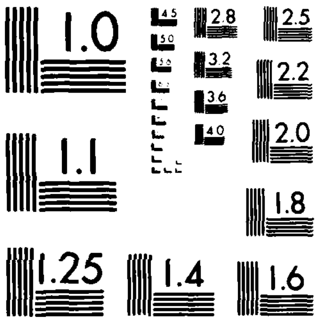
AD-A149 693 CONNECTION MANAGEMENT AND FILE TRANSFER PROTOCOLS FOR THE DTNSRDC (DAVID..(U) DAVID W TAYLOR NAVAL SHIP RESEARCH AND DEVELOPMENT CENTER BET.. D P ALLEN ET AL. JUL 84 DTNSRDC/CMLD-84-16 1/1

UNCLASSIFIED F/G 9/2 NL

END



MICROCOPY RESOLUTION TEST CHART
NATIONAL BUREAU OF STANDARDS-1963-A

(12)

CMLD-84-16

Connection Management and File Transfer Protocols for the DTNSRDC Hyperchannel Network

DAVID W. TAYLOR NAVAL SHIP RESEARCH AND DEVELOPMENT CENTER

Bethesda, Maryland 20084

Connection Management and File Transfer Protocols for the DTNSRDC Hyperchannel Network

by

Daniel P. Allen
Robert W. Tinker

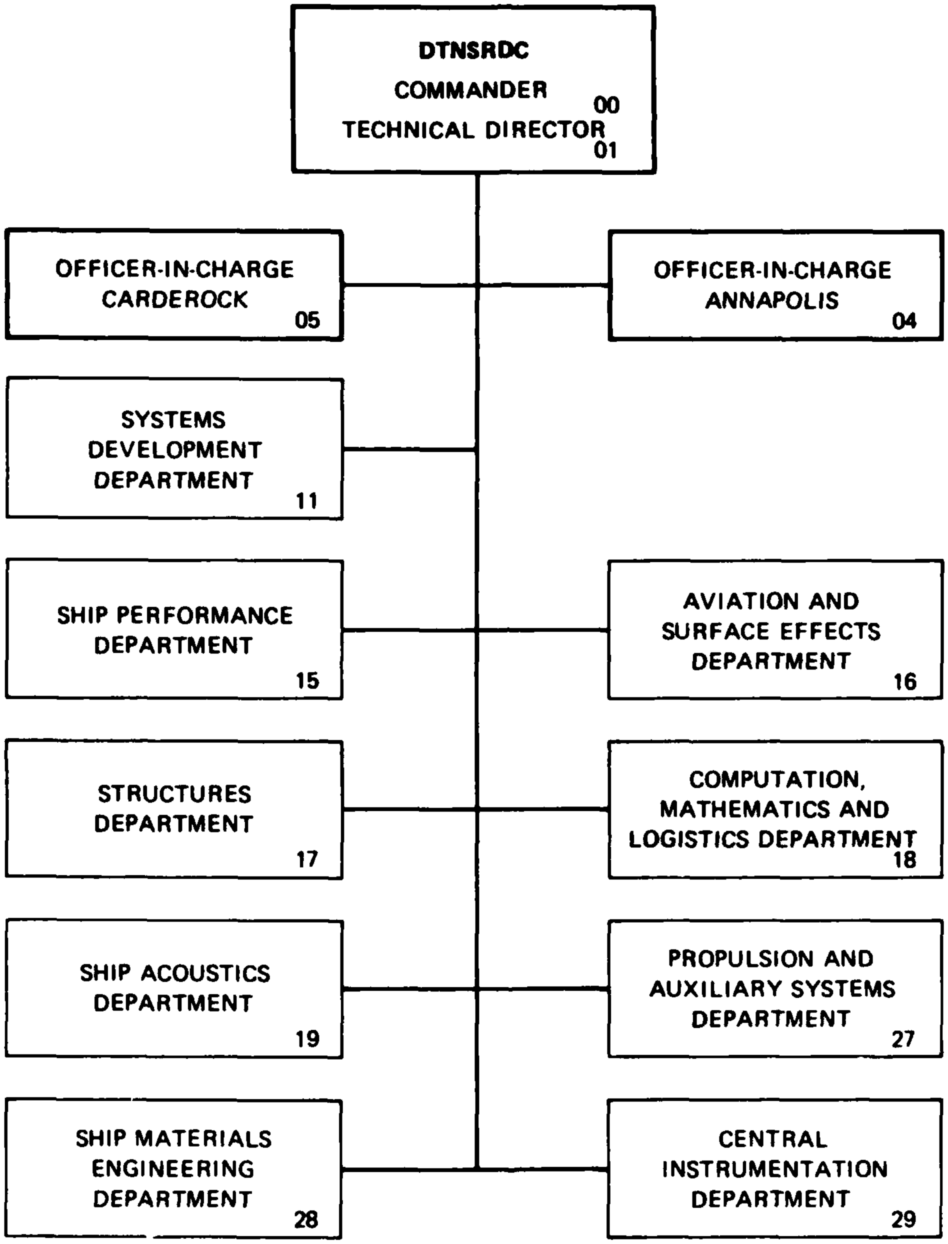
APPROVED FOR PUBLIC RELEASE: DISTRIBUTION UNLIMITED

Computation, Mathematics, and Logistics Department
Departmental Report

July 1984 CMLD-84-16

NDW DTNSRDC 5602.30 (2-80)
(supersedes 3960-46)

MAJOR DTNSRDC ORGANIZATIONAL COMPONENTS



SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

REPORT DOCUMENTATION PAGE

READ INSTRUCTIONS BEFORE COMPLETING FORM

1. REPORT NUMBER: CMLD-84-16
2. GOVT ACCESSION NO.: AD-A149 693
3. RECIPIENT'S CATALOG NUMBER:
4. TITLE (and Subtitle): Connection Management and File Transfer Protocols for the DTNSRDC Hyperchannel Network
5. TYPE OF REPORT & PERIOD COVERED: Final
6. PERFORMING ORG. REPORT NUMBER:
7. AUTHOR(s): Daniel P. Allen, Robert W. Tinker
8. CONTRACT OR GRANT NUMBER(s):
9. PERFORMING ORGANIZATION NAME AND ADDRESS: David Taylor Naval Ship R&D Center, Advanced Systems Development Group, Code 189.2, Bethesda, MD
10. PROGRAM ELEMENT, PROJECT, TASK AREA & WORK UNIT NUMBERS:
11. CONTROLLING OFFICE NAME AND ADDRESS: Computation, Mathematics & Logistics Department, Computer Facilities Division (189)
12. REPORT DATE: July 1984
13. NUMBER OF PAGES: 24
14. MONITORING AGENCY NAME & ADDRESS (if different from Controlling Office):
15. SECURITY CLASS. (of this report): UNCLASSIFIED
15a. DECLASSIFICATION/DOWNGRADING SCHEDULE:
16. DISTRIBUTION STATEMENT (of this Report): Approved for Public Release: Distribution Unlimited
17. DISTRIBUTION STATEMENT (of the abstract entered in Block 20, if different from Report):
18. SUPPLEMENTARY NOTES:
19. KEY WORDS (Continue on reverse side if necessary and identify by block number)

(1) Local Area Networking
(2) Transport Protocol
(3) File Transfer

20. ABSTRACT (Continue on reverse side if necessary and identify by block number)

This report describes two sets of protocols implemented on DTNSRDC's Hyperchannel based local area network. The first protocol set facilitates local area network connections among arbitrary processes running on computers attached to DTNSRDC's Hyperchannel network. The second protocol set provides a mechanism for allowing local area network hosts (computers connected via the Hyperchannel) to store, retrieve, and delete files on other network hosts.

DD FORM 1473, 1 JAN 73 — EDITION OF 1 NOV 65 IS OBSOLETE
S/N 0102-LF-014-6601

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

TABLE OF CONTENTS

List of Figures

Appendixes

ABSTRACT

This report describes two sets of protocols implemented on DTNSRDC's Hyperchannel based local area network. The first protocol set facilitates local area network connections among arbitrary processes running on computers attached to DTNSRDC's Hyperchannel network. The second protocol set provides a mechanism for allowing local area network hosts (computers connected via the Hyperchannel) to store, retrieve, and delete files on other network hosts.

ADMINISTRATIVE INFORMATION

The work described in this report was a joint effort of the Systems Software Group (Code 1892.3) and the Advanced Systems Development Group (Code 189.2) of the Computation, Mathematics and Logistics Department, *David W. Taylor Naval Ship Research and Development Center* under sponsorship of the DTNSRDC Computer Center (Code 189).

1. Introduction

The Computation, Mathematics, and Logistics Department (CMLD) at the David Taylor Naval Ship Research and Development Center (DTNSRDC) operates a variety of digital computer systems ranging in size from small minicomputers, such as the PDP 11/34, to larger general purpose mainframes such as the CDC Cyber 170-750. These systems support a diverse workload in the areas of engineering and scientific research projects, management and financial record keeping, and office automation systems. No single system available in the Department can provide all the services required by the user community. At the same time many of the applications operational on a particular system require resources that are available only on one of the other processors. The use of online data storage devices is a prime example of this problem.

The Department has acquired a large Mass Storage Facility (MSF) from Control Data Corporation (CDC) which serves the users of the existing CDC computer systems. At the same time the Department is operating two VAX 11/780-UNIX based office automation systems which are suffering from a lack of online storage space. Access to the MSF has been made available to foreign systems in general, and to the two VAX's in particular as a back-end data storage system. Access to the MSF has been achieved using Network Systems Corporation's Hyperchannel as the transport medium.

To support this effort two protocols were developed and are described here. The first protocol facilitates the control of network connections thereby allowing arbitrary processes on the network to inter-communicate. The second protocol provides for storing, retrieving, deleting, and tracking files on remote network hosts via a network connection.

2. Network Connection Protocol (NCP)

2.1. General Description

The base protocol implemented on the Hyperchannel is a general purpose network connection management protocol. The network connection protocol (NCP) is intended to allow any network host to support up to 255 simultaneous logical connections between itself and remote hosts. This protocol makes no specific provisions to ensure data integrity or flow control. Instead, the Hyperchannel hardware is relied upon to provide these services. In most cases this will be sufficient. However, where it is not, application level protocols must be invoked to handle these services on a case by case basis.

The NCP relies on the concept of a "logical connection." A logical connection is a means of identifying a "conversation" between two logical processes residing on hosts on the network. Each end of a logical connection is identified by the hardware address of the local

Hyperchannel adapter, the Hyperchannel adapter port number to which the local host computer is attached ("0" for single port adapters), and a "link number" assigned by the local host's connection management (NCP) routines when the logical connection is established. Represented symbolically, a connection-end identifier consists of five hexadecimal digits of the form HHPLL where HH is the Adapter's thumbwheel selectable hardware address, P is the port number, and LL is the link number. The complete logical connection is uniquely identified by the pair of connection-end identifiers.

For example, if two hosts are connected to the network via Hyperchannel adapters with hardware addresses 11 and 22, respectively, and through adapter ports 0 and 1, respectively, the connection-end identifiers for the first host will be of the form 110xx (where xx is the locally assigned link number for a given connection). Similarly, 221yy will be the form of the second host's connection-end identifiers. Now if, for a particular connection between the two hosts, the first host assigned link number 6, and the second host assigned link number 4, then the complete logical connection is identified by the combination of the two connection end identifiers, viz. 11006/22104. Note that each host is free to assign a link number to each of its connection ends regardless of any link number assignments by other hosts. Each message which traverses the network contains the complete logical connection identifier (hardware addresses and link numbers) to unambiguously denote the sending and receiving processes.

Link number zero is reserved by each host's NCP process for exchange of the connection management protocol. This link is used to establish and dis-establish other links, or to reject attempts by other hosts to establish links when it is desirable or necessary to do this.

2.2. NCP Functions and Formats

This section details the link 0 (connection management) protocol functions and format as implemented on the Hyperchannel network. Note that each connection management protocol message is sent as a Hyperchannel "message proper" with no "associated data." Moreover, the first eight bytes (0-7) of each message have specific meaning to the Hyperchannel Adapter hardware. The reader is referred to Network Systems Corporation's publication Hyperchannel - Systems Description Manual (publication number A01-0000-02) for an understanding of these terms and fields.

In the following descriptions all values for field offsets and parameter values are given as hexadecimal constants unless specifically stated otherwise.

2.2.1. Open (OPN) Function

This function is issued by a host to open a logical connection and "echoed" by the receiver to signal its acceptance of the connection request. If the request cannot be honored, the RJT function will be sent in reply. When a network host initially requests a connection to a remote host, it assigns a logical link number for its end of the connection to be established and places this link number in parameter field p1. It also clears parameter field p2. The receiver of such a function completes the connection by assigning a logical link number for its end of the connection. It places this in field p1, the initiator's link number in field p2, and sends a confirming OPN message. Figure 2-1 describes the format of the OPN message block.

0	1	2	3	4	5	6	7	8	9
trk	ctl	acc	acc	rad	rpn	sad	spn	0	op

A	B	C	D	E	F	10	11	12	13
sln	rln	p1	p2	0	0	0	0	0	0

14	15	16	17	18	19	1A	1B	1C	1D
pid	pid	pid	pid	pid	pid	pid	pid	pid	pid

trk - trunk selection bits
ctl - control bits
acc - access code
rad - receiver's adapter address
rpn - receiver's adapter port number (zero if single port adapter)
sad - sender's adapter address
spn - sender's adapter port number (zero if single port adapter)
op - operation code - OPN function (01)
sln - sender's link number (always zero for NCP messages)
rln - receiver's link number (always zero for NCP messages)
p1 - sender's link number to be used for the newly established connection
p2 - receiver's link number to be used for the newly established connection (zero for requesting OPN, initiator's link number for confirming OPN)
pid - name of the server process to which this connection attempt is directed

Figure 2-1 - NCP Open (OPN) Message Format

2.2.2. Close (CLS) Function

This function is issued by a host to close a logical connection. The initiator of the close inserts the link number for his connection-end in parameter p1 and the remote host's link number for the remote connection-end in parameter p2. The receiving system acknowledges the request by transmitting an answering CLS message after reversing p1 and p2. Note that this scheme allows each host to simultaneously and unilaterally initiate a close on the same logical connection.

0	1	2	3	4	5	6	7	8	9
trk	ctl	acc	acc	rad	rpn	sad	spn	n/u	op

A	B	C	D
sln	rln	p1	p2

trk - trunk selection bits
ctl - control bits
acc - access code
rad - receiver's adapter address
rpn - receiver's adapter port number
sad - sender's adapter address
spn - sender's adapter port number
op - operation code - CLS function (02)
sln - sender's link number (always zero for NCP messages)
rln - receiver's link number (always zero for NCP messages)
p1 - sender's link number of the connection to be closed
p2 - receiver's link number of the connection to be closed

Figure 2-2 - NCP Close (CLS) Message Format

2.2.3. Reject (RJT) Function

This function is used to refuse OPN requests or to reject ill-formed link zero messages such as illegal op code, etc. It requires no acknowledgment.

0	1	2	3	4	5	6	7	8	9
trk	ctl	acc	acc	rad	rpn	sad	spn	0	op

A	B	C	D
sln	rln	rc	pl

trk - trunk selection bits
ctl - control bits
acc - access code
rad - receiver's adaptor address
rpn - receiver's port number
sad - sender's adapter address
spn - sender's port number
op - operation code - RJT function (03)
sln - sender's link number (always zero).
rln - receiver's link number (always zero).
rc - reason code for connection rejection
    01 - illegal op code
    02 - invalid link
    03 - no links available
pl - if rc = 3, link number on which user tried to open a connection.

Figure 2-3 - NCP Reject (RJT) Message Format

3. File Transfer Protocol (FTP)

3.1. General Description

The objective of this protocol is to permit network hosts to access the file system of other hosts on the network via a Hyperchannel connection. Each host with a user FTP process can store, retrieve, delete, or list directory information from the file system of any other host with a server FTP process.

This protocol consists of eight functions used to control the transfer of files or directory information between the two FTP processes. All connections are initiated by the user FTP process and maintained until closed by the user FTP process.

To facilitate the efficient use of the MSF cartridge system and to provide support for the transfer of CDC system binary images and program libraries, the concepts of system End of Record (EOR), End of File (EOF), and End of Information (EOI) are provided. These concepts provide a mechanism for partitioning a data set into discrete subsets. Using these structures a remote system can consolidate its files to make more efficient and cost effective use of the MSF allocation mechanisms. This feature can also serve to minimize the quantity of data moved over the network when it is desired to retrieve a sub-file rather than an entire data set. For non-CDC systems the implementation of these data set partitioning mechanisms will be system dependent.

The protocol relies on the data validation performed by the Hyperchannel hardware to ensure file integrity. No checksumming is performed in the software. No mechanism is provided in the protocol for flow control. It is assumed that the status returns from the Hyperchannel hardware and their data buffering capabilities are sufficient to recover data overrun conditions. Because the connection between remote systems is full duplex, it is incumbent on both the sending and receiving systems to monitor the Hyperchannel for incoming messages.

Basic protocol information is passed as part of the Hyperchannel message proper. Byte 9 of this message always contains one of the eight protocol-defined function codes. Bulk data associated with each FTP message block is sent as an associated data block. The protocol places no restriction other than an absolute maximum of 20,000 bytes on the length of each associated data block. As a result it is the responsibility of the user and server FTP processes to cooperate on block sizes such that differences in word size and memory addressability between hosts of different hardware architecture do not corrupt the integrity of associated data.

3.2. FTP Function Descriptions

The following pages detail the format and content of the FTP protocol messages. All relative block offsets and parameter values are given as hexadecimal constants unless explicitly stated otherwise.

3.2.1. FTP Data Function (DAT)

This function is used to transfer file or directory data between the server and user FTP process. No response is required. Once initiated, the transfer of FTP data blocks will be repeated until the logical end of information is reached. The end of information is always signaled by a DAT message with the EOI flag set. This message may or may not have associated data. If the receiver of the data wishes to interrupt the transfer for some reason, he should do so by transmitting an FTP negative acknowledgment message with an appropriate reason code. The NAK will abort any additional processing for the FTP function in progress. Figure 3-1 details the format and content of a DAT message block.

0	1	2	3	4	5	6	7	8	9
trk	ctl	acc	acc	rad	rpn	sad	spn	0	op

A	B	C
sln	rln	flg

trk - trunk selection bits
ctl - control bits
acc - access code
rad - receiver's adapter address
rpn - receiver's port number
sad - sender's adapter address
spn - sender's port number
op - operation code - DAT function (10).
sln - sender's link number
rln - receiver's link number
flg - file structure flag bits
    01 - End of record (EOR)
    02 - End of file (EOF)
    04 - End of information (EOI)

Figure 3-1 - FTP Data Message Format

3.2.2. Positive Acknowledgment Function (ACK)

This function is used to signal a positive acknowledgment of FTP requests. It is sent only by the server FTP process. Figure 3-2 details the format and content of an ACK message block.

0	1	2	3	4	5	6	7	8	9
trk	ctl	acc	acc	rad	rpn	sad	spn	0	op

A	B
sln	rln

trk - trunk selection bits
ctl - control bits
acc - access code
rad - receiver's adapter address
rpn - receiver's port number
sad - sender's adapter address
spn - sender's port number
op - operation code - ACK function (11)
sln - sender's link number
rln - receiver's link number

Figure 3-2 - FTP Positive Acknowledgment Message Format

3.2.3. Negative Acknowledgment Function (NAK)

This function is issued by the server FTP process in response to a request from a user FTP process that cannot be honored or by either the server or user FTP process to abort a data *transfer*. The *rc field* contains an error code for the error condition. Bytes F - 3F contain the ASCII text of an appropriate error message. Figure 3-3 details the format and content of a NAK message block.

0	1	2	3	4	5	6	7	8	9
trk	ctl	acc	acc	rad	rpn	sad	spn	0	op

A	B	C	D	E	F	10		3F
sln	rln	rc	0	0	txt	txt	....	0

trk - trunk selection bits
ctl - control bits
acc - access code
rad - receiver's adapter address
rpn - receiver's port number
sad - sender's adapter address
spn - sender's port number
op - operation code NAK function (12).
sln - sender's link number
rln - receiver's link number
rc - reason code (hexadecimal).
    01 - illegal function
    02 - invalid user number/password
    03 - File access error
    04 - I/O error
    05 - FTP protocol sequence error
    06 - File busy (retry later)
    07 - File being staged from secondary storage to disk (retry later).
txt - ASCII text describing the error condition. The text is terminated by a zero byte (NUL).

Figure 3-3 - FTP Negative Acknowledgment Message Format

3.2.4. User Validation Function (USR)

This function must be the first FTP function issued by a user FTP process after establishing a network connection. The server FTP process will respond with positive acknowledgment (ACK) if the user number/password is successfully validated. If validation fails, the server FTP process will send negative acknowledgment (NAK). This function may be issued as appropriate after initial validation to switch to new user numbers for processing of additional files. Figure 3-4 details the format and content of a USR message block.

0	1	2	3	4	5	6	7	8	9
trk	ctl	acc	acc	rad	rpn	sad	spn	0	op

A	B	C	D	E	F	10	11	12	13
sln	rln	0	0	0	0	0	0	0	0

14	15	16	17	18	19	1A	1B	1C	1D
uid	uid	uid	uid	uid	uid	uid	0	0	0

1E	1F	20	21	22	23	24	25	26	27
pwd	pwd	pwd	pwd	pwd	pwd	pwd	0	0	0

trk - trunk selection bits
ctl - control bits
acc - access code
rad - receiver's adapter address
rpn - receiver's port number
sad - sender's adapter address
spn - sender's port number
op - operation code - USR function (13).
sln - sender's link number
rln - receiver's link number
uid - user identifier (ASCII), left adjusted with zero fill.
pwd - user's account password (ASCII), left adjusted with zero fill.

Figure 3-4 - FTP User Message Format

3.2.5. Retrieve a Remote Data Set (GET)

This function is used to request a copy of a remote data set. The file specification string for the data set accompanies the message as an associated data block. The file specification consists of an ASCII text string terminated with a zero (NUL) character and is host dependent. The server FTP process is responsible for parsing the file specification. Legal responses are NAK and ACK. If the request is acknowledged, the server FTP process will begin the data transfer immediately after the ACK. Figure 3-5 details the format and content of a GET message block.

0	1	2	3	4	5	6	7	8	9
trk	ctl	acc	acc	rad	rpn	sad	spn	0	op

A	B	C	D		1E	1F	20	21
sln	rln	fty	0	...	fno	fno	fct	fct
1E	1F	20	21					

trk - trunk selection bits
ctl - control bits
acc - access code
rad - receiver's adapter address
rpn - receiver's port number
sad - sender's adapter address
spn - sender's port number
n/u - not used
op - operation code - GET function (14).
sln - sender's link number
rln - receiver's link number
fty - File type
    0 - ASCII text
    1 - CDC display code text
    2 - Binary data
fno - relative file offset of first file to be retrieved
fct - number of files to be retrieved.

Figure 3-5 - FTP Get Message Format

3.2.6. Store a Data Set Remotely (PUT)

This function is used to request that a data set be stored on the remote system. The file specification to be used in creating the file on the remote host is passed in an associated data block. The file specification is an ASCII text string terminated with a zero (NUL) character and is host dependent. It is the responsibility of the server FTP process to parse the file specification. Valid responses are NAK and ACK. Upon receipt of an ACK the requesting host may begin the data transfer. Figure 3-6 details the format and content of a PUT message block.

0	1	2	3	4	5	6	7	8	9
trk	ctl	acc	acc	rad	rpn	sad	spn	0	op

A	B	C
sln	rln	fty

trk - trunk selection bits
ctl - control bits
acc - access code
rad - receiver's adaptor address
rpn - receiver's port number
sad - sender's adapter address
spn - sender's port number
op - operation code - PUT function (15).
sln - sender's link number
rln - receiver's link number
fty - File type
  0 - ASCII text
  1 - CDC display code text
  2 - Binary data

Figure 3-6 - FTP Put Message Format

3.2.7. Delete a Remote Data Set (PUR)

This function is issued to remove a file from a remote host's file system. The file specification for the file to be removed is passed in an associated data block. The file specification consists of an ASCII text string terminated with a zero (NUL) character and is host dependent. Possible responses are ACK and NAK. Figure 3-7 details the format and content of a PUR message block.

0	1	2	3	4	5	6	7	8	9
trk	ctl	acc	acc	rad	rpn	sad	spn	0	op

A	B	C	D	E	F	10	11	12	13
sln	rln	0	0	0	0	0	0	0	0

trk - trunk selection bits
ctl - control bits
acc - access code
rad - receiver's adapter address
rpn - receiver's port number
sad - sender's adapter address
spn - sender's port number
op - operation code - PUR function (16).
sln - sender's link number
rln - receiver's link number

Figure 3-7 - FTP Purge Message Format

3.2.8. Retrieve Remote Directory Information (DIR)

This function is issued to request directory information from a remote host's file system. A directory specification may accompany the request in an associated data block. The directory specification consists of an ASCII text string terminated with a zero (NUL) character and is host dependent. It is the responsibility of the receiving server FTP process to parse the string. The remote system will respond with positive acknowledgment (ACK) if the directory is successfully accessed and follow immediately with data (DAT) blocks containing the directory data. Directory data consist of one or more ASCII text lines, formatted by the serving FTP process, and terminated with a zero (NUL) character. If directory access fails the server FTP process will send negative acknowledgment (NAK). Figure 3-8 details the format and content of a DIR message block.

0	1	2	3	4	5	6	7	8	9
trk	ctl	acc	acc	rad	rpn	sad	spn	0	op

A	B	C	D	E	F	10	11	12	13
sln	rln	0	0	0	0	0	0	0	0

trk - trunk selection bits
ctl - control bits
acc - access code
rad - receiver's adapter address
rpn - receiver's port number
sad - sender's adapter address
spn - sender's port number
op - operation code - DIR function (17).
sln - sender's link number
rln - receiver's link number

Figure 3-8 - FTP Directory Message Format

Appendix A - Sample NCP Protocol Exchange Sequences

This appendix provides examples of common NCP protocol exchanges. In the interest of brevity, the first 9 bytes of each message are ommitted since they are dictated by the hardware and are not a part of the NCP protocol. Each message description starts with the NCP operation code, given as a mnemonic, and is followed by a list parameters, separated by commas (,) which correspond positionally to the fields described in Section 2 of this document. Fields which contain character data are represented as character strings enclosed in quotes ("). The entire message description is bracketed by the characters < and >.

Example 1 - Connection open

Host A	Host B
<OPN,0,0,1,0,"SRVRFTP">	
	<OPN,0,0,2,1>

Example 2 - Connection refusal for insufficient links

Host A	Host B
<OPN,0,0,1,0,"SRVRFTP">	
	<RJT,0,0,3,1>

Example 3 - Connection refusal for process unknown or unavailable

Host A	Host B
<OPN,0,0,1,0,"BADPID">	
	<RJT,0,0,3,1>

Example 4 - Connection close exchange

Host A	Host B
<CLS,0,0,1,2>	
	<CLS,0,0,2,1>

Appendix B - Sample FTP
Protocol Exchange Sequences

This appendix provides examples of valid FTP protocol exchanges. In the interest of brevity, the first 9 bytes of each message are ommitted since they are dictated by the hardware and are not a part of the NCP protocol. Each description presumes the prior establishment of logical connection via the appropriate NCP protocol exchange. Each message description starts with the FTP operation code, given as a mnemonic, and is followed by a list of parameters, separated by commas (,) which correspond positionally to the fields described in Section 3 of this document. Fields which contain character data are represented as character strings enclosed in quotes ("). The entire message description is bracketed by the characters < and >. Associated data accompanying the message is represented as a string of text enclosed in the characters [and], and appended to the end of the message description.

Example 1 - User validation

Host A	Host B
<USR,1,2,"userid","paswrd">	
	<ACK,2,1>

Example 2 - User validation failure

Host A	Host B
<USR,1,2,"userid","paswrd">	
	<NAK,2,1,2,"Invalid user">

Example 3 - Retrieve remote file

Host A	Host B
<GET,1,2,0,0,0> [ANYFILE]	
	<ACK,2,1>
	<DAT,2,1,0> [First block of file]
	<DAT,2,1,0> [Second block of file]
	.
	.
	.
	<DAT,2,1,4> [Last block of file]

Example 4 - Store file remotely

```
      Host A                                        Host B

 <PUT,1,2,1> [ANYFILE/AC=1234567890]
                                             <ACK,2,1>
 <DAT,1,2,0> [First file block]
 <DAT,1,2,0> [Second file block]
          .
          .
          .
 <DAT,1,2,4> [Last file block]
```

Example 5 - Delete remote file.

```
      Host A                                        Host B

 <PUR,1,2> [ANYFILE/UN=HIS]
                                           <ACK,2,1>
```

Example 6 - Get remote directory information

```
      Host A                                        Host B

 <DIR,1,2> [/ETC/BIN/*]
                                           <ACK,2,1>
                                           <DAT,2,1,0> [First directory block]
                                           <DAT,2,1,0> [Second directory block]
                                           <DAT,2,1,4> [Last directory block]
```

INITIAL DISTRIBUTION

COPIES	
12	DIRECTOR DEFENSE TECHNICAL INFORMATION CENTER (DTIC)

CENTER DISTRIBUTION

COPIES		
1	18/1809	GLEISSNER, G. H.
2	1808	WILDY, D.
1	189	GRAY, G.
1	189.1	HIBBERT, D.
1	189.2	HAYDEN, H.
5	189.2	ALLEN, D.
1	1892.1	STRICKLAND, J.
1	1892.2	SOMMER, D.
5	1892.3	TINKER, R.
1	1894	SEALS, W.
1	1811	STABENAU, K.
1	1896	GLOVER, A.
1	182	CAMARA, A.
1	184	SCHOT, J.
1	185	CORIN, T.
1	187	ZUBKOFF, M.
1	522.1	LIBRARY, CARDEROCK
1	522.2	LIBRARY, ANNAPOLIS
1	93	PATENT COUNSEL

DTNSRDC ISSUES THREE TYPES OF REPORTS

1. DTNSRDC REPORTS, A FORMAL SERIES, CONTAIN INFORMATION OF PERMANENT TECHNICAL VALUE. THEY CARRY A CONSECUTIVE NUMERICAL IDENTIFICATION REGARDLESS OF THEIR CLASSIFICATION OR THE ORIGINATING DEPARTMENT.

2. DEPARTMENTAL REPORTS, A SEMIFORMAL SERIES, CONTAIN INFORMATION OF A PRELIMINARY, TEMPORARY, OR PROPRIETARY NATURE OR OF LIMITED INTEREST OR SIGNIFICANCE. THEY CARRY A DEPARTMENTAL ALPHANUMERICAL IDENTIFICATION.

3. TECHNICAL MEMORANDA, AN INFORMAL SERIES, CONTAIN TECHNICAL DOCUMENTATION OF LIMITED USE AND INTEREST. THEY ARE PRIMARILY WORKING PAPERS INTENDED FOR INTERNAL USE. THEY CARRY AN IDENTIFYING NUMBER WHICH INDICATES THEIR TYPE AND THE NUMERICAL CODE OF THE ORIGINATING DEPARTMENT. ANY DISTRIBUTION OUTSIDE DTNSRDC MUST BE APPROVED BY THE HEAD OF THE ORIGINATING DEPARTMENT ON A CASE-BY-CASE BASIS.

END

FILMED

2-85

DTIC