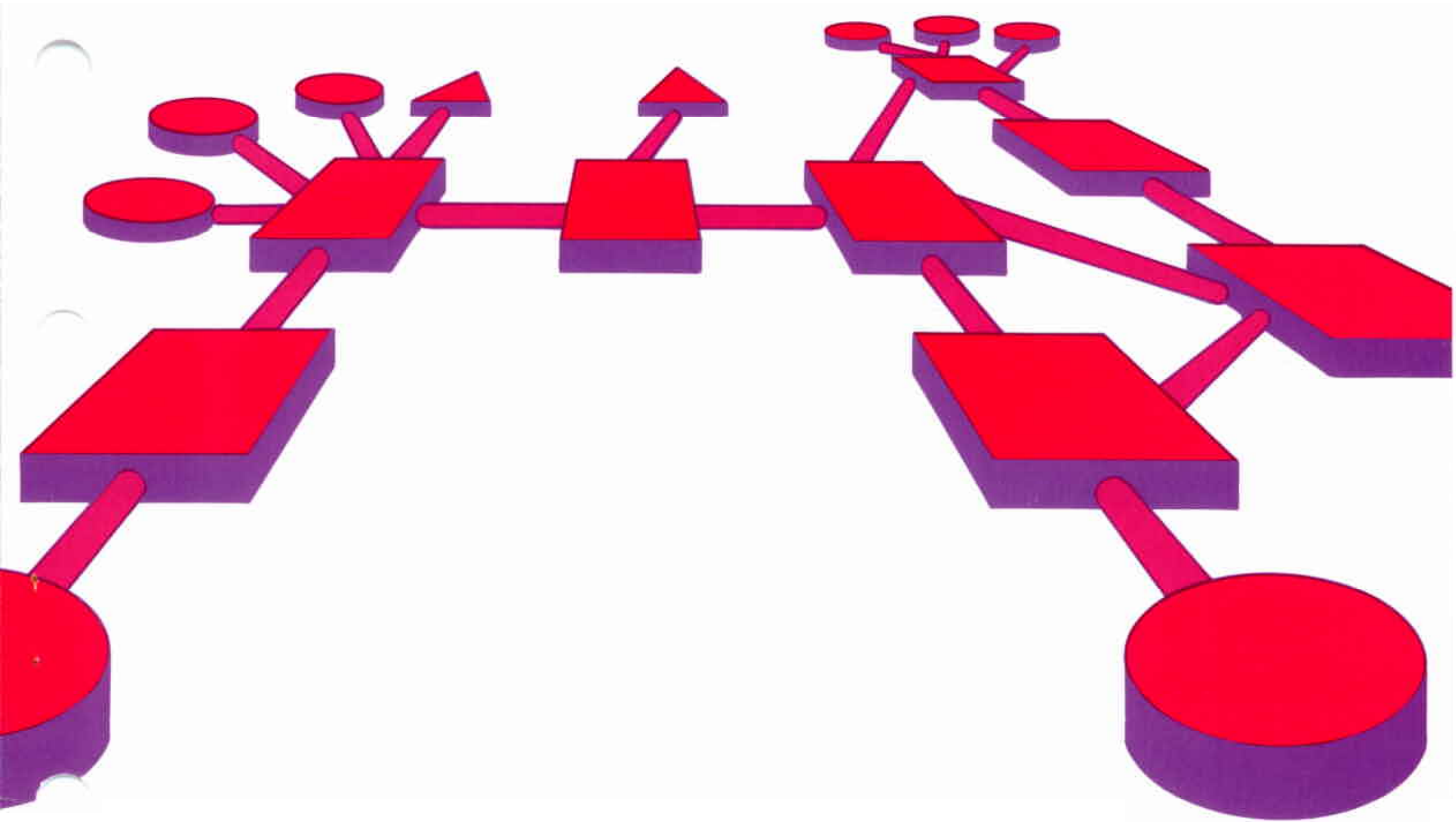
Order No. AA-K182A-TK

DECnet
DIGITAL NETWORK ARCHITECTURE

SESSION CONTROL
FUNCTIONAL SPECIFICATION

Version 1.0.0



DECnet
DIGITAL NETWORK ARCHITECTURE
(PHASE III)

SESSION CONTROL
FUNCTIONAL SPECIFICATION

Order No. AA-K182A-TK
Version 1.0.0

November 1980

This document describes the functions, interfaces, operation, and protocols of Session Control. Session Control models the DECnet implementation software that provides the system-dependent functions related to logical link operations.

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

digital equipment corporation · maynard, massachusetts

First Printing, November 1980

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.

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.

CONTENTS

FIGURES

CONTENTS (Cont.)

Page

TABLES

1.0 INTRODUCTION

This document describes the functions, interfaces, operation, and protocols of Session Control. As part of the DIGITAL Network Architecture (DNA), Session Control models the software that provides system-dependent functions related to creating, maintaining, and destroying logical links. A logical link is a virtual connection between two user-level software modules (end users) in the same node or in different nodes of the same network.

DNA is the model on which DECnet implementations are based. A DECnet network is a family of software modules, data bases, hardware components and facilities used to tie DIGITAL systems together for resource sharing, distributed computation or remote system communication.

DNA is a layered structure. Modules within each layer perform distinct functions. Modules within a single layer (but typically in different computer systems) communicate using specific protocols. Modules in different layers (but typically in the same computer system) interface using subroutine calls or some other system-dependent method. In this document, interfaces are described in terms of calls to subroutines.

This specification describes Phase III Session Control. In Phase II DNA, Session Control was part of the Network Services layer. Phase III DNA, however, logically separates Session Control from Network Services and defines the interface between the newly distinct layers.

A glossary at the end of this document defines many Session Control terms.

This document assumes that the reader is familiar with computer communications and DECnet. The implementors of DECnet systems compose the primary audience for this document, which may also be of interest to anyone wishing to know details of DECnet structure. The other DNA Phase III functional specifications are:

DNA Data Access Protocol (DAP) Functional Specification, Version 5.4.0, Order No. AA-D601B-TC

DNA Digital Data Communications Message Protocol (DDCMP) Functional Specification, Version 4.2.0, Order No. AA-D599B-TC

DNA Maintenance Operations Protocol (MOP) Functional Specification, Version 2.2.0, Order No. AA-D602B-TC

DNA Transport Functional Specification, Version 1.3.0, Order No. AA-J059A-TK

DNA Network Management Functional Specification, Version 2.0.0, Order No. AA-J060A-TK

DNA Network Services (NSP) Functional Specification, Version 3.2.0, Order No. AA-D600B-TC

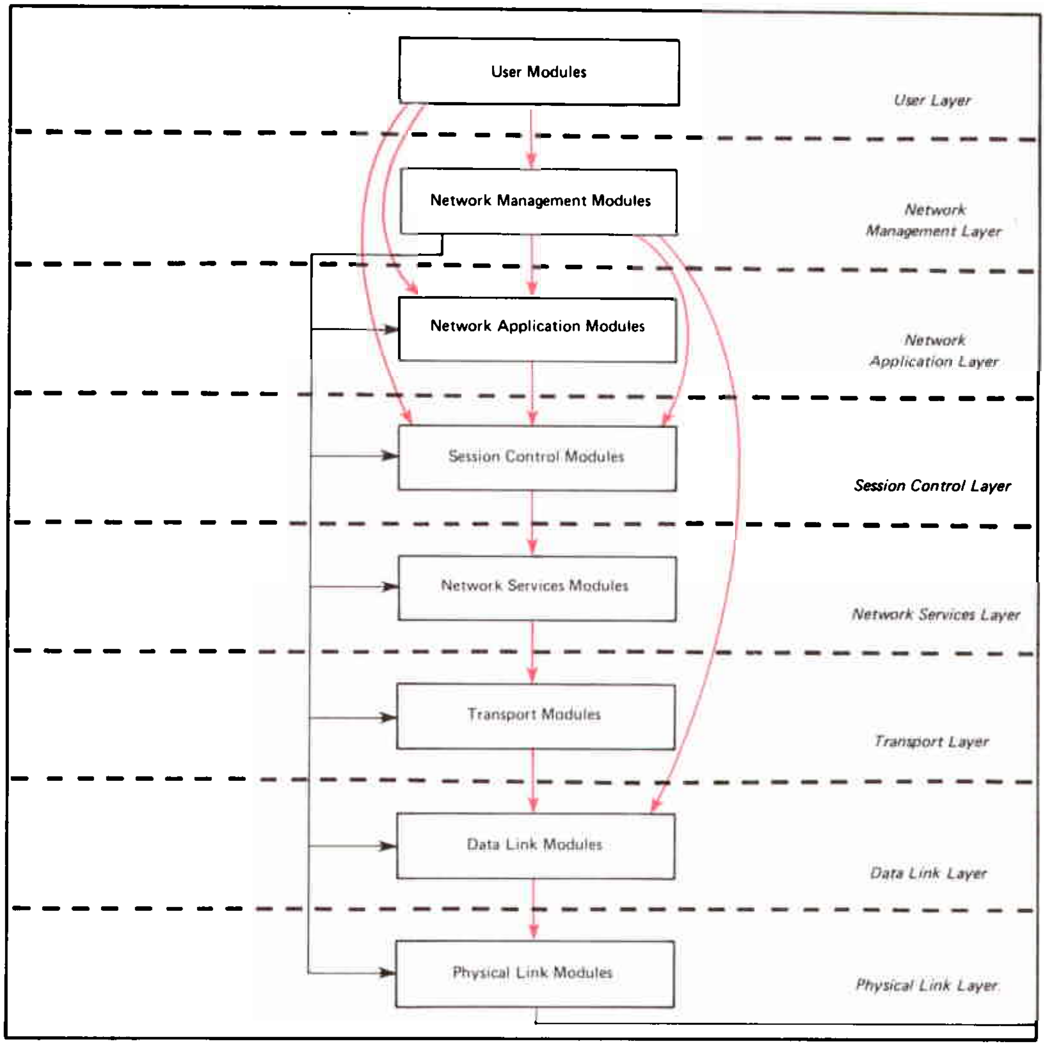
The DNA General Description (Order No. AA-H202B-TK) provides an overview of the network architecture and an introduction to each of the functional specifications.

1.1 Relation to DIGITAL Network Architecture

Figure 1 shows the relationship of Session Control to the DNA hierarchy. Each layer in DNA consists of functional modules and protocols.

Generally, modules provide services to the next higher layer and use the services of the next lower layer. The service relationship is reflected in the way the interfaces are modeled as calls to subroutines. Note, however, that the Network Management layer directly interfaces with each of the lower layers. Also, all the layers above Session Control interface directly with it.

Modules of the same type in the same layer communicate with each other to provide their services. The rules concerning this communication and the messages required constitute the protocol for those modules. Typically, these messages are exchanged between equivalent modules in different nodes. However, equivalent modules within the same node can also exchange messages.



Black arrows show direct access for control and examination of parameters, counters, etc. Red arrows show interfaces between layers for normal user operations such as file access, down-line load, up-line dump, end-to-end looping, and logical link usage.

Figure 1 Relation of Session Control to DNA

The functional components of Session Control are as follows:

User layer. The highest layer, it supports user services and programs.

Network Management layer. This layer is the only one that has direct access to each lower layer for control purposes. Modules in this layer provide user control over and access to network parameters and counters. These modules also perform up-line dumping, down-line loading, and testing functions.

Network Application layer. Modules in this layer support network functions, such as remote file access and file transfer, used by the two higher layers.

Session Control layer. This layer defines the system-dependent aspects of logical-link communication, which allows messages to be sent from one node to another in a network.

Network Services layer. This layer defines the system-independent aspects of logical link communication.

Transport layer. Modules in this layer route messages, called packets, between source and destination nodes.

Data Link layer. This layer defines the protocol concerning data integrity and physical channel management.

Physical Link layer. This layer encompasses a part of the device driver for each communications device plus the communications hardware itself. The hardware includes interface devices, modems, and the communications lines.

1.2 Functional Description

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

An end user communicates directly with Session Control to request logical link service. The form of this communication, which is the end user interface, varies from one operating system to another. However, the functions that this interface provides an end user, regardless of the local system, include:

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

These functions are similar to those that Session Control requests from Network Services. In response to requests from end users, Session Control makes parallel requests to Network Services. Unlike the end user interface, however, the Network Services interface is not system-dependent. It is specified in detail (Section 2.1) and standard for all implementations of DNA.

Session Control facilitates Network Services by performing the following system-dependent tasks:

- **Mapping node names to node addresses.** A Session Control module maintains a node name mapping table that defines the correspondence between a node name and either a node address or a channel number. (The channel number is used only for loopback testing.) The table enables the Session Control module to select the destination node address or channel number for outgoing connect requests to Network Services. For incoming connect requests from Network Services, the Session Control module uses the table to identify the node from which the request originated.

- **Identifying end users.** A Session Control module executes a system-dependent algorithm to determine if an existing end user corresponds to the destination end user specified in an incoming connect request. It also performs additional functions related to passing a connect request to an existing end user. See Section 5.2.

- **Validating incoming connect requests.** A Session Control module uses access control information included in an incoming connect request to perform system-dependent validation functions.

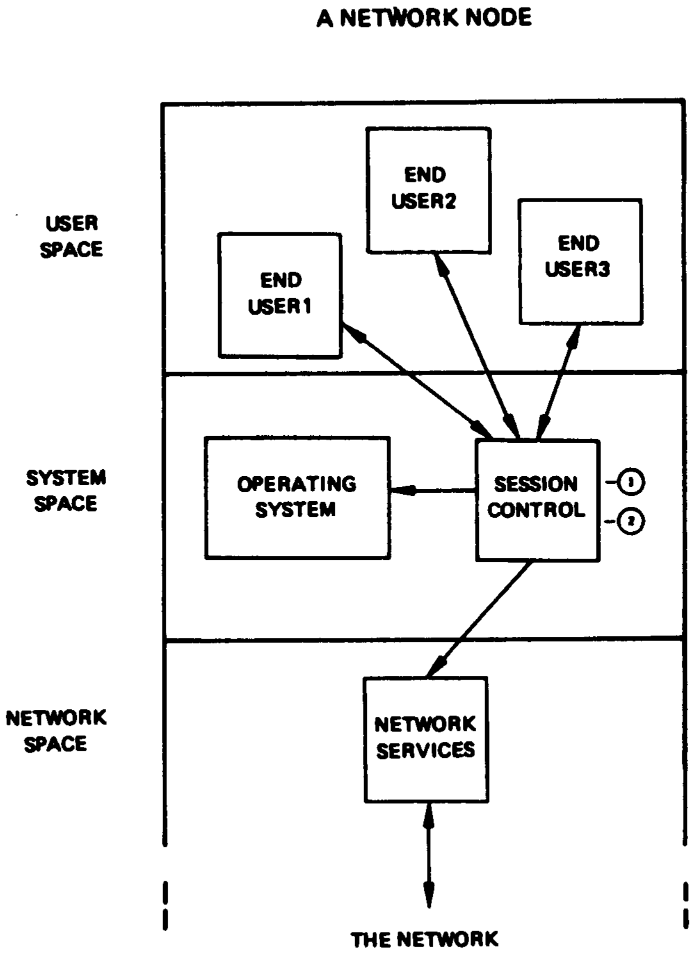
Section 5.0, which discusses Session Control operation in detail, describes how Session Control performs these functions.

1.3 A Session Control Model

The model defined by the Session Control layer provides an interface to Network Services for end users that reside in a "user" space created by an operating system. It also provides an interface that can be used by Network Management. The relation to, or interface with, the operating system is system-dependent.

Unlike modules that implement other DNA layers, a Session Control module is not self-contained -- it cannot be specified in isolation from non-DECnet modules. Session Control represents the point (or one of the points) at which DECnet is integrated with an operating system. Figure 2 represents a model of Session Control operating within a network node.

As the figure shows, Session Control requires two data bases, a node-name mapping data base and a data base that contains the state of Session Control and optional default connection timers. Any other data bases or components required by a Session Control module are system-dependent.



COMMENTS

- End users are User, Network Application, and Network Management* modules.

- Session Control is an interface to Network Services for end users. It functions in conjunction with the operating system. ① and ② are data bases used by Session Control.

- Network Services provides logical link service to Session Control. Its functions are not dependent on individual operating systems.

* Network Management interfaces with Session Control in two ways: (1) to obtain logical link service and (2) to monitor and control Session Control operations. See Figure 1.

Figure 2 A Session Control Model

2.0 SESSION CONTROL INTERFACES

Session Control maintains four interfaces between itself and its environment:

- Interface to Network Services
- Interface to end users
- Interface to Network Management
- Interface to the resident operating system

This section describes the functions of the first three interfaces. The Network Services interface is described first because it is more precisely defined than the end user interface. Furthermore, the end user interface is basically derived from the Network Services interface. The operating system interface is entirely system-dependent and therefore cannot be modeled in this specification.

The Network Services and Network Management interface functions are expressed as calls to subroutines in the following format:

```
FUNCTION (input;  output)
```

In general, there are two types of subroutines:

1. Those performing a function that is completed immediately.
2. Those queuing a buffer for transmitting or receiving data.

For buffer-queuing calls, additional calls are defined to allow polling to obtain "buffer returned" notifications. A "buffer" argument denotes a system-dependent buffer descriptor that contains location and length information. A "port id" is a system-dependent number identifying a port. (See the following section for a definition of a port.) Although not described in the following functions, an invalid port identifier causes an error.

NOTE

An implementor is not required to code the interfaces as subroutines. The calls specify functions only.

2.1 The Network Services Interface

The Network Services interface provides Session Control with logical link service. Using this service, Session Control can create one or more logical links to other Session Control modules in the same network.

Network Services uses individual databases called ports to manage the logical links it creates. Network Services and Session Control modules refer to logical links in terms of their associated ports. As a result, many of the functions described below include references to port ids and port states. See Appendix A for a definition of ports and other related terms.

In the following description of the Network Services interface, the terms "source" and "destination" distinguish the requestor of a function from the receiver of the request. The source and destination can be within a single Session Control module or in two separate Session Control modules. Thus, at a single node, a Session Control module can communicate with itself via a logical link; between two nodes, two Session Control modules can communicate with each other via a logical link.

The calls, described by function, are as follows:

STATUS (; status)

returns:	Network Services is halted.
	Network Services is running; the implementation's minimum receive buffer size is returned

This function reads the status of Network Services and obtains a minimum receive buffer size if Network Services is running. This is the one Network Services interface function that does not involve the use of a logical link.

OPEN (source, buffer ; return)

source:	a 16-bit buffer to contain the logical link requestor node address when this node receives a connect request
buffer:	a buffer to receive incoming connect data
returns:	port allocated; port identifier returned
	port not allocated; insufficient resources
	port not allocated; Network Services halted

This function allocates a port in Network Services for receiving a logical link connect request. The source variable and the buffer receive the node address of the requesting node and the incoming connect data, respectively. When the port state indicates an incoming connect request is received, Session Control receives the source node address in the source variable and the incoming data in the buffer. Appendix A describes port states.

CLOSE (port id)

This function deallocates a port. When a port is closed, Network Services immediately returns all transmit and receive buffers to Session Control (see DATA-XMT and DATA-RCV calls). Once a port is closed, its associated port identifier is undefined. Any subsequent call issued with such a port identifier results in an error return.

Session Control may close a port at any time regardless of the port's state. However, doing so may create ambiguities for the Session Control module at the other end of the logical link (see Appendix A).

CONFIDENCE (port id; confidence)

returns: network probably connected

network probably disconnected

This function obtains Network Services' assessment of connectivity. Network Services periodically tests a logical link once it is formed to ascertain if the network supporting the logical link is connected. The testing determines Network Services' assessment of connectivity. The assessment is not guaranteed to be accurate.

Session Control may issue this call to determine when to disconnect a link on behalf of a program at the user level.

STATE (port id; state)

returns: the state of the associated logical link

This function obtains the state of a port that is in any state other than CLOSED. Appendix A describes the port states.

Because Session Control's operation is not necessarily synchronized with that of Network Services, it is possible that this call will not detect every state transition. This is especially true for state transitions that occur very quickly. However, this is not a problem because the intervening undetected states can be logically deduced as described in Appendix A.

CONNECT-XMT (destination, channel, buffer; return)

destination: destination node address

channel: an internal Network Services mechanism selector used to enable loop testing. Channel is either unspecified (for normal use) or a system-dependent line number representing the line Network Services is to use for its messages establishing this logical link (for Network Management loop tests)

buffer: a buffer containing up to 200 bytes of data to be transmitted.

returns: port allocated; port id returned

port not allocated; insufficient resources

port not allocated; Network Services halted

This function allocates a port and requests a logical link connection. After a logical link has been successfully formed, Session Control can put a load on a particular physical link for loop test purposes provided that the channel argument specified the physical link. This enables testing of the physical link and all of the DECnet modules from Session Control or higher layers by sending and receiving data on the resulting logical link. For normal use the channel argument is set to "unspecified."

CONNECT-STATUS (port id, buffer; return)

returns: connect request accepted by destination; port in RUNNING state

connect request rejected by destination; port in REJECTED state

port in neither RUNNING nor REJECTED state

This function obtains accept or reject data returned as a result of a previous connect request. If the return is one of the first two, Session Control receives any available accept or reject data from Network Services. Once this is done, a Network Services implementation may discard its copy of the accept or reject data so that a subsequent connect status function would not return data.

In cases where state transitions occur very rapidly, Session Control may not be able to perceive some intervening states. Consequently, this call may not be accepted (see Appendix A).

Accept data will be lost if the rapid state transitions end with a transition to the DISCONNECT-NOTIFICATION state and this call was never executed in the RUNNING state. No data is lost otherwise.

If the connect request is accepted, up to 16 bytes of accept data may be returned in the buffer. If the connect request was rejected, up to 18 bytes of reject data may be returned in the buffer (see the ACCEPT and REJECT calls).

ACCEPT (port id, buffer; return)

returns: link accepted

port not in CONNECT-RECEIVED state

This function accepts a connect request from a remote Session Control module. The call supplies a buffer containing up to 16 bytes of accept data.

REJECT (port id, buffer; return)

returns: link rejected

port not in CONNECT-RECEIVED state

This function rejects a connect request from a Session Control module. The call supplies a buffer containing up to 18 bytes of reject data.

DISCONNECT-XMT (port id, buffer; return)

returns: call accepted

call rejected; port not in RUNNING state

This function requests the disconnection of a logical link that is in the RUNNING state. The call supplies a buffer containing up to 18 bytes of disconnect data.

The remote Session Control module receives any data transmitted by the disconnecting Session Control module prior to this call. Session Control disconnects a link when it has no more data to send and wants to insure that the link will be properly disconnected, not aborted.

ABORT-XMT (port id, buffer; return)

returns: call accepted

call rejected; port not in RUNNING state

This function requests the immediate disconnection of a logical link that is in the RUNNING state. The remote Session Control module may not have received all previously transmitted data before receiving the abort notification.

The call supplies a buffer containing up to 18 bytes of abort data.

DISCONNECT-RCV (port id, buffer; return)

returns: disconnect data available

no disconnect data available

port not in DISCONNECT-NOTIFICATION state

This function receives disconnect data returned to the local Session Control module as a result of a DISCONNECT-XMT or ABORT-XMT call from the remote Session Control module. Session Control detects a logical link disconnection or an abort when a STATE call returns a DISCONNECT-NOTIFICATION. Up to 18 bytes of data may be returned in the buffer.

DATA-XMT (port id, buffer, xmtflag; return)

xmtflag: a flag indicating whether the last byte in the buffer is the last byte of a Session Control message. Its value is one of:

- end-of-message
- not-end-of-message

returns: buffer queued

buffer not queued; insufficient resources

port not in RUNNING state

This function queues a transmit buffer to a port for transmitting normal data on a logical link. Network Services refuses to queue the buffer either if it lacks the resources to do so or if the port is not in the RUNNING state.

XMT-POLL (port id; return)

returns: no transmit complete

transmit complete; buffer descriptor returned

This function causes Network Services to notify Session Control whether or not a transmission has completed. If a transmission completed, Network Services returns a transmit buffer to Session Control.

DATA-RCV (port id, buffer, rcvflag; return)

rcvflag: a flag indicating whether data truncation is allowed. It may have either of the following values:

- no truncation allowed
- truncation allowed

returns: buffer queued

buffer not queued; insufficient resources

buffer not queued; buffer too small and no truncation was specified in rcvflag

port not in RUNNING or DISCONNECT-INITIATE state

This function queues a receive buffer to a port to receive normal data. A "buffer too small" return indicates the buffer size is smaller than the minimum receive buffer, NSPbuf (see STATUS).

Session Control may provide a buffer to a port in the DISCONNECT-INITIATE state to avoid a Session Control deadlock in which each end of the logical link is in the DISCONNECT-INITIATE state. However, this is an implementation-dependent issue. Refer to Appendix A.

RCV-POLL (port id; return)

returns: no buffer returned (Either no receive buffers are queued to the port or there is no receive data available.)

buffer returned; no data lost, end-of-message

buffer returned; data lost, end-of-message

buffer returned; no data lost, not end-of-message

buffer returned; data lost, not end-of-message

buffer returned empty; port not in RUNNING, DISCONNECT-INITIATE, DISCONNECT-COMPLETE, or DISCONNECT-NOTIFICATION states

This function obtains a "receive complete" notification for a receive buffer previously queued via a DATA-RCV call. Network Services returns receive buffers along with buffer descriptors to Session Control in the order in which data was placed in them. (See the Network Services Functional Specification for further details.)

INTERRUPT-XMT (port id, buffer; return)

returns: data accepted

data not accepted

port not in RUNNING state

This function sends up to 16 bytes of high priority data to the destination Session Control module. The data has no sequential relationship to normal data transferred on a logical link. Network Services may refuse a request to send interrupt data if it is unable to queue the data internally. The buffer may be up to 16 bytes long.

INTERRUPT-RCV (port id, buffer; return)

returns: data returned

no data returned

port not in running state

This function obtains available interrupt data. Interrupt data is delivered in the order transmitted by the INTERRUPT-XMT function. Interrupt data has no sequential relationship to normal data transferred on a logical link.

2.2 The End User Interface

Because the form of Session Control's interface to end users is system-dependent, this specification does not use subroutine calls to model the functions of this interface as it does for the Network Services and Network Management interfaces. In general, however, the functions available at the end user interface are similar to those provided by the Network Services interface.

The remainder of this section describes how the end user interface can differ in detail from the Network Services interface. Specifically, the length of some fields may be limited to a value less than that supported by Network Services to allow for the insertion of Session Control header information. In addition, Session Control may accept additional flags or variables at the end user interface. Session Control functions differ from Network Services interface functions in one of two ways: the maximum length of data allowed is different, or additional arguments are possible. Specific instances of the ways in which Session Control and Network interface functions differ are described below.

- **Node name mapping.** The end user refers to nodes by name rather than by address. On outgoing connections, Session Control translates the destination node name into a node address or, in the case of loopback testing, into a channel number. On incoming connections, Session Control translates the source node address into a node name if the node address is in the node name mapping table. Otherwise, no remote node identification is passed to the user.

- **Opening a port.** The function equivalent to opening a Network Services port to receive an incoming request may allow an end user to supply a destination end user name.

- **Connect data.** In a connect request, an end user can supply up to 16 bytes of connect data. This value is less than the amount of connect data that Network Services will accept because Session Control must use some of the connect data it gives to Network Services to contain destination and source end user names as well as access control information. The additional arguments that an end user may pass to Session Control in a connect request are as follows. Section 5.0 explains how these arguments may be used.

  - Destination end user name

  - Source end user name

  - Access control information

  - One or more "error tolerance" variables

- **Incoming connect requests.** When Session Control delivers an incoming connect request to an end user, it may deliver the destination end user name, the source end user name, and the received access control information, as well as the end user connect data. The maximum amount of end user connect data that Session Control can deliver is 16 bytes.

- Connect reject or disconnect request data. The maximum amount of data that an end user can include with a connect reject or disconnect request is 16 bytes. Session Control uses the additional 2 bytes available from Network Services for a disconnect or reject reason code. Similarly, the maximum amount of reject or disconnect data that can be delivered to the end user requesting a connection is 16 bytes.

- Error tolerance arguments. The function that accepts an incoming connect request may allow the end user to supply one or more "error tolerance" arguments.

2.3 The Network Management Interface

Using its interface to Session Control, Network Management can exert control in three ways:

1. It can control and monitor the state of Session Control.

2. It can modify the node name mapping table.

3. It can optionally set and read two Session Control parameters used in handling outgoing and incoming connect requests.

In addition, the Network Management Event Logger can monitor the two events logged by Session Control. These events are described in Section 2.3.4.

2.3.1 Session Control States - All Session Control modules support at least two operational states: OFF and ON. Furthermore, a Session Control module may support either or both of two additional states: RESTRICTED and SHUT. Table 1 below defines these states.

Table 1
Session Control States

State	Description
OFF	Session Control is halted and no logical links are operational.
ON	Session Control is running and able to provide logical link service.
SHUT	Session Control is running and supporting existing logical links but it will refuse any new connect requests, either incoming or outgoing.
RESTRICTED	Session Control is running and supporting existing logical links. It will attempt to initiate new logical links on request (CONNECT-XMT). However, it will not accept any new logical links (ACCEPT) unless the requesting user has sufficient privilege. This determination is system-dependent.

Figure 3 below describes the possible transitions from one Session Control state to another. A single arrow represents a transition caused by a Network Management request. A double arrow represents a transition caused either by a Network Management request or by an internal Session Control transition.

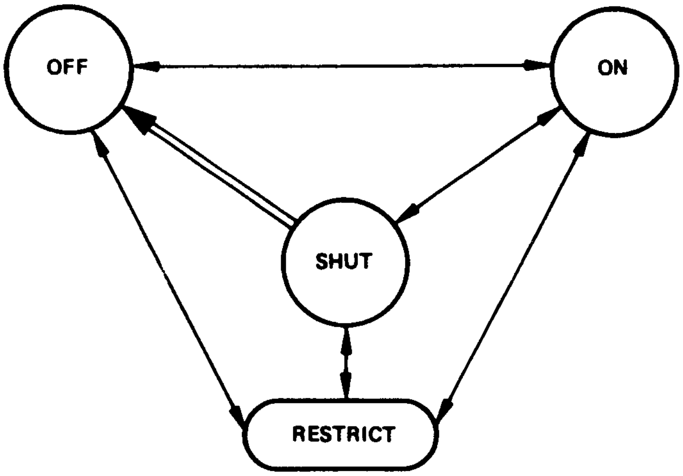


Figure 3 Session Control State Diagram

Because a Session Control module is usually implemented as part of, or closely coupled to, an operating system, its initialization is often coupled to the operating system's initialization and is not modeled in this specification. The state of a Session Control module following such initialization is system-dependent.

The calls described below represent Network Management requests to change the state of Session Control. A Session Control module has the option of not supporting any of these calls while it is in the ON state after initialization.

OFF (; return)

returns: success

failure

This function halts Session Control by changing it to the OFF state.

ON (; return)

returns: success

failure

This function enables Session Control to provide logical link service by changing its state to ON.

SHUT (; return)

returns: success

failure

This function changes Session Control's state to SHUT. If a Session Control module accepts the SHUT call, then it moves automatically to the OFF state after the last logical link has disconnected.

RESTRICTED (; return)

returns: success

failure

This function changes Session Control's state to RESTRICTED.

STATUS (; state)

state: OFF

ON

SHUT

RESTRICTED

This function reports the current state of Session Control to Network Management.

2.3.2 Defining the Node Name Mapping Table - Network Management can define entries in the node name mapping table. Each table entry associates a node name with a node address or channel number. Section 4.0 explains how Session Control uses this table. The maximum length of the table is system-dependent, but restrictions that do apply include the following:

- No node name can be in the table more than once.
- No node address can be in the table more than once.
- No channel number can be in the table more than once.

Network Management uses the following functions to access Session Control's node name mapping table:

ADD-TO-TABLE (name, number; return)

name: a node name

number: a node address or channel number

returns: success

failure; the node name, node address, or channel number is already in the table

This function adds an entry to the table. The way an implementation distinguishes a node address from a channel number is system-dependent.

REMOVE-FROM-TABLE (name; return)

name:	a node name
returns:	success
	failure; the node name is not in the table.

This function removes an entry from the node name mapping table.

CHANGE-TABLE (name, number; return)

name:	a node name
number	a node address or channel number
returns:	success
	failure; the node specified is not in the table
	failure; the node address or channel number is already in the table

This function modifies an existing table entry. Network Management can use the function to change the node address or channel number associated with a node name.

READ-TABLE (number; return)

number:	a node address or a channel number
returns:	the node name mapped to the specified node address or channel number
	failure; the table does not contain the node address or channel number specified

2.3.3 **Default Connection Timers** - A Session Control module can optionally have two default values for connection timers: one for incoming connect requests and one for outgoing connect requests. At the option of Session Control, Network Management may set and read the default timer values. Network Management can therefore perform the following functions if Session Control uses the default timer values and also allows Network Management to control them.

SET-OUTGOING-TIMER (time;)

time:	the default outgoing timer in seconds

This function sets a default timer value for outgoing connect requests. Session Control starts a timer when it issues a connect request to Network Services. If the port that Network Services opens for the request does not enter the NO-RESOURCES, NO-COMMUNICATION, REJECTED, or RUNNING state (Appendix A) before the timer expires, Session Control issues a timeout rejection to the end user requesting the connection and Session Control closes the port. Such timer processing is system-dependent.

If the end user includes an "error tolerance" argument in the connect request, that argument may override the default timer set by Network Management.

READ-OUTGOING-TIMER (; time)

time: the current default out-going timer in seconds

This function reports to Network Management the default value currently set for the outgoing timer.

SET-INCOMING-TIMER (time;)

time: the default incoming timer in seconds

This function allows Network Management to set Session Control's default incoming timer value. When Session Control receives an incoming connect request, it can optionally start an incoming timer. If the destination end user does not accept or reject the connect request before the timer expires, Session Control issues a connect reject to Network Services, including two bytes of reject data as a reject code (see Section 6.0). Such timer processing is system-dependent.

READ-INCOMING-TIMER (; time)

time: the current default incoming timer in seconds

This function reports to Network Management the default value currently set for the incoming timer.

2.3.4 **Session Control Events** - Network Management can read and clear Session Control's internal event log, which is a queue of event records. The two events logged by Session Control are:

- **Node State Change.** Session Control has changed its state. The new state is logged.

- **Access Control Failure.** Session Control has received a connect request from Network Services which it rejects for access control reasons. The event log records the source node address and all connect request data except the user connect data and password.

3.0 SESSION CONTROL MESSAGES

Session Control's message protocol defines messages sent on a logical link as connect data, reject data, and disconnect data. These Session Control messages are described in Sections 3.2 and 3.3 according to the notation described below. This notation is the same as that used in other DIGITAL Network Architecture specifications.

3.1 Message Format Notation

The following notation is used to describe the messages contained herein:

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 (octets)

2. A number followed by a "B" meaning number of bits

3. The letters "EX-n" meaning extensible field. n is a number that specifies the maximum length of 8-bit bytes in the protocol before interpretation, as described below. If no number is specified, the current maximum length is 1 byte. Extensible fields are variable in length consisting of 8-bit bytes. The high-order bit of each byte indicates whether the next byte is part of the same field. A 1 means the next byte is part of this field. A 0 indicates the next byte is the last byte. The low-order 7-bits of each byte are information bits. Extensible fields can be binary or bit-map. If they are binary, then 7-bits from each byte are concatenated into a single binary field. If they are bit-map, then 7-bits from each byte are used independently or in groups as information bits.

   NOTE

   The bit definitions define the information bits after removing the extension bits and compressing the bytes.

4. The letters I-n indicate an image field; n is a number specifying the maximum length of 8-bit bytes in the image. A 1-byte count of the length of the remainder of the field precedes the image. Image fields are variable in length and may be null (count=0). All 8 bits of each byte are information bits. The meaning and interpretation of each image field is defined with that specific field.

5. In addition, the notation *-19 means that the field is not more than 19 bytes long and is defined further along in the message description.

CODING the representation type used as follows:

A	7-bit ASCII
B	binary
BM	bit map (each bit or group of bits has independent meaning)
C	constant
null	interpretation data dependent

CONVENTIONS

1. If both the length and coding are omitted, the field represents a generic field with a number of subfields specified in the description.

2. Any bit or field that is stated to be "reserved" must be zero unless otherwise specified.

3. All numeric values in this section are decimal unless otherwise noted.

4. Bits are numbered with bit 0 on the right (low-order, least-significant bit) and bit 7 on the left (high-order, most-significant bit). For convenience, when the graphic form of a 2-byte field is given, it will be shown converted to a 16-bit word. When a subfield of a message field contains more than one bit, it should be considered a binary value.

5. Unless otherwise specified, the numbers that appear at the top of the message formats represent bit positions.

6. Brackets around a field indicate its presence in a message is optional.

3.2 Connect Data

A Connect Data message has the following form:

DSTNAME SRCNAME MENUVER RQSTRID PASSWRD ACCOUNT USRDATA

DSTNAME (*-19) : Is the destination end user name (see below)

SRCNAME (*-19) : Is the source end user name (see below)

MENUVER (EX) : BM Is the field format and version information:

Bit 0=1 RQSTRID, PASSWRD, ACCOUNT fields included

Bit 1=1 USRDATA field included

Bits 2-4=0 Reserved

Bits 5,6 Is the Session Control version:

0 = Version 1.0
1-3 = Reserved

RQSTRID (I-39) : B Is the source user identification for access verification

PASSWRD (I-39) : B Is the access verification password

ACCOUNT (I-39) : B Is the link or service account data

USRDATA (I-16) : B Is the end user connect data

An end user name has the following form:

FORMAT OBJTYPE NAME

FORMAT (1) : B Is the end user name format:

0 = Format 0
1 = Format 1
2 = Format 2
3-255 = Reserved

OBJTYPE (1) : B Is the object type. It may not be 0 for format 0; it must be 0 for formats 1 and 2.

For format 0:

NAME (null) Is not present

For format 1:

NAME (I-16) : B Is an end user descriptor

For format 2:

NAME Is GRPCODE, USRCODE, DESCRPT

GRPCODE (2) : B Is a group code

USRCODE (2) : B Is a user code

DESCRPT (I-12) : B Is an end user descriptor

3.3 Reject and Disconnect Data

A Reject or Disconnect message has the following form:

REASON DATA-CTL

REASON (2) : B a reason code (see Section 6.2)

DATA-CTL (I-16) : B user data. The length of this field is ascertained from the total length of reject or disconnect data received from Network Services. It may be null.

4.0 DATA BASES

The Session Control model includes two data bases which are described in the following section:

- The node name mapping table
- The Session Control state and default connection timer table

4.1 The Node Name Mapping Table

The node name mapping table defines the correspondence between node names and node addresses or channel numbers. Session Control uses the table to identify either the destination node named in an outgoing connect request or the source node that has sent an incoming connect request.

The table should contain an entry for:

- Every destination node that an end user will specify to Session Control in an outgoing connect request
- For every node likely to send a connect request to Session Control. Network Management defines the table entries and can change, remove, and read existing entries. These functions are described in Section 2.3.2.

Section 5.0 describes the mapping table's role in processing incoming and outgoing connect requests.

4.2 The Session Control State and Default Timer Table

This table is an internal data base that contains the current state of Session Control and optionally contains default incoming and outgoing connection timers. Network Management can access this table by means of the interface functions described in Sections 2.3.1 and 2.3.3. Section 5.0 explains how Session Control uses the default connection timers if it chooses to implement them.

5.0 SESSION CONTROL OPERATION

Session Control performs the following basic operations:

- Requests logical link connections on behalf of end users.
- Receives connect requests addressed to end users.
- Sends and receives logical link data.
- Disconnects and aborts logical links.
- Optionally monitors logical links.

This section describes how the Session Control model performs these operations.

5.1 Requesting a Logical Link Connection

Session Control performs four tasks when it receives a logical link connection request from an end user (the source):

1. Identifies the destination node address or channel number. It may use a node name mapping table to identify the destination node address or channel number for Network Services when the end user specifies a node name. A Session Control module also has the option of maintaining an "alias" node name mapping table, which maps an additional set of node names to those entered in the mandatory table. The existence, maintenance, and use (except as just defined) of either a node name mapping table or an alias mapping table is system-dependent.

2. Formats connect data to be passed to Network Services. Section 3.2 describes how Session Control formats a connect data message. How Session Control obtains the destination and source end user names, the access control information, and the end user connect data that make up the message is system-dependent.

3. Issues a connect request to Network Services. If sufficient resources are available, Network Services opens a port for the connection requested by Session Control. The request is unsuccessful if Network Services then places the port in the NO-RESOURCES, NO-COMMUNICATION, or REJECTED state (Appendix A). In this case, Session Control returns information to the source end user indicating that the connect request has failed.

   The extent to which Session Control gives the end user specific information regarding the failure is system-dependent. The information that Session Control can provide is based on either the port's state if it was NO-RESOURCES or NO-COMMUNICATION or the first two bytes of reject data if the port's state was REJECTED. In the latter case, Session Control must make user reject data, if any, available to the source end user. If Network Services places the port in the RUNNING state, Session Control then indicates to the source end user that the connect request succeeded. Session Control must also make accept data, if any, available to the end user.

4. **Optionally starts an outgoing connection timer.** This timer has the following effect on Session Control's operation. If Network Services does not place the port associated with the request in the NO-RESOURCES, NO-COMMUNICATION, REJECTED, or RUNNING state before the timer expires, Session Control returns a timeout rejection to the source end user and closes the port. As stated previously, the outgoing connection timer is optional and timer processing is system-dependent. If a Session Control module chooses to use the default timer and supports the SET-OUTGOING-TIMER function described in Section 2.3.3, the default timer value is that most recently set by Network Management. Session Control may also allow the end user to provide its own timer value in an "error tolerance" argument included in its connect request. Such an argument overrides the default value set by Network Management.

5.2 Receiving a Connect Request

Session Control maintains one or more ports in the OPEN state in order to detect incoming connect requests. (See the Network Services interface OPEN function, described in Section 2.1.) Network Services notifies Session Control of an incoming request by changing the state of one of these ports to CONNECT-RECEIVED. When Session Control detects the port's new state (using the STATE function, Section 2.1), it performs the following six tasks:

1. **Obtains data for the incoming connect request.** It obtains the destination end user name, source end user name, access control information, and end user connect data for the incoming connect request. Session Control obtains this information by parsing the connect data received from Network Services, which is formatted according to the message protocol described in Section 3.2.

2. **Validates access control information.** It validates the access control information in a system-dependent manner. For example, one Session Control module may log the logical link onto the local system. Another Session Control module may perform no validation of its own and pass the information directly to the end user for validation. These are only examples; there are no restrictions on how to process access control information.

3. **Identifies, creates, or activates the destination end user.** It either maps the destination end user name to an existing end user or creates or activates a destination end user to receive the connect request. A destination end user name is in one of two forms:

   - A system-dependent name, indicated by a zero (0) object type. The name itself consists of a sequence of bytes interpreted in a system-dependent way by the destination Session Control module.

   - A system-independent service, indicated by a non-zero object type (a one-byte number) and no name.

To identify the destination end user, Session Control executes a system-dependent algorithm. The destination end user name and/or the access control information received in the connect data may be inputs to the algorithm. Executing the algorithm may result in one of the following outcomes:

- The destination end user exists and can receive a new connect request.
- The destination end user exists but cannot receive a new connect request (for example, its "mailbox" queue may be full or it may already be using its maximum number of "network logical units").
- The destination end user does not exist. In this case, Session Control can optionally create or activate an end user to receive the connect request. How Session Control activates or creates the end user is system-dependent.

If Session Control cannot identify, create, or activate a destination end user, it issues a REJECT to Network Services, including two bytes of reject data (see Section 6.0).

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

5. **Delivers the incoming connect request to the end user.** If Session Control has identified, created, or activated a destination end user, it delivers the incoming connect request to the end user in a system-dependent way. Session Control also delivers information that identifies the source node of the connect request. The form that this information takes is system-dependent, but it may be one of the following: the source node's name, the node identification "unknown," or a channel number.

   If the end user accepts the connection, Session Control issues an ACCEPT to Network Services and passes along any user accept data (up to 16 bytes). If the end user rejects the connection, Session Control issues a REJECT to Network Services, including a two-byte reject code (see Section 6.0) and up to 16 bytes of reject data, if the destination end user supplied any.

6. **Optionally starts an incoming connection timer.** It can optionally start an incoming connection timer when it makes the connect request available to the destination end user. If the end user does not accept or reject the connect request before the timer elapses, Session Control issues a reject to Network Services, including two bytes of reject data (see Section 6.0). Such timer processing is system-dependent. However, if a Session Control module uses such a timer and supports the SET-INCOMING-TIMER function described in Section 2.3.3, the value of the incoming connect timer is the default value most recently set by Network Management.

5.3 Sending and Receiving Data

Because the interface between end users and Session Control is system-dependent, Session Control can handle requests to send and receive data in several ways. Section 5.3.1 describes two data buffering models and Section 5.3.2 describes three possible data transfer interfaces.

5.3.1 Data Buffering Models - The two models described are the end user buffering model and the Session Control buffering model.

The End User Buffering Model -- This model is characterized as follows:

1. **When an end user transmits a buffer of data,** Session Control temporarily receives ownership of the buffer. To the end user, it appears that the data is being transmitted directly out of the buffer -- while Session Control "owns" the buffer, the end user cannot modify the data without nullifying the guarantees of data integrity. To regain buffer ownership, the end user polls Session Control for a "transmit complete."

2. **When an end user supplies a buffer to receive data,** Session Control temporarily receives ownership of the buffer. To the end user, it appears that the data is being received directly into the buffer. As also applies to a transmit buffer, the end user cannot modify the buffer while Session Control owns it without nullifying the guarantees of data integrity. To regain ownership of the buffer and to receive the data it contains, the end user polls Session Control for a "receive complete."

The Session Control Buffering Model -- This model is characterized as follows:

1. **When an end user attempts to transmit a buffer of data,** Session Control either accepts or rejects the transmit request. If Session Control accepts the request, the data in the transmit buffer appears to have been moved to a Session Control transmit buffer while the request was being processed. The end user does not need to poll Session Control to regain ownership of the buffer nor can it in any way nullify the guarantee of data integrity.

2. **When an end user supplies a receive buffer,** Session Control either returns the buffer empty with a "no data" indication or returns the buffer with data and a "data returned" indication. If data is returned, it appears to the end user that the data was moved into the buffer from a Session Control receive buffer while the receive request was being processed. The end user does not need to poll Session Control to regain ownership of the buffer nor can it in any way nullify the guarantee of data integrity.

5.3.2 Data Transfer Interfaces - A Session Control module sends and receives data via one of three data transfer interfaces: message, segment, or stream.

1. **The Message Interface.** This interface, which is unconcerned with network packet size, allows an end user to send or receive a large amount of information. On transmission, Session Control uses Network Services' DATA-XMT function with "xmtflag" set to "end-of-message" to send the entire buffer of end user data. To receive data via the message interface, the end user passes a receive buffer to Session Control. In turn, Session Control uses the DATA-RCV function with "rcvflag" set to "truncation allowed" to pass the buffer to Network Services. If the receive buffer is smaller than the corresponding transmit buffer, the transmitted data that overflows the receive buffer is lost.

   Session Control can use either of the buffering models described above to operate the message interface.

2. **The Segment Interface.** This interface ensures that a receiving end user never loses any information from data truncation. Using the segment interface, an end user sends and receives data in buffers that correspond to Network Services packets. In other words, a transmitting end user cannot send a buffer larger than the packet size, and a receiving end user cannot provide a buffer smaller than the packet size. When transmitting data, an end user provides the "xmtflag" value that Session Control passes to Network Services. When an end user makes a request to receive data, Session Control sets "rcvflag" to "no truncation allowed". In addition, an end user has access to a function that returns the segment (that is, Network Services packet) size.

   Session Control can use either of the buffering models described above to operate the segment interface.

3. **The Stream Interface.** When using this interface, an end user views the data it sends and receives as a sequence (that is, stream) of bytes, words, and so on, occasionally separated by an "end-of-message" marker. The stream interface is basically the same as the segment interface except that the size of the Network Services packet does not restrict the size of end user transmit or receive buffers. Transmit buffers can be larger than the packet size and receive buffers can be smaller than the packet size.

   Session Control must use the Session Control buffering model to operate the stream interface.

5.4 Disconnecting and Aborting a Logical Link

This section describes the Session Control models for sending a disconnect or abort request and receiving a disconnect or abort request.

1. **Sending disconnect or abort requests.** When an end user instructs Session Control to disconnect or abort a logical link, Session Control issues either a DISCONNECT-XMT or an ABORT-XMT call, respectively, to Network Services. Either call passes a two-byte disconnect code (see Section 6.0) and up to 16 bytes of disconnect data if the requesting end user supplied any. After requesting a disconnection or abort, Session Control may continue to provide receive buffers until Network Services places the appropriate port in the DISCONNECT-INITIATE-COMPLETE state. Upon detection of this state, Session Control closes the port.

   Because Network Services does not guarantee to complete a disconnect or an abort within a bounded period of time, Session Control can optionally start a timer when it issues its request. If Network Services does not place the relevant port in the DISCONNECT-INITIATE-COMPLETE state before the timer elapses, Session Control closes the port. The timer value may be a default or it may be an "error tolerance" argument provided by an end user in a connect request or connect accept function (see Section 2.2). Depending on the system, Session Control may or may not return completion status to the end user that requested the disconnection or abort.

2. **Receiving a disconnect or abort request.** When a Session Control module has issued a DISCONNECT-XMT or an ABORT-XMT to Network Services, Network Services forwards the request to a destination Network Services module. The destination Network Services then places the appropriate port in the DISCONNECT-NOTIFICATION state. In turn, the destination Session Control module detects the state of the port, obtains disconnect data from Network Services, and closes the port.

   The first two bytes of the disconnect data constitute a reason code for the disconnection. How Session Control interprets the reason code depends on the type of node in which the end user that requested the disconnection resides:

   - If the node is a Phase III system, the first two bytes of disconnect data equal either 0 (zero) or 9. Zero indicates a disconnect request from the remote end user; 9 indicates an abort request from the remote end user.

   - If the node is a Phase II system, the two bytes may equal zero, which indicates a disconnect request, or they may be nonzero, which indicates an abnormal disconnection. It depends on the system whether or not Session Control supplies the destination end user with additional information about the reason for an abnormal disconnection.

5.5 Monitoring a Logical Link

A Session Control module can optionally monitor a logical link in a system-dependent manner. The purpose of such monitoring would be to detect the probable disconnection of the network between the nodes at either end of a logical link or the failure of Network Services to deliver transmitted data in a timely fashion. How Session Control performs this optional function, is system-dependent. Furthermore, the information that a Session Control module uses to monitor a link is also system-dependent.

End users might be able to control logical link monitoring by supplying an error tolerance variable in a connect request or connect accept issued to Session Control (see Section 2.2). One possible monitoring operation is described below:

Example of Monitoring Operation:

Session Control can monitor the confidence variable for a logical link. The confidence variable is Network Services' assessment of a logical link's connectivity, that is, whether the network supporting the logical link is connected. (See the CONFIDENCE function, described in Section 2.1.) If the variable is false, indicating a probable disconnection of the network, Session Control can start a timer, either a default value or a user-supplied "error tolerance" variable. If the confidence variable does not return to true before the timer expires, Session Control can close the port associated with the logical link.

6.0 ERROR CODES

This section describes the types of errors that Session Control returns to end users and to other Session Control modules.

- Errors that Session Control returns to end users are described in Section 6.1.
- Errors that one Session Control module sends to another in the REASON field of a disconnect or reject message (Section 3.3) are described in Section 6.2.

NOTE

> The term "object," which appears in several of the error descriptions below, means end user.

6.1 Error Codes Returned to End Users

The error code numbers and corresponding reasons (capitalized phrases) given below are standard for all implementations of Session Control. Every implementation of Session Control can generate error codes 0, 4, 5, 6, 9, and 34. Two error codes (0 and 38) have two reasons for the same code number. In these cases, the state of the logical link, as perceived by the user, determines which reason (and therefore which phrase) applies to the error.

The following description groups the errors into two categories:

- Errors returned in response to a connect request
- Errors returned with a disconnect notification.

Connect request errors:

Code Number **Reason**

0 REJECTED BY OBJECT

The destination end user rejected a connect request.

1 NETWORK RESOURCES

No resources are available for a new logical link. (The missing resources may be in the local or in the remote system.)

2 UNRECOGNIZED NODE NAME

The destination node name specified in a connect request cannot be translated -- the node name mapping table does not contain a corresponding entry.

3 REMOTE NODE SHUT DOWN

The destination Session Control is in the SHUT or RESTRICTED state (Section 2.3.1) and therefore cannot handle incoming connect requests.

4 UNRECOGNIZED OBJECT

The destination end user specified in a connect request does not exist.

5 INVALID OBJECT NAME FORMAT

The specification of the destination end user in a connect request was not in correct object name format. (This error is returned only in those systems in which the source end user can control the format of the destination process name.)

6 OBJECT TOO BUSY

The destination end user does not have sufficient resources to handle a new logical link.

10 INVALID NODE NAME FORMAT

The destination node name in a connect request had an invalid format.

11 LOCAL NODE SHUT DOWN

The local Session Control is in the OFF or SHUT state (Section 2.3.1) and therefore cannot handle outgoing connect requests. (Some implementations may return this reason only in the SHUT state if Session Control is unavailable in the OFF state.)

34 ACCESS CONTROL REJECTION

The access control information in a connect request was invalid and therefore rejected by the destination Session Control.

38 NO RESPONSE FROM OBJECT

The destination end user did not respond to a connect request. (It may have received the connect request but crashed before accepting or rejecting it.)

39 NODE UNREACHABLE

The node to which a connect request has been sent is unreachable.

Disconnect notification errors:

Code
Number **Reason**

0 DISCONNECT BY OBJECT

The end user disconnected a running logical link.

8 ABORT BY MANAGEMENT

The logical link was disconnected by third party (by a party other than the communicating end user).

9 ABORT BY OBJECT

The remote end user has aborted the link.

38 NODE OR OBJECT FAILED

A third party has aborted the link; or the end user or node at the opposite end of the link has crashed.

6.2 Reject and Disconnect Data Error Codes

A Session Control module specifies one of the codes defined below in the REASON field of a reject or disconnect data message. The REASON field consists of the first two bytes of the data. The codes are decimal.

Table 2 below shows the correspondence between the Session Control to Session Control error codes described in this section and the error codes Session Control returns to end users, described in Section 6.1.

Code Number **Reason**

0 No error.

3 The node is shutting down.

4 The destination end user does not exist.

5 A connect request contained an invalid destination end user name.

6 The destination end user does not have sufficient resources to handle a new logical link.

7 Unspecified error.

8 A third party (that is, a process other than either end user) has aborted the logical link.

9 An end user has aborted the logical link.

32 The node does not have sufficient resources to handle a new logical link.

33 The destination end user does not have sufficient resources to handle a new logical link.

34 A connect request contained an unacceptable RQSTRID or PASSWORD field; Session Control therefore rejected the request.

36 A connect request contained unacceptable ACCOUNT information. The requesting end user is not authorized or the account balance is unacceptable.

38 The end user has aborted, timed out, or cancelled a connect request.

43 The image data field (RQSTRID, PASSWORD, ACCOUNT, USRDATA) of a connect data message was too long.

Table 2
Session Control to Session Control Codes Mapped to Session Control to End User Codes

Session Control to Session Control	Session Control to End User
Reject codes:	
0	0
3	3
4	4
5	5
6	6
34,36,43	34
38	38
Disconnect or abort codes:	
0	0
8	8
9	9
38	38

APPENDIX A

PORT AND LOGICAL LINK STATES

Ports and porc states are important factors in Session Control's interface to Network Services. This appendix explains what ports are and defines the states that Network Services can assign to them. For more detailed information on this subject, refer to the DNA Network Services Functional Specification.

A.1 PORTS

Every logical link set up by Network Services is distinguishable from all other links. To differentiate one link from another, Network Services assigns a port to each logical link it sets up, where a port is a space in memory (generally in a dedicated or shared pool) that contains control variables for managing the link. Each node within a network has a number of available ports, which are allocated and controlled by Network Services. Because each end of a logical link has its own port, the creation of a link can be thought of as the association of one port with another.

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

Network Services also maintains a "confidence" variable in each port that has been opened. Session Control can access this variable, which is useful in detecting network failures.

A.2 PORT STATES

At any given time, each end or port of a logical link is in a particular state, which is determined by Session Control requests and Network Services messages pertaining to the link. The possible states at the ends of a link are called port states. The state that Network Services associates with a port is represented by a variable in the port data base. Table 3 below defines all the possible port states. At any time, a port is in only one state.

Table 3
Port States

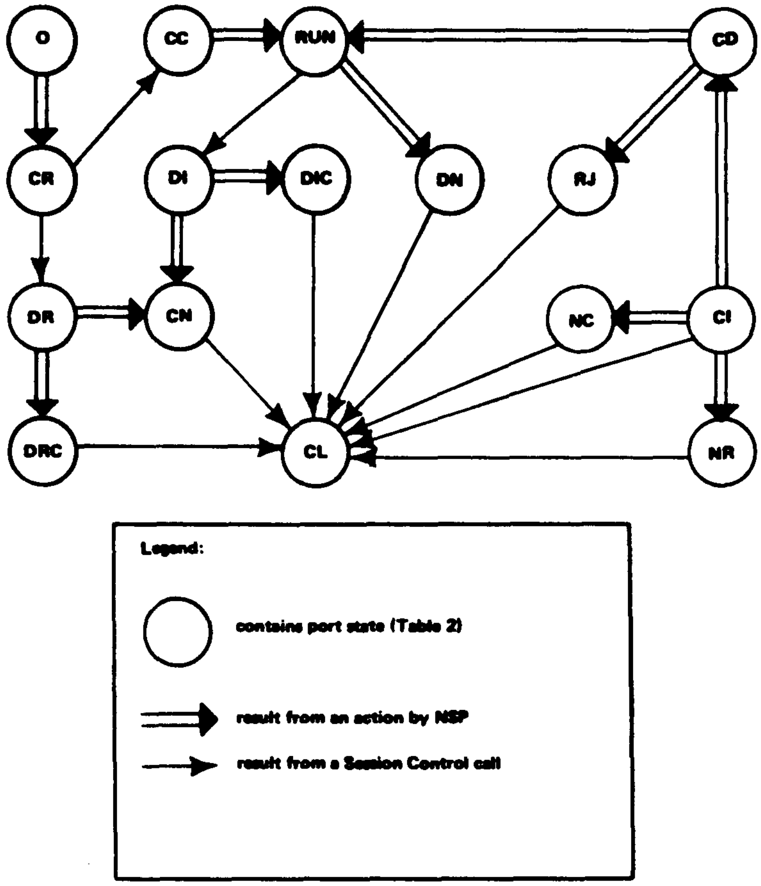
(Symbol) State	Explanation
(O) OPEN	The local Session Control has issued an OPEN call which allocated the port.
(CR) CONNECT-RECEIVED	NSP has received a Connect Initiate message.
(DR) DISCONNECT-REJECT	The local Session Control has issued a REJECT call while the port was in the CONNECT-RECEIVED state.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

(continued on next page)

Table 3 (Cont.)
Port States

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

Figure 4 below summarizes the normal port state transitions. These transitions correspond to those described in Table 3 above.


O
CC
RUN
CD
CR
DI
DIC
DN
RJ
DR
CN
NC
CI
DRC
CL
NR
Legend:
contains port state (Table 2)
result from an action by NSP
result from a Session Control call

NOTES

1. A state from which an exit can be made by a double arrow is a potentially unstable state.
2. A state from which the only exits are single arrows are stable states.
3. A state from which an exit can be made by more than one double arrow is a state from which the exit is non-deterministic.

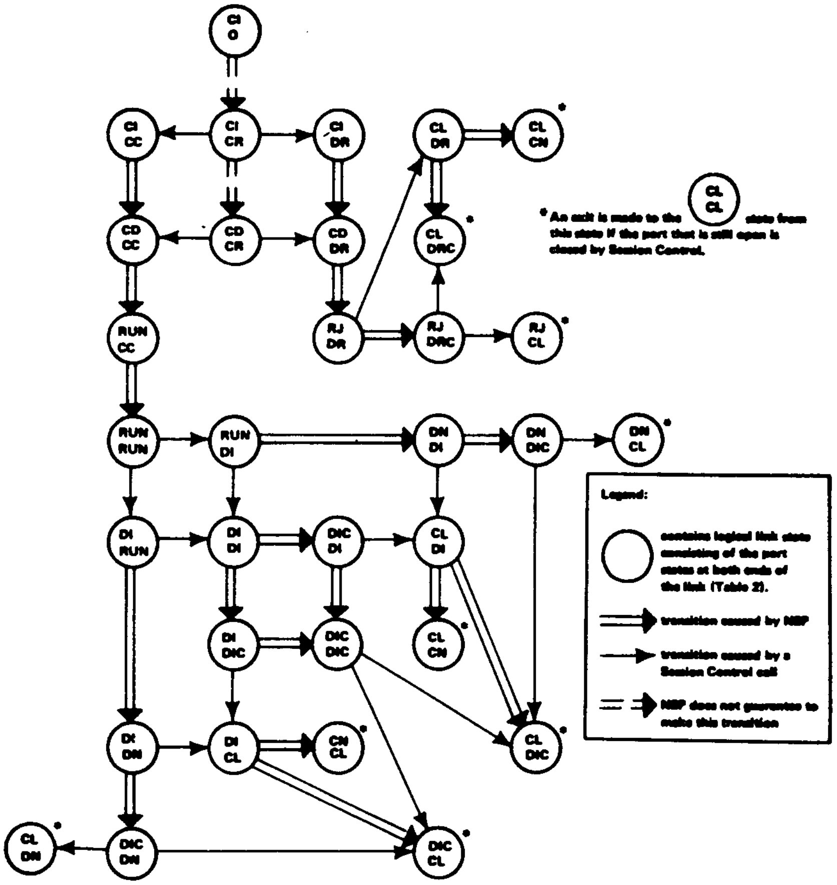
Figure 4 Port State Diagram

A.3 LOGICAL LINK STATES

At any given time, a logical link is also associated with a particular state, which is determined by the combination of possible port states at each end of the link. The product of the two sets of port states is called the logical link state.

When one Session Control module attempts to form a connection to a second Session Control module, Network Services places the requesting port in the CONNECT-INITIATE (CI) state. Network Services then attempts to assocate the source, local port with a destination port that is in the OPEN (O) state. If the association between the two ports is successful, a logical link exists and its initial state is CI/O.

Figure 5 below shows the normal logical link state transitions. At any time, a logical link can be in only one state.


CI O
CI CC
CI CR
CI DR
CL DR
CL CN
* An exit is made to the CL CL state from this state if the port that is still open is closed by Session Control.
CD CC
CD CR
CD DR
CL DRC
RUN CC
RJ DR
RJ DRC
RJ CL
RUN RUN
RUN DI
DN DI
DN DIC
DN CL
Legend:
contains logical link state consisting of the port states at both ends of the link (Table 2).
transition caused by NSP
transition caused by a Session Control call
NSP does not guarantee to make this transition
DI RUN
DI DI
DIC DI
CL DI
DI DIC
DIC DIC
CL CN
DI DN
DI CL
CN CL
CL DIC
CL DN
DIC DN
DIC CL

NOTES

1. A state from which an exit can be made by a double arrow is a potentially unstable state.
2. A state from which the only exits are single arrows are stable states.
3. A state from which an exit can be made by more than one double arrow is a state from which the exit is non-deterministic.
4. The logical link states presented above describe the disconnection or abortion of the link from the RUN state when requested by either Session Control module. This is true because the Session Control requesting a disconnection could be either the Session Control that requested the logical link or the module that accepted the logical link.
5. If a logical link enters either the DI/RUN or RUN/DI state because of a disconnect request by one of the Session Control modules, then an NSP exit from the DI/RUN or RUN/DI states is possible only if the Session Control module in the RUN state has provided a sufficient number of receive buffers to receive all data transmitted by the other Session Control module. The unbroken double arrow exit from either of these states means that NSP guarantees to make the exit eventually only if this constraint has been met. Similarly, an NSP exit from the DI/DI state is possible only if one of the Session Control modules sharing the logical link has met this constraint. This constraint does not apply when the DI port state is entered because of an abort request.

Figure 5 Logical Link State Diagram

APPENDIX B

OBJECT TYPES

The object type code values that have been defined are listed below. The numbers listed are decimal.

Object Type	Process Type
0	General task, user process
1	File access (FAL/DAP-Version 1)
2	Unit Record Service (URDs)
3	Application Terminal Services (ATS)
4	Command Terminal Services (CTS)
5	RSX-11M Task Control-Version 1
6	Operator Services Interface
7	Node Resource Manager
8	IBM 3270-BSC Gateway
9	IBM 2780-BSC Gateway
10	IBM 3790-SDLC Gateway
11	TPS Application
12	RT-11 DIBOL Application
13	TOPS-20 Terminal Handler
14	TOPS-20 Remote Spooler
15	RSX-11M Task Control-Version 2
16	TLK Utility
17	File Access (FAL/DAP-Version 4 and any subsequent version)
18	RSX-11S Remote Task Loader
19	NICE Process

Object Type	Process Type
20	RSTS/E Media Transfer Program (NETCPY)
21	RSTS/E Homogeneous Network Command Terminal Handler
22	Mail Listener (DECnet-based electronic mail system)
23	Host Terminal Handler
24	Concentrator Terminal Handler
25	Loopback Mirror
26	Event Receiver
27	VAX/VMS ,Personal Message Utility
28	FTS

GLOSSARY

confidence

A Network Services variable (CONFIDENCE) that indicates the probable connectedness of the physical network supporting a logical link. See Sections 2.1 and 5.5.

end user

A module that runs in the "user space" of a network node and communicates with Session Control to obtain logical link service. In relation to the DNA hierarchy, an end user module resides in the User, Network Application, or Network Management layer.

incoming connection timer

A timer that Session Control can optionally start when it receives an incoming connect request. If the timer (measured in seconds) expires before the destination end user responds, Session Control issues a connect reject to Network Services. See Section 2.3.3.

logical link

A virtual channel between two end users in the same node or in separate nodes. Session Control acts as an interface between an end user requiring logical link service and Network Services which actually creates, maintains, and destroys logical links. See Section 1.2.

Network Management

The DNA layer that enables operator control over and observation of network parameters and variables. Network Management also provides down-line loading, up-line dumping, and testing functions.

Network Services (NSP)

The DNA layer immediately below Session Control which enables the creation, maintenance, and destruction of logical links, and which performs data flow control and end-to-end error control, and which segments and reassembles messages sent across the logical links.

node name mapping table

A table that defines the correspondence between node names and node addresses or channel numbers. Session Control uses the table to identify destination nodes for outgoing connect requests and source nodes for incoming connect requests. See Sections 2.3.2 and 4.1.

outgoing connection timer

A timer that Session Control can optionally start when it issues a connect request to Network Services. If the port associated with the request does not enter one of four specific states before the timer expires, Session Control issues a timeout rejection to the requesting end user and asks Network Services to close the port. See Section 2.3.3.

port

A collection of control variables and parameters for managing logical links. Each logical link has a port at each end. Each Network Services at each node has a number of available ports. When Session Control requests a logical link or requests a port be opened to receive an incoming connect request, NSP allocates a port if sufficient resources are available. See Sections 1.2.1 and 1.2.2 and Appendix A.

Session Control

The DNA layer directly above NSP. Session Control defines the system-dependent aspects of logical link communication. Session Control provides functions such as name-to-address translation, process addressing, and in some systems, access control.

Transport

The DNA layer directly below NSP that provides NSP with routing, congestion control, and packet lifetime control services.

DECnet DIGITAL
Network Architecture
Session Control
Functional Specification
AA-K182A-TK

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 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____________

Please cut along this line.

— — — —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
146 MAIN STREET ML 5-5/E39
MAYNARD, MASSACHUSETTS 01754

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

Cut Along Dotted Line

DECnet DIGITAL
Network Architecture
Session Control
Functional Specification
AA-K182A-TK

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 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__________

Please cut along this line.

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

digital

No Postage
Necessary
if Mailed in the
United States

SOFTWARE DOCUMENTATION
146 MAIN STREET ML 5-5/E39
MAYNARD, MASSACHUSETTS 01754

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

Cut Along Dotted Line