

**DECnet Digital Network Architecture
Phase IV
Ethernet Node
Product Architecture Specification**
Order No. AA-X440A-TK

digital
software

**DECnet Digital Network Architecture
Phase IV
Ethernet Node
Product Architecture Specification
Order No. AA-X440A-TK**

December 1983

This document specifies the minimum required functions for a DEC node connected to an Ethernet. A node meeting these requirements will be maintainable and usable to build additional product specific functions.

SUPERSESSION/UPDATE INFORMATION: This is a new manual.

VERSION: 1.0.0

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

digital equipment corporation - maynard, massachusetts

The information in this document is subject to change without notice and should not be construed as a commitment by Digital Equipment Corporation. Digital Equipment Corporation assumes no responsibility for any errors that may appear in this document.

The software described in this document is furnished under a license and may only be used or copied in accordance with the terms of such license.

No responsibility is assumed for the use or reliability of software on equipment that is not supplied by DIGITAL or its affiliated companies.

Copyright © 1983 by Digital Equipment Corporation

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

The following are trademarks of Digital Equipment Corporation:

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

Distributed Systems Publications typeset this manual using DIGITAL's
TMS-11 Text Management System.

MGTPEALL

CONTENTS

1	Introduction	4
1.1	Functions	4
1.2	Requirements, Goals, And Non-goals	5
1.2.1	Requirements	5
1.2.2	Goals	5
1.2.3	Non-goals	5
2	Models	-
2.1	Relation to Digital Network Architecture	7
2.2	Ethernet Node Type Model	8
3	Required Functions	9
3.1	All Nodes	9
3.1.1	Data Link Layer Requirements	9
3.1.2	Network Management Layer Requirements	10
3.1.3	User Layer Requirements	11
3.2	Remote Control Nodes	11
3.2.1	Network Management Layer Requirements	11
3.3	Remote Dump/Load Nodes	12
3.3.1	Network Management Layer Requirements	12
3.4	Remote Diagnosis Nodes	12
3.4.1	Network Management Layer Requirements	12
3.4.2	User Layer Requirements	12
4	Network Auto-configuration	13
4.1	Goals	13
4.2	Algorithm	13

1 Introduction

This document specifies the minimum required functions for a DEC node connected to an Ethernet. A node meeting this specification will be maintainable and usable to build additional product specific functions.

Within the context of this specification, a node is defined as a collection of hardware and software that appears to other nodes as a single functional unit. This node is attached to a common coaxial cable and uses this cable as a network communication medium according to the Digital, Intel, Xerox Ethernet Specification.

This specification assumes reader familiarity with the following documents:

The Ethernet, A Local Area Network, Data Link Layer and Physical Layer Specifications, Version 2.0, (Digital, Intel, and Xerox), Order No. AA-K759B-TK

DNA Ethernet Data Link Functional Specification, Version 1.0.0, Order No. AA-Y298A-TK

DNA Maintenance Operations Functional Specification, Version 3.0.0, Order No. AA-X436A-TK

DECnet Digital Network Architecture (Phase IV) General Description, Order No. AA-N149A-TC

DNA Network Management Functional Specification, Version 4.0.0, Order No. AA-X437A-TK

DNA Routing Layer Functional Specification, Version 2.0.0, Order No. AA-X435A-TK

DNA Network Services Protocol (NSP) Functional Specification, Version 4.0.0, Order No. AA-X439A-TK

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

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

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

1.1 Functions

This specification addresses the following functional areas:

- o Initialization and self-test -- the processing that a node performs to declare itself operational.
- o Communication service -- the basic Ethernet communication service.
- o Communication diagnosis -- testing the ability of nodes to communicate over the Ethernet and isolating communication problems to specific nodes.
- o System diagnosis -- testing the node itself and isolating faults that prevent a node from functioning properly.
- o Down-line load and up-line dump -- using the services of a host node for storage and retrieval of system software over the Ethernet.

1.2 Requirements, Goals, and Non-goals

This section describes characteristics that the Ethernet Node design must have, that it will attempt to have, and that it will not have. These constraints vary somewhat according to the node's individual product requirements. They must thus be considered in that context.

The various classes of products are discussed in the Model section.

1.2.1 Requirements

This Ethernet Node design must have the following characteristics:

- o All Ethernet capabilities needed for higher level functions are available.
- o A network manager can control and observe the node and the network as they function.
- o The design complies with other applicable specifications.
- o The design can be adapted to the varied requirements of legitimately different products.

1.2.2 Goals

This Ethernet Node design attempts to have the following characteristics:

- o Be efficient in usage of processor, memory, and network.
- o Required functions are simple to implement.
- o Usage is predictable, simple, and consistent.
- o Node and network operation continue in the face of software, hardware, or management problems.
- o Security will not be decreased from the low level that is already present in a broadcast network.

1.2.3 Non-goals

The following are not goals of the Ethernet Node design:

1. High level functions addressed beyond those required for minimal maintenance. General network functions and most management functions are left for higher level definition.
2. Isolation of failing components within a node. This is left to the specific diagnostics for that product. This architecture defines only primitive access to the node, usable by specific diagnostics.
3. Functional partitions for implementations. For example, this document does not address the division of labor between a system processor and a communication processor.
4. Functions within a node that do not affect the network or need not be available across the network.
5. Security beyond the earlier statement of goals.

2 Models

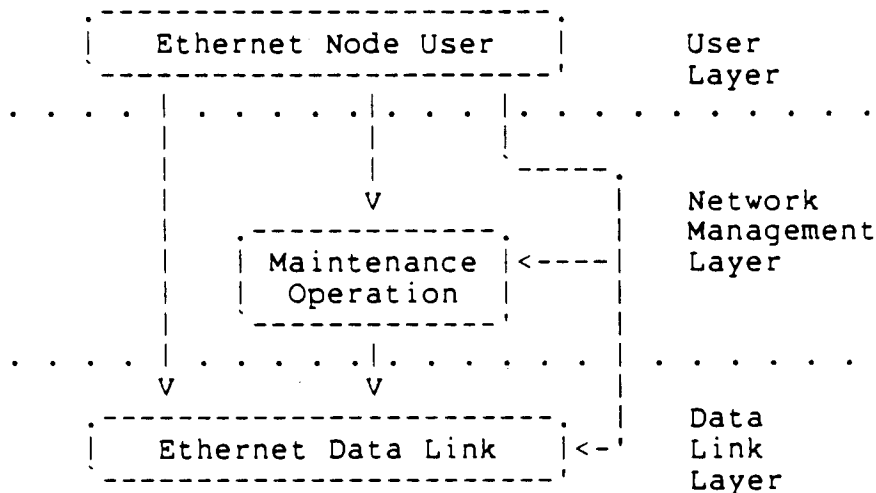
This section describes the relationship of the Ethernet Node architecture to other network layers and modules. It also defines the model used to classify nodes so that their functional requirements can be stated. Although this specification only describes how the Ethernet Data Link fits into the Digital Network Architecture (DNA), the Ethernet Data Link could be integrated into any layered network architecture (for example, Digital's System Communication Architecture).

2.1 Relation To Digital Network Architecture

The functions addressed in this specification are found in the User, Network Management, and Data Link Layers. Those that are in the Network Management Layer are defined in the DNA Maintenance Operations Functional Specification. The Data Link functions are in the DNA Ethernet Data Link Functional Specification. The User functions are defined in this specification.

Note that although there are other DNA layers, and other high level functions in the Network Management layer that use them, their existence is not directly relevant to this specification. The high level functions they represent may be provided by DNA or any other similar architecture.

The following diagram shows the relationship of the above mentioned modules.



Arrows indicate flow of control. Vertical arrowheads indicate User Interfaces, horizontal arrowheads indicate Network Management Interfaces.

2.2 Ethernet Node Type Model

This section defines a model for viewing Ethernet Node types. The model is in terms of how the node is maintained and controlled as related to the scope of this specification. This model is used later in this specification to define required functions for different node types in different states.

There are three important characteristics that must be considered:

1. Control -- causing a node to load or dump its system image. Causing a node to load is called "booting".
2. System image location -- where copies of the system image are actually kept. Transfer of the image is the process of up-line dump or down-line load.
3. Diagnosis -- detection and isolation of failures in the node.

Each of these necessary functions may be performed locally or remotely, that is, within the node itself, or on its behalf by some other node. Since these three functions may independently be performed locally or remotely, they can be combined into eight different configurations.

Any of the node types may be in one of three states relative to maintenance operations. Changes between these states are a matter of system operation, sometimes in implementation-specific ways and sometimes due to outside control.

1. Primitive -- the node supports only the absolute minimum maintenance functions for its type. For example, these functions may be implemented in read-only memory, while all other functions are available only when explicitly loaded.
2. Maintenance -- the node supports all maintenance functions appropriate to its type. This state is appropriate to smaller nodes that may not have room for all maintenance and normal functions simultaneously.
3. Normal -- the node is in its normal operating mode.

3 Required Functions

This section specifies the required interface functions for the various node types and states. Unless specified otherwise, functions are required regardless of node states as described in the Models section.

Any functions not required are optional. Inclusion of optional functions must be based on specific product requirements. For example, a node that is to be able to do active communication testing must implement the Loop Requester.

3.1 All Nodes

The functions in this section are required regardless of node type.

The following statements of principle must be met:

- . All protocol type usage must be in compliance with the applicable protocol specification.
- . The broadcast address must not be used unless a frame is intended for all nodes, regardless of function or manufacturer.
- . No two channels on the same cable may have the same physical address.
- . In any case of a node with multiple Ethernet channels, the User Layer must resolve all potential conflicts. For example, a node with multiple channels may have to resolve conflicts between multiple attempts to take control of its remote console.

3.1.1 Data Link Layer Requirements

The following User Interface functions must be implemented:

- . Open
- . Enable-protocol
- . Disable-protocol
- . Enable-multicast
- . Disable-multicast
- . Close
- . Transmit
- . Transmit-poll
- . Receive
- . Receive-poll
- . Receive-abort

The following Network Management functions must be implemented:

- . Read-channel
- . Enable-channel
- . Disable-channel
- . Read-counters

The Network Management Set-address function must be implemented on nodes that have more than one channel. It must also be implemented on all nodes that do not automatically have the correct DECnet address as described below.

All channel counters must be implemented. All portal counters must be implemented except for the maintenance protocols for loopback, console, and dump/load. In the case of the maintenance protocols, portal counters are optional.

All event information must be implemented. All event information must be available through either the User transmit and receive functions or the Network Management Read-event function.

3.1.2 Network Management Layer Requirements

The Loop Server must be implemented. Furthermore, it must be enabled whenever the Data Link state is "on". The only optional Loop Server functions are Enable-assistance and Disable-assistance. Note that all required functions must operate regardless of such special states as "promiscuous receive".

The Console Server Identify-self function must be implemented. The Console Server must respond to a remote Read-identity function. Furthermore, these functions must be enabled whenever the Data Link state is "on".

In normal state, the Console Server must respond to a remote Read-counters function.

In normal state, a DECnet Phase IV node must set its physical address to a function of its DECnet address. Referring to transmission order, the first three bytes of the address consist of one of the address groups assigned to Digital by Xerox. The fourth byte is a zero. The fifth byte is the low order byte of the 16 bit DECnet address, and the sixth byte is the high order byte of the 16 bit DECnet address.

So, for example, DECnet node number 14 would have the Ethernet address AA-00-04-00-0E-00 (this format for address display is further discussed in a later section).

3.1.3 User Layer Requirements

When enabling the Data Link, the node must perform sufficient initialization and self test, short of using the network, to believe that it can communicate properly. If it fails this self test, it must not come onto the network.

The node must report all event information in some way, not necessarily as events.

The node must implement some means of displaying the hardware address. The node must implement some means of displaying the physical address if it can be different from the hardware address. These requirements can be met, for example, with a printed label or a machine-accessible output.

The format for display of an Ethernet address must be according to the inter-company standard as found in the Ethernet Specification. That standard is repeated here for reader convenience. Each byte of the address is displayed as a hexadecimal number. The bytes are displayed from left to right, in the order that they are transmitted, separated by hyphens. The bits within the bytes are transmitted from right to left.

As an example, consider the address AB-CD-EF-01-23-45. The first byte transmitted is AB, the last is 45. The first bit transmitted is the low order bit of AB, a one, therefore the example is a multicast address. The last bit transmitted is the high order bit of 45, a zero. The first three bytes (AB, CD, and EF) are assigned by Xerox. The last three bytes (01, 23, and 45) are assigned by the manufacturer.

Whenever the Data Link state is "on", the node must periodically call the Console Server Identify-self function. For the precise definition of this period and an explanation of the use of this function, see the section on Network Auto-configuration.

3.2 Remote Control Nodes

This section states additional requirements for a node that is remotely controlled. This is a node that can be forced by a remote node to load or dump itself.

3.2.1 Network Management Layer Requirements

The node must implement response to the Console Server Boot function in primitive and maintenance states. The node must implement this response in normal state if it does not reliably and automatically go to primitive or maintenance state when it malfunctions.

3.3 Remote Dump/Load Nodes

This section states additional requirements for a node that uses remote storage of its system image. This is a node that is down-line loaded or up-line dumped.

3.3.1 Network Management Layer Requirements

The node must implement the Dump/Load Requester Load-self function in the primitive state. The Load-self function should start as far into the multi-stage load process as possible. In other words it should not go through the load of a secondary or tertiary loader program unless required by implementation constraints.

3.4 Remote Diagnosis Nodes

This section states additional requirements for nodes that are to be remotely diagnosed.

3.4.1 Network Management Layer Requirements

The node must implement Console Server response to the console carrier functions.

The required Console Server functions need not be available in primitive state if the node is remotely controlled, down-line loadable, and the node's self test covers all resources necessary to support them.

3.4.2 User Layer Requirements

The node must either keep the Console Server available for remote use or implement some reliable way of making it available.

4 Network Auto-configuration

The functions required above allow the automatic determination of the configuration of an Ethernet network. This section explains the algorithm for using the required primitives and the goals used in selecting the algorithm. This procedure is specified as part of general maintenance operations because it is Ethernet specific.

4.1 Goals

The following goals were used in the selection of an auto-configuration algorithm:

1. Configuration available within one hour.
2. With a 0.999 probability, configuration includes all nodes whose Data Link state is "on".
3. The configuration information includes physical addresses and system identification information.
4. The process is of low overhead for the nodes being configured and the overall network.
5. The process is simple for the nodes being configured to implement.
6. The process works correctly when multiple nodes attempt to determine configuration.
7. The process is simple and of low overhead on the node determining the configuration when this goal is not in conflict with the other goals.

4.2 Algorithm

Simply stated, every node periodically sends a system identification message to the remote console service multicast address. A node wishing to determine network configuration listens to this multicast address for a multiple of the transmission period and builds the configuration list from the received messages. The minimum requirement to implement this is that each system periodically call the Console Server Identify-self function.

The periodic transmission algorithm is:

```
Compute wait time.  
WHILE Data Link state = "on"  
    CALL Console Server Identify-self.  
    Wait.  
ENDWHILE
```


The wait time base is ten minutes. On a thousand node network this creates an average traffic of 100 frames/minute. Assuming a network capacity of 60,000 frames/minute, this represents an overhead of less than 0.2% of network bandwidth.

In order to avoid overwhelming a listener with synchronized messages, each node modifies its timer value so that nodes started at the same time will not transmit at the same time. The timer modification is a 16 bit signed number. The number must be random or pseudo-random based on a seed that is unique across similar implementations (such as the node address).

The resulting number is treated as milliseconds divided by 4. This number is added to the number of milliseconds divided by 4 in 10 minutes (150,000) resulting in a timer value of 10 minutes plus or minus about 2 minutes 11 seconds, to a resolution of 4 milliseconds.

The node then truncates this to its nearest timer resolution, and uses it as the time between sending identification messages.

Assuming a random distribution of modifications, a thousand node network that all came up at the same time would transmit over four minutes, a rate of about 4 frames/second.

A node collecting a list should listen for at least 40 minutes.

A node that has a list and wishes to confirm it can use the Console Requester Read-identity function to each node on its list.

READER'S COMMENTS

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

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

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

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

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

Name _____ Date _____

Organization _____

Street _____

City _____ State _____ Zip Code _____

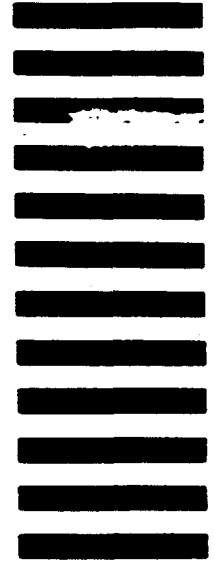
or
Country

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

digital



No Postage
Necessary
if Mailed in the
United States



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

POSTAGE WILL BE PAID BY ADDRESSEE

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

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

Cut Along Dotted Line

