        FILE TRANSFER PROTOCOL (FTP) STATUS AND FURTHER COMMENTS

A number of HOSTs have working server and user FTPs now. The
following reflects the status of FTP implementations to the best of
my knowledge:

    BBN(A and B), SRI-ARC, UTAH, CASE, USC-ISI, CCA, MIT-AI MIT-
    Mathlab, MIT-DMCG, CMU, AMES-67, and SU-AI have fully functionning
    server and user FTPs.

    MIT-Multics has user and server FTPs but the server does not
    listen on socket 3 (it can be started by normal login and the
    command ftp_server). UCSB will soon have user and server FTP's.

The servers at all the TENEX systems are more or less identical
(developed by Bob Clements at BBN). The servers at MIT-AI and MIT-ML
are also identical (developed by Pitts Jarvis of MAC). Others
currently involved with FTP include Arvola Chan (AC@MIT-DMCG), Ken
Pogran (Multics), Greg Hicks (HICKS@UTAH), Wayne Hathaway (AMES-67),
Ralph Gorin (SU-AI), Rick Werme (CMU), and Ron Stoughton (UCSB).

The User-FTP or the user interface to FTP is where desirable and
interesting features can be put in. An example of such a features is
the BBN (and other TENEXes) "SNDMSG USER@HOST" feature which allows a
local user to send messages (or mail) to other network users. If the
remote host is not up, the message is stored as "--UNSENT-MAIL--
USERHOST" in the user's directory and a background job periodically
checks for such files to send mail. MIT-AI and MIT-ML have a "TRANS"
command which allows convenient transfer of files. At MIT-DMCG we
have developed under the "CALICO" subsystem, generalized commands
which allow local users to send mail, copy files efficiently, and
list users and directories over the network in a manner similar to
local usage (that is without having to explicitly connect, login, and
send commands to a remote HOST). We also allow TELNET, FTP, and RJS
users to automatically "login" and perform other command sequences
from an "initial" file.

It should be noted that file transfer between PDP-10's in "Image 36"
is an order of magnitude faster (and more efficient) than in "ASCII
8". Note also that it is useful to provide a "Quote" or "talk" mode
in user-FTP, to enable a user to input commands directly to the FTP
server (i.e. commands not implemented in user-FTP). It is desirable

that user and server FTP features and desirable modes of usage be
documented and reported via the RFC mechanism.

The following suggestions and additions pertain to the File Transfer
Protocol as stated in NWG/RFC 354 and NWG/RFC 385. After receiving
comments to this RFC, I will have the three RFC's combined into a
single document and have it issued as the ARPANET Official File
Transfer Protocol, very soon. It should however be noted that FTP is
an open-ended protocol with room for experimentation. New commands,
reply codes, data representation types, and file structures may be
defined in future. If two sites agree, they can define their own
experimental set of commands, data types, file structures, and/or
transfer modes. Such additions to the protocol should be well
documented and clearly specified so that other sites can also make
use of the same.

1) The FTP assumes line-at-a-time interaction with local acho. The
   server is not obliged to provide remote echo and may ignore TELNET
   control characters. A server however should not give error or bad
   response on receiving TELNET control characters.

   The server does not explicitly provide any editing capability such
   as character delete or line kill characters. All editing is
   assumed to be local. TIP users should use FTP in line mode and
   send both <CR> and <LF> (by TIP commands @T O L and @I L). In
   such a mode the TIP user can flush his current input line by the
   flush command (@F).

   The server should respond to the TELNET "SYNCH" by flushing the
   current command line and waiting for user input such as an "ABOR"
   command. Other commands such as "BYE" or "STAT" may also
   constitute an acceptable input.

2) Commands such as "STAT" which will produce more than one line of
   output over the TELNET connection, require some way of positively
   indicating the end of status information. It is proposed that a
   "200 status complete" reply give a positive indication for end of
   status information. The reply to STAT should begin with a line
   starting with 1xx (where x=digit), with the following lines not
   having a digit as their first character, and the status ending
   with the 200 reply. (Note that the requirement of three spaces is
   dropped in favour of the less restrictive requirement of the first
   character not being a digit.) This change would make operations
   much easier for both user and server FTPs.

3) A reminder that BYE<CR><LF> is legal. A space after a command
   name is not required if there is a null argument.

4) The following response are proposed to the "STAT" and "LIST"
   commands (this was not clearly specified specially for the null
   argument case). Responses to "STAT" and "LIST" shall always be
   over the TELNET and Data connections, respectively. The "LIST"
   command with null argument should produce a list of files in
   user's current working or default directory. The "STAT" command
   with null argument should (as suggested by Wayne Hathaway) produce
   tha status of all file transfer parameters (user, byte, size, data
   type, transfer mode, and file structure) if used between file
   transfers (i.e. no transfer in progress). If STAT is sent during
   a file transfer operation (accompanied with TELNET synch), the
   server should respond with the status of the operation in
   progress. If the argument of the "LIST" and "STAT" commands is a
   pathname, then a list associated with that pathname should be
   sent.

5) Two new commands are hereby proposed. First is a "HELP" command
   which should send to the user helpful hints about using the server
   and its implementation status (news). The information will be
   sent over the TELNET connection starting with type 100 reply and
   ending with a type 200 reply (completion). It is suggested that
   the use of this command and the "MAIL" and "BYE" commands be
   allowed without the user having to "login" (i.e., supplying valid
   user, password, and account).

   The other command (suggested by Bob Clements) is a new directory
   listing command called "NLST" which sends only the names of files
   (as valid pathname strings separated by CRLF) and no other useful
   but confusing information, so that it is possible to copy a whole
   directory automatically using this command and the store and
   retrieve commands. The syntax and format of this command is
   identical to the "LIST" command (for some HOSTs they may be
   identical commands).

6) Although the minimum implementation does not require the TYPE,
   BYTE, MODE, and STRU commands, it is suggested that these commands
   be accepted with the default values by even those having a minimum
   implementation. This would avoid some of the "ugly" error
   responses to input such as "TYPE A" and "STRU F", when these are
   perfectly acceptable to the server.

7) In using the "MLFL" and "LIST" commands, it is the user's (or
   user-FTP's) responsibility to ensure that the TYPE is ASCII (8-bit
   bytes). If the TYPE is other than ASCII, the server may send an
   error response and refuse the command. The user (or user-FTP)
   should therefore send the server "TYPE A" command if type is other
   than ASCII before sending the "MLFL" or "LIST" type commands.

8) A useful suggestion is to allow multiple user names in the "MAIL"
   and "MLFL" commands. Often a user wishes to send the same mail to
   a number of users at particular site. It would be very convenient
   if he can do this by doing a single transfer and command. It is
   strongly urged that server sites implement this option.

9) Another suggestion that has been made is to standardize pathname
   syntax in FTP. It appears that subdirectories will soon be
   introduced in the TENEX system. Perhaps that will have some
   bearing on the standard pathname syntax. The requirements of any
   pathname standard scheme are that it should allow convenient use
   of local pathname conventions, and not conflict with it. A
   reasonable proposal seems to be to have the standard pathname
   start with a special character such as ">" (as in Multics), and to
   use this special character to separate the elements of a pathname.
   If the special character happens to ba valid part of a pathname
   element, use the literal quote convention of "'>" (single quote to
   precede the special character).

   Examples of pathnames under this convention would be:

       >udd>CNet>Doe>foo_bar   (for Multics)
       >DSK>JFD>foo bar   (for ITS)
       >DOE>foo.bar1   (for TENEX)

10) The requirement of account numbers by TENEXes has caused a
    certain problems for automatons using FTP, under the present
    reply code sequences. Therefore two new reply codes are defined
    to handle the account requirement. A reply code of "331 Enter
    Account" shall be used if an account is required as part of user
    "login" sequence. A reply code of "433 Cannot store files
    without valid account. Enter account." be used if an account is
    required only for a particular operation such as store.

11) The following suggestions made by Wayne Hathaway (RFC
    forthcoming) seem reasonable and should be included in the
    Protocol:

    i) The following End-of-Record condition should be explicit on
    last record, and not implied in an End-of-File. This change
    would simplify server implementation and improve reliability
    (better error control).

    ii) Implementors of user-FTP's should note that it is trivial for
    them to implement record structures in ASCII type and Stream mode
    (the default CRLF convention for end-of-record). All user-FTPs
    should allow store or retrieve of record structured files with
    ASCII type and stream mode.

iii) It is possible to send record strutured "print-file" types
(in addition to ASCII type) in either stream or text modes. (RFC
354 was not clear on this issue).

iv) The TELNET synch mechanism should be extended to other
commands such as BYE and STAT in addition to ABOR.

v) Comments are invited on the desirability of NOOP, CLSE, and
SRVR commands. In my opinion a STAT command with null argument
serves the purpose of NOOP (to see if server is still alive), and
BYE serves the purpose of CLSE (USER command should be used to
change user name). SRVR is a useful command.

12) Bob Clements raised the old issued of error detection and control
    again. To handle this we can define two new descriptor codes in
    the Block mode, one that signals start of block check, and the
    other that indicates end of block check (and includes the block
    check bytes). These codes may be ignored by any site not wishing
    to implement the error detection scheme. Your comments on the
    error check scheme and the desirability of its inclusion in FTP
    are solicited.

     [This RFC was put into machine readable form for entry]
     [into the online RFC archives by Helene Morin, Via Genie, 12/99]