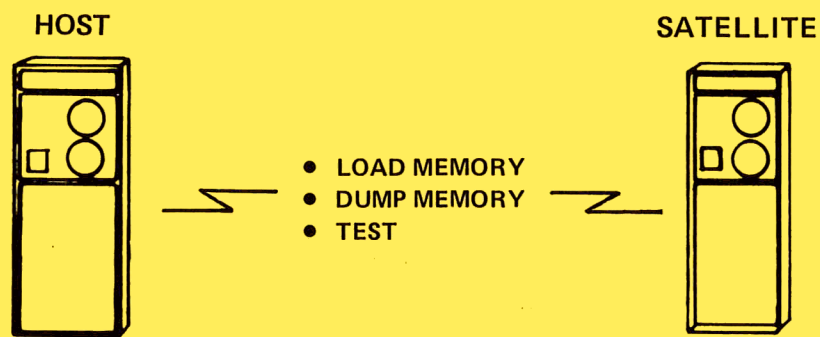


MOP



To order additional copies of this document, contact the Software Distribution Center, Digital Equipment Corporation, Maynard, Massachusetts 01754.

digital

DECNET
DIGITAL NETWORK ARCHITECTURE

Maintenance Operation Protocol
(MOP)

Functional Specification
Version 2.0

March 1978

First Printing, March 1978

Copyright (c) 1978 by Digital Equipment Corporation

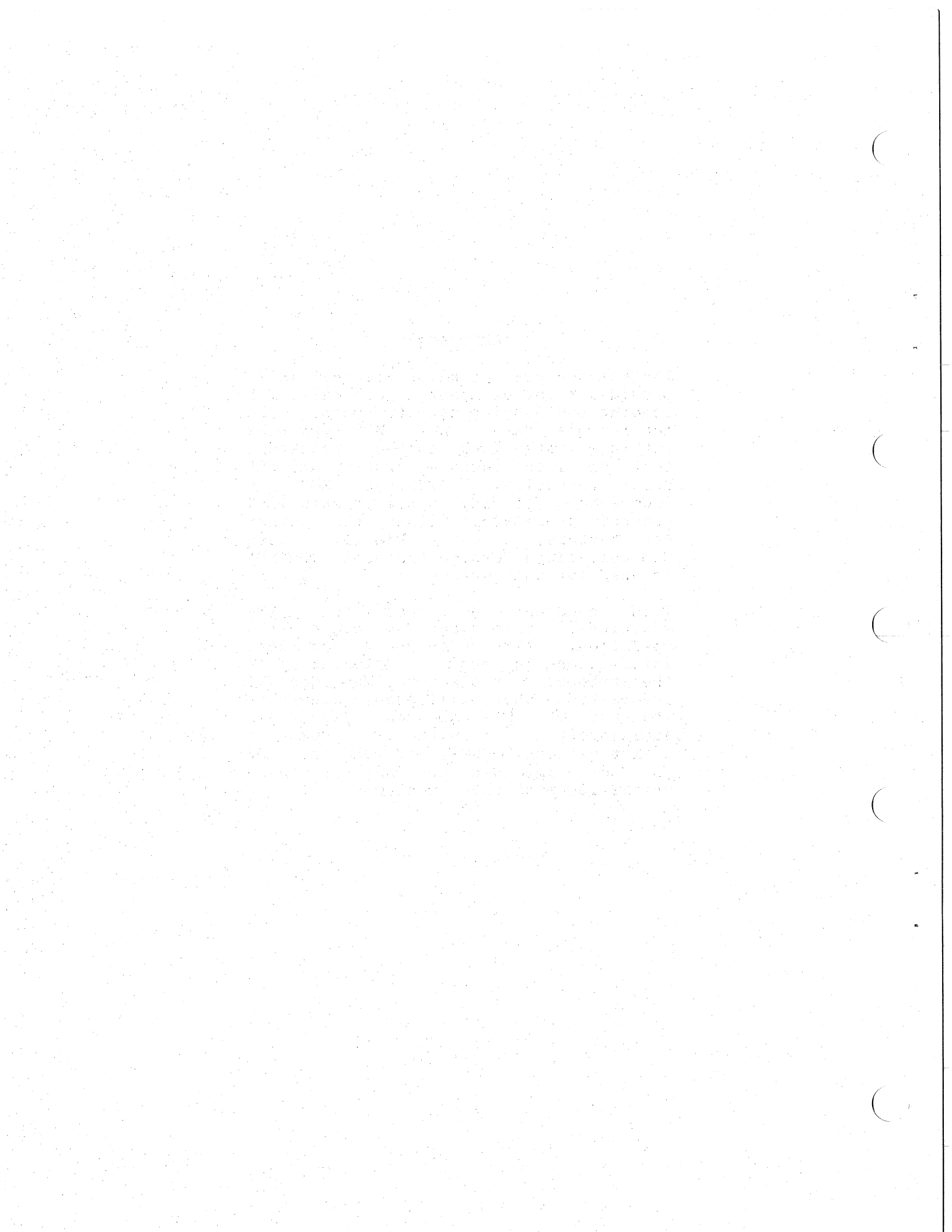
This material may be copied, in whole or in part, provided that the above copyright notice is included in each copy along with an acknowledgment that the copy describes the MOP protocol developed by Digital Equipment Corporation.

This material may be changed without notice by Digital Equipment Corporation, and Digital Equipment Corporation is not responsible for any errors which may appear herein.

ABSTRACT

The Maintenance Operation Protocol (MOP) provides a set of messages and rules for loading and dumping computer memory and testing a data link. MOP operates within a data link control procedure that provides message framing and bit error detection. Although MOP is independent of any specific data link control procedure, within the DECnet Architecture, the Digital Data Communications Message Protocol (DDCMP) is used for that purpose.

This document describes the MOP functions, interfaces, messages, and operation. It provides technical information to assist implementors of the protocol. It also provides general information that will enable others to evaluate MOP. The document assumes a familiarity with computer communications, DDCMP, and DECnet. It is not intended to instruct those unfamiliar with these concepts.



CONTENTS

SECTION	Page
1.0 INTRODUCTION	1
2.0 FUNCTIONAL DESCRIPTION	2
2.1 MOP Primary Mode	3
2.2 Memory Loading	3
2.3 Memory Dumping	4
2.4 Link Testing	4
2.5 Unattended System Control	4
2.6 Link Control Procedure	5
3.0 INTERFACE	6
3.1 Commands to DDCMP	6
3.2 Responses from DDCMP	7
4.0 MESSAGES	8
4.1 Message Format Notation	8
4.2 MOP Message Formats	9
4.2.1 Memory Load with Transfer Address (Deposit Memory and Transfer)	9
4.2.2 Memory Load without Transfer Address (Deposit Memory)	10
4.2.3 Request Memory Dump (Examine Memory)	11
4.2.4 Enter MOP Mode	11
4.2.5 Request Program	12
4.2.6 Request Memory Load	12
4.2.7 MOP Mode Running	13
4.2.8 Memory Dump Data (Examine Memory)	13
4.2.9 Reserved for Remote-11 Use	14
4.2.10 Reserved for Remote-11 Use	14
4.2.11 Parameter Load with Transfer Address	14
4.2.12 Loopback Test	16
5.0 OPERATION	17
5.1 MOP Primary Mode Operation	18
5.1.1 Primary Program Operation	18
5.2 Loading Computer Memory	19
5.3 Dumping Computer Memory	20
5.4 Link Testing	20
5.5 Unattended Control	21
6.0 MOP STATE TABLES	22
6.1 States	22
6.2 Satellite State Table	23
6.3 Host State Table	25

CONTENTS (Cont.)

	Page
APPENDIX A EXAMPLES	27
APPENDIX B DDCMP MAINTENANCE MODE FORMAT	29
APPENDIX C PROGRAM TYPES	30
APPENDIX D REVISION HISTORY	31

1.0 INTRODUCTION

The Maintenance Operation Protocol (MOP) is a set of messages and rules used to load and dump computer memory as well as test a communication link on a computer system directly connected to another system via that link. It operates within the data field of a data link control procedure that provides the functions of message framing (i.e., locating the beginning and end of a message) and bit error detection. Within DECnet, this data link control procedure is usually provided by the maintenance mode of the DDCMP protocol.

MOP allows the selection and initiation of functions from either computer on the link. It has been designed to execute with a minimum amount of code so that basic functions can be conveniently stored in a read-only memory (ROM) module of the system to be loaded or dumped.

MOP functions may be handled directly by the two adjacent computers involved in the MOP message exchange or by remote DECnet facilities (i.e., non-adjacent nodes). The control of MOP functions by remotely located nodes within a network is beyond the scope of this document.

2.0 FUNCTIONAL DESCRIPTION

MOP is a protocol used to: a) down-line load the memory of a computer system; b) dump the contents of that memory usually upon a failure; and c) test the data link and/or its hardware components (modems and interfaces). It performs these functions by exchanging messages over a data link connecting two computer systems. One system is designated the host, the other the satellite. The satellite is the object of the down-line load and the source of data for the dump. The satellite is used to loop back messages when MOP is in the link test mode. The host provides the data for the load and receives data from the dump. The host also initiates the link testing function. In some implementations, hosts may use the network to access files containing load and dump images rather than using their local storage facilities. Functionality and simplicity rather than performance have been the main objectives of MOP, so that basic functions take very little code and may be coded into read-only memories resident in the satellite system.

MOP operates in an alternate half-duplex mode. That is, first a message is sent in one direction, then in the other, alternately. MOP operates over a single data link (between two computers) within an error detection and message framing link control procedure envelope. The envelope provides the function of error detection. Only messages without any bit errors are passed to MOP. Message acknowledgments and retransmissions (if necessary) are handled at the MOP level via MOP messages.

When the host is connected into a DECnet network, it may get its down-line load images from another node via DECnet functions (rather than use a locally stored image). Similarly, it may send the dump to another node for storage. In this way, the loading and dumping functions are extended to nodes other than those directly connected to the satellite.

Hosts are required to implement all the MOP functions: loading, dumping and link testing. If a host does not have enough mass storage to store load and dump images, it may communicate with other nodes in the network to provide storage for these images. Satellite systems should implement what they are capable of. If only minimal amounts of read-only memory are available, then the satellite may implement only primary mode. If more read-only memory on a local load device is used to load read/write memory, then any subset of the secondary loading, dumping, or loopback link testing may be implemented. Those functions not directly implemented in a satellite will be down-line loaded from the host in a bootstrapping mode.

The functions provided by MOP are:

1. Load memory (deposit);
2. Dump memory (examine);
3. Loop testing;
4. Forced entry into the MOP mode (from running systems); and
5. Transferring control to programs resident in memory (whether loaded by MOP or by other means).

2.1 MOP Primary Mode

Most MOP functions execute at a level called the MOP secondary mode. The satellite may start in secondary mode by either having a large ROM containing the secondary code or by having the secondary code loaded via a local load device (e.g., cassette). Alternately, the satellite may contain a small ROM containing only MOP primary code. This code is then used to down-line load the secondary code into the system, effectively bootstrapping itself up to secondary level.

If the primary code is running it operates in the following manner:

- a. The primary code sends a Request Program Message to the host and expects in return a single Program Load with Transfer Address Message containing the secondary entire code.
- b. This secondary code is loaded into memory and started.
- c. All other functions then take place at this secondary level. Some functions may require an even larger program than can be handled by the secondary level program (which must be loaded in a single message). In these cases, the secondary level program is used to load the required program for the required functions (as is done for any load memory request).

Function requests may be initiated by the host or the satellite. If satellite-initiated, the satellite will send request messages to the host (e.g., the Request Program Message and Request Memory Load Message). If host-initiated, the satellite will ask the host for a request by sending a MOP Mode Running Message, and the host will reply with an appropriate request command to the satellite (e.g., the Memory Load Message and the Request Memory Dump Message). All message exchanges operate one message at a time alternately between the host and satellite.

2.2 Memory Loading

Memory loading is initiated either by the satellite sending a Request Program Message to the host, and the host responding with a Memory Load Message to the satellite; or the host sending a Memory Load Message to the satellite without having received such a request. The choice depends on the initiating computer and the previous function performed. Each load is numbered and acknowledged by having the satellite return a Request Memory Load Message to the host in response to each load, which acknowledges that load and requests the next one. The final load will be either a Memory Load with Transfer Address Message or a Parameter Load with Transfer Address Message, which will cause execution to be transferred to the loaded program after loading any necessary running parameters. After transfer, the new program may:

- a. continue to run MOP;
- b. run another protocol or data link control procedure on the link; or
- c. choose not to use the link at all and run stand alone.

Thus, MOP may be exited following a transfer to a loaded program.

2.3 Memory Dumping

Memory dumping is always initiated by the host. It sends Request Memory Dump Message commands to the satellite. The satellite, in turn, responds to these commands with Memory Dump Data Messages containing the memory image requested. This will usually continue until the host has completed the dump, at which time it will usually start a memory load operation as described above by sending a Memory Load Message.

2.4 Link Testing

Link testing is accomplished via the Loopback Test Message. The host sends such a message out on the link and waits for the message to be returned. The satellite returns the message exactly as sent. By using the identical message the satellite can be replaced by a loopback plug or modem looping facility. In this way problems can be diagnosed by moving the loopback point and isolating components. When the message is returned to the host, the host remembers that it originally sent the message and does not return it again (if returned, this could cause continuous retransmission).

2.5 Unattended System Control

The Enter MOP Mode Message is used to control an unattended system. This message, together with the appropriate hardware, enables a satellite computer to halt current operation and begin operating in either the MOP primary or secondary mode. This is accomplished by transferring control to a resident MOP program or bootstrap. The hardware is used to recognize this message and force the computer system to transfer control to the MOP program, usually residing in a read-only memory. The password in this message protects the system from being controlled and loaded by an unauthorized host. Only messages with a matching password will cause the system to enter MOP mode.

A detailed description of the MOP operation including error recovery is provided in Section 5.0 Operation.

2.6 Link Control Procedure

MOP operates within the envelope of a data link control procedure that provides error-free transmission and reception on a data link connecting the host and satellite systems. This procedure provides two functions:

1. Message framing - locating the beginning and end of a message at the receiving end of a link. This procedure will usually involve bit, byte, and message synchronization. MOP only sends and receives messages that are multiples of 8-bit bytes.
2. Bit error detection - the detecting of one or more bit errors introduced by the communication medium. A requirement of the link control procedure is that only good messages (without bit errors) are passed to MOP. Messages that arrive with data bit errors may be optionally flagged to MOP to speed up error recovery rather than have MOP wait for timeouts. The detection mechanism assumes that messages are completely received in memory by the link control procedure before they are checked for bit errors and passed to MOP. The previous contents of the buffer used to receive the message will, therefore, be destroyed and the old contents will not be available for use (e.g., dump). The location of the receive buffer should be chosen to cause the minimum interference with MOP operation. If this checking is not done through a receive buffer and a load message is directly loaded into its specified load address, then if the message were detected to be in error, the load address might be in error and the image loaded in the wrong place, destroying other information.

The link control procedure must also perform all other functions necessary for proper operation of the channel. These include link management functions for turning around half-duplex links and selecting and addressing multipoint stations. Any timers necessary for proper operation of these link management functions are provided by the link control procedure and are transparent to MOP. The selection of a station and turning around of the link are triggered by the MOP transmit and receive commands described below. No specific commands are issued for link management functions. The link control procedure may handle modem control signals directly or pass them through the MOP interface to be handled by MOP directly. The link control procedure may record errors on the link but is not required to pass them to MOP. The link control procedure may have a special mode for use with MOP. This mode will provide the functions listed above necessary for MOP operation. It may be different from other operating modes of the link control procedure, usually trading off performance for simplicity. There must be ways to both enter and exit this special mode. See the DDCMP Specification for a more detailed description of how DDCMP provides these functions.

3.0 INTERFACE

This section describes how MOP interfaces to the data link control procedure used for framing and error detection. It utilizes DDCMP as an example of such a data link procedure. References to DDCMP imply a reference to the actual data link procedure used.

The interface between MOP and DDCMP consists of a number of commands to DDCMP and responses from DDCMP used to transmit and receive MOP messages to and from the data link. The actual interface depends heavily on the features and capabilities within the operating systems running MOP and DDCMP. Mechanisms used for passing commands and receiving responses might include shared tables, calls with parameter lists, I/O registers, and/or interrupt mechanisms.

The information passed between MOP and DDCMP are messages. Its description usually consists of a starting buffer address and a length or byte count, or a chain of addresses and counts.

3.1 Commands to DDCMP

1. Enter Maintenance Mode. This command initializes DDCMP into the maintenance mode before transmitting or receiving maintenance mode messages. This is the special mode used for MOP operation. If a MOP message is received while not in this mode the protocol will halt, inform the user that such a message was received, and allow the user to restart the protocol in this mode via the Enter Maintenance Mode command.
2. Transmit Message. This command gives a message to DDCMP for transmission in the maintenance mode.
3. Receive Message. This command gives an empty buffer to DDCMP for reception of the next maintenance message. Alternately, MOP might supply DDCMP with a pool of buffers and have the protocol select one. In this mode there will be a command to Return Empty Buffers to the pool so they may again be assigned by DDCMP.
4. Halt Protocol. This command will stop transmission and reception by DDCMP. Optionally, there may be a way to control the modem signals and force the modem to hang-up in the dial case. The protocol may also halt by receiving a maintenance mode message while in normal mode or receiving a normal mode message while in the maintenance mode. After halting, the protocol may be started again in the maintenance mode or in the normal mode.

3.2 Responses from DDCMP

1. Message Transmitted. Response to the Transmit Message command. The message has been sent out on the link. The response is returned after actual transmission is completed and includes any delay due to selection, link turnaround and transmission time.
2. Message Received. The next maintenance message has been received. Either MOP supplied a buffer in a Receive Message command, or a buffer will be taken from a pool previously supplied to DDCMP. In some implementations, if DDCMP has not been initialized into the maintenance mode and a maintenance mode message is received, there may be a separate response indicating that the protocol at the other end of the link is in maintenance mode. At that point, the protocol will halt, and MOP will have to initialize DDCMP into the maintenance mode prior to transmitting and receiving maintenance messages. In these implementations, the original received MOP message may be lost and not actually passed to MOP.
3. Message Received in Error. An optional reply to MOP indicating that a maintenance message was received with a data CRC (block check) error. DDCMP was able to frame the message but it had one or more bit errors in the data field. This response is useful in reducing the length of time MOP will wait for a reply to a previously sent message. Without this response MOP will wait for a MOP timeout interval. The length of this interval is implementation specific, but must include processing delays and transmission time. Typically, the value would be the same as the select or response timer value used by the data link control procedure.

4.0 MESSAGES

4.1 Message Format Notation

All MOP messages are sent embedded within a physical link control protocol. Within DECnet, the DDCMP maintenance mode envelope is generally used for this purpose. This section presents the general format of the MOP messages sent within the data field of that envelope. The DDCMP Maintenance Mode Format is specified in Appendix B.

The following notation will be used to describe the MOP messages:

Field (length): coding = description of field

Where:

Field = the name of the field being described

length = the length of the field as:

1. A number meaning number of 8-bit bytes:
2. The letters "I-n" meaning image field with n being a number that is the maximum length in 8-bit bytes of the image. The image is preceded by a 1-byte count of the length of the remainder of the field. Image fields are variable length and may be null (count=0). All 8 bits of each byte are used as information bits. If the length is "R" the field may extend the remainder of the message (maximum length is determined from maximum message length).

coding = the representation type used:

B = Binary
BM = bit map (each bit has independent meaning)
A = ASCII
Null = interpretation depends on data representation

Notes:

1. All numeric values are shown in decimal representation unless otherwise noted.
2. All fields are transmitted low-order or least-significant bit first on the data links unless otherwise specified.
3. Bits in a MOP field are numbered from 0 to n where 0 is the low-order or least-significant bit.
4. Fields that refer to memory on specific computers are numbered according to the conventions on that computer system.
5. The same names used in fields in separate messages have the same meaning and format.

4.2 MOP Message Formats

The general format of MOP messages is:

CODE	OPERAND
------	---------

where:

CODE = the MOP message type code (1 byte).

OPERAND = the operand information specific to each message type.

4.2.1 Memory Load with Transfer Address (Deposit Memory and Transfer)

This message causes the contents of the image data to be loaded into memory at the load address, and the system to be started at (the PC set to) the transfer address.

CODE	LOADNUM	LOADADDR	IMAGE DATA	TRANSFER ADDR
------	---------	----------	------------	---------------

Where:

CODE (1):B = 0

LOADNUM (1):B = the load number for multiple load images. This message may be preceded by Memory Load without Transfer Address Messages. The load number starts at zero and is incremented for each load message sent in a loading sequence. A load number of zero is always valid and resets the expected load number. Zero should not be used for all load numbers in a sequence of load messages, as it nullifies the sequence checking of the MOP protocol. LOADNUM is modulo 256. After load number 255 is load number 0.

LOADADDR (4):B = the memory load address for storage of the data image.

IMAGE DATA = the image to be stored into computer memory. The form sent will be machine dependent and will vary with the type and word length of the system:

PDP-11 - each byte represents 1 memory byte.

VAX-11/780 - each byte represents 1 memory byte.

PDP-8 - each 3 bytes represents 2 memory words.

byte 1 = low 8 bits of memory word 1 (4-11)
byte 2 = low 8 bits of memory word 2 (4-11)
byte 3 = low 4 bits of byte are high 4 bits
of word 1 (0-3), high 4-bits of
byte are high 4-bits of word 2
(0-3).

DECSYSTEM 10/20 - each 5 bytes represents one 36-bit word.

byte 1 = highest numbered (i.e., low-order) 8 bits of word (28-35)
byte 2 = next 8 bits (20-27)
byte 3 = next 8 bits (12-19)
byte 4 = next 8 bits (4-11)
byte 5 = low 4 bits of byte are highest order 4 bits of word (0-3); highest 4 bits of byte are unused and set to zero.

TRANSFER ADDR (4):B = the starting address of the image just loaded.

NOTE

IMAGE DATA or LOADADDR and IMAGE DATA may be omitted. Valid MOP message lengths (DDCMP count values) will be 6 (LOADADDR and IMAGE DATA omitted), 10 (IMAGE DATA omitted), or greater than 10.

4.2.2 Memory Load without Transfer Address (Deposit Memory)

This message causes the contents of the image data to be loaded into memory at the load address.

CODE	LOADNUM	LOADADDR	IMAGE DATA
------	---------	----------	------------

Where:

CODE (1):B = 2

Refer to Section 4.2.1 for a description of other fields.

NOTE

IMAGE DATA may be omitted. Valid MOP message lengths may be 6 (IMAGE DATA omitted), or greater than 6. Messages without IMAGE DATA cause nothing to be loaded, however, the LOADNUM value is still incremented for the next load.

4.2.3 Request Memory Dump (Examine Memory)

This message requests a dump of a portion of memory to be returned in a memory data dump message.

CODE	MEMADDR	NUMLOCS
------	---------	---------

Where:

CODE (1):B = 4

MEMADDR (4):B = the starting memory address for the dump.

NUMLOCS (2):B = the number of locations to dump.

where:

PDP-11 = 1 location is 1 byte

VAX-11/780 = 1 location is 1 byte

PDP-8 = 1 location is 1 word

DECSYSTEM-10/20 = 1 location is 1 word

NOTE

This request will result in a single Memory Dump Data Message. A dump should not be requested for more data than can be reliably sent in a single reply on the link type used (i.e., the maximum length is limited by the maximum link control procedure message length for a given link).

4.2.4 Enter MOP Mode

This message causes a system not in the MOP mode to enter the MOP mode if the password matches. This usually means transferring control on the satellite to a MOP program.

CODE	PASSWORD
------	----------

Where:

CODE (1):B = 6

PASSWORD (4):B = a password that must match before the receiving station will enter the MOP mode. If the password is sent over a DDCMP link in the maintenance mode, the link will enter the DDCMP maintenance mode (due to the maintenance mode envelope), but the node will only enter the MOP mode (e.g., respond to load requests) if the PASSWORD matches.

4.2.5 Request Program

This message requests a program to be sent in some number of memory load messages.

CODE	DEVTYPE	MOPVER	PGMTYPE	SOFTID
------	---------	--------	---------	--------

Where:

CODE (1): B = 8

DEVTYPE (1): B = the device type at the requesting system. Used to cause the proper requested program to be loaded if it is device specific. Types are:

- 0 DP11
- 2 DU11 or DUV11
- 4 DL11-E or DLV11-E
- 6 DQ11
- 8 DA11-B
- 10 DUP11
- 12 DMC11
- 20 DTE20
- 32 KL8J

MOPVER (1): B = the MOP format version number. For specifications version 1 and 2 the format version number is 1.

PGMTYPE (1): B = the generic type of program requested (see Appendix C for a description of these types):

- 0 (or omitted) - secondary loader
- 1 - tertiary loader
- 2 - operating system

SOFTID (I-R): A = identification of the software being requested. Specified as an image field of ASCII characters. This information is not needed if PGMTYPE is enough to specify the requested program. Also omitted if PGMTYPE is omitted.

4.2.6 Request Memory Load

This message requests the next load in a loading sequence and provides error status on the previous load.

CODE	LOADNUM	ERROR
------	---------	-------

Where:

CODE (1): B = 10

LOADNUM (1): B = the load number being requested

ERROR (1): B = error indicator for previous load numbered segment:

- 0 (or omitted) - no error
- 1 - "Memory Load" Message
Image data not properly
loaded (e.g., memory boundary
problem).

4.2.7 MOP Mode Running

This message indicates to a host that the system is in the MOP mode and supports the features indicated.

CODE	DEVTYPE	MOPVER	MEMSIZE	FEATURES
------	---------	--------	---------	----------

Where:

CODE (1):B = 12

DEVTYPE = refer to Section 4.2.5

MOPVER = refer to Section 4.2.5

MEMSIZE (4):B = the size of physical machine memory in number of locations. See Section 4.2.3 for location definition.

FEATURES (1):BM = features available at this station according to the bit(s) set.

<u>Bit</u>	<u>Feature</u>
0	Multiblock load. Accepts the following messages: 1. Memory Load With Transfer Address message; 2. Memory Load Without Transfer Address message; and 3. Parameter Load With Transfer Address message.
1	Dump. Accepts the Request Memory Dump message.
2	Loopback. Accepts the Loopback Test message.
3-7	Reserved

4.2.8 Memory Dump Data (Examine Memory)

The reply to a Request Memory Dump Message sending back the requested memory image.

CODE	MEMADDR	IMAGE DATA
------	---------	------------

Where:

CODE (1):B = 14

MEMADDR (4):B = the starting address of the dump

IMAGE DATA = the dump of memory in the same form as IMAGE DATA in Section 4.2.1.

4.2.9 Reserved for Remote-11 Use

Remote-11, a networking product using RT-11 uses this message type.

CODE	MESSAGE
------	---------

Where:

CODE (1):B = 16

MESSAGE = Refer to Remote-11 Documentation

4.2.10 Reserved for Remote-11 Use

Remote-11, a networking product using RT-11 uses this message type.

CODE	MESSAGE
------	---------

Where:

CODE (1):B = 18

MESSAGE = Refer to Remote-11 Documentation

4.2.11 Parameter Load with Transfer Address

Used instead of memory load with transfer address (as the last load) to load system parameters before transferring control to the loaded program.

CODE	LOADNUM	PARAMETERS	TRANSFER ADDR
------	---------	------------	---------------

Where:

CODE (1):B = 20

LOADNUM = refer to Section 4.2.1

PARAMETERS = ENTRY, ENTRY, ... ,ENDMARK

ENTRY = PARTYPE, PARLENGTH, PARVALUE
(ENTRY fields are optional)

PARTYPE (1):B = parameter type number

PARLENGTH (1):B = number of bytes in PARVALUE

PARVALUE = parameter value according to
PARTYPE and PARLENGTH

PARTYPE	PARLENGTH	PARVALUE
1	1 to 6	ASCII node name
2	1 to 2	Binary node number
3	1 to 6	ASCII name of host assigned to node (e.g., control host for task loading of core-only systems).

ENDMARK (1):B = 0

TRANSFER ADDR = Refer to Section 4.2.1

NOTE

The parameter message, if it is used, will be the last in a multiblock load. The minimum MOP message length is 7. The following techniques are to be used to pass the parameters to the operating systems:

PDP-8 = Currently Unspecified

PDP-11 = The parameters are stored at the top of physical memory in the following locations:

NOTE

"top of memory" refers to the first (even) address beyond available memory. Its value is the same as the value returned in the MEMSIZE field (refer to Section 4.2.7).

top of memory - 64. bytes = a 2-byte node number. If none then zeros.

top of memory - 62. bytes = a 6-byte node name padded with spaces. If none, then first 2 bytes are zeros.

top of memory - 56. bytes =
a 6-byte host node name padded
with spaces. If none, then
first 2 bytes are zeros.

top of memory - 50. bytes =
all bytes through top of
memory reserved.

The registers are set as
follows:

R0 = load device CSR address
R1 = -1
R2 = -1
R3 = unit number of load
device
R4 = 2-byte ASCII load device
mnemonic:

XL = DL11E, DLV11E
XU = DU11, DUV11
XW = DUP11
XM = DMC11
XB = DA11B
XP = DP11
XQ = DQ11

VAX-11/780 = Currently
Unspecified

DECSYSTEM-10/20 = Currently
Unspecified

4.2.12 Loopback Test

Used to test a link by echoing the message sent. The message is sent by the host and echoed by the satellite. When the original sender (host) receives the message back it does not echo it again and cause an infinite loop. The message is returned by the satellite exactly as sent. In this way, a loopback turnaround connector or loopback modem switch may be used instead of a satellite computer to test the link at different points and isolate a problem.

CODE	LOOPDATA
------	----------

Where:

CODE (1):B = 24

LOOPDATA = data to be looped back. This message is returned exactly as sent. The maximum length is limited by the maximum link control procedure message length for a given link.

5.0 OPERATION

MOP messages are sent within a link control procedure providing the functions of message framing and bit error detection. It operates between two computer systems directly connected by a data link, which is dedicated to the MOP operation. On multipoint links the subchannel connecting the two systems is dedicated to MOP operation. MOP messages are sent alternately between the computers. One system is designated the host and the other the satellite. The host controls MOP operation and usually initiates the error recovery procedures. The host provides the data for the memory load, receives the data from the memory dump, and performs the link test function. The satellite is the object of the load, responds with data for the dump, and echoes messages for the link testing function. All message acknowledgment, timeout, and retransmission functions are handled at the MOP level via MOP messages and functions. The link control procedure only provides a bit error check. A complete description of the link control procedure functions is provided in a previous section.

Within DECnet, the link control procedure is the DDCMP maintenance mode. A complete description of the operation of DDCMP can be found in the DDCMP Specification. DECnet facilities have extended the capabilities of MOP by providing mechanisms whereby the host can get load images from and send dump images to other nodes in a DECnet network, rather than using local storage for these images. This allows the control of loading and dumping functions to be extended to nodes other than the nodes directly connected to the Satellite. In either case, whether the host gets data from and sends data to a local device (e.g., disk) or another node in the network (e.g., DECnet node), the operation between the host and the satellite is the same.

In general, the link control procedure should buffer incoming messages, check them for bit errors, and then pass them to MOP. The buffer area should be chosen so that it does not conflict with areas of memory requiring MOP access (e.g., loading or dumping). Load images should not be loaded directly into the specified load address in load messages. Messages with bit errors might have load addresses in error, which would cause them to be loaded into the wrong area of memory, possibly destroying important information.

The only exception to this is the loading of the secondary loader, which is known in advance (by specification) to be loaded in a particular memory area.

Error recovery is handled at the MOP level via timeouts. That is, when a command is sent, a timer is started. If a response is not received within a timeout interval, the timer will expire and error recovery procedures will be initiated. The timeout function is usually handled by the host system. The exception to this is MOP primary mode initialization where the satellite is in the MOP mode and the host is not. In this case, timeouts and retransmission will occur from the satellite. The link control procedure will pass messages without bit errors to MOP and ignore all others. Those with bit errors will be recovered by the timeout mechanism. Optionally, the link control procedure may notify MOP that a message was received with one or more bit errors. In this case, MOP may initiate error recovery procedures prior to the expiration of the response timeout. This may improve the performance of MOP on links with high error rates. The error recovery procedures are the same, whether initiated in response to a timeout or notification from the link control procedure of message reception with bit errors.

5.1 MOP Primary Mode Operation

MOP usually operates at the secondary level where program loads, dumps, and link testing is performed. If there is enough space within the satellite to contain the entire secondary program (either within ROM, or loaded from a local load device), the satellite will start in the MOP secondary mode, otherwise a minimum program is used (primary mode) to bootstrap the satellite system to the secondary mode. The dialogue to achieve secondary mode operation is:

1. If the primary program is running, it sends a Request Program Message to the host system requesting the secondary loader.
2. When the host receives this message, it sends the entire secondary program in the form of a single Memory load with Transfer Address Message.
3. If this is in error (no message received), the primary program requests again after a timeout period.
4. Once the secondary loader is loaded and running, the secondary loader may send either:
 - a. The Request Program Message - to request a special function program (e.g., tertiary loader) or an operating system, or
 - b. The MOP Mode Running Message - to tell the host what functions are available and wait for the host to initiate action.

The choice of message will depend on the capabilities of the specific secondary program that was loaded. If it is an advanced loader program only (e.g., tertiary loader) then it will send a Request Program message. If it is a more general purpose secondary program capable of multiple functions (e.g., load, dump, test), then it will send the MOP Mode Running Message. This flexibility allows control requests to come from either the satellite or the Host depending on the operating requirements.

If the secondary loader already exists in the satellite system, the dialogue will start at Step 4 above. The action taken in step 4 determines whether (1) the satellite wants to initiate action (e.g., load memory) or (2) the host selects the MOP function (the Satellite simply tells the host that the MOP mode is running).

There may be many secondary loaders with different capabilities. The specific network and host will determine which is the appropriate one to use. Once the secondary loader has been loaded it may load other loaders using the loading memory function.

5.1.1 Primary Program Operation

The primary programs are specific to each computer family. That is, they exchange the same messages but may operate internally in different ways. The required operation of the primary program for each computer family is as follows:

- a. PDP-8 - Currently Unspecified
- b. PDP-11 - The primary program sends a Request Program message for the secondary loader (PGMTYPE and SOFTID in message are

omitted). It expects the secondary loader to be returned in a single Memory Load with Transfer Address message. In this message, the fields must have the following values:

```
LOADNUM=0
LOADADDR=6
TRANSFER ADDR=6
```

The MOP message (from CODE to TRANSFER ADDR) is loaded at location 0. That is, the secondary loader (IMAGE DATA field) is loaded in a single message block, starting at location 6 in physical memory with a starting transfer address to location 6. Upon transfer, the PDP-11 registers will have the following values:

R1=load device CSR address

The primary loader starts a timer while waiting for the Memory Load Message. If the secondary loader is received in error, no loader is received or any other message is received, the primary program will timeout and restart, retransmitting the Request Program Message again. If this cycle occurs some threshold number of times the satellite will assume the host is not answering and may disconnect the link (e.g., "hang up" a switched circuit).

- c. VAX-11/780 - Currently Unspecified
- d. DECSYSTEM 10/20 - Currently Unspecified

5.2 Loading Computer Memory

The loading of memory occurs in the MOP secondary mode. The loading may be initiated by the Satellite secondary program or by the host as described in Step 4 (see Section 5.1). In either case, loading occurs by the exchange of Memory Load messages from the Host and Request Memory Load responses from the satellite. Loading starts with load number 0 and is incremented for each successive load. The Host sends a Memory Load with a load number of 0. If the Memory Load message has no bit errors, it is loaded into memory and a Request Memory Load for load number 1 is returned by the Satellite. This both acknowledges load 0 and requests the next load (load 1). Any error occurring at the satellite in load 0 is reported in this message as well.

If the load message is in error, the satellite does nothing and waits for the host to timeout and retransmit the load again. If the wrong load number is received the Satellite responds with a request for the expected load number. Alternately, if the satellite is confused (e.g., wrong load received too many times) it may go back to its initial state and send either a MOP Mode Running Message or a Request Program Message back to the host. The host is then responsible for restarting the load from load 0 again. In general, it is assumed the host is capable of more intelligence than the satellite, and the burden of the error recovery is put on the host. If the host receives a message with bit errors, it should send its last message again, either upon a timeout or optionally, on notification of the error from the link control procedure.

The final load will cause the secondary mode loader to transfer to the designated transfer address after, optionally, acknowledging that last load with a Request Memory Load message for the next expected load (the load after the last one). The acknowledgment is omitted when it

is known that the loaded program will itself be sending MOP messages (e.g., as in the case of the loading of a higher level MOP function). If the program requested was a loader, the final load is acknowledged, since the loader will send a Request Program Message when it starts, acknowledging its proper operation. The final load, including the transfer address, may be either a Program Load with Transfer Address Message or a Parameter Load with Transfer Address Message.

If a MOP Mode Running or Request Program Message is received by the host following the sending of the message with a transfer address, the received message should be examined to determine if it was sent by the program to which the host intended to transfer control or was sent by the MOP loader to which the transfer message was sent. This determination is necessary to distinguish between the initial load request of a new loader and the restarting of the current loader. This determination involves comparing the program request or features fields of the received message against the current loading parameters to determine if they came from the old or new program. If they came from the old program then loading should start again from the beginning. If they came from the new program then successful loading has occurred and new MOP operation may commence.

Loads that exceed memory limits are not loaded and cause the next load to be requested with an error code (ERROR field) included in that request. In some cases, the secondary loader will request an advanced loader (e.g., a tertiary loader), which will then load the operating system. One MOP program may load another via the load memory procedures and thus transfer from one MOP program to another in a series of MOP operations.

5.3 Dumping Computer Memory

The dumping function is initiated from the host system. Request Memory Dump Messages are sent to the satellite system, which responds with a Memory Dump Data message. If no reply is returned, the host will timeout and repeat the request. If the satellite receives a message with bit errors, it may either ignore them and let the host timeout or return a MOP Mode Running Message to cause retransmission of the request at the host prior to the timer expiring. Requests outside memory bounds result in a Memory Dump Data message being sent with zeros for image data. Requests for more than can be sent in a single block will result in the maximum that can be sent in a block, the remainder of the request being ignored.

5.4 Link Testing

The Loopback Test message is used to test the condition of a data link. The message is sent by the host and simply turned around and returned on the data link. The actual looping back may be done by a program in the satellite or by a loopback box or switch placed somewhere between the host and satellite systems. With this technique, problems can be pinpointed by moving the loopback point and isolating components.

The system that sends the Loopback Test Message does not echo it again upon return as this would cause a loop condition. By varying the data bit patterns, many problems may be diagnosed.

5.5 Unattended Control

The Enter MOP Mode Message is used to control unattended systems. If sent to a running satellite that implements this message, it will cause the satellite to transfer control to MOP primary or secondary mode programs in the satellite. If implemented in hardware, the hardware will search for the proper Enter MOP Mode Message string. Upon a match, the hardware will cause a transfer to the MOP program, usually residing in read only memories. This will allow a system that is in a loop or halted to be remotely controlled and rebooted.

A password is part of the Enter MOP Mode Message to avoid improper or accidental use. Only a message with a matching password is considered valid and processed. All others are ignored. Upon entering MOP mode, the primary or secondary program will execute as described above under primary operation.

6.0 MOP STATE TABLES

This section presents the possible states, events, and actions to be taken by host and satellite systems running the MOP protocol.

6.1 States

Satellite States:

non-MOP - The satellite is not in MOP mode.

PRIMARY MODE - The satellite is running the basic primary mode program and trying to load the secondary program from the Host.

SECONDARY MODE - The satellite is running the secondary mode program and waiting for a command from the Host.

LOADING MODE - The satellite is running in the secondary loading mode where it is requesting loads from the host and loading them into memory.

HOST states:

non-MOP - The host is not in MOP mode with respect to the connected satellite.

Loading mode - the host is loading the memory of the satellite.

Dumping mode - the host is dumping the memory of the satellite.

Link test mode - the host is testing the data link.

6.2 Satellite State Table

State	Event	New State	Action
non-MOP	receive valid Enter MOP Mode	Primary, Secondary, or Loading	1.1
	User requests MOP mode	Primary or Secondary	1.1
Primary	receive valid Memory load with Transfer address	Secondary or Loading	2.1
	receive other message	Primary	2.2
	timeout	Primary	2.3
Secondary	receive Memory Load	Loading	3.1
	receive Transfer	Non-Mop	3.1
	receive Dump request	Secondary	3.2
	receive Loopback Test Message	Secondary	3.3
	receive other message	Secondary	3.4
Loading	receive valid Memory Load	Loading	4.1
	receive invalid Memory Load	Loading	4.2
	receive Transfer	non-MOP	4.3
	receive other message	Loading or Secondary	4.4

Satellite State Table Actions

- 1.1 If primary code is available in satellite, then send Request Program Message for secondary code and enter Primary state. If secondary code is available, then send either:
 - (a) the Request Program Message for a tertiary loader or operating system (if secondary is only a loader) and enter Loader State or,
 - (b) the MOP Mode Running Message (if other functions are supported) and enter secondary state.
- 2.1 Valid Memory Load is per requirements of primary code for each computer family. If loaded then take action as for secondary code under 1.1
- 2.2 Send Request Program message again and wait for secondary code.
- 2.3 Take action as in 2.2 until some threshold is exceeded, then reset, hanging up link if a dial-type connection.
- 3.1 If load number 0, then handle as if message had been received in Loading state.
- 3.2 Send Memory Dump Data response.
- 3.3 Echo Loopback Test Message exactly as received.
- 3.4 Either ignore or send messages as described for secondary under 1.1
- 4.1 Valid Memory Load per requirements of computer family, LOADNUM field must be as expected (or zero). If valid then load into memory, send a Request Memory Load for received LOADNUM + 1.
- 4.2 Either send Request Memory Load for expected LOADNUM or ignore.
- 4.3 If Memory Load with Transfer, first load memory part. If Parameter Load with Transfer, set up parameter list. If the transfer is to an operating system (or other non-MOP program), send Request Memory Load for LOADNUM + 1 and then transfer to TRANSFER ADDR.
- 4.4 Either send Request Memory Load for expected LOADNUM or return to Secondary state and take action as under 1.1 for secondary code.

6.3 Host State Table

State	Event	New State	Action
non-MOP	receive valid Request Program	Loading	1.1
	receive valid MOP Mode Running	Loading, dumping, or link test	1.2
Loading	receive valid Request Memory Load	Loading or non-MOP	2.1
	receive MOP Mode Running	Loading, Dumping or Link Test	2.2
	timeout	Loading	2.3
Dumping	receive valid Request Program	Loading	1.1
	receive Memory Dump Data	Dumping, loading, or Link Test	3.1
	timeout	Dumping	3.2
Link test	receive Looptest	Link Test	4.1
	timeout	Link Test	4.2

Host State Table Actions

- 1.1 Send first block of program requested, start timer, enter Loading state. If sending secondary code as a single message then do not start timer and stay in non-MOP state.
- 1.2 Determine function to be performed. If loading, take action as in 1.1; if dumping, send a Request Memory Dump request, start timer, and enter Dumping state; or if link testing, send a Loopback Test message, start timer, and enter Link test state.
- 2.1 If request for either last load number sent or next load, send:
(a) Memory Load message either with or without transfer address or
(b) send Parameter Load with Transfer Address if last load and parameters are required. If request for another load, restart loading from load 0 again. Restart timer when sending response. If received Request Load is in response to transfer message (last load) then enter non-MOP state and do not start timer.
- 2.2 If the MOP Mode Running Message indicates that the desired features are not present, restart loading from load 0. This could occur if the satellite received an invalid message and desired to reinitialize.

If the desired features are present, then the action described in 1.2 should be taken.
- 2.3 Send last load again, restart timer.
- 3.1 If more memory is required to be dumped then send another Request Memory Dump, start timer. If loading is to be started, then take action as described under 1.2. If link testing is to be started, then take action as described under 1.2.
- 3.2 Send Request Memory Dump again and restart timer.
- 4.1 If more link testing is to be done, send another Loopback Test message and start timer. For other functions, take action as described under 1.2.
- 4.2 Resend Loopback Test and restart timer.

APPENDIX A

EXAMPLES

A.1 Satellite running ROM bootstrap primary program wants to be loaded

<u>Step</u>	<u>Event</u>
1	Satellite primary program sends Request Program Message for secondary loader.
2	Host sends Memory Load with Transfer Address of secondary program image.
3	Satellite sends Request Program for operating system.
4	Host sends Memory Load without Transfer Address load 0.
5	Satellite sends Request Memory Load 1. Repeat Steps 4 and 5 for each successive load, eventually the last load (number n) is received with transfer address.
6	Satellite sends Request Memory Load for load number $n + 1$ and transfers to program.

Notes:

1. Last request is used only to ACK.
2. Satellite may request tertiary loader (after Step 2) which would then request the operating system. In this case the last Request Memory Load Message is used and the ACK is omitted.

A.2 Host wants to examine the memory of a satellite (secondary program running)

<u>Step</u>	<u>Event</u>
1	Satellite sends MOP Mode Running with dump function supported.
2	Host sends Request Memory Dump.
3	Satellite sends Memory Dump Data.

APPENDIX B

DDCMP MAINTENANCE MODE FORMAT

In the DDCMP maintenance mode, the link is dedicated to performing MOP functions. The DDCMP message provides a CRC block check on the MOP data, but does not provide any acknowledgment of receipt or sequence check. The DDCMP maintenance mode envelope is similar in format to a DDCMP numbered data message. See the DDCMP specification for details on this format and operation. All numeric values are shown in decimal representation. All bytes are 8 bits long. DDCMP maintenance messages have the following format:

SYNSEQ DLE COUNT Q S FILL FILL ADDR CRC1 MAINTMSG CRC2

Where:

SYNSEQ = the DDCMP sync sequence:
Synchronous links - 4 or more bytes; =150. (226 octal)
Asynchronous links - none

DLE = the maintenance message header start byte; =144. (220 octal)

COUNT = the count field (14 bits); number of bytes in the MAINTMSG data field.

Q = the QSYNC flag (1 bit); = 1.

S = the SELECT flag (1 bit); = 1.

FILL = a fill byte; = 0.

FILL = a fill byte; = 0.

ADDR = the tributary station address byte; for point-to-point operation = 1; for multipoint operation = 1-255.

CRC1 = the header CRC computed on DLE through ADDR (16-bit CRC-16).

MAINTMSG = the maintenance message data field. Contains the MOP messages (COUNT bytes in length).

CRC2 = the data CRC computed on the MAINTMSG data field only (16-bit CRC-16).

The MOP messages are sent within the MAINTMSG field within the DDCMP maintenance mode envelopes.

APPENDIX C
PROGRAM TYPES

In the Request Program Message subfield PGMTYPE (see Section 4.2.5) the generic type of program must be specified. This may be: 0 (or omitted) for the secondary loader; 1 for the tertiary loader; or 2 for the operating system. A brief description of the PDP-11 secondary and tertiary loaders is provided below.

Type 0 (secondary loader) - The secondary loader is sent in a single Memory Load with Transfer Address Message. The secondary loader must, therefore, fit into a single message block. It is loaded at location 6. Current secondary loaders are between 400 and 600 bytes in length, depending upon the device type used. They use the stack address set up by the primary loader. For current loaders this will be between 17400 and 17776. The secondary loader assigns its buffer space below the stack. The secondary loader accepts Memory Load with and without Transfer Address Messages. It is, therefore, capable of doing multi-block loads into absolute addresses without memory management. It requests a tertiary boot to be loaded.

Type 1 (tertiary loader) - The tertiary loader is loaded by the secondary in a multi-block load starting at location 10000. It will run with memory management on if it exists on the system. The tertiary loader moves itself to the top of physical memory and assigns its stack and buffer space just below itself. It is, therefore, capable of multi-block loads from location 0 up to its buffer address, usually the last 1-2K of physical memory. It requests the operating system to be loaded. The current tertiary loaders do not specify any specific operating system. The choice of system to send is established by prior agreement and by command at the host system.

APPENDIX D

REVISION HISTORY

This Appendix provides a list of changes that have been made to the Maintenance Operation Protocol Version 1.1 (dated January 16, 1976). It is intended to aid those who have implemented MOP Version 1.1 by providing a summary of the principal differences between Version 1.1 and 2.0.

1. Removed all references to MOP being used directly to non-adjacent systems over DECnet links. The NICE protocol performs MOP like functions within DECnet, using actual MOP protocol only over a physical link.
2. Decoupled MOP from DDCMP maintenance mode. The protocol specifies the requirements of a link control procedure to be used with MOP. DDCMP maintenance mode is one such procedure.
3. Deleted the following messages not needed in MOP. These are now handled by NICE. Code 20 - Examine data by name, Code 22 - clear data by name, Code 26 - Examined data by name.
4. Clarified the description of the fields in all MOP messages. Clarified and expanded the operational details of MOP and added a state table for operation.
5. Added a detailed description of the requirements of the data link control procedure to be used by MOP and a detailed description of the interface, set of commands and responses, to that procedure.
6. Added VAX and DECSYSTEM 10/20 information in message formats where necessary.
7. Changed message 8 - (Request MOP secondary mode program) to Request Program. It is now used to request all program loads in MOP, not just the secondary program. The STADDR field is removed and replaced by a MOP version number field. Added DTE20 to DEVTYPE field. PGMTYPE field is changed and SOFTID is added.
8. Changed message 10 - Request memory load to remove NODE and SOFTID, function now part of message 8 described above. Added an ERROR field to return any errors on previous load.
9. Changed message 12 - to MOP mode running from secondary mode running. Removed STADDR and replaced with MOP version number. Added a FEATURES field to describe the MOP features a node supports.

10. Added a new message code 20 - Parameter load with transfer addr. This message is used to load a parameter block before transferring control to a just loaded program.
11. Added a detailed description of primary mode and the operation of loading the secondary program.

digital

digital equipment corporation