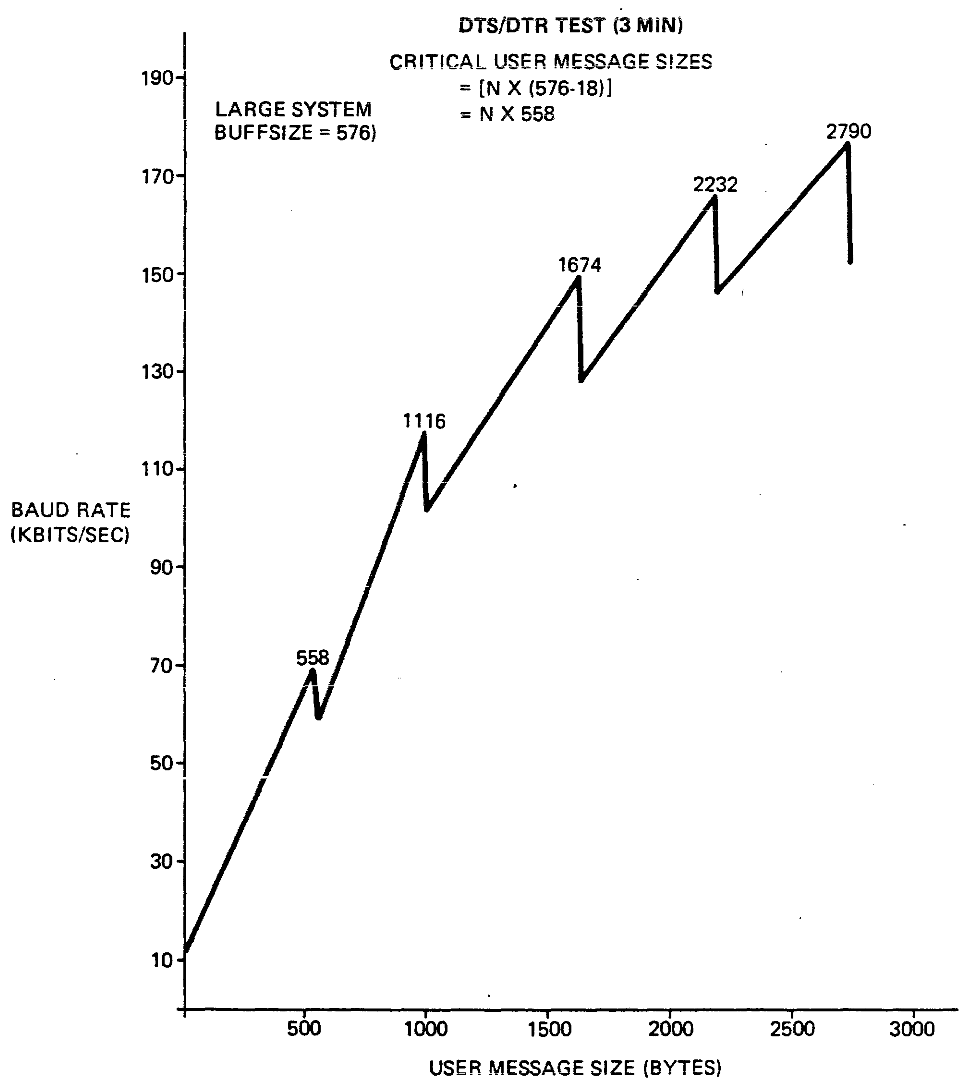
NFT Results

NFT tests show a marked difference in file transfer times between DECnet M and DECnet M+. DECnet 2.1 is anywhere between 7-50% faster than DECnet 4.0 (see Chart 4). DECnet M tends to use less CPU time/block than DECnet M+, except when small (< 128 blocks) files are moved (see Graphs 30 and 31).



GRAPH 1

MKV84-1125



GRAPH 2

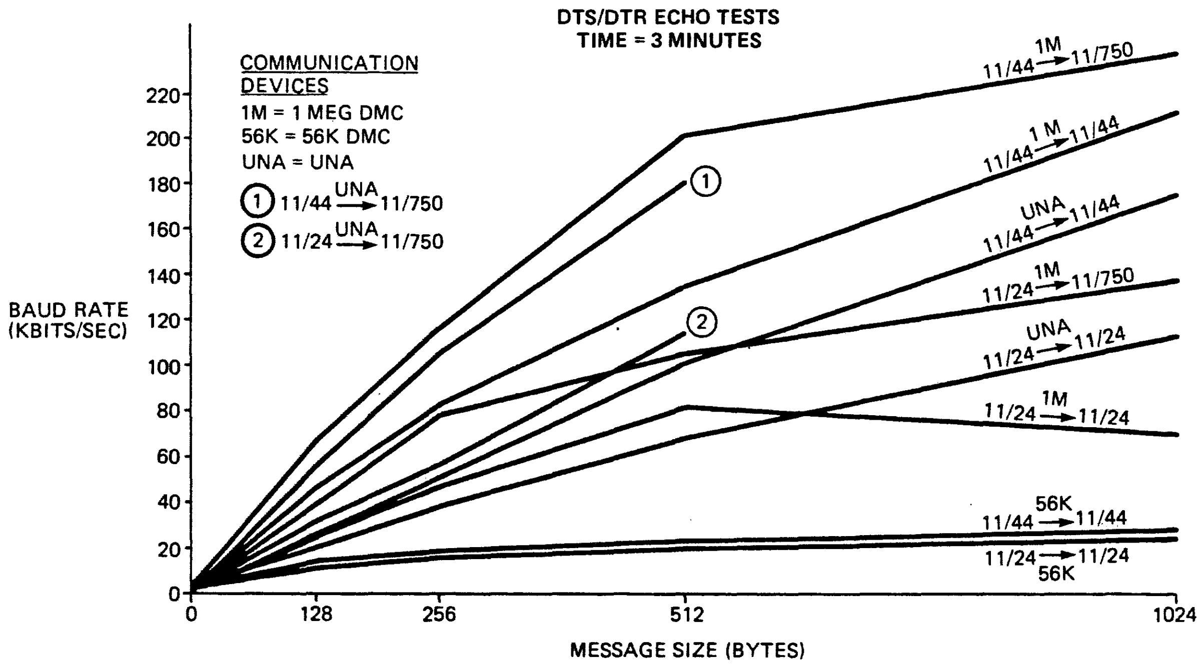
MKV84-1126

Table 3 Three Minute DTS/DTR ECHO Tests

Processors	Comm. Device	Mess. Size (bytes)	Baud	CPU Util.	CPU/mess. (ms/mess.)
11/24-->11/24	UNA	1			
		128	21053	69.4	16.84
		256	44637	75.72	17.37
		512	75912	67.82	18.3
		1024	62376	36.93	24.24
11/24-->11/24	56K DMC	1	128	54.89	16.49
		128	10443	34.38	16.83
		256	15072	25.29	17.18
		512	18432	19.25	21.39
		1024	21480	13.89	26.57
11/44-->11/44	UNA	1	240	73.8	12.18
		128	29672	73.16	12.62
		256	47634	60.17	12.96
		512	92573	62.37	13.79
		1024	162579	70.42	17.74
11/44-->11/44	56K DMC	1	160	37.73	9.19
		128	12128	2.62	9.54
		256	16792	16.03	9.78
		512	20736	10.3	10.17
		1024	22752	7.34	13.22
11/44-->11/44	1 Meg DMC	1	371	83.96	8.94
		128	43661	83.04	9.74
		256	77637	75.09	9.9
		512	125520	65.18	10.62
		1024	196544	67.01	13.96
11/24-->11/750	UNA	1	208	99.14	18.52
		128	27427	99.14	18.5
		256	51317	99.14	19.78
		512	105309	99.14	19.28
		1024			
11/24-->11/750	1 Meg DMC	1	296	98.14	13.04
		128	35595	99.14	14.23
		256	73701	99.14	13.77
		512	98304	99.14	20.66
		1024	126675	98.7	31.86

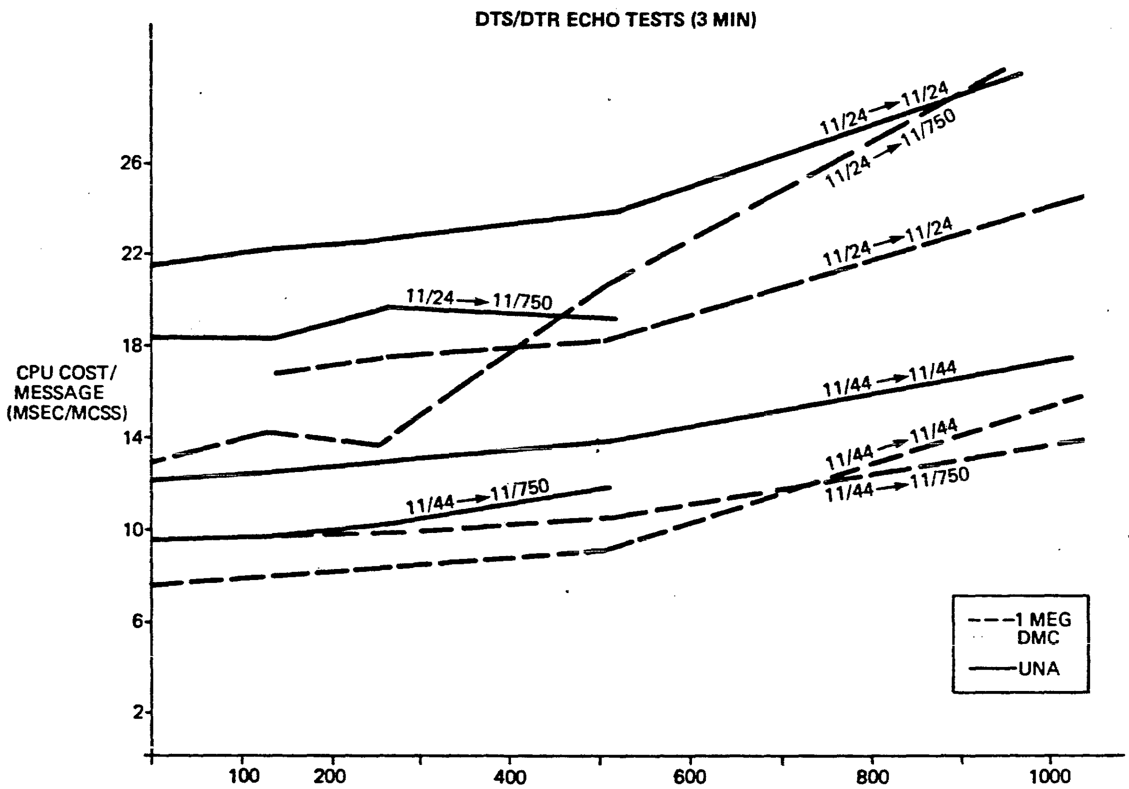
Table 3 Three Minute DTS/DTR ECHO Tests (Cont)

Processors	Comm. Device	Mess. Size (bytes)	Baud	CPU Util.	CPU/mess. (ms/mess.)
11/44-->11/750	UNA	1			
		128	403	96.24	9.64
		256	98877	98.7	9.981
		512	169488	98.83	10.23
		1024	na	98.37	11.89
11/44-->11/750	1 Meg DMC	1	477	93.22	7.87
		128	61227	96.09	8.03
		256	109749	89.37	8.34
		512	187325	84.09	9.18
		1024	220725	84.81	15.74



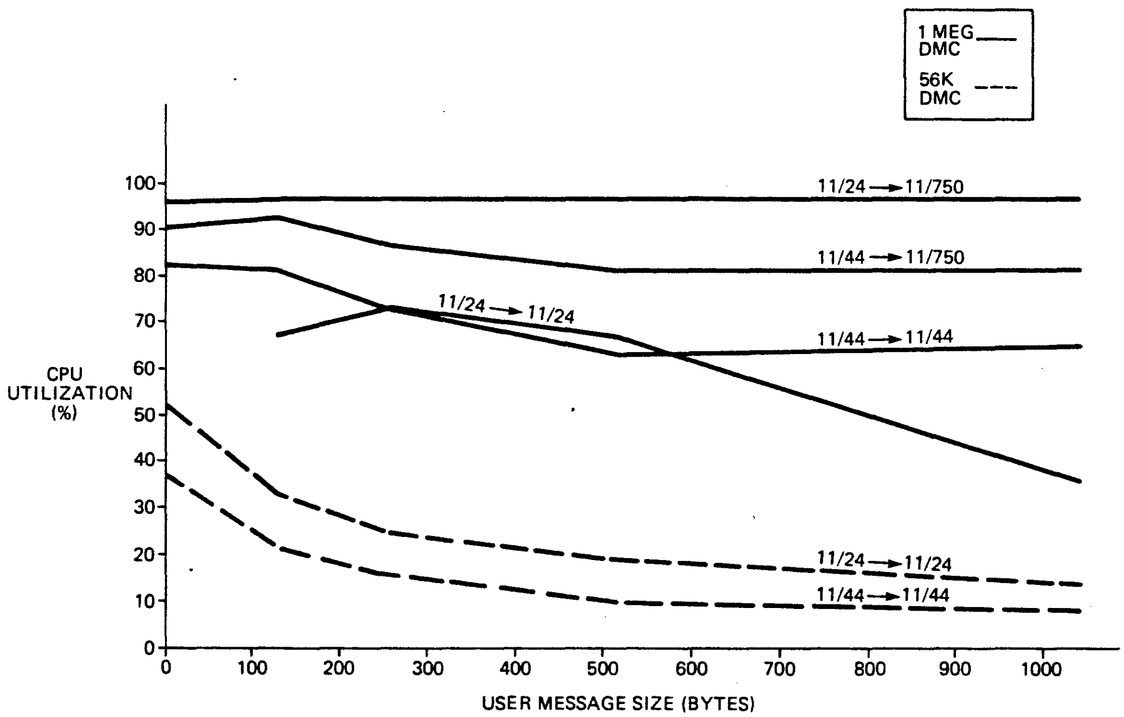
GRAPH 3

MKV84-1127



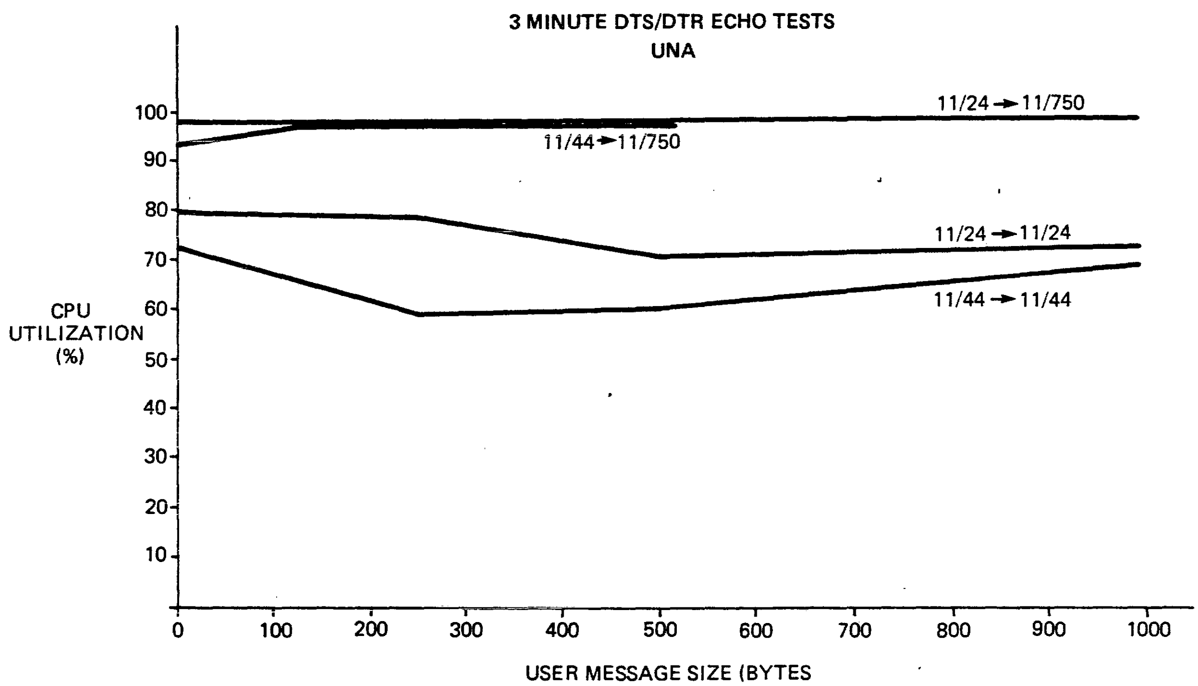
GRAPH 4

MKV84-1128



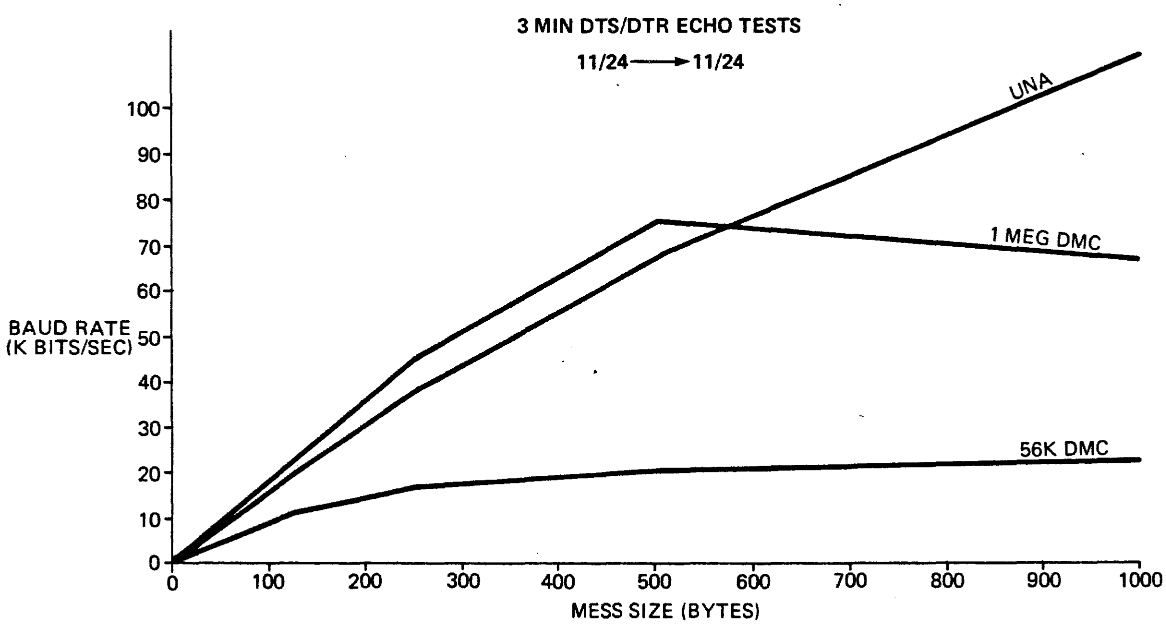
GRAPH 5

MKV84-1129



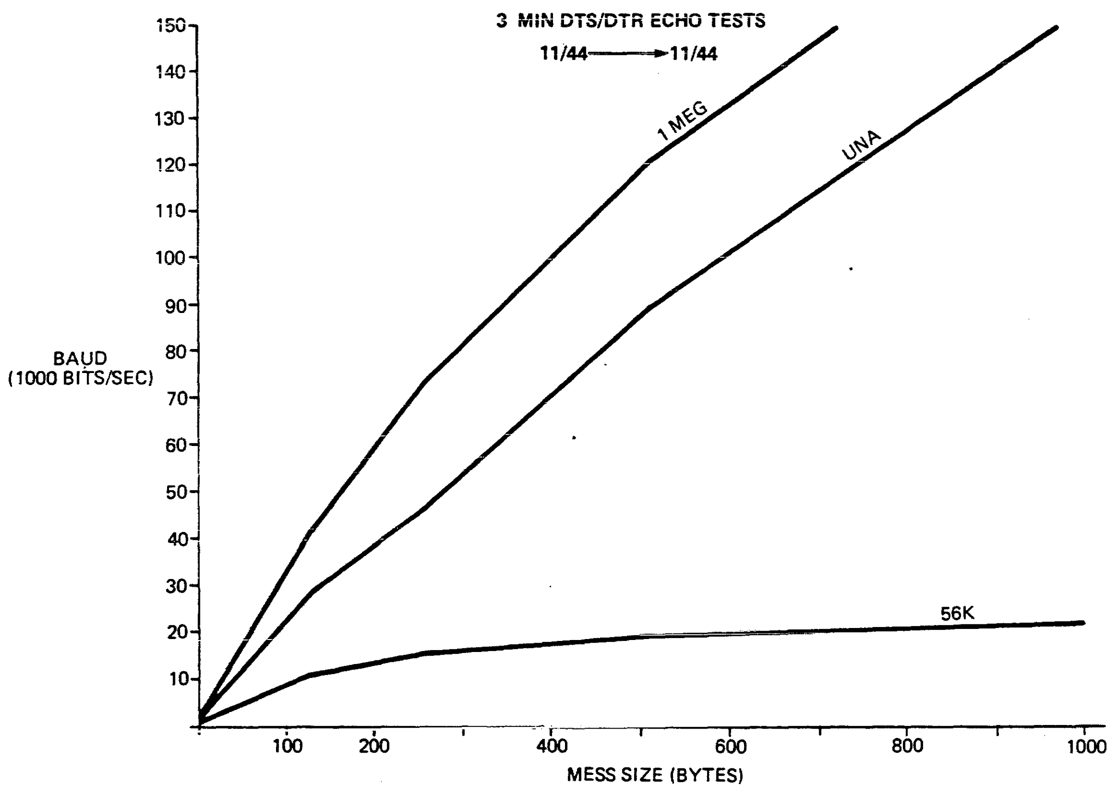
GRAPH 6

MKV84-1130



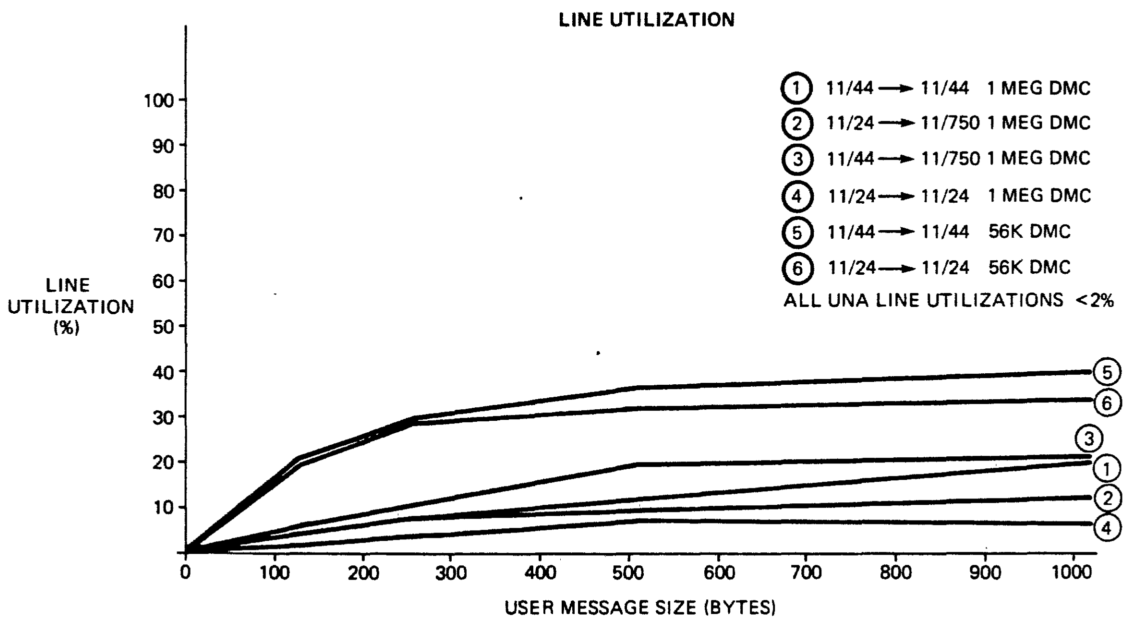
GRAPH 7

MKV84-1131



GRAPH 8

MKV84-1132


LINE UTILIZATION
1 11/44 → 11/44 1 MEG DMC
2 11/24 → 11/750 1 MEG DMC
3 11/44 → 11/750 1 MEG DMC
4 11/24 → 11/24 1 MEG DMC
5 11/44 → 11/44 56K DMC
6 11/24 → 11/24 56K DMC
ALL UNA LINE UTILIZATIONS <2%
LINE UTILIZATION (%)
100
90
80
70
60
50
40
30
20
10
0
100
200
300
400
500
600
700
800
900
1000
USER MESSAGE SIZE (BYTES)
MKV84-1133

GRAPH 9

Table 4 NFT Transfers -- FCS versus RMS

NFT File Transfers (FCS FAL)

Time (In Seconds)

(FCS FAL)

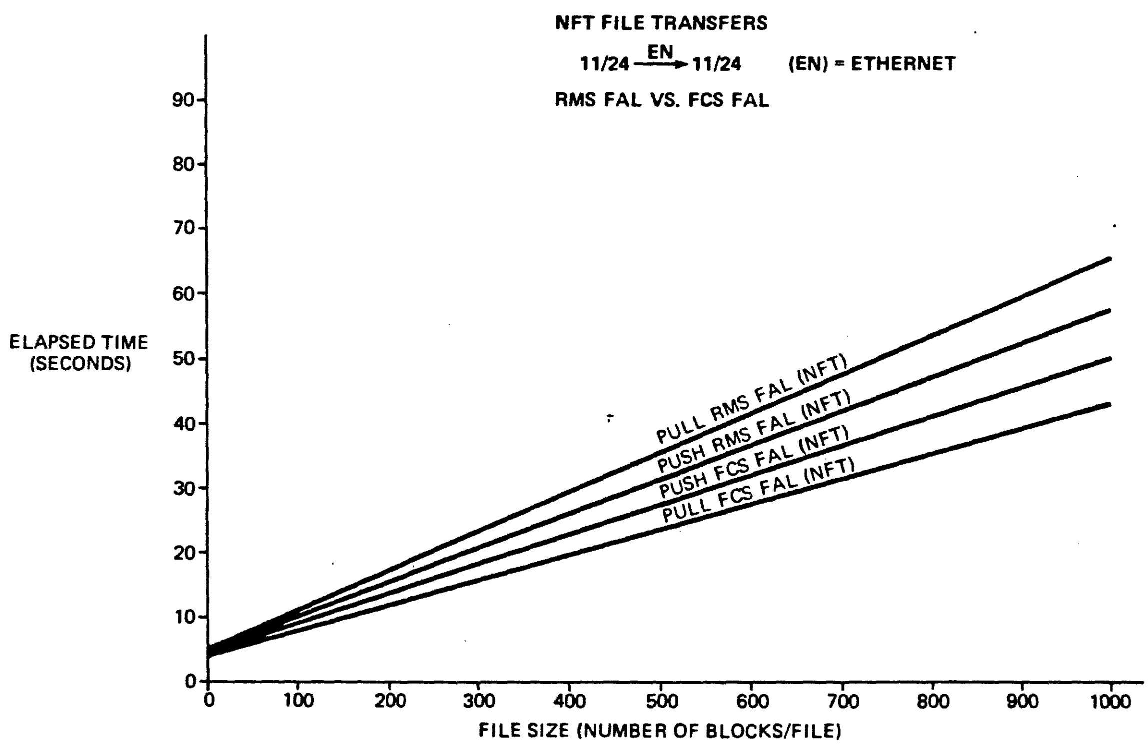
		Push					Pull				
		File Size (Blocks)									
Processors	**Comm Device**	**1**	**128**	**256**	**512**	**1024**	**1**	**128**	**256**	**512**	**1024**
11/44->11/750	1 Meg DMC	4	6	10	17		4	7	10	17	31
11/44->11/750	UNA	4	7	10	19	35	3	9	14	25	47
11/24->11/750	1 Meg DMC	4	8	12	21		3	10	16	29	56
11/44->11/44	UNA	6	9	12			5	8	12	19	
11/24->11/750	UNA	5	8	13	23		3	10	17	30	57
11/44->11/44	1 Meg DMC	5	9	23	14		5	8	12	19	
11/24->11/24	1 Meg DMC	6	12	22	32	71	5	12	17	29	53
11/24->11/24	UNA	6	14	22	39	72	5	13	20	35	64
11/24->11/24	56K DMC	5	17	29	54	103	5	17	29	54	103

NFT File Transfers (RMS FAL)

Time (In Seconds)

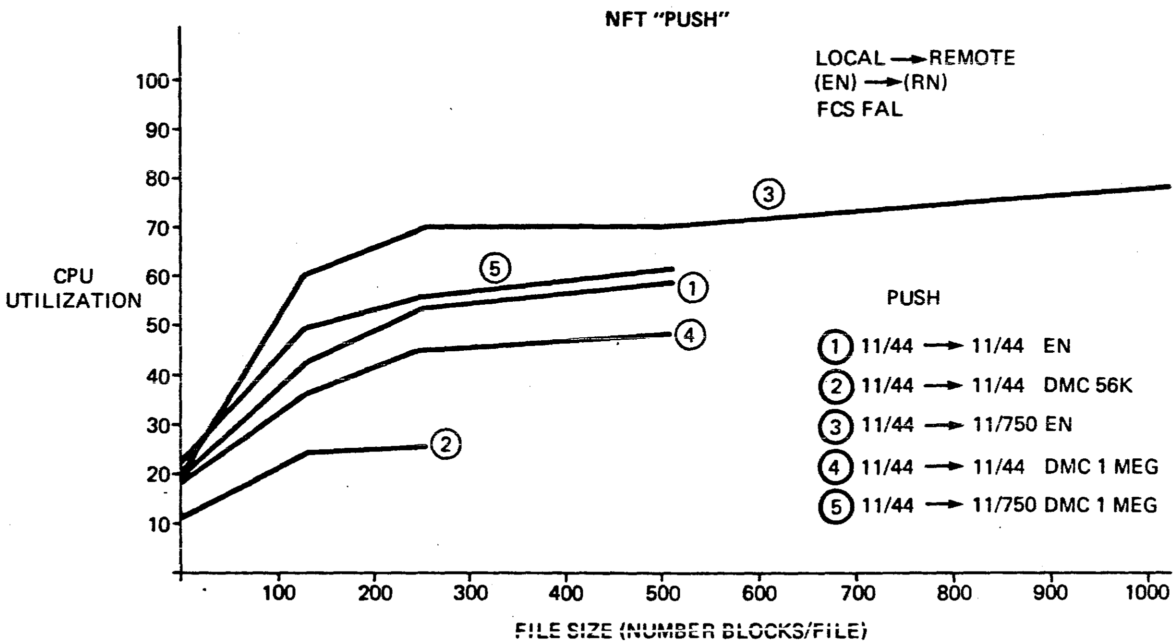
(RMS FAL)

		Push					Pull				
		File Size (Blocks)									
Processors	**Comm Device**	**1**	**128**	**256**	**512**	**1024**	**1**	**128**	**256**	**512**	**1024**
11/44->11/750	UNA	4	6	10	17		3	7	10	17	31
11/44->11/750	1 Meg DMC	4	6	10	17		3	6	10	17	31
11/24->11/750	1 Meg DMC	4	8	13			3	10	17	30	
11/44->11/44	UNA	6	9	13			5	9	13	21	
11/24->11/750	UNA	5	8	13	23		3	10	17	30	57
11/44->11/44	1 Meg DMC	6	9	12	16		5	8	13	19	
11/24->11/24	1 Meg DMC	6	13	16	21	49	5	11	17	28	58
11/24->11/24	UNA	6	16	25	46	85	6	16	27	50	94
11/44->11/44	56K DMC	5	10	28			6	17	29	53	
11/24->11/24	56K DMC	6	18	30	55	105	5	18	30	56	108



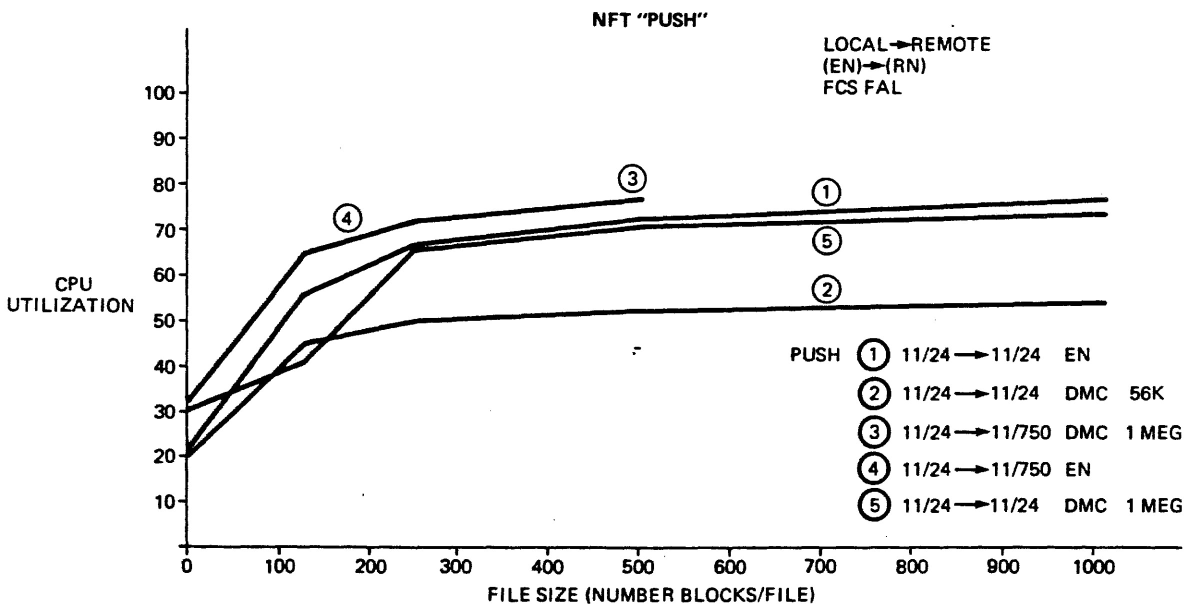
GRAPH 10

MKV84-1134



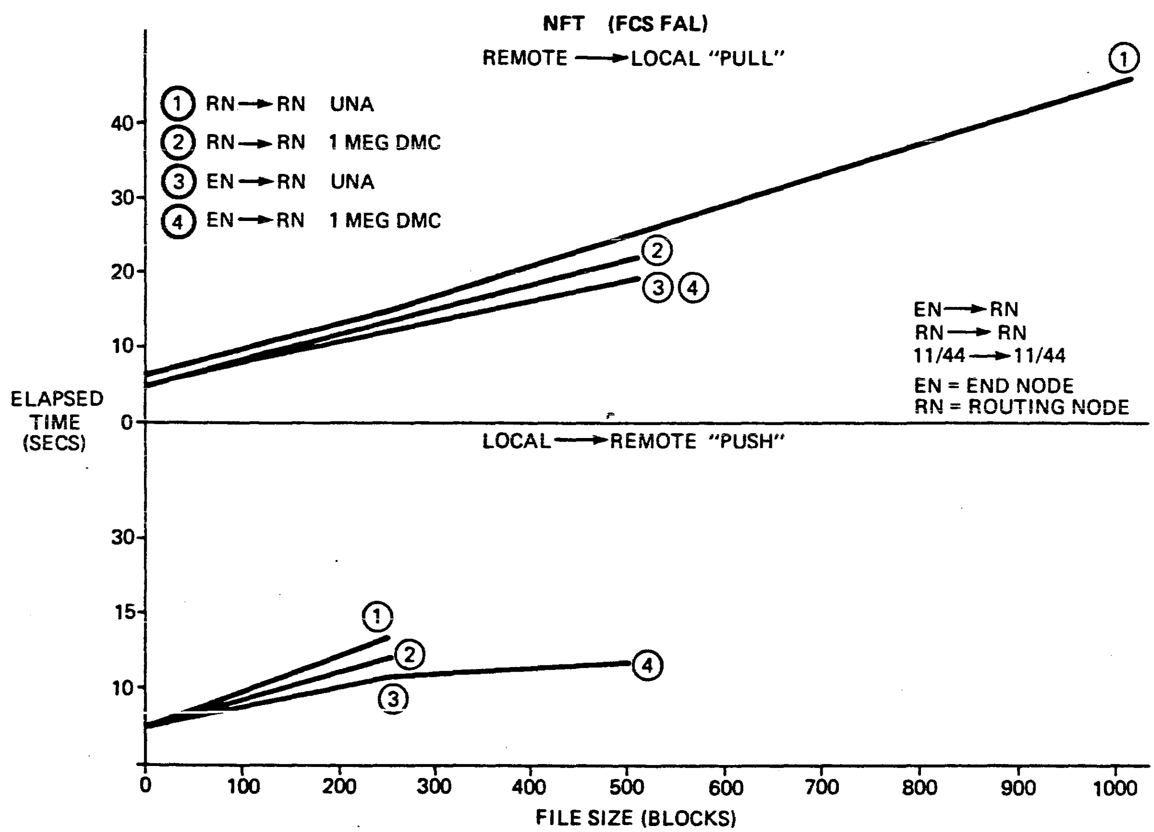
GRAPH 11

MKV84-1135



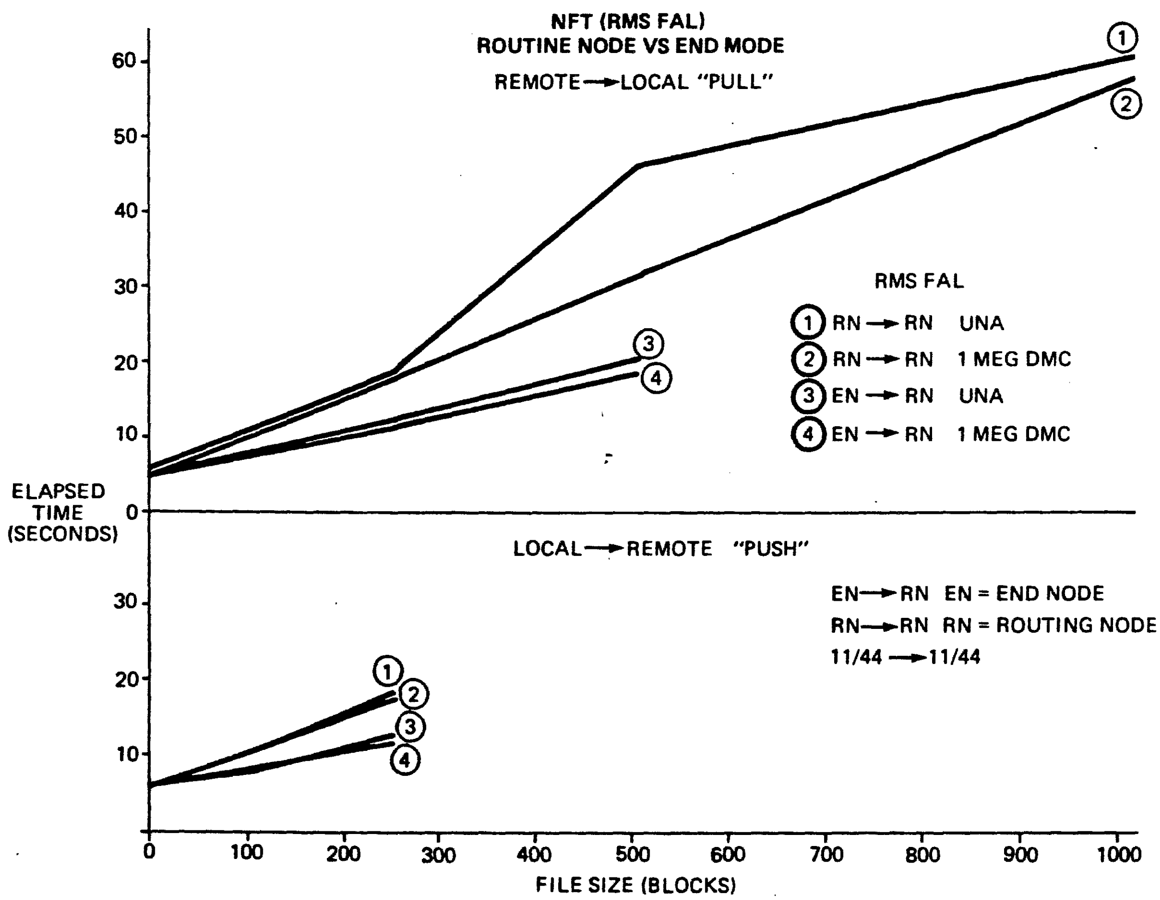
GRAPH 12

MKV84-1136



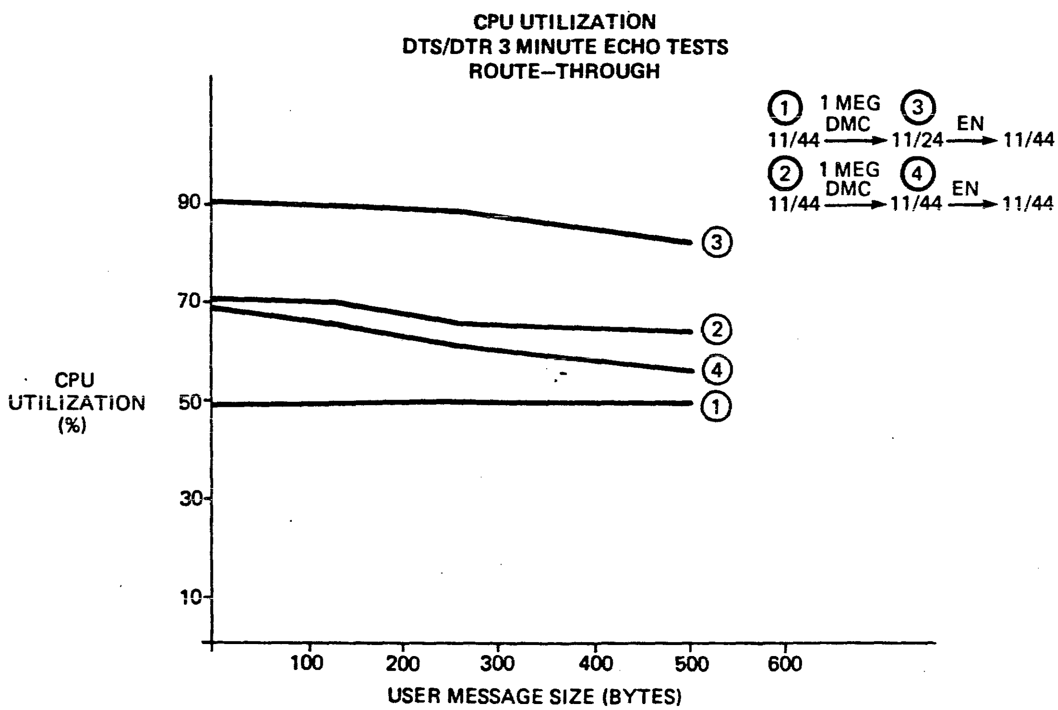
GRAPH 13

MKV84-1137



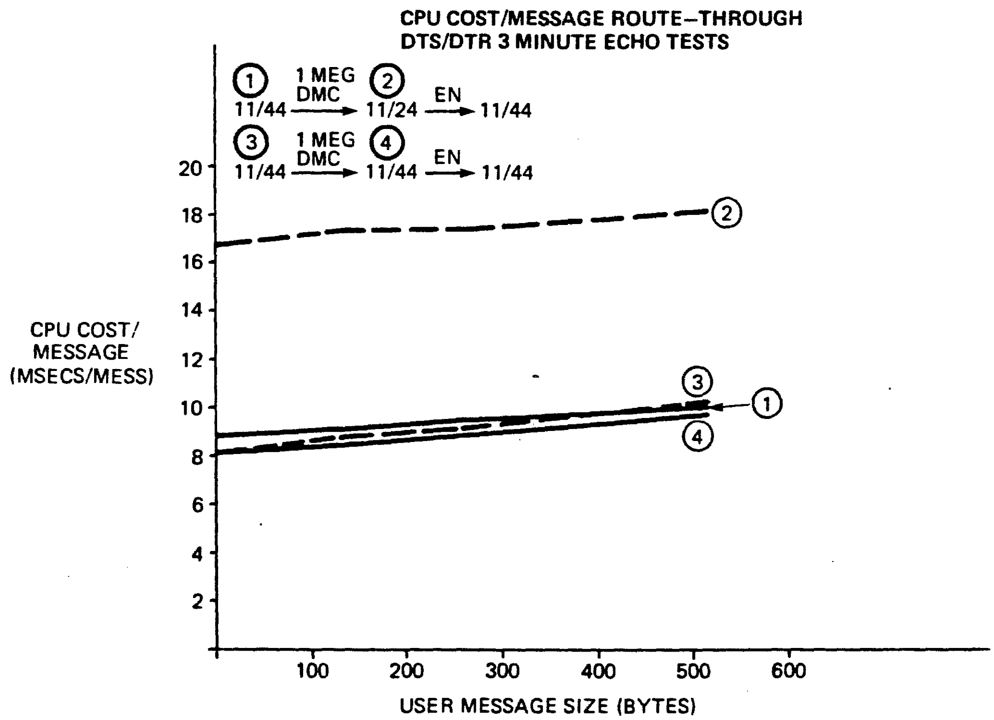
GRAPH 14

MKV84-1138



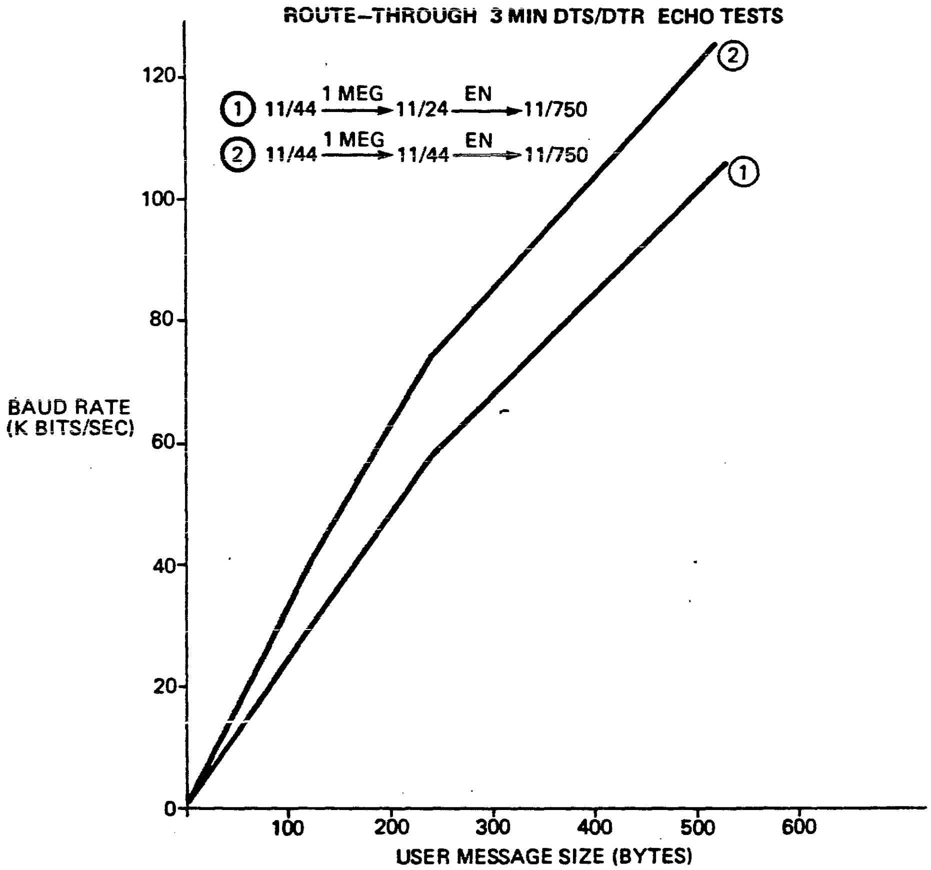
GRAPH 15

MKV84-1139



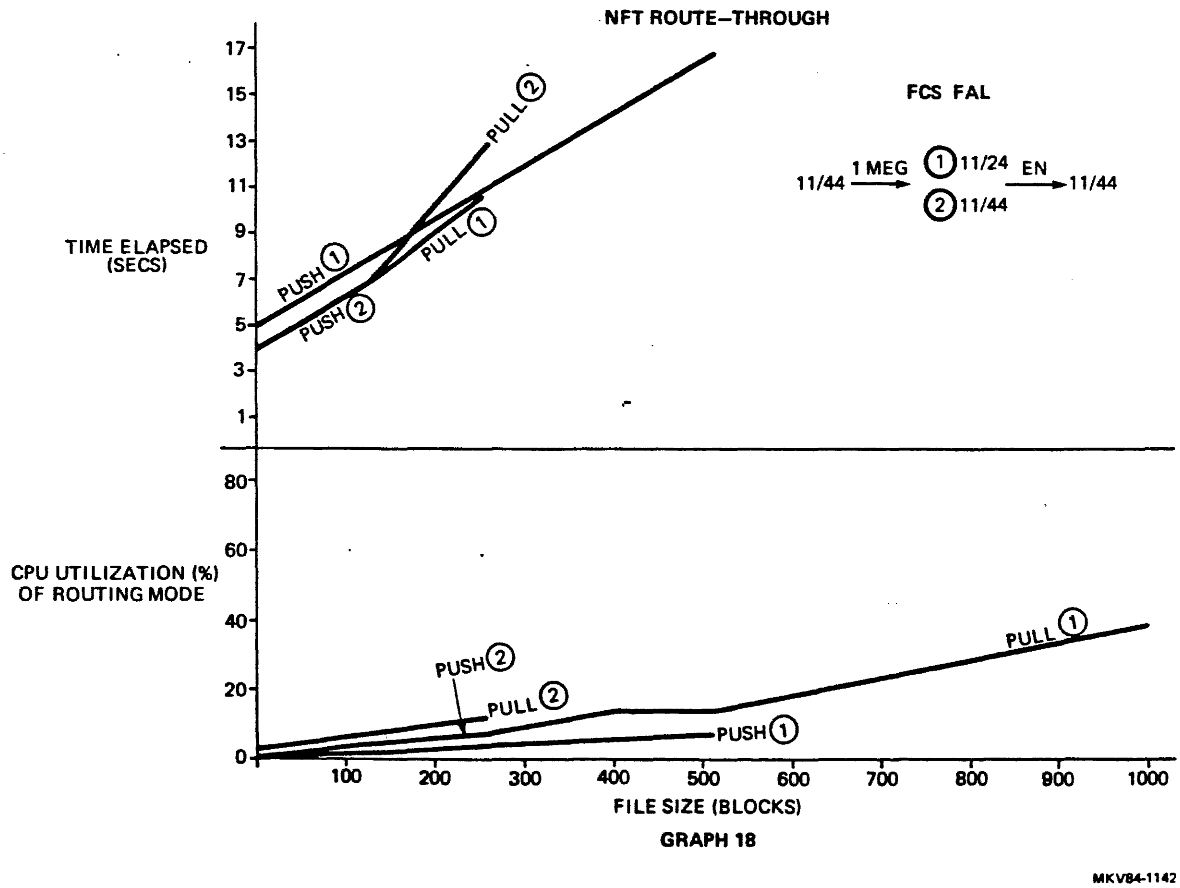
GRAPH 16

MKV84-1140

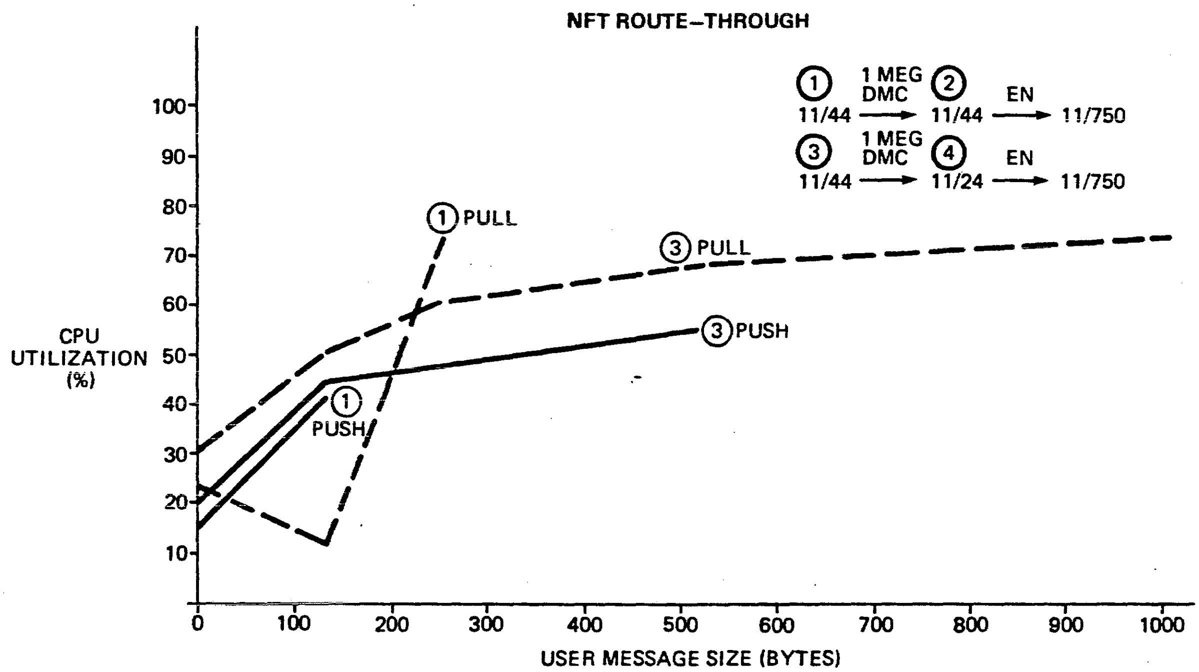


GRAPH 17

MKV84-1141

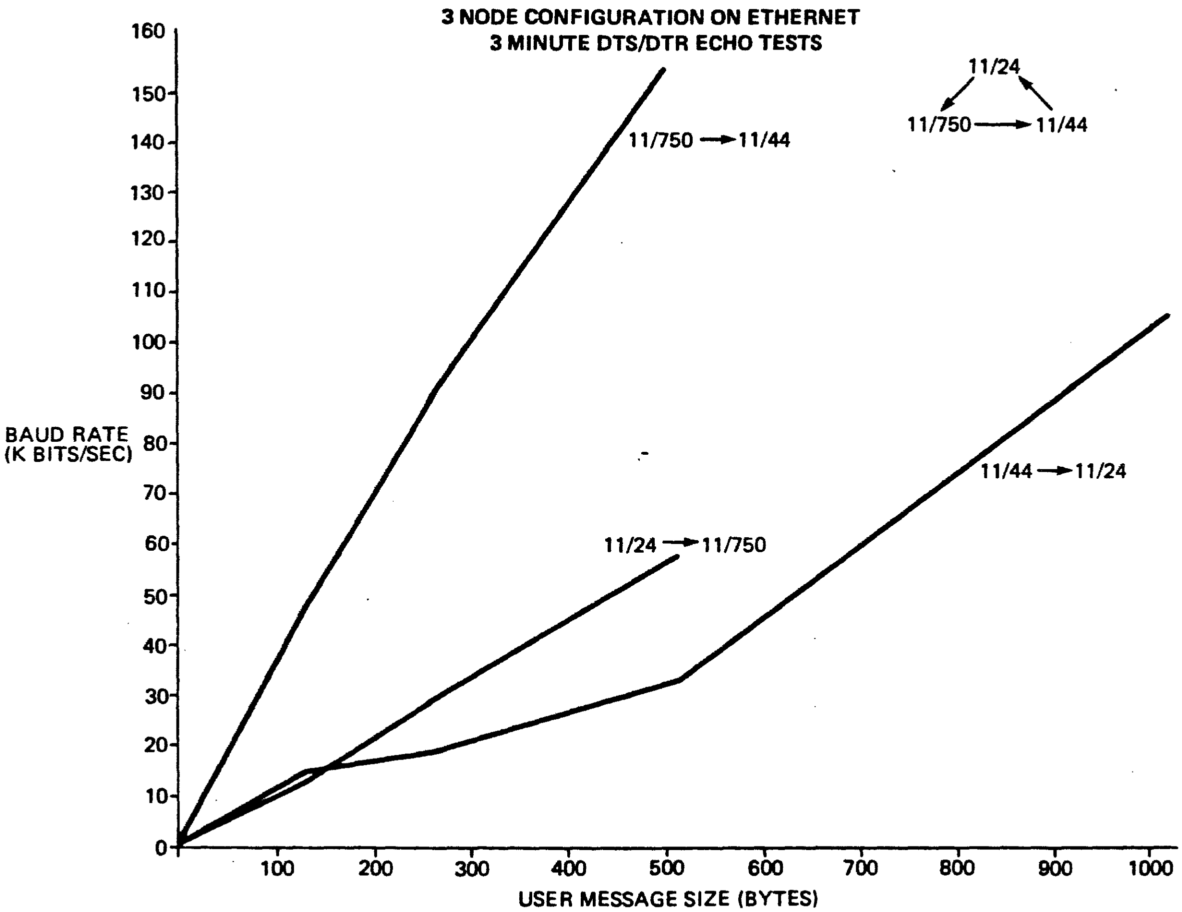

NFT ROUTE–THROUGH
FCS FAL
11/44
1 MEG
1 11/24
2 11/44
EN
11/44
TIME ELAPSED (SECS)
17
15
13
11
9
7
5
3
1
PULL 2
PUSH 1
PULL 1
PUSH 2
CPU UTILIZATION (%) OF ROUTING MODE
80
60
40
20
0
PUSH 2
PULL 2
PUSH 1
PULL 1
100
200
300
400
500
600
700
800
900
1000
FILE SIZE (BLOCKS)
MKV84-1142

GRAPH 18



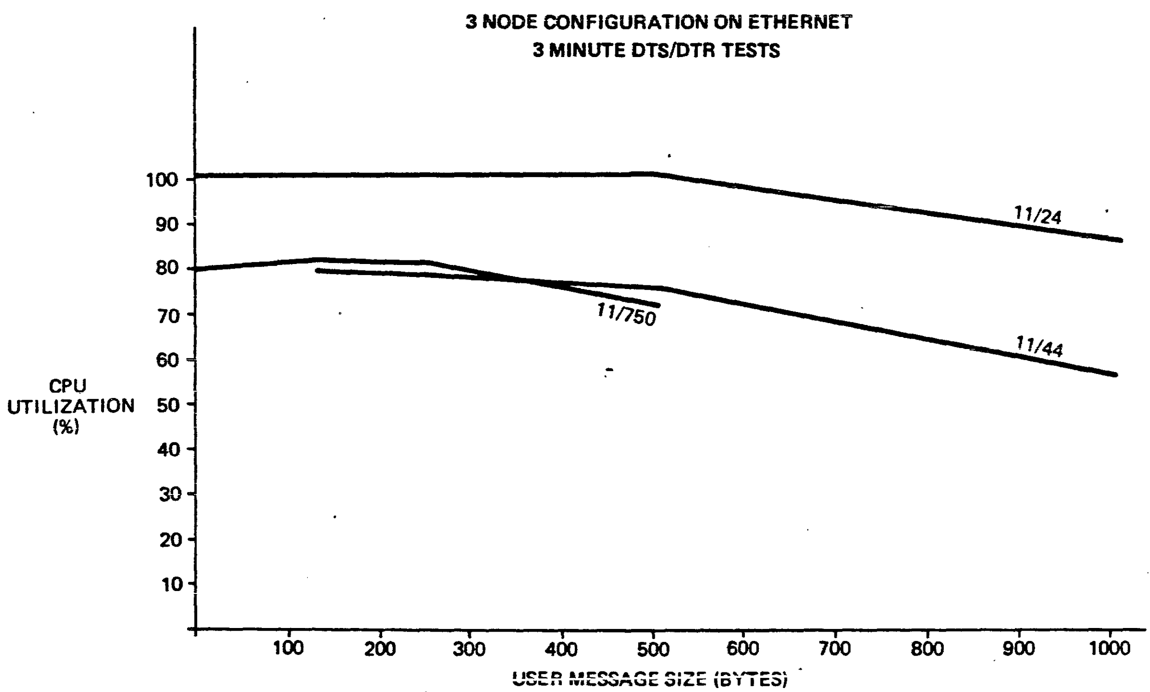
GRAPH 19

MKV84-1143



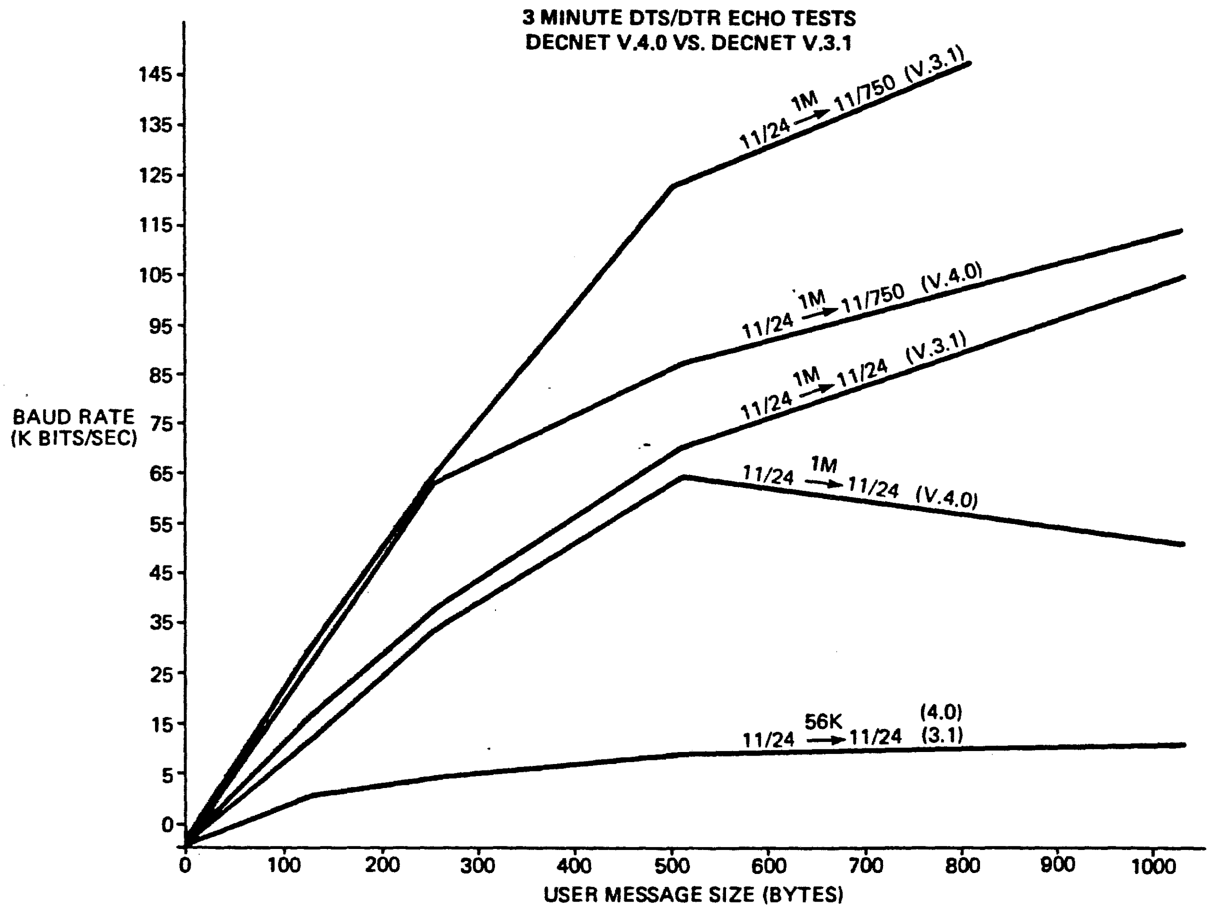
GRAPH 20

MKV84-1144



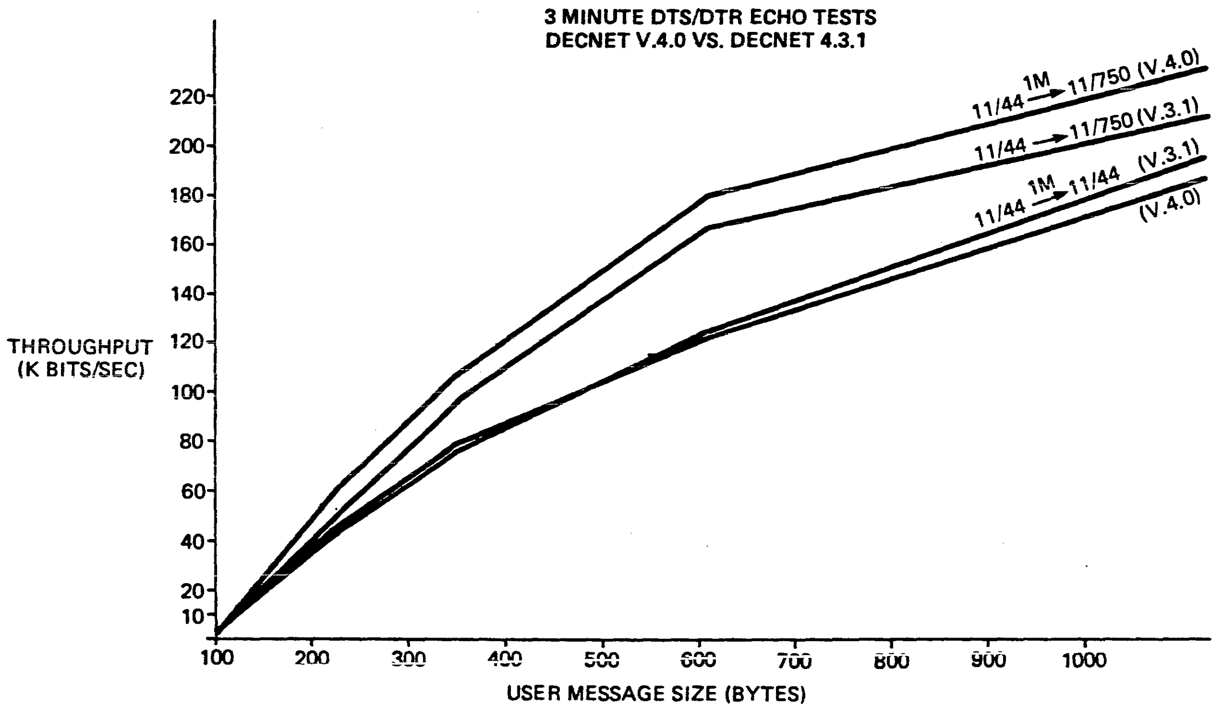
GRAPH 21

MKV84-1145



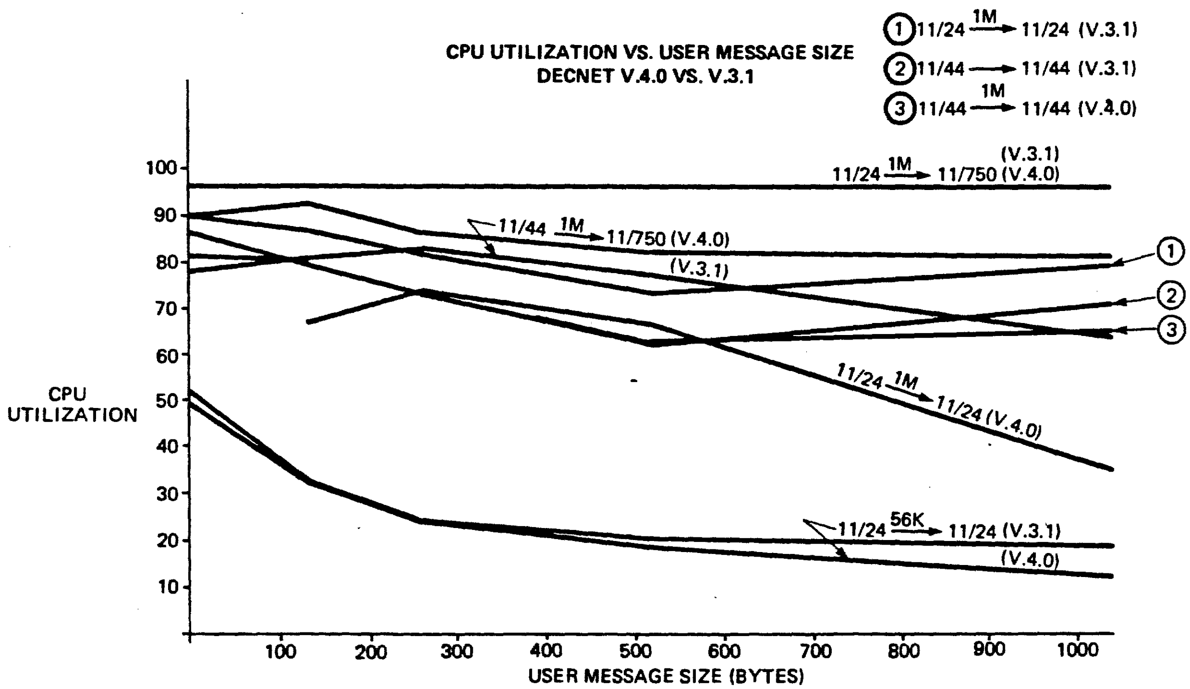
GRAPH 22

MKV84-1146



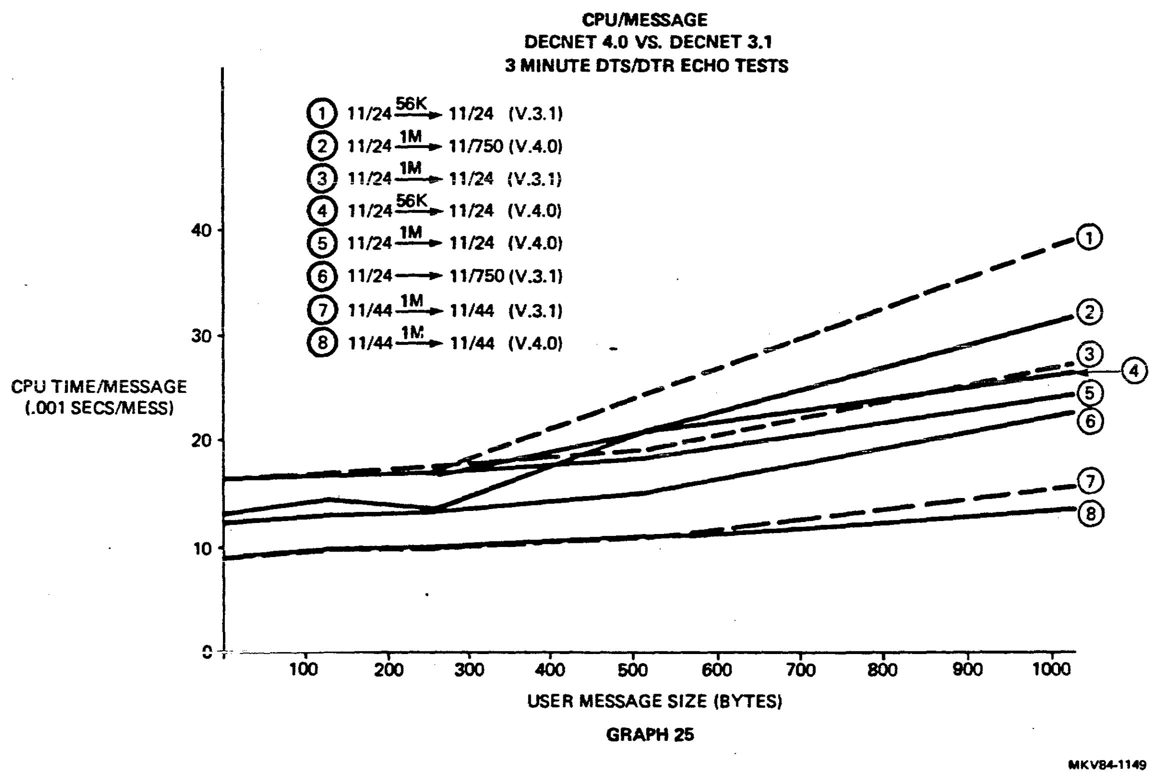
GRAPH 23

MKV84-1147



GRAPH 24

MKV84-1148


CPU/MESSAGE
DECNET 4.0 VS. DECNET 3.1
3 MINUTE DTS/DTR ECHO TESTS
1 11/24 56K → 11/24 (V.3.1)
2 11/24 1M → 11/750 (V.4.0)
3 11/24 1M → 11/24 (V.3.1)
4 11/24 56K → 11/24 (V.4.0)
5 11/24 1M → 11/24 (V.4.0)
6 11/24 → 11/750 (V.3.1)
7 11/44 1M → 11/44 (V.3.1)
8 11/44 1M → 11/44 (V.4.0)
CPU TIME/MESSAGE
(.001 SECS/MESS)
40
30
20
10
0
100
200
300
400
500
600
700
800
900
1000
USER MESSAGE SIZE (BYTES)
MKV84-1149

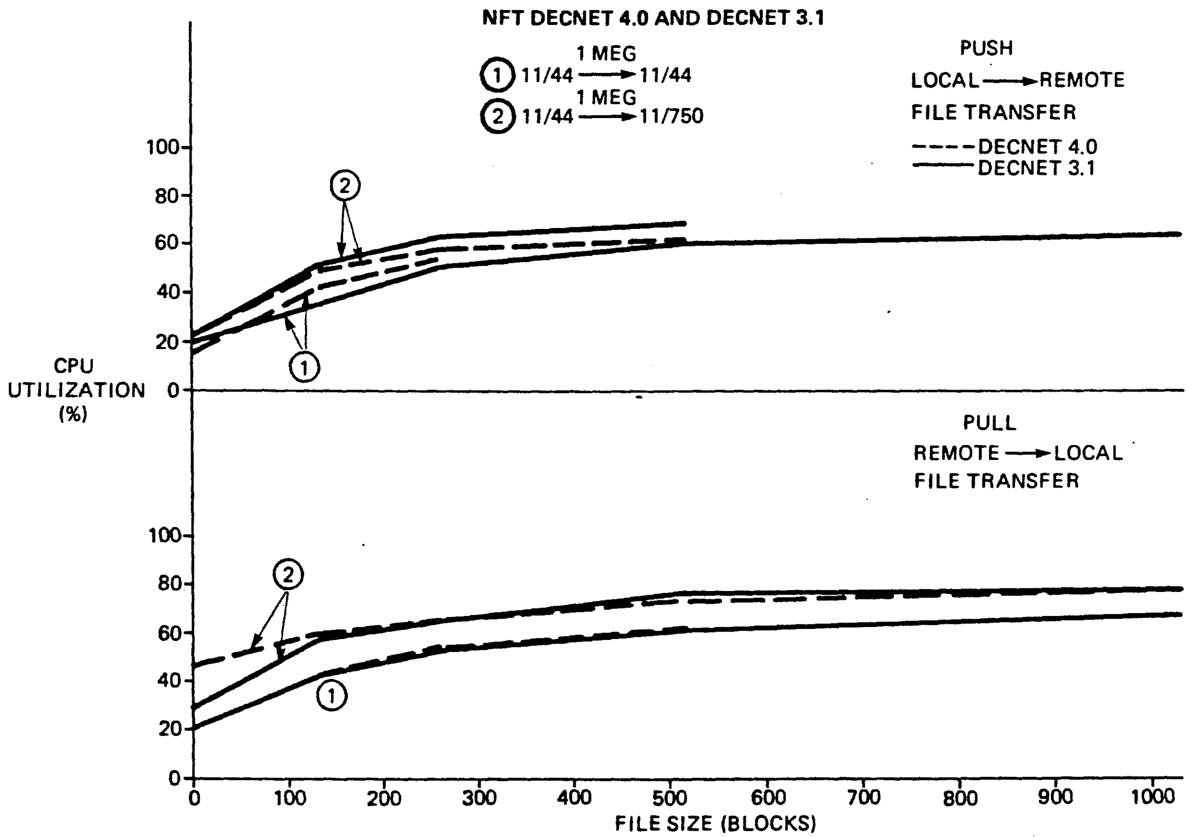
GRAPH 25

Table 5 DECnet 4.0 versus DECnet 3.1
NFT File Transfers (FCS FAL)

	"PUSH" (Local -> Remote)					"PULL" (Remote -> Local)			
Configuration and DECnet Version	File Size (Blocks	Time (Seconds)	CPU/Block (msec/bl.)	CPU/Util. (%)	Line Util. (%)	Time (Seconds)	CPU Util. (%)	Line Util.	Line Util. (%)
56K DMC 11/24-->11/24									
DECnet 4.0	1	5	.948	18.97	1.46	5	1.29	25.86	1.46
	128	17	.059	44.22	55.07	10	.06	45.23	55.07
	256	29	.057	49.45	63.83	29	.056	48.86	63.83
	512	54	.055	52.27	69.34	54	.054	51.63	69.35
	1024	103	.054	53.65	72.95	103	.053	52.73	72.95
DECnet 3.1	1	6	1.09	18.1	1.22	5	20.8	1.37	18.1
	128	18	.048	34.39	52.01	17	37.83	54.01	52.01
	256	30	.046	38.9	61.73	30	40.72	63.12	61.73
	512	58	.05	44.14	64.19	54	42.85	69.35	64.19
	1024	104	.043	41.89	72.02	103	44.1	72.48	72.02
1 MEG DMC 11/24-->11/24									
DECnet 4.0	1	6	1.33	21.05	.06	5	1.03	20.69	.08
	128	12	.043	45.4	4.37	12	.044	48.28	4.49
	256	22	.038	44.36	4.77	19	.039	51.91	5.52
	512	32	.036	57.8	6.62	29	.035	61.79	7.15
	1024	71	.032	46.45	5.93	53	.034	65.09	7.86
DECnet 3.1	1	6	1.1	19.37	.07	6	1.2	22.41	.077
	128	11	.049	55.12	4.63	10	.044	54.12	5.07
	256	16	.039	61.75	6.55	16	.04	65.88	6.69
	512	26	.036	70.86	7.96	26	.038	75.13	0.07
	1024	47	.035	75.96	8.86	46	.036	80.1	9.12

Table 5 DECnet 4.0 versus DECnet 3.1
NFT File Transfers (FCS FAL) (Cont)

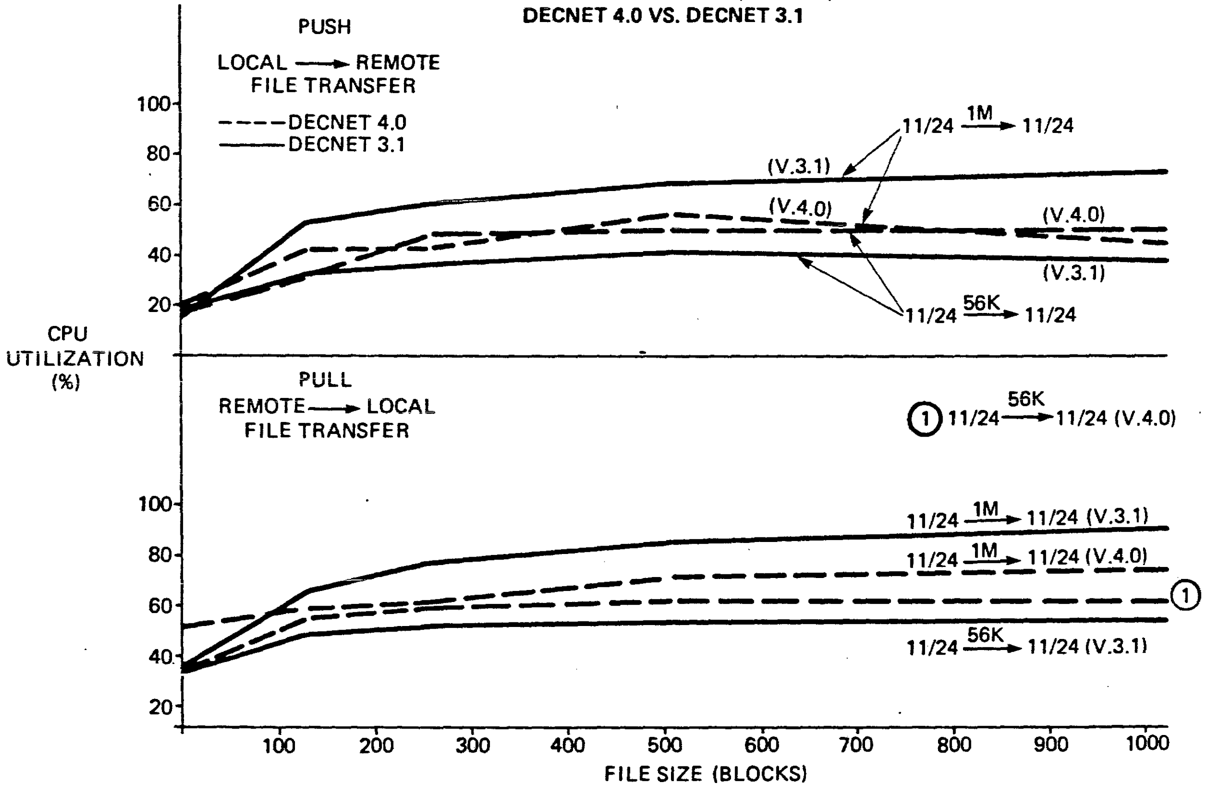
	"PUSH" (Local -> Remote)					"PULL" (Remote -> Local)			
Configuration and DECnet Version	File Size (Blocks	Time (Seconds)	CPU/Block (msec/bl.)	CPU/Util. (%)	Line Util. (%)	Time (Seconds)	CPU Util. (%)	Line Util.	Line Util. (%)
1 Meg DMC 11/44-->11/44 DECnet 4.0	1	5	.814	16.28	.082	5	.95	20.36	.007
	128	9	.031	41.31	5.42	9	.033	44.67	5.62
	256	12	.03	52.52	7.15	14	.029	54.48	7.67
	512					22	.027	62.3	9.53
	1024								
DECnet 3.1	1	5	1.07	20.04	.077	5	1.15	21.51	.077
	128	9	.027	37.54	5.6	9	.03	43.57	6.05
	256	12	.024	50.51	8.5	12	.025	53.7	8.74
	512	19	.022	59.12	10.85	19	.023	60.94	10.85
	1024	37	.021	64.21	12.46	33	.022	66.8	12.71
1 Meg DMC 11/44-->11/750 DECnet 4.0	1	4	.811	22.11	.11	4	1.79	48.78	.11
	128	6	.024	49.35	8.28	7	.032	60.68	7.86
	256	10	.022	56.57	10.49	10	.027	67.92	10.49
	1024					31	.023	77.93	13.68
DECnet 3.1	1	4	.831	20.77	.102	4	1.08	29.58	.11
	128	7	.03	51.98	7.15	7	.034	58.92	7.15
	256	11	.026	62.59	9.83	11	.03	66.86	9.25
	512	18	.024	67.35	11.44	19	.028	75.48	11.23
	1024					34	.026	79.5	7.77



GRAPH 26

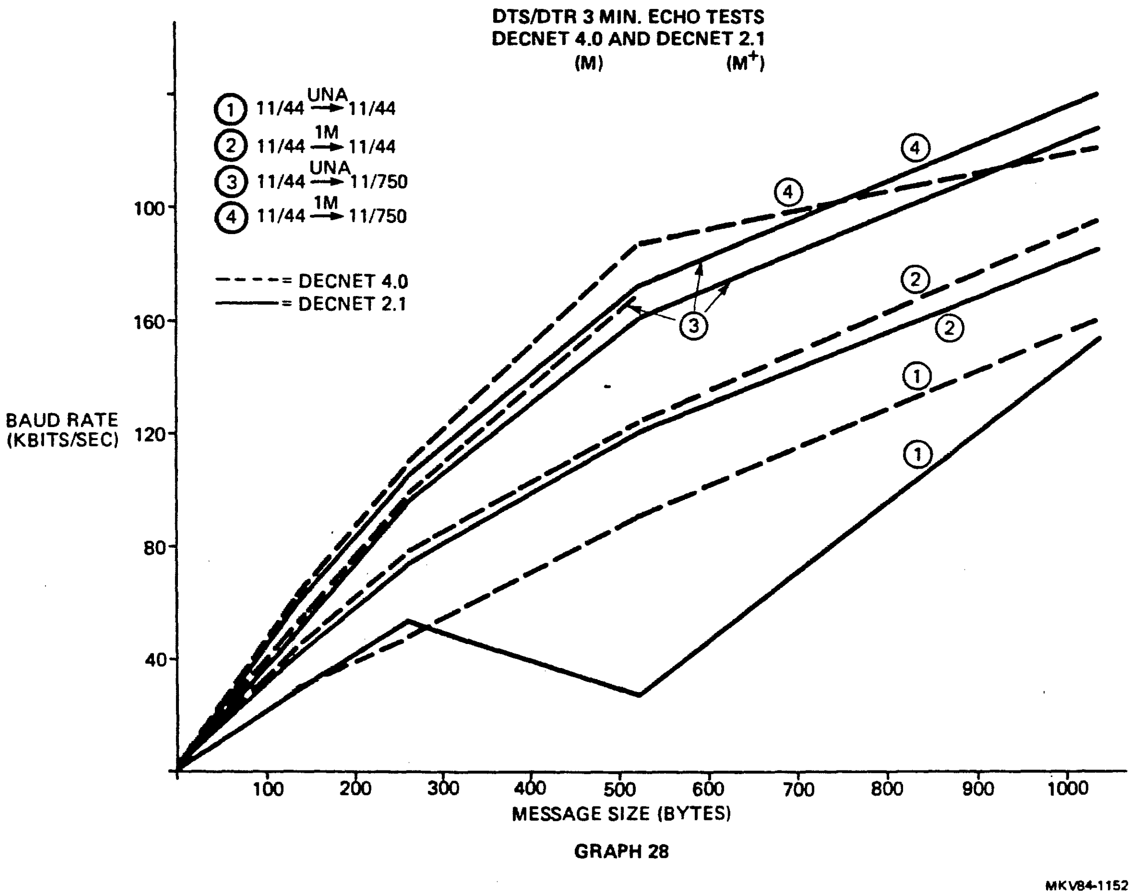
MKV84-1150

NFT FILE TRANSFERS (FCS FAL)
DECNET 4.0 VS. DECNET 3.1

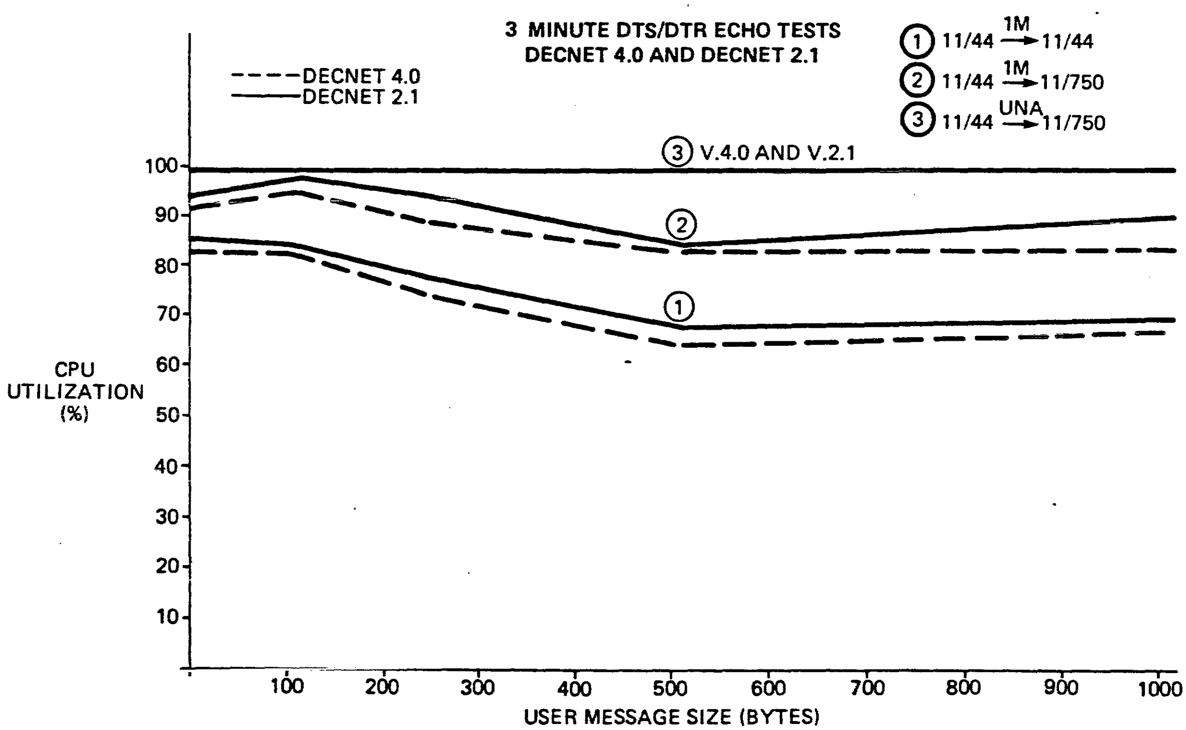


GRAPH 27

MKV84-1151


DTS/DTR 3 MIN. ECHO TESTS
DECNET 4.0 AND DECNET 2.1
(M) (M+)
1 11/44 UNA → 11/44
2 11/44 1M → 11/44
3 11/44 UNA → 11/750
4 11/44 1M → 11/750
---- = DECNET 4.0
—— = DECNET 2.1
BAUD RATE (KBITS/SEC)
40
80
120
160
100
100
200
300
400
500
600
700
800
900
1000
MESSAGE SIZE (BYTES)
MKV84-1152

GRAPH 28

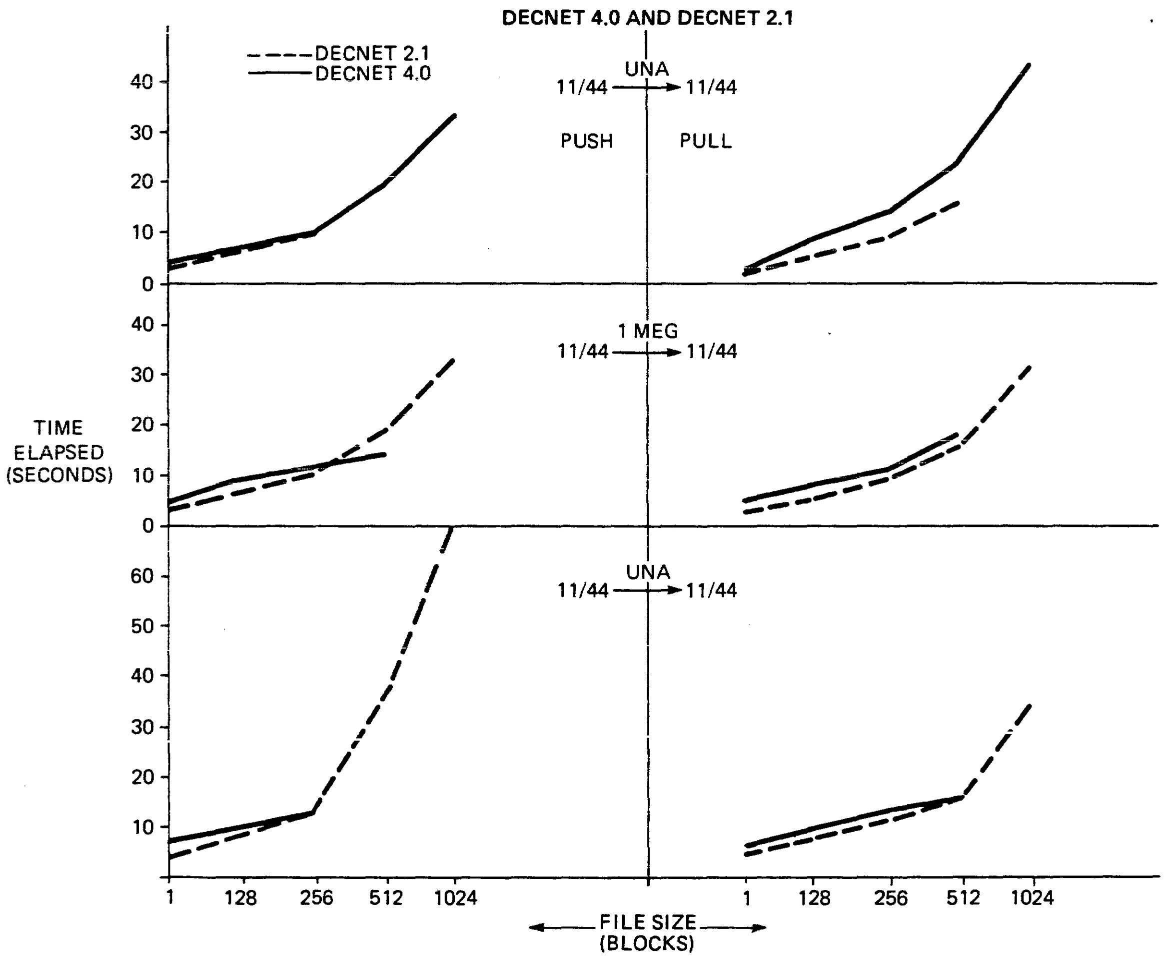


GRAPH 29

MKV84-1153

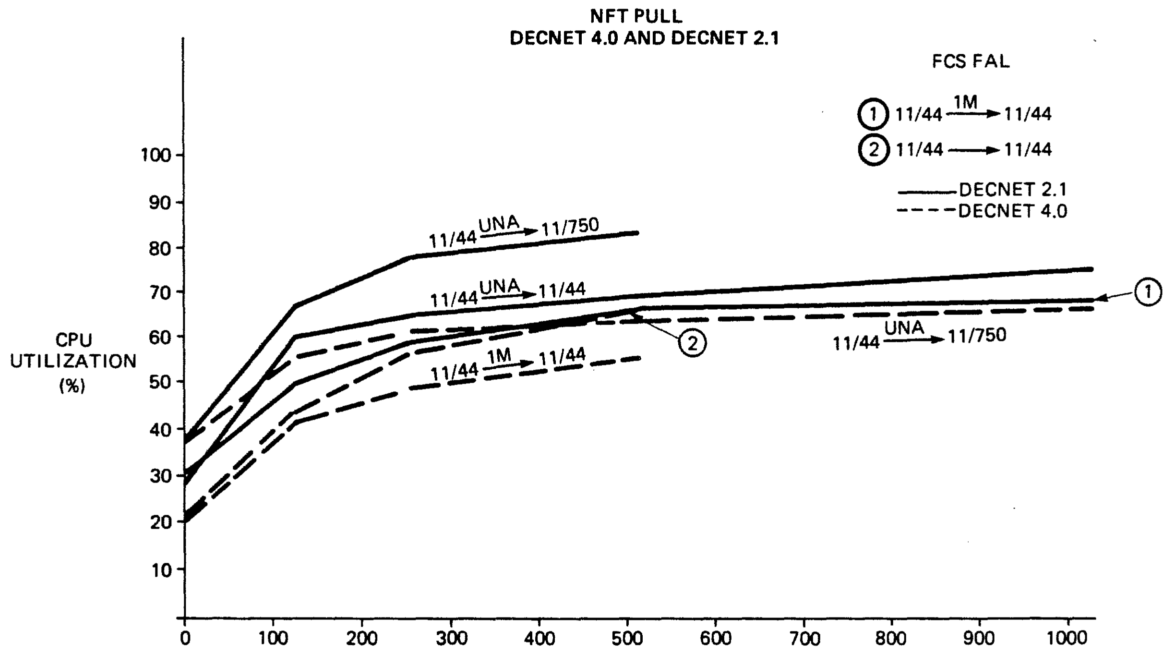
Table 6 DECnet Version 4.0 versus DECnet Version 2.1
(DECnet M versus DECnet M-Plus)
NFT File Transfers (FCS FAL)

		"PUSH" (Local -> Remote)				"PULL" (Remote -> Local)			
Configuration and DECnet Version	File Size (Blocks	Time (Seconds)	CPU/Block (msec/bl.)	CPU/Util. (%)	Line Util. (%)	Time (Seconds)	CPU Util. (%)	Line Util.	Line Util. (%)
UNA									
11/44-->11/44									
DECnet 4.0	1	6	1.25	21.98	.007	5	1.08	21.75	.008
	128	9	.05	43.57	.6	8	.03	46.01	.629
	256	12	.027	56.22	.85	12	.027	57.29	.87
DECnet 2.1	1	3	.81	26.7	.014	3	.95	28.4	.012
	128	7	.031	54.65	.715	7	.033	60.32	.75
	256	12	.029	62.83	.874	10	.026	67.14	1.05
	512	35	.028	67.36	.983	19	.026	72.12	1.12
	1024	70	.027	69.64	1.04	32	.025	77.46	1.25
1 Meg DMC									
11/44-->11/44									
DECnet 4.0	1	5	.696	13.93	.082	5	.95	20.36	.088
	128	9	.026	38.6	6.05	8	.028	42.25	6.29
	256	12	.023	48.49	8.74	12	.024	51.1	8.74
	512	14	.015	53.87	14.63	19	.021	57.45	11.23
	1024								
DECnet 2.1	1	3	.65	21.75	.137	3	.926	30.88	.137
	128	7	.03	52.52	7.15	6	.026	52.43	8.27
	256	11	.026	61.23	9.53	10	.023	61.55	10.85
	512	19	.024	65.82	11.04	17	.023	67.95	12.1
	1024	35	.024	69.98	12.1	33	.023	70.12	12.71
DMA									
11/44-->11/750									
DECnet 4.0	1	4	.846	19.52	.01	3	1.3	38.87	.012
	128	7	.431	60.03	.766	9	.039	57.95	.605
	256	10	.026	70.04	1.02	14	.034	62.77	.749
	512	19	.026	70.92	1.1	25	.032	65.37	.85
	1024	35	.026	77.17	1.21	47	.032	66.74	.89
DECnet 2.1	1	3	.75	22.54	.012	2	.73	31.25	.018
	128	6	.028	6.72	.93	6	.033	69.35	.87
	256	9	.025	68.82	1.12	9	.029	79.88	1.12
	512					17	.028	86.15	1.26
	1024								



GRAPH 30

MKV84-1154



GRAPH 31

MKV84-1155

APPENDIX B
Software Product Description

Software Product Description

PRODUCT NAME: DECnet-11M, Version 4.0 **SPD 10.75.11**
RSX-11M Network Software

DESCRIPTION:

DECnet-11M allows a suitably configured RSX-11M system to participate as a routing or non-routing (end) node in DECnet computer networks. DECnet-11M is a Phase IV network product and is warranted for use only with supported Phase III and Phase IV products supplied by DIGITAL.

DECnet Phase IV networks can contain up to 1023 nodes given proper network planning. Phase III nodes participating in Phase III/IV networks are limited to the Phase III routing capability of 255 nodes. Phase II nodes are not supported.

DECnet-11M offers task-to-task communications, utilities for network file operations, homogeneous network command terminal support, and network resource-sharing capabilities using the DIGITAL Network Architecture (DNA) protocols. DECnet-11M communicates with adjacent nodes over synchronous and asynchronous communication lines, Ethernet Local Area Networks (LANs) and parallel interfaces. Communications using X.25 circuits over selected Packet Switched Data Networks (PSDN) is also possible. This requires DECnet-11M to be configured with RSX-11 PSI/M product. Refer to the RSX-11 PSI/M SPD (10.42.xx) for further information. Access to DECnet-11M is supported for RSX-11M user programs written in MACRO-11, FORTRAN IV, FORTRAN-77, BASIC-PLUS-2, and COBOL.

The functions available to an RSX-11M user depend, in part, on the configuration of the rest of the network. Each DECnet product offers its own level of functionality and its own set of features to the user. Networks consisting entirely of DECnet-11M nodes can have the full functionality described in this SPD. Networks that mix DECnet-11M nodes with other DECnet products can limit the functions available to the DECnet-11M user because some DECnet-11M features are not supported by all DECnet products. Some supplied optional features require hardware configurations larger than the minimum supported systems.

The DECnet products and functions available to users on mixed networks can be determined by comparison of the SPDs for the component products.

Adaptive Routing

Adaptive routing is the mechanism by which one or more nodes in a network can route or forward messages between another pair of nodes in the same network. This routing capability will forward such messages even if no direct physical link exists between the pair of nodes apart from the sequence of physical links that includes the routing nodes.

A DECnet-11M node must function as a routing node whenever multiple lines are used simultaneously by that node. DECnet-11M end nodes provide all the capabilities of DECnet-11M routing nodes except that end nodes cannot route messages on behalf of other nodes in the network. Since end nodes do not route messages, they do not need to store or update routing databases. Consequently, end nodes use less system resource and generate less network traffic than routing nodes.

For this same reason, end node operation consumes less processing power than routing node operation. The Full Function DECnet-11M software must be installed on a node in order for that node to operate as a routing node. For a node to operate as an end node either the Full Function or the End Node DECnet-11M software must be installed on that node. Full Function DECnet-11M software allows a node to be set up as either a routing node or as an end node.

Although two adjacent routing nodes can be connected by more than a single physical link, messages will be sent over only one of the links. All other lines will serve as "hot standbys," such that the least cost path available between two nodes is the one that will be used for message traffic. A line cost parameter set by the system manager determines the line over which all messages will be sent from node to adjacent node.

Task-To-Task Communication

Using DECnet-11M, an RSX-11M user program written in MACRO-11 or one of the supported high level languages can exchange messages with other network user programs. These two programs can be on the same node or on any other Phase III or Phase IV node in the network. The messages sent and received by the two user programs can be in any data format.

The DECnet-11M software will optionally verify the access control privileges of a task requesting communication with a DECnet-11M task. The RSX-11M System Account File is used to determine access privileges. The results can either be passed on to the receiving task or used to reject the request by the network software.

Network Resource Access

File Transfer Utilities

Using DECnet-11M utilities, a user can transfer sequential ASCII files between DECnet nodes. Files can be transferred in both directions between the locally supported RSX-11M File Control System (FCS) devices and the file system of other DECnet nodes. Wild cards can be used for the user identification code, file name, file type and version number for local to remote file transfers. Transfer of other file types is supported provided the source and destination DECnet systems support the same file type. Directory listings are also a supported DECnet-11M feature.

The DECnet-11M file transfer utilities support file transfers for both FCS and RMS files, where formats are compatible between the DECnet nodes.

Additional facilities allow system command files to be submitted to a remote node where the list of commands is in a format acceptable to the node responsible for the execution. DECnet-11M also allows RSX-11M command files to be received from other systems and executed.

A utility is also provided with DECnet-11M which allows the user to queue file operation requests for execution at a specified time. The user has the ability to monitor, list, and delete entries from this queue.

Network users must specify the appropriate user identification and password in order to access the files on a DECnet-11M node. Access to local files using the DECnet software can be controlled through the RSX-11M System Account File.

File Access

File access is supported to and from remote DECnet systems by explicit subroutine calls in the supported high level languages. READ, WRITE, OPEN and CLOSE, and DELETE operations can be initiated by local tasks for sequential and random access files residing on the local system or at remote DECnet systems. Other nodes supporting File Access can exercise this capability for files located on the RSX-11M node. Fixed and variable length record formats are supported. Files accessed remotely can contain either ASCII or binary information.

Access to RMS file organizations from other DECnet products is supported by DECnet-11M.

Network Command Terminal

DECnet-11M provides a utility that allows a terminal user to establish a virtual connection to another Phase IV DECnet-11M or DECnet-11M-PLUS system. This connection makes the terminal appear as if it were physically connected to the other system and the operator can use standard system and network utilities supported by that system. This utility is particularly useful for doing remote program development and allows terminal users on small application oriented systems to utilize the resources of larger development oriented systems.

DECnet-11M also provides an unsupported utility that allows a terminal user to establish a virtual connection to Phase III/IV DECnet-VAX systems. This utility is provided only as a courtesy from DIGITAL, with no implied support services offered, as is the case with other mentioned DECnet-11M-PLUS capabilities.

Down-Line System Loading

Initial memory images for DECnet-11S nodes in the network can be stored on RSX-11M file system devices and loaded into nodes across point-to-point, multipoint (DMP/DMV only) and Ethernet links. Load requests can come from the local RSX-11M operator or from the remote node. Generation of initial memory images of DECnet-11S systems for down-line loading is supported by RSX-11M.

Upline Dumping

Memory images of adjacent RSX-11S nodes connected by DECnet can be written onto a file on a DECnet-11M system. This facility helps a programmer understand what may have caused the RSX-11S system to crash.

Down-Line Task Loading

Programs to be executed on DECnet-11S nodes in the network can be stored on the DECnet-11M system and loaded on request into DECnet-11S nodes. In addition, programs already executing on DECnet-11S nodes can be checkpointed to the host file system and later restored to main memory of the DECnet-11S node. These features simplify the operation of network systems that do not have mass storage devices.

Network Management

The Network Control Program (NCP) performs three primary functions: displaying statistical and error information, controlling network components, and testing network operation. These functions can be performed locally or executed at remote Phase IV nodes that support this feature. In either case, the output resulting from a command can be directed to a local file or to the user's terminal.

An operator can display the status of DECnet activity at the local node and other Phase IV nodes. The user can choose to display statistics related to both node and communication lines, including data on traffic and errors. The local console operator can also perform many network control functions such as loading and unloading DECnet components, starting and stopping lines, activating the local node, and down-line loading DECnet-11S systems.

DECnet-11M also provides local network event logging to the console device, a remote node, or a user written program. The NCP utility can be used to enable and disable the logging of specific events as well as to enable and disable the event logging facility.

Communications

DECnet-11M, supports the DIGITAL Data Communications Messages Protocol (DDCMP) for full- or half-duplex transmission in point-to-point and multipoint operation using serial synchronous or asynchronous facilities. DDCMP provides error detection/correction and physical link management facilities. In addition, an auto-answer capability is provided.

Multipoint and auto-answer function with EIA-type devices only. Parallel communication devices use special link protocols (not DDCMP) optimized for their characteristics.

The DEUNA, when used in conjunction with the H4000 or DELNI transceiver, allows DECnet-11M to utilize Ethernet as a data link transmission medium.

RSX-11 PSI/M is the software product that provides an interface to X.25 Packet Switched Data Networks. When DECnet-11M is used in conjunction with RSX-11 PSI/M, DECnet-11M can utilize the PSDN as though it was a standard datalink to transmit messages between DECnet nodes.

The maximum number of physical links that can be supported by a DECnet-11M node is sixteen depending on CPU memory, type of communications interface, and speed of interfaces.

A maximum of 32 X.25 virtual circuits is supported when DECnet-11M is used in conjunction with RSX-11 PSI/M for DECnet communication through an X.25 PSDN. However, when the total number of DECnet circuits (DLM and other) is greater than 24, the maximum node address supported is less than 1023 (#circuits * maximum address <25,000).

DECnet-11M multipoint will support up to a maximum of twelve tributaries on a single multipoint line. Aggregate bandwith of tributaries is limited to that of the control station device. The communication path to each tributary counts as a circuit with respect to the limits on number of circuits specified above. Multipoint line configurations are supported for the following devices:

Multipoint Devices

Devices	Multipoint Control Station (Master)	Multipoint Tributary (Slave)
DL11/DLV11	YES	YES
DUP11	YES	YES
DU11/DUV11	YES	YES
DPV11	YES	YES
DZ11/DZV11	YES	NO
KMC11 (DZ11)	YES	NO
KMC11 (DUP11)	YES	YES
DV11	YES	YES
DMP11*	YES	YES
DMV11*	YES	YES

*Multipoint communication hardware device

Direct Line Access

User written MACRO-11 tasks will be provided with Direct Line Access (DLX) support to all supported devices (including Ethernet Controller). DLX will allow direct control of the communications lines, bypassing the logical link control and transport mechanism provided by the DECnet software. User programs will be required on both ends of the link in order to use this interface.

Modification of the generation procedures is required to generate DLX-only nodes. This interface is useful in applications where the user desires to have minimum protocol overhead on a physical link and does not require the logical link management and other network services provided by the DECnet software for a particular line.

DECnet-11M Configuration

The process of configuring a DECnet-11M node is based primarily on trade-offs of cost, performance, and functionality, within the realm of satisfying the user's application requirements. It can be readily expected that network applications will run the full gamut from low-speed, low-cost situations to those of relatively high performance and functionality. The performance of a given DECnet node is a function not only of the expected network traffic and resultant processing ("global" conditions), but also of the amount of concurrent processing specific to that node (local conditions). Thus, node performance depends on many factors, including:

- CPU type and Memory Size
- Number of device interrupts per unit time
- Communication line characteristics
- Number and size of buffers
- Message size and frequency of transmission
- Local applications
- Size and frequency of route-through traffic

Note that the rate at which user data can be transmitted (throughput) over a communications line may sometimes approach, but will never equal or exceed, the actual line speed. The reason is that the actual throughput is a function of many factors, such as the network application(s), topology, protocol overhead and line quality, as well as the factors cited above.

NOTE: Careful analysis is required when configuring routing nodes with 124K words or less.

Five basic groups of communications interfaces are presented in the tables below. They differ in many respects, particularly in their effect upon CPU utilization.

- With character interrupt devices such as the DUP11, CPU cycles are required for not only the line protocal processing such as DDCMP or the optional X.25 protocol, but also for each character sent and received.
- Devices such as the DV11 are direct memory access (DMA) devices. Since DDCMP is in the PDP-11 software, CPU cycles are required for DDCMP line protocol processing.
- Devices such as the DMC11, DMR11, DMP11, DMU11, and KMS11 are direct memory access (DMA) devices. The line protocol is executed in microcode, thus off-loading the PDP-11 CPU. The only DECnet load the processor sees is completed incoming and outgoing messages.
- The PCL11-B is a high speed DMA device which uses local parallel communications lines. It has its own line protocol and does not use DDCMP. CPU cycles are only required for processing of incoming and outgoing data messages and to perform control functions.
- The DEUNA, unibus to Ethernet controller, is a high speed DMA controller supporting CSMA/CD protocol. CPU cycles are only required for processing of incoming and outgoing messages.

Device Groups

Device Group	Maximum Line Speed (Kilobits/sec)
Character Interrupt	
DL11/DLV11	9.6/4.8
DUP11/DPV11	9.6(1)/4.8
DU11/DUV11	9.6(1)/4.8
DZ11/DZV11	9.6
DMA	
DV11	9.6
KMC11 (DZ11)	9.6
KMC11 (DUP11)	19.2
KMS11-BD/E*	19.2
DMA/DDCMP	
DMC11-AR-DA	19.2
DMC11-AR-FA	56.0
DMC11-AL-MD	56.0
DMC11-AL-MA	1000.0
DMR11-AA	19.2
DMR11-AB,AC,AE	1000.0
DMP11-AA, DMV11-AA	19.2(2)
DMP11-AB, DMV11-AB,AC	56.0
DMP11-AE,AC	1000.0(3)
Parallel	
PCL11-B	4000.0
Ethernet	
DEUNA	10000.0

*Optional, requires the RSX PSI/M product.

1. 4.8K bps for 11 24 processor

2. Up to 56.0K bps for RS423-A interfaces.

3. 500K bps for full-duplex.

These tables describe the physical hardware configurations supported by DECnet-11M in terms of CPU class and communication interface device group. It should be noted that the attachment of such devices as A/D converters and timesharing terminals can reduce the maximum number of communication lines which can effectively be supported. When used with RSX-11 PSI/M, the number of devices supported on any CPU will be dictated by the limits supported by the PSI product (see SPD 10.42.xx).

It is strongly recommended that DECnet-11M be configured as an end node if it is used on a Micro/PDP-11.

NOTE: In the following table, the rated bandwidth is stated for a single device type. The maximum bandwidth for an intermix of device types cannot be calculated from these tables.

Maximum Line Configurations on 11/23 11/23-PLUS, MICRO/PDP-11 CPUs**

Device Group	Max. No. of Lines	Maximum Device Bandwidth (Kilobits/sec)	Mode
Character	8	14.4	FDX
Interrupt	8	28.8	HDX
DMV11*	2	112.0	FDX,HDX

*Not supported on 11/23

**Requires RL02 configuration for system and network generation.

Maximum Line Configurations on 11/24,11/34,11/40,11/45,11/55,11/60 CPUs

Device Group	Max. No. of Lines	Maximum Device Bandwidth (Kilobits/sec)	Mode
Character	8	14.4	FDX
Interrupt	8	28.8	HDX
DMA	8	30.6	FDX
	8	61.2	HDX
DMA/DDCMP			
DMC11-AR-DA	16	307.2	FDX,HDX
DMC11-AR-FA	6	336.0	FDX,HDX
DMC11-AL-MD	6	N/A	FDX,HDX
DMC11-AL-MA	2	N/A	FDX,HDX
DMR11-AA	16	307.2	FDX,HDX
DMR11-AB-AC-AE	6*	336.0	FDX,HDX
PCL11-B	1	N/A	Parallel
DEUNA	2**	10000	Ethernet

*Two at 1M bps. One at 1M bps for 11/24

**Supported on 11 24, 11 34A only. Must be on Independent Ethernet Networks.

Maximum Line Configurations On 11/44 and 11/70 CPUs

Device Group	Max. No. of Lines	Maximum Device Bandwidth (Kilobits/sec)	Mode
Character	8	19.2	FDX
Interrupt	8	38.4	FDX
DMA	16	40.8	FDX
	16	81.6	HDX
DMA/DDCMP			
DMC11-AR-DA	16	307.2	FDX,HDX
DMC11-AR-FA	6	336.0	FDX,HDX
DMC11-AL-MD	6	N/A	FDX,HDX
DMC11-AL-MA	1	N/A	FDX,HDX
DMR11-AA	16	307.2	FDX,HDX
DMR11-AB-AC-AE	6*	336.0	FDX,HDX
PCL11-B	1	N/A	Parallel
DEUNA	2**	10000	Ethernet

*One at 1M bps for 11 44, 11 70

**Only one DEUNA on 11 70 CPU, must be independent Ethernet networks.

Maximum Line Configurations Guidelines (Multipoint)

Maximum Line Speed (Kilobits per Second, half- or full-duplex)

Device Group	19.2	56	250	500	1000
DMV11-AA	2/8	2/8			
-AB	2/8	2/8			
-AC	2/8	2/8			
DMP11-AA	+4/8	2/8			
-AB	+4/8	2/8			
-AC	2/8	1/12	1/12	1/12*	
-AE	+4/8	2/8	1/12	1/12	1/12*

*half-duplex
+ 11/24 is limited to 2 controllers

NOTE: Left side of slash (/) indicates number of controllers per node and right side indicates total number of tributaries per control node.

Total number of circuits not to exceed 16 per node.

Number of tributaries on lines should be carefully configured for performance considerations.

In order to achieve a viable configuration, the user and/or a DIGITAL software specialist must perform a level of application analysis which addresses the factors above. In the preceding tables, the columns have the following meanings:

Maximum Number of Lines – The largest number of physical lines that can be attached and driven by the DECnet-11M system.

Maximum Device Bandwidth – The maximum total number of bits per second which can be handled by a CPU when all communication devices of a single given type, such as character interrupt, are added together. For example, DECnet-11M on a PDP-11/44 can accommodate two half-duplex character-interrupt devices at 19.2Kb, or eight at 4.8 Kb, or eight at 2.4 Kb (constrained by the maximum number of lines rather than bandwidth). Maximum device bandwidth should be calculated for all lines known to operate concurrently. Maximum bandwidth on a multipoint line is limited to that supported by the Control Station device.

Maximum Line Speed – The fastest clock rate at which the device can be driven under DECnet-11M. If specific devices have the ability to operate at a maximum rate, they must be configured subject to the "maximum device bandwidth" restriction above.

Mode – This indicates whether the line is operating in either half-duplex (a single bit stream) or full-duplex (two concurrent bit streams) mode. In some instances in the tables, a half-duplex line is quoted as having maximum bandwidth approximately double that of the comparable full-duplex line. This reflects the single bit stream character of half-duplex lines, and the fact that two of them place a load on the CPU roughly equivalent to one full-duplex line with traffic in both directions.

MINIMUM HARDWARE REQUIRED:

Any valid RSX-11M system configuration with:

- The following additional memory available.

  DECnet-11M end node – 16KW
  DECnet-11M routing node – 18KW
  DECnet-11M end node with RSX-11 PSI/M – 22KW
  DECnet-11M routing node with RSX-11 PSI/M – 30KW
  Ethernet support will add 3KW to the above memory requirements

- The following additional number of disk blocks available for DECnet-11M network software

  DECnet-11M end node – 4000
  DECnet-11M routing node – 4500
  DECnet-11M end node with RSX-11 PSI/M – 4500
  DECnet-11M routing node with RSX-11 PSI/M – 5000

- PDP 11/24, PDP11/44, or PDP11/70 central processor with one of the following communications devices:

  DUP11-DA low-speed synchronous interface (5)
  DU11-DA low-speed synchronous interface (4)
  DMC11-AR-DA remote synchronous V.24/EIA RS-232-C interface (4)
  DMC11-AR-FA remote synchronous V.35/DDS interface (4)
  DMC11-AL-MD high-speed local synchronous interface (4)
  DMC11-AL-MA high-speed local synchronous interface (4)
  DMP11-AA synchronous UNIBUS interface RS232-C/RS423-A (5)
  DMP11-AB synchronous UNIBUS interface CCITT V.35/DDS (5)
  DMP11-AC local synchronous UNIBUS interface (5)
  DMP11-AE synchronous UNIBUS interface RS422 (5)
  DMR11-AA synchronous UNIBUS interface RS232-C/CCITT V.24 (5)
  DMR11-AB synchronous UNIBUS interface CCITT V.35/DDS (5)
  DMR11-AC local synchronous UNIBUS interface (5)
  DMR11-AE synchronous UNIBUS interface RS449/422 (5)
  DL11-E asynchronous EIA interface with modem control (5)
  DL11-C/WA asynchronous 20mA current loop interface (1,5)
  DZ11-A/B multiline asynchronous EIA interface (2,5)
  DZ11-C/D multiline asynchronous 20mA current loop interface. (1,2,5)
  DV11-AA/BA multiline NPR synchronous interface (2,5)
  PCL11-B multiple CPU link
  DEUNA Unibus to Ethernet controller
  KMS11-BD/E, synchronous UNIBUS multiplexer interface (3)

- PDP 11/23, 11/23 Plus or Micro/PDP-11 central processor with one of the following communications devices. (Micro/PDP-11 Configuration must include RL02 for system and network generation.)

  DMV11-AA synchronous QBus interface RS232-C/RS423A (5)
  DMV11-AB synchronous QBus interface V.35/DDS for 11/23 PLUS (5)

DMV11-AC local synchronous QBus interface for 11/23 PLUS (5)
DUV11-DA low-speed EIA synchronous interface (5)
DLV11-E asynchronous EIA interface with modem control (5)
DLV11-F asynchronous 20 mA current loop interface (1,4)
DZV11-B multiline asynchronous EIA interface (2,5)
DPV11-DB synchronous QBUS interface (5)

NOTES:

(1) Requires either the H319 option for optical isolation or one side of the 20mA line to be in passive mode.
(2) All lines on this interface must be dedicated as DECnet links.
(3) Requires RSX-11 PSI/M product.
(4) These products are no longer marketed by DIGITAL and may not be supported in future releases of DECnet-11M.
(5) Or FCC eqivalent hardware option.

OPTIONAL HARDWARE:

- Additional lines and/or communication interfaces (as listed above) up to maximum as defined in Maximum Line Configurations tables for mapped systems.
- KG11-A Communications Arithmetic Element (may be used in conjunction with DV11, DZ11, and DL11)
- KMC11-A (without the RSX-11 PSI/M product, can be used in conjunction with up to eight DUP11s or with up to a sixteen line DZ11)

PREREQUISITE SOFTWARE:

RSX-11M Operating System

Refer to the RSX-11M Optional Software Cross Reference Table (SPD 20.98.xx) for the required version.

OPTIONAL SOFTWARE:

COBOL 81
FORTRAN-77/RSX
PDP-11 BASIC-PLUS-2 for RSX-11M
RSX-11 PSI/M

SOFTWARE WARRANTY

Warranty for this software product is provided by DIGITAL with the purchase of a license for the product. There is no additional charge. This software product is warranted to conform to the Software Product Description (SPD). This means that DIGITAL will remedy any nonconformance when it is reported to DIGITAL by the customer during the warranty period.

The warranty period is ninety (90) days. It begins when the software is installed or thirty (30) days after delivery to the end user, whichever occurs first and expires ninety (90) days later or one hundred eighty (180) days after release of a subsequent version of the software.

Warranty Service

DIGITAL provides software warranty service worldwide. Service is provided in the country of purchase and is not transferable between countries.

DIGITAL will provide a service location which will accept reporting (in a format prescribed by DIGITAL) of a nonconformance problem caused when using the licensed software under normal conditions as defined by the SPD. DIGITAL will respond to a nonconformance problem in the current unaltered release of the licensed software by issuing correction information such as: correction documentation, corrected code, or notice of availability of corrected code; or a restriction or a bypass. The customer will be responsible for the preparation and submission of the problem report to the service location.

Warranty Exclusion

DIGITAL DOES NOT WARRANT THAT THE SOFTWARE LICENSED TO CUSTOMER SHALL BE ERROR FREE, THAT THE SOFTWARE SHALL OPERATE WITH ANY HARDWARE AND SOFTWARE OTHER THAN AS SPECIFIED IN THIS SPD, THAT THE SOFTWARE SHALL SATISFY CUSTOMER'S OWN SPECIFIC REQUIREMENTS, OR THAT COPIES OF THE SOFTWARE OTHER THAN THOSE PROVIDED OR AUTHORIZED BY DIGITAL SHALL CONFORM TO THE SPD.

DIGITAL MAKES NO WARRANTIES WITH RESPECT TO THE FITNESS AND OPERABILITY OF MODIFICATIONS NOT MADE BY DIGITAL.

IF THE SOFTWARE FAILS TO FUNCTION FOR REASONS STATED ABOVE, THE CUSTOMER'S WARRANTY WILL BE INVALIDATED AND ALL SERVICE CALLS WILL BE BILLABLE AT THE PREVAILING PER CALL RATES.

INSTALLATION

Only experienced customers should attempt installation of this product. DIGITAL recommends that all other customers purchase DIGITAL's Installation Services. These services provide for installation of the software product by an experienced DIGITAL Software Specialist.

DIGITAL's Installation Services can be purchased as a separate service.

Installation Services

For a fixed price a DIGITAL Software Specialist will assure that
the customer's system is ready for installation, install the software, and familiarize the customer with its operation.

Installation for DECnet-11M will consist of the following:

- Verification that all components of DECnet-11M have been received.
- Verification that the necessary versions of the RSX-11M software and documentation are available.
- Verification of the appropriate sysgen parameters.

NOTE: Should a software specialist be required to modify the previously installed operation system parameters, a time and materials charge will apply.

- Install DECnet-11M software
- Define and create a local node DECnet database.
- Modify the system's start-up command procedure including starting up the DECnet-11M network.
- Verify the proper installation of DECnet-11M by running a series of tests to show connectivity (demonstrated by the use of the post installation checkout procedure) to a designated node.

Connectivity to all other nodes within the network is the responsibility of the customer.

Pre-Installation Procedures Required

Before DIGITAL can install the software, the customer must:

- Ensure that system meets the minimum hardware and software requirements (as specified in the SPD.)
- Obtain, install and demonstrate as operational any modems and other equipment and facilities necessary to interface DIGITAL's communication equipment.
- For multi-node networks designate one adjacent node to verify installation/connectivity.
- Make available for a reasonable period of time, as mutually agreed upon by DIGITAL and the customer, all hardware communication facilities and terminals that are to be used during installation.

Delays caused by any failure to meet these responsibilities will be charged at the prevailing rate for time and materials.

COURTESY INSTALLATION SERVICE: This software product will be installed by DIGITAL at no additional charge if you purchase Installation Service for the host Operating System and you install both software products concurrently.

ORDERING INFORMATION

DIGITAL provides a wide range of material and service options supporting this software product. Each option is described below. **IF YOU ARE ALREADY FAMILIAR WITH THESE OPTIONS YOU MAY OBTAIN THE ORDERING INFORMATION DIRECTLY FROM THE SOFTWARE OPTIONS CHART.** In most cases you will want to review the following descriptions to determine what options you require.

You will need a separate license for each CPU on which you will be using the software product (except as otherwise specified by DIGITAL). Then you will select the materials and service options you need to utilize the product effectively.

You order the license, materials and services using order numbers of the form: Qxxxx-X?. "Qxxxx" refers to the specific software product, "X" is the license code or material service option and "?" is the selectable media code for machine readable materials.

Single-Use licensed software is furnished under the licensing provisions of DIGITAL's Standard Terms and Conditions of Sale, which provide in part that the software and any part thereof may be used on only the single CPU on which the software is first installed, and may be copied, in whole or in part (with the proper inclusion of DIGITAL's copyright notice and any proprietary notices on the software) for use on such CPU.

LICENSE OPTIONS

Single-Use License Option

The Single-Use License is your right to use the software product on a single CPU and it includes your 90 day warranty.

For your first installation of this software product you must purchase as a **minimum**:

- Single-Use License Option, and
- Distribution and Documentation Option

The license gives you the right to use the software on a single CPU and the Distribution and Documentation option provides the machine readable software and related documentation.

To use this software product on additional CPU's, you must purchase as a **minimum**:

- Single-Use License Option

In addition to the right to use, the license gives you the right to copy the software from your original CPU installation to the additional CPU. Therefore, the Distribution and Documentation option is not required, but optional.

MATERIALS AND SERVICE OPTIONS

Distribution and Documentation Option

The Distribution and Documentation option provides the software object code in binary form and the basic documentation. You must have, or order, a Single-Use License to obtain this option. You will need this option to install the software for the first time. You will also need this option to obtain revised versions of the software product when they become available.

If you prefer to receive automatic distribution of revised versions for this product, you must purchase a Software Product Service Agreement.

Software Revision Right-To-Copy Option

The Right-To-Copy option allows a customer with multiple CPU's to copy a revised version of a software product from one CPU to another. Each CPU must be licensed for that product. You first install the revised software on one CPU; then you can make copies for additional CPU's by purchasing the Right-To-Copy option for each additional CPU.

If you prefer to automatically obtain the right-to-copy, you must purchase a Service Right-to-Copy for each additional CPU; this is a service added to a Software Product Service Agreement.

Documentation-Only Option

You can obtain one copy of the basic documentation by purchasing the Documentation-Only option.

Installation Service Option

Installation Service includes those services provided by a DIGITAL Software Specialist to successfully install a software product. Digital's Installation Service accelerates your productive use of this product. (If Operating System parameters must be changed, system regeneration may not be included in the Installation Service.)

Installation Service will be provided at no additional charge (Courtesy Installation) under the conditions described in the Installation section above.

NETstart Services Options

NETstart Services help you "start-up" operation of your DECnet networking environment quickly by accelerating the learning process of your staff. It includes direct assistance, documentation review, discussion and hands-on experience provided on-site by a DIGITAL Networking Specialist. NETstart is designed to benefit first time users, as well as experienced network computer users.

There are two levels of NETstart Services. NETstart I is the most comprehensive start-up service, followed by NETstart II.

You should be experienced with your appropriate operating systems prior to scheduling these services. Contact your DIGITAL representative for more details on NETstart Services.

Software Product Service Agreements

DIGITAL offers you a choice of three Software Product Service Agreements to support your software. These agreements provide a set of the following services depending on the agreement selected: remedial and maintenance service, revised versions of the software and documentation, telephone support and newsletters or dispatches containing suggested fixes for known problems.

The level of service you need depends primarily on three factors: (1) your available manpower, (2) your staff's level of expertise, and (3) the importance to you of having your system current and operating at peak efficiency. Our DIGITAL representative can help you select the service best suited to your needs. The agreement types are listed below.

DECsupport Service is designed for people requiring high availability of their system and for customers who need regular access to DIGITAL's technical expertise. It is the most comprehensive support service, offering on-site preventative maintenance and remedial support for critical problems. In addition, it also provides revised versions of the software and documentation, fast response, telephone support and newsletters or dispatches.

BASIC Service is ideal for customers who have their own technical staff but need fast answers to operational questions. The telephone support provided in this service gives you timely answers and solves most software problems. In addition, you get revised versions of the software and documentation, and newsletters or dispatches.

Self-Maintenance Service is designed for customers who require only minimal support but wish to receive revised versions of the software and documentation information from DIGITAL. In addition, the service provides newsletters or dispatches.

All agreements are available to licensed DIGITAL customers on an annual, contractual basis.

The Prerequisite Software, as specified by this SPD, must have the equivalent level Software Product Service.

A variety of service options may be added to an existing Software Product Service Agreement as follows:

- Customers who want to copy the revised software received under their Software Product Service Agreement onto additional CPUs running that same product may order a Service Right-to-Copy for each CPU, Qxxxx-3Z.
- Customers who have a Software Product Service Agreement can obtain an additional copy of the documentation supplied under the Agreement by ordering the Documentation Update Service, Qxxxx-KZ.
- Customers who have a Basic Service or DECsupport Service Agreement can add additional names to the three who, under the Agreement, may call the Telephone Support Center by ordering the Additional Telephone Support Center Contact Service, Qxxxx-6Z.

TRAINING FROM EDUCATIONAL SERVICES

To ensure customer success with DIGITAL products, Educational Services sells training for the installation, maintenance and/or management of DIGITAL software. Course formats vary from seminars to packaged training materials that include self-paced instruction and computer-based instruction to traditional lecture/labs at DIGITAL's 27 worldwide Training Centers.

All course schedules, availability and purchasing information is listed in Educational Services' *DIGEST*, a quarterly publication designed to assist customers in planning their training programs.

Professional Software Services

DIGITAL Software Specialists are available on a per-call or resident contract basis to help in all phases of software development or implementation. Specialists are available to serve as technical consultants, decision support consultants or business systems analysts. Resources are available to:

- Supplement your programming staff
- Assume project management responsibility
- Develop software
- Assure successful start-up and performance with product specific packaged services

Contact your DIGITAL representative for additional information and ordering details.

SOFTWARE OPTIONS CHART

The distribution Media Codes used in the Software Options Chart are described below. You specify the desired Media Code at the end of the Order Number, e.g. QJ764-HD = binaries on 9-track 800 BPI Magtape (NRZI).

D = 9-track 800 BPI Magtape (NRZI)
H = RL02 Disk Cartridge
M = 9-track 1600 BPI Magtape (PE)
Q = RL01 Disk Cartridge
T = RK06 Disk Cartridge
V = RK07 Disk Cartridge
Z = No hardware dependency

NOTE: The availability of these software product options and services may vary by country. Customers should contact their local DIGITAL office for information on availability.

OPTIONS	ORDER NUMBER	ORDER NUMBER
LICENSE OPTIONS: A LICENSE IS REQUIRED FOR EACH CPU.	FULL FUNCTION	END NODE
Single-Use License	QJ764-UZ	QJ765-UZ
MATERIALS AND SERVICE OPTIONS:		
Distribution and Documentation Option	QJ764-HD QJ764-HH QJ764-HM QJ764-HQ QJ764-HT QJ764-HV	QJ765-HD QJ765-HH QJ765-HM QJ765-HQ QJ765-HT QJ765-HV
Software Revision Right-To-Copy Option	QJ764-HZ	QJ765-HZ
Documentation Only Option	QJ764-GZ	QJ765-GZ
Installation Service Option	QJ764-ID QJ764-IH QJ764-IM QJ764-IQ QJ764-IT QJ764-IV	QJ765-ID QJ765-IH QJ765-IM QJ765-IQ QJ765-IT QJ765-IV
SOFTWARE PRODUCT SERVICE AGREEMENTS:		
DECsupport Service	QJ764-9D QJ764-9H QJ764-9M QJ764-9Q QJ764-9T QJ764-9V	QJ765-9D QJ765-9H QJ765-9M QJ765-9Q QJ765-9T QJ765-9V
Basic Service	QJ764-8D QJ764-8H QJ764-8M QJ764-8Q QJ764-8T QJ764-8V	QJ765-8D QJ765-8H QJ765-8M QJ765-8Q QJ765-8T QJ765-8V
Self-Maintenance Service	QJ764-3D QJ764-3H QJ764-3M QJ764-3Q QJ764-3T QJ764-3V	QJ765-3D QJ765-3H QJ765-3M QJ765-3Q QJ765-3T QJ765-3V