MAIL TRANSFER PROTOCOL

PREFACE

   This is a first draft of this protocol and comments are very
   definitely requested.

INTRODUCTION

   The objective of Mail Transfer Protocol (MTP) is to transfer mail
   reliably and efficiently.

   This paper assumes knowledge of the following protocols described in
   the ARPA Internet Protocol Handbook. The reader will note strong
   similarities to portions of the File Transfer Protocol; in part, this
   is due to the original ARPA Network implementation of computer mail
   as a feature of FTP.

      The ARPANET Host-to-Host Protocol [Network Control Protocol] (NCP)

      The Transmission Control Protocol (TCP)

      The TELNET Protocol (TELNET)

      The File Transfer Protocol (FTP)

DISCUSSION

   In this section, the terminology and the MTP model are discussed.
   The terms defined in this section are only those that have special
   significance in MTP. Some of the terminology is very specific to the
   MTP model; some readers may wish to turn to the section on the MTP
   model while reviewing the terminology.

   TERMINOLOGY

      ASCII

         The ASCII character set as defined in the ARPA Internet
         Protocol Handbook. In MTP, ASCII characters are defined to be
         the lower half of an eight-bit code set (i.e., the most
         significant bit is zero) and is called NVT-ASCII.

control connection

   The TCP full-duplex communication path or two NCP simplex
   communication paths between a sender-MTP and a receiver-MTP for
   the exchange of commands, replies, and mail text. The control
   connection operates according to the TELNET Protocol.

data mode

   The mail is transmitted over the control connection as a stream
   of octets. (In FTP terminology this is called stream mode.)

data structure

   The internal structure of mail is considered to be a continuous
   sequence of data octets. (In FTP terminology this is called
   file-structure.)

data representation

   The internal representation of all data (i.e., mail) is in
   NVT-ASCII.

host

   A computer in the internetwork environment on which mailboxes
   reside.

MTP commands

   A set of commands which comprise the control information
   flowing from the sender-MTP to the receiver-MTP.

mail

   An ordered set of computer data of arbitrary length, which
   conforms to the standard set in RFC 733 (Standard for the
   Format of ARPA Network Text Messages).

mailbox

   A character string (address) which identifies a user to whom
   mail is to be sent. Mailbox normally consists of the host and
   user specifications. The standard mailbox naming convention is
   defined to be "user@host". Additionally, the "container" in
   which mail is stored.

NVT

   The Network Virtual Terminal as defined in the TELNET Protocol.

octet

   Bytes in MTP are octets (8 bits). This is not necessarily the
   same byte size in which data is stored in a host.

reply

   A reply is an acknowledgment (positive or negative) sent from
   receiver to sender via the control connection in response to a
   MTP command. The general form of a reply is a completion code
   (including error codes) followed by a text string. The codes
   are for use by programs and the text is usually intended for
   human users.

receiver-MTP process

   A process which transfers mail in cooperation with a sender-MTP
   process. It "listens" on its port/socket L for a connection
   from a sender-MTP and establishes a control connection using
   the TELNET Protocol. It receives MTP commands from the
   sender-MTP, sends replies, and governs the transfer of mail.

sender-MTP process

   A process which transfers mail in cooperation with a
   receiver-MTP process. A local language may be used in the user
   interface command/reply dialogue. The sender-MTP initiates the
   control connection from its port/socket U to the receiver-MTP
   process. It initiates MTP commands, receives replies, and
   governs the transfer of mail.

user

   A human being (or a process on behalf of a human being) wishing
   to obtain mail transfer service. In addition, a recipient of
   computer mail.

THE MTP MODEL

   With the above definitions in mind, the following model (shown in
   Figure 1) may be diagrammed for an MTP service.
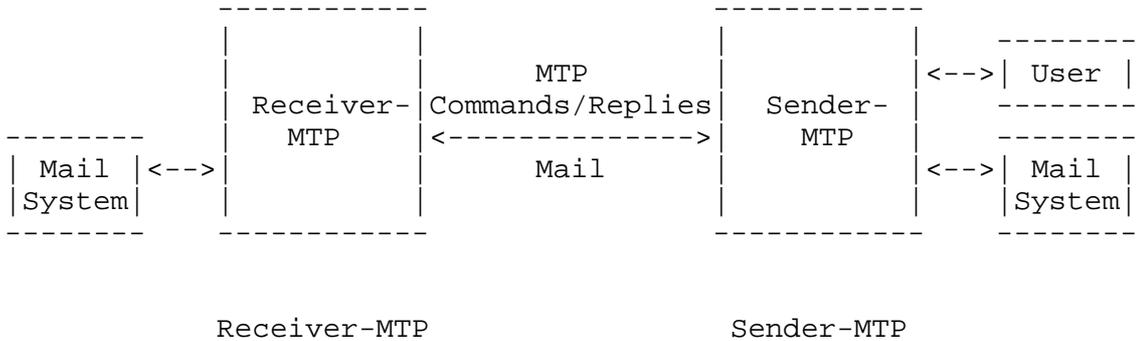
```
            ------------                    ------------
            |          |                    |          |   --------
            |          |        MTP         |          |<-->| User |
            | Receiver-|Commands/Replies| Sender-  |   --------
 --------   |   MTP    |<--------------->|   MTP    |   --------
 | Mail |<-->|          |        Mail        |          |<-->| Mail |
 |System|   |          |                    |          |   |System|
 --------   ------------                    ------------   --------


        Receiver-MTP                      Sender-MTP

                    Model for MTP Use

                        Figure 1
```

   In the model described in Figure 1, the sender-MTP initiates the
   TCP/NCP control connection which follows the TELNET Protocol. At
   the initiation of the user, standard MTP commands are generated by
   the sender-MTP and transmitted to the receiver-MTP via the control
   connection. Standard replies are sent from the receiver-MTP to
   the sender-MTP over the control connection in response to the
   commands. In addition, mail is sent over the control connection.

MAIL TRANSFER FUNCTIONS

   The control connection is used for the transfer of commands which
   describe the functions to be performed, the replies to commands, as
   well as the actual transfer of mail. Mail is transferred only via
   the control connection.

   The communication channel from the sender-MTP to the receiver-MTP is
   established by a TCP/NCP control connection from the sender to a
   standard receiver port/socket. The sender-MTP is responsible for
   sending MTP commands, interpreting the replies received, and sending
   the mail; the receiver-MTP interprets commands, sends replies, and
   receives the mail.

MAIL REPRESENTATION AND STORAGE

Mail is transferred from a storage device in the sending host to a
storage device in the receiving host. It may be necessary to
perform certain transformations on the mail because data storage
representations in the two systems are different. For example,
NVT-ASCII has different data storage representations in different
systems. PDP-10's generally store NVT-ASCII as five 7-bit ASCII
characters, left-justified in a 36-bit word. 360's store
NVT-ASCII as four 8-bit EBCDIC codes in a 32-bit word. Multics
stores NVT-ASCII as four 9-bit characters in a 36-bit word.

For the sake of simplicity, all data must be represented in MTP as
NVT-ASCII. This means that characters must be converted into the
standard NVT-ASCII representation when transmitting text,
regardless of whether the sending and receiving hosts are
dissimilar. The sender converts the data from its internal
character representation to the standard 8-bit NVT-ASCII
representation (see the TELNET specification). The receiver
converts the data from the standard form to its own internal form.
In accordance with this standard, the <CRLF> sequence should be
used to denote the end of a line of text.

The mail in MTP has no internal structure and is considered to be
a continuous sequence of data octets.

ERROR RECOVERY AND RESTART

There is no provision for detecting bits lost or scrambled in data
transfer; this level of error control is handled by the TCP/NCP.
In addition, there is no restart procedure provided to protect
senders from gross system failures (including failures of a host,
an MTP-process, or the underlying network).

MTP COMMANDS

COMMAND SEMANTICS

The MTP commands define the mail transfer or the mail system
function requested by the user. The syntax of mailboxes must
conform to receiver site conventions (with standard defaults
applicable). In response to an MTP transfer command, the mail
shall always be transferred over the control connection.

The Mail Transfer Protocol follows the specifications of the
TELNET Protocol for all communications over the control

connection. Although the language used for TELNET communication
can be a negotiated option, the "TELNET language" and the
corresponding "TELNET end of line code" are required to be
NVT-ASCII and <CRLF> respectively. No other specifications of the
TELNET Protocol will be cited.

MTP commands are NVT-ASCII strings terminated by <CRLF>. The
command codes themselves are alphabetic characters terminated by
the character <SP> (space) if parameters follow and <CRLF>
otherwise.

The MTP commands are discussed below. In the description of a few
of the commands in this section the possible replies are given
explicitly. MTP replies are discussed in the next section.

   MAIL (MAIL)

      This command allows a sender-MTP to send mail over the
      control connection. The argument field contains a sender
      and optional path sequence. If the path sequence is
      present, it consists of an optional list of hosts and a
      destination mailbox. When the list of hosts is present, it
      is source routing information and indicates that the mail
      must be forwarded to the first host on the list. Following
      this command line the receiver treats all subsequent
      characters as mail text from the sender. The mail text is
      terminated by the character sequence "CRLF.CRLF".

      As mail is forwarded along the path sequence, each
      forwarding host must remove itself from the list. When mail
      reaches its ultimate destination (the path sequence has only
      a (possibly empty) destination mailbox), the receiver
      inserts it into the destination mailbox in accordance with
      its host mail conventions. If the second argument field is
      blank (one or more spaces) or empty (<CRLF>), the mail is
      destined for a printer or other designated place for site
      general delivery mail. The mail may be marked as sent from
      the sender as specified by the first argument field.

   MAIL RECIPIENT SCHEME QUESTION (MRSQ)

      This MTP command is used to select a scheme for the
      transmission of mail to several users at the same host. The
      schemes are to list the recipients first, or to send the
      mail first.

MAIL RECIPIENT (MRCP)

This command is used to identify the individual recipients
of the mail in the transmission of mail for multiple users
at one host.

HELP (HELP)

This command causes the receiver to send helpful information
regarding its implementation status over the control
connection to the receiver. The command may take an
argument (e.g., any command name) and return more specific
information as a response. The reply is type 211 or 214.

QUIT (QUIT)

This command specifies that the receiver must close the
control connection.

NOOP (NOOP)

This command does not affect any parameters or previously
entered commands. It specifies no action other than that
the receiver send an OK reply.

COMMAND SYNTAX

The commands (and their functions and semantics) are TELNET
NVT-ASCII strings transmitted over the control connection. The
functions and semantics of commands are described in the section
on MTP Commands. The reply sequences are discussed in the section
on Sequencing of Commands and Replies. Scenarios illustrating the
use of commands are provided in the section on Typical MTP
Scenarios. The command syntax is specified in this section.

The commands begin with a command code followed by an argument
field. The command codes are four alphabetic characters. Upper
and lower case alphabetic characters are to be treated
identically. Thus any of the following may represent the mail
command:

    MAIL   Mail   mail   MaIl   mAIl

This also applies to any symbols representing parameter values,
such as R or r for RECIPIENT first. The command codes and the
argument fields are separated by one or more spaces.

The argument field consists of a variable length character string
ending with the character sequence <CRLF>. It should be noted
that the receiver is to take no action until the end of line code
is received.

The syntax is specified below in NVT-ASCII. All characters in the
argument field are ASCII characters. Square brackets denote an
optional argument field. If the option is not taken, the
appropriate default is implied.

The following are the MTP commands:

    MAIL <SP> FROM:<sender> [<SP> TO:<path>] <CRLF>

    MRSQ [<SP> <scheme>] <CRLF>

    MRCP <SP> TO:<path> <CRLF>

    HELP [<SP> <string>] <CRLF>

    QUIT <CRLF>

    NOOP <CRLF>

The syntax of the above argument fields (using BNF notation where
applicable) is given below. The "..." notation indicates that a
field may be repeated one or more times.

    <sender> ::= "<" <mailbox> ">"

    <path> ::= "<" ["@" <host> "," ...] <mailbox> ">"

    <scheme> ::= "R" | "T" | "?"

    <string> ::= <char> | <char><string>

    <mailbox> ::= <user> "@" <host>

    <host> ::= <string>

    <user> ::= <string>

    <char> ::= any of the 128 ASCII characters except <CR> and <LF>

     CONTROL FUNCTIONS

        Most time-sharing systems provide mechanisms to allow a terminal
        user to regain control of a "runaway" process. When used locally,
        such systems have access to all user-supplied signals, whether
        these are normal characters or special "out of band" signals.
        When terminals are connected to the system through the network,
        the system does not necessarily have access to all user signals;
        the network's flow control mechanisms may cause such signals to be
        buffered elsewhere, for example in the user's host.

        To counter this problem, the TELNET "Synch" mechanism is used. A
        Synch signal consists of a TCP Urgent or an NCP Interrupt
        notification, coupled with the TELNET command DATA MARK (DM).
        This notification, which is not subject to the flow control
        pertaining to the TELNET connection, is used to invoke special
        handling of the data stream by the process which receives it. In
        this mode the data stream is immediately scanned for a TELNET
        Interrupt Process (IP) command. (The rationale for the use of the
        TELNET IP command is to allow an existing server TELNET module to
        sit "under" the MTP. If this code were directly implemented in
        the MTP the IP command would be unnecessary.) The TELNET command
        DM is the synchronizing mark in the data stream which indicates
        that any special signal has already occurred and the recipient can
        return to normal processing of the data stream. For a more
        complete understanding of this mechanism, see the TELNET Protocol
        Specification in the Internet Protocol Handbook.

        The effect of this mechanism is to to discard all characters (up
        to the DM) between the sender of the Synch and its recipeint.
        Thus, all characters in the control connection are ignored until
        the TELNET command DM is received. The full sequence is
        illustrated below. Each vertical bar (|) represents the boundary
        between data octets; IAC refers to the TELNET command code
        Interpret As Command.

```
                  Old                         New
                 -+-+-+-+-+-----+---+--+---+--+-
              ...|M|A|I|L|  ... |IAC|IP|IAC|DM|...
                 -+-+-+-+-+-----+---+--+---+--+-
```

MTP REPLIES

   Replies to Mail Transfer Protocol commands are devised to ensure the
   synchronization of requests and actions in the process of mail
   transfer, and to guarantee that the sender-MTP always knows the state
   of the receiver. Every command must generate at least one reply,
   although there may be more than one. In the latter case, the
   multiple replies must be easily distinguished. Additionally, some
   commands must occur sequentially, such as MRSQ T->MAIL->MRCP or
   MRSQ R->MRCP->MAIL. Replies to these sequences show the existence of
   an intermediate state if all preceding commands have been successful.
   A failure at any point in the sequence necessitates the repetition of
   the entire sequence from the beginning.

      The details of the command-reply sequence are made explicit in the
      section on State Diagrams.

   An MTP reply consists of a three digit number (transmitted as three
   alphanumeric characters) followed by some text. The number is
   intended for use by automata to determine what state to enter next;
   the text is meant for the human user. It is intended that the three
   digits contain enough encoded information that the sender-MTP will
   not need to examine the text and may either discard it or pass it on
   to the user, as appropriate. In particular, the text may be
   receiver-dependent, so there are likely to be varying texts for each
   reply code.

   Formally, a reply is defined to be the sequence: a three-digit code,
   space <SP>, one line of text (where the maximum line length is 65),
   and a terminal <CRLF>. Occasionally the text is longer than a single
   line; in these cases the complete text must be bracketed so the
   sender-MTP knows when it can stop reading the reply. This requires a
   special first line format to indicate a multiple line reply, and
   another on the last line to so designate it. Both lines will contain
   the appropriate reply code which indicates the transaction state.

      Thus the format for multi-line replies is that the first line will
      begin with the exact required reply code, followed immediately by
      a Hyphen, "-" (also known as minus), followed by text. The last
      line will begin with the same code, followed immediately by space
      <SP>, optionally some text, and <CRLF>.

        For example:
                              123-First line
                              Second line
                                234 A line beginning with numbers
                              123 The last line

    The sender-MTP then simply needs to search for the second
    occurrence of the same reply code followed by <SP> (space> at the
    beginning of a line, and ignore all intermediary lines. If an
    intermediary line begins with a three-digit number, the receiver
    must pad the front to avoid confusion.

        This scheme allows standard system routines to be used for
        reply information, with "artificial" first and last lines
        tacked on. In the rare cases where these routines are able to
        generate three digits and a space at the beginning of any line,
        the beginning of each text line should be offset by some
        neutral text, like space.

        This scheme assumes that multi-line replies may not be nested. In
        general, reply nesting will not occur except for random system
        messages (also called spontaneous replies) which may interrupt
        another reply. System messages (i.e., those not processed by the
        receiver-MTP) will NOT carry reply codes and may occur anywhere in
        the command-reply sequence. They may be ignored by the sender-MTP
        as they are only information for the human user.

    The three digits of the reply each have a special significance. This
    is intended to allow a range of very simple to very sophisticated
    response by the sender-MTP. The first digit denotes whether the
    response is good, bad or incomplete. (Referring to the state
    diagram) an unsophisticated sender-MTP will be able to determine its
    next action (proceed as planned, redo, retrench, etc.) by simply
    examining this first digit. A sender-MTP that wants to know
    approximately what kind of error occurred (e.g., mail system error,
    command syntax error) may examine the second digit, reserving the
    third digit for the finest gradation of information.

        There are five values for the first digit of the reply code:

        1yz   Positive Preliminary reply

            The requested action is being initiated; expect another
            reply before proceeding with a new command. (The sender-MTP
            sending another command before the completion reply would be

in violation of protocol. However, receiver-MTP processes
should queue any commands that arrive while a preceding
command is in progress.)

2yz   Positive Completion reply

The requested action has been successfully completed. A new
request may be initiated.

3yz   Positive Intermediate reply

The command has been accepted, but the requested action is
being held in abeyance, pending receipt of further
information. The sender-MTP should send another command
specifying this information. This reply is used in command
sequence groups.

4yz   Transient Negative Completion reply

The command was not accepted and the requested action did
not occur. However, the error condition is temporary and
the action may be requested again. The sender should return
to the beginning of the command sequence (if any). It is
difficult to assign a meaning to "transient" when two
different sites (receiver- and sender- MTPs) must agree on
the interpretation. Each reply in this category might have
a different time value, but the sender-MTP is encouraged to
try again. A rule of thumb to determine if a reply fits
into the 4yz or the 5yz category (see below) is that replies
are 4yz if they can be repeated without any change in
command form or in properties of the sender or receiver.
(E.g., the command is repeated identically; the receiver
does not put up a new implementation).

5yz   Permanent Negative Completion reply

The command was not accepted and the requested action did
not occur. The sender-MTP is discouraged from repeating the
exact request (in the same sequence). Even some "permanent"
error conditions can be corrected, so the human user may
want to direct the sender-MTP to reinitiate the command
sequence by direct action at some point in the future (e.g.,
after the spelling has been changed, or the user has altered
his/her directory status.)

The second digit encodes responses in specific categories:

    x0z   Syntax -- These replies refer to syntax errors,
           syntactically correct commands that don't fit any
           functional category, and unimplemented or superfluous
           commands.

    x1z   Information -- These are replies to requests for
           information, such as status or help.

    x2z   Connections -- These are replies referring to the control
           connection.

    x3z   Unspecified as yet.

    x4z   Unspecified as yet.

    x5z   Mail system -- These replies indicate the status of the
           receiver mail system vis-a-vis the requested transfer or
           other mail system action.

The third digit gives a finer gradation of meaning in each
category specified by the second digit. The list of replies below
will illustrate this. Each reply text is recommended rather than
mandatory, and may even change according to the command with which
it is associated. On the other hand, the reply codes must
strictly follow the specifications in this section. Receiver
implementations should not invent new codes for slightly different
situations from the ones described here, but rather adapt codes
already defined.

    A command such as NOOP whose successful execution does not
    offer the sender-MTP any new information will return a 200
    reply. The response is 502 when the command requests an
    unimplemented non-site-specific action. A refinement of that
    is the 504 reply for a command that IS implemented, but that
    requests an unimplemented parameter.

REPLY CODES BY FUNCTION GROUPS

    200 Command okay
    500 Syntax error, command unrecognized
       [This may include errors such as command line too long]
    501 Syntax error in parameters or arguments
    502 Command not implemented
    503 Bad sequence of commands

211 System status, or system help reply
214 Help message
   [Information on how to use the receiver or the meaning of a
   particular non-standard command; this reply is useful only to
   the human user]
215 <scheme> is the preferred scheme

120 <host> Service ready in nnn minutes
220 <host> Service ready for new user
221 <host> Service closing control connection
421 <host> Service not available, closing control connection
   [This may be a reply to any command if the service knows it
   must shut down]

151 User not local; will forward to <user>@<host>
152 User unknown; mail will be forwarded by the operator
250 Requested mail action okay, completed
450 Requested mail action not taken: mailbox unavailable
   [E.g., mailbox busy]
550 Requested action not taken: mailbox unavailable
   [E.g., mailbox not found, no access]
451 Requested action aborted: local error in processing
452 Requested action not taken: insufficient system storage space
552 Requested mail action aborted: exceeded storage allocation
   [For current mailbox location]
553 Requested action not taken: mailbox name not allowed
354 Start mail input; end with <CR><LF>.<CR><LF>

   NUMERIC ORDER LIST OF REPLY CODES

   120 <host> Service ready in nnn minutes
   151 User not local; will forward to <user>@<host>
   152 User unknown; mail will be forwarded by the operator
   200 Command okay
   211 System status, or system help reply
   214 Help message
      [Information on how to use the receiver or the meaning of a
      particular non-standard command; this reply is useful only to
      the human user]
   215 <scheme> is the preferred scheme
   220 <host> Service ready for new user
   221 <host> Service closing control connection
   250 Requested mail action okay, completed
   354 Start mail input; end with <CR><LF>.<CR><LF>

        421 <host> Service not available, closing control connection
            [This may be a reply to any command if the service knows it
            must shut down]
        450 Requested mail action not taken: mailbox unavailable
            [E.g., mailbox busy]
        451 Requested action aborted: local error in processing
        452 Requested action not taken: insufficient system storage space
        500 Syntax error, command unrecognized
            [This may include errors such as command line too long]
        501 Syntax error in parameters or arguments
        502 Command not implemented
        503 Bad sequence of commands
        550 Requested action not taken: mailbox unavailable
            [E.g., mailbox not found, no access]
        552 Requested mail action aborted: exceeded storage allocation
            [For current mailbox location]
        553 Requested action not taken: mailbox name not allowed

DISCUSSION OF MAIL TRANSFER

   The basic command for transmitting mail is MAIL. This command causes
   the transmitted data to be entered into the recipient's mailbox.

      MAIL <SP> "FROM:" <sender> [<SP> "TO:" <path>] <CRLF>

         <sender> is a mailbox and <path> is a source routing list of
         hosts and destination mailbox. If accepted, it returns a 354
         reply and considers all succeeding lines to be the message
         text. It is terminated by a line containing only a period,
         upon which a 250 completion reply is returned. Various errors
         are possible.

   There are two possible preliminary replies that a receiver may use to
   indicate that it is accepting mail for a user whose mailbox is not at
   that receiver.

      151 User not local; will forward to <user>@<host>

         This reply indicates that the receiver knows the user's mailbox
         is on another host and will take responsibility for forwarding
         the mail to that host. For example, at BBN (or ISI) there are
         several hosts. Each has a list of many of the users on the
         hosts. Each host can accept mail for any user on their list
         and forward it to the correct host.

                                   15

   152 User Unknown; mail will be forwarded by the operator

      This reply indicates that the host does not recognize the user
      name, but that it will accept the mail and have the operator
      attempt to deliver it. This is useful if the user name is
      misspelled, but may be a disservice if the mail is really
      undeliverable.

If forwarding by the operator is unacceptable or if the user would
prefer to send the mail directly to the recipient's actual host, the
dialogue may be terminated upon receipt of one of these preliminary
responses.

There are two MTP commands which allow the text of a message to be
mailed to several recipients simultaneously; such message
transmission is far more efficient than the practice of sending the
text again and again for each additional recipient at a site. In
one, all recipients are specified first, and then the text is sent.
In the other, the order is reversed and the text is sent first,
followed by the recipients. Both schemes are necessary because
neither by itself is optimal for all systems, as will be explained
later. To select a particular scheme, the MRSQ command is used; to
specify recipients after a scheme is chosen, MRCP commands are given;
and to furnish text, the MAIL command is used.

SCHEME SELECTION: MRSQ

   MRSQ is the means by which a sender-MTP can test for MRSQ/MRCP
   implementation, select a particular scheme, reset its state, and
   even do some rudimentary negotiation. Its format is as follows:

      MRSQ [<SP> <scheme>] <CRLF>

      <scheme> is a single character. The following are defined:
         R Recipients first. If this is not implemented, T must be.
         T Text first. If this is not implemented, R must be.
         ? Request for preference. This must always be implemented.

         No argument means a "selection" of none of the schemes (the
         default).

      Possible replies are:
         200 OK, we'll use specified scheme
         215 <scheme> This is the scheme I prefer
         501 I understand MRSQ but can't use that scheme
         5xx Command unrecognized or unimplemented

There are three aspects of MRSQ. The first is that an MRSQ with
no argument must always return a 200 reply and restore the default
state of having no scheme selected. Any other reply implies that
MRSQ and hence MRCP are not understood or cannot be performed
correctly.

The second is that the use of "?" as a <scheme> asks the MTP
receiver to return a 215 reply in which the receiver specifies a
"preferred" scheme. The format of this reply is simple:

    215 <SP> <scheme> [<SP> <arbitrary text>] <CRLF>

    Any other reply (e.g., 4xx or 5xx) implies that MRSQ and MRCP
    are not implemented, because "?" must always be implemented if
    MRSQ is.

The third important point about MRSQ is that it always has the
side effect of resetting all schemes to their initial state. This
reset must be done no matter what the reply will be -- 200, 215,
or 501. The actions necessary for a reset will be explained when
discussing how each scheme actually works.

MESSAGE TEXT SPECIFICATION: MAIL

Regardless of which scheme (if any) has been selected, a MAIL
command with a non-null "TO" argument will behave exactly as
before; the MRSQ/MRCP commands have no effect on it. However, a
normal MAIL command does have the same side effect as MRSQ; it
"resets" the current scheme to its initial state.

It is only when the "TO" argument is null (e.g., MAIL FROM:<X@Y>
<CRLF>) that the particular scheme chosen is important. Rather
than producing an error (as most receivers currently do), the
receiver will accept message text for this "null" specification.
What it does with it depends on which scheme is in effect, and
will be described in the section on Scheme Mechanics.

RECIPIENT SPECIFICATION: MRCP

In order to specify recipient names (i.e., mailboxes) and receive
some acknowledgment (or refusal) for each name, the following
command is used:

    MRCP <SP> TO:<path> <CRLF>

    Reply for no scheme:
        503 No scheme specified yet; use MRSQ
    Replies for scheme T are identical to those for MAIL.
    Replies for scheme R (recipients first):
        200 OK, name stored
        452 Recipient table full, this name not stored
        553 Recipient name rejected
        4xx Temporary error, try this name again later
        5xx Permanent error, report to sender

Note that use of this command is an error if no scheme has been
selected yet; an MRSQ <scheme> must have been given if MRCP is to
be used.

SCHEME MECHANICS: MRSQ R (RECIPIENTS-FIRST)

In the recipients-first scheme, MRCP is used to specify names
which the MTP receiver stores in a list or table. Normally the
reply for each MRCP will be either a 200 for acceptance or a
4xx/5xx rejection code. All 5xx codes are permanent rejections
(e.g., user not known) which should be reported to the human user,
whereas 4xx codes in general connote some temporary error that may
be rectified later. None of the 4xx/5xx replies impinge on
previous or succeeding MRCP commands, except for 452 which
indicates that no further MRCPs will succeed unless a message is
sent to the already stored recipients or a reset is done.

Sending message text to stored recipients is done by giving a MAIL
command with no "TO" argument; that is, just MAIL <SP> <sender>
<CRLF>. Transmission of the message text is exactly the same as
for normal MAIL. However, a positive acknowledgment at the end of
transmission means the message has been sent to ALL recipients
that were remembered with MRCP, and a failure code means that it
should be considered to have failed for ALL of these specified
recipients. This applies regardless of the actual error code.
Regardless of what the reply signifies, all stored recipient names
are flushed and forgotten -- in other words, things are reset to
their initial state. This purging of the recipient name list must
also be done as the reset side effect of any use of MRSQ.

A 452 reply to an MRCP can be handled by using MAIL to specify the
message for currently stored recipients, and then sending more
MRCPs and another MAIL, as many times as necessary. For example,
if a receiver only had room for 10 names this would result in a
50-recipient message being sent 5 times, to 10 different
recipients each time.

If a sender attempts to specify message text (MAIL with no "TO"
argument) before any successful MRCPs have been given, this should
be treated exactly as a "normal" MAIL with a null recipient would
be; some receivers return an error, such as "550 Null recipient".

See the example in Appendix A for a mail transfer using MRSQ R.

SCHEME MECHANICS: MRSQ T (TEXT-FIRST)

In the text-first scheme, MAIL with no "TO" argument is used to
specify message text, which the receiver stores away. Succeeding
MRCPs are then treated as if they were MAIL commands, except that
none of the text transfer manipulations are done; the stored
message text is sent to the specified recipient, and a reply code
is returned identical to that which an actual MAIL would invoke.
(Note that ANY 2xx code indicates success.)

The stored message text is not forgotten until the next MAIL or
MRSQ, which will either replace it with new text or flush it
entirely. Any use of MRSQ will reset this scheme by flushing
stored text, as will any use of MAIL with a non-null argument.

If an MRCP is seen before any message text has been stored, the
sender in effect is trying to send a null message; some receivers
might allow this, others would return an error code.

See the example in Appendix B for a mail transfer using MRSQ T.

WHY TWO SCHEMES ANYWAY?

Because neither by itself is optimal for all systems. MRSQ R
allows more of a "bulk" mailing because everything is saved up and
then mailed simultaneously. This is very useful for systems such
as ITS where the MTP-receiver does not itself write mail directly,
but hands it on to a central mailer demon of great power. The
more information (e.g., recipients) associated with a single
"hand-off", the more efficiently mail can be delivered.

By contrast, MRSQ T is geared to receiver-MTPs which want to
deliver mail directly, in one-by-one incremental fashion. For
each given recipient this scheme returns an individual
success/failure reply code which may depend on variable mail
system factors such as exceeding disk allocation, mailbox access
conflicts, and so forth. If these receiver-MTPs tried to emulate
MRSQ Rs bulk mailing, they would have to ensure that a success
reply to the MAIL indeed meant that it had been delivered to ALL
recipients specified -- not just some.

NOTES:

* Because these commands are not required in the minimum
  implementation of MTP, one must be prepared to deal with sites
  which don't recognize either MRSQ or MRCP. "MRSQ" and "MRSQ ?"
  are explicitly designed as tests to see whether either scheme is
  implemented. MRCP is not designed as a test, and a failure
  return of the "unimplemented" variety could be confused with "No
  scheme selected yet", or even with "Recipient unknown".

* There is no way to indicate in a positive response to "MRSQ ?"
  that the preferred "scheme" for a receiver is that of the
  default state; i.e., none of the multi-recipient schemes. The
  rationale is that in this case, it would be pointless to
  implement MRSQ/MRCP at all, and the response would therefore be
  negative.

   * One reason that the use of MAIL is restricted to null "TO"
     arguments with this multi-recipient extension is the ambiguity
     that would result if a non-null "TO" argument were allowed. For
     example, if MRSQ R was in effect and some MRCPs had been given,
     and a MAIL FROM:<X@Y> TO:<FOO><CRLF> was done, there would be no
     way to distinguish a failure reply for mailbox "FOO" from a
     global failure for all recipients specified. A similar
     situation exists for MRSQ T; it would not be clear whether the
     text was stored and the mailbox failed, or vice versa, or both.

   * "Resets" are done by all MRSQs and "normal" MAILs to avoid
     confusion and overly complicated implementation. The MRSQ
     command implies a change or uncertainty of status, and the MAIL
     command would otherwise have to use some independent mechanisms
     to avoid clobbering the data bases (e.g., message text storage
     area) used by the T/R schemes. However, once a scheme is
     selected, it remains "in effect" just as an FTP "TYPE A" remains
     selected. The recommended way for doing a reset, without
     changing the current selection, is with "MRSQ ?". Remember that
     "MRSQ" alone reverts to the no-scheme state.

   * It is permissible to intersperse other MTP commands among the
     MRSQ/MRCP/MAIL sequences.

DECLARATIVE SPECIFICATIONS

   MINIMUM IMPLEMENTATION

      In order to make MTP workable without needless error messages, the
      following minimum implementation is required for all receivers:

         COMMANDS -- QUIT
                     MAIL
                     NOOP

      In terms of FTP, the values of the transfer parameters must be:

         TYPE -- ASCII
         MODE -- STREAM
         STRU -- FILE-STRUCTURE

      All hosts must use the above values for mail transfer.

   CONNECTIONS

      The receiver-MTP shall "listen" on Port L. The sender-MTP shall
      initiate the TCP/NCP control connection. The control connection
      consists of a full-duplex connection under TCP; it is two simplex
      connections under NCP. Receiver- and sender- MTPs should follow
      the conventions of the TELNET Protocol as specified in the ARPA
      Internet Protocol Handbook. Receivers are under no obligation to
      provide for editing of command lines and may specify that it be
      done in the sender host. The control connection shall be closed
      by the receiver at the sender's request after all transfers and
      replies are completed.

   SEQUENCING OF COMMANDS AND REPLIES

      The communication between the sender and receiver is intended to
      be an alternating dialogue. As such, the sender issues an MTP
      command and the receiver responds with a prompt primary reply.
      The sender should wait for this initial primary success or failure
      response before sending further commands.

      Certain commands require a second reply for which the sender
      should also wait. These replies may, for example, report on the
      progress or completion of mail transfer. They are secondary
      replies to mail transfer commands.

      One important group of informational replies is the connection

greetings. Under normal circumstances, a receiver will send a 220
reply, "awaiting input", when the connection is completed. The
sender should wait for this greeting message before sending any
commands. If the receiver is unable to accept input right away,
it should send a 120 "expected delay" reply immediately and a 220
reply when ready. The sender will then know not to hang up if
there is a delay.

   Note: all the greeting type replies have the official name of
   the server host as the first word following the reply code.

The table below lists alternative success and failure replies for
each command. These must be strictly adhered to; a receiver may
substitute text in the replies, but the meaning and action implied
by the code numbers and by the specific command reply sequence
cannot be altered.

COMMAND-REPLY SEQUENCES

   In this section, the command-reply sequence is presented. Each
   command is listed with its possible replies; command groups are
   listed together. Preliminary replies are listed first (with
   their succeeding replies indented under them), then positive
   and negative completion, and finally intermediary replies with
   the remaining commands from the sequence following. The 421
   reply (service not available, closing control connection) may
   be given at any point if the MTP-receiver knows it must shut
   down. This listing forms the basis for the state diagrams,
   which will be presented separately.

       CONNECTION ESTABLISHMENT
          120
             220
          220
          421
       MAIL ACTION COMMANDS
          MAIL
             151, 152
                354
                   250
                   451, 552
             354
                250
                451, 552
             450, 550, 452, 553
             500, 501, 502, 421

                    MRSQ
                       200, 215
                       500, 501, 502, 421
                    MRCP
                       151, 152
                          200
                       200
                       450, 550, 452, 553
                       500, 501, 502, 503, 421
                    QUIT
                       221
                 INFORMATIONAL COMMANDS
                    HELP
                       211, 214
                       500, 501, 502, 421
                 MISCELLANEOUS COMMANDS
                    NOOP
                       200
                       500 421

STATE DIAGRAMS

   Here we present state diagrams for a very simple minded MTP
   implementation. Only the first digit of the reply codes is used.
   There is one state diagram for each group of MTP commands.

   The command groupings were determined by constructing a model for
   each command and then collecting together the commands with
   structurally identical models.

   For each command there are three possible outcomes: "success" (S),
   "failure" (F), and "error" (E). In the state diagrams below we use
   the symbol B for "begin", and the symbol W for "wait for reply".

We first present the diagram that represents the most MTP commands:


```
                          1,3     +---+
                       ---------->| E |
                       |          +---+
                       |
 +---+     cmd     +---+     2     +---+
 | B |---------->| W |---------->| S |
 +---+           +---+           +---+
                       |
                       |   4,5     +---+
                       ---------->| F |
                                   +---+
```
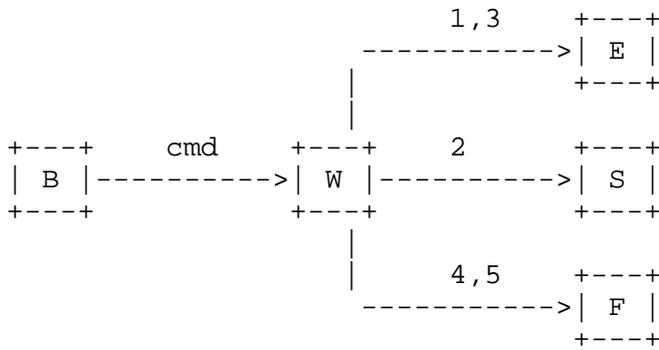

This diagram models the commands:

   HELP, MRCP, MRSQ, NOOP, QUIT.

A more complex diagram models the MAIL command:

```
                    ----   1
                   |    |
     +---+   cmd    -->+---+      2      +---+
     | B |---------->| W |---------->| E |
     +---+            +---+         -->+---+
                       | |          |
             3         | | 4,5      |
       -------------   ------       |
      |                       | |    +---+
      |               -------------->| S |
      |                | 1,3 | |     +---+
      |              2|   --------
      |               | |    |
      V               | |    |
     +---+   text     +---+ 4,5 ----->+---+
     |   |---------->| W |---------->| F |
     +---+            +---+           +---+
```
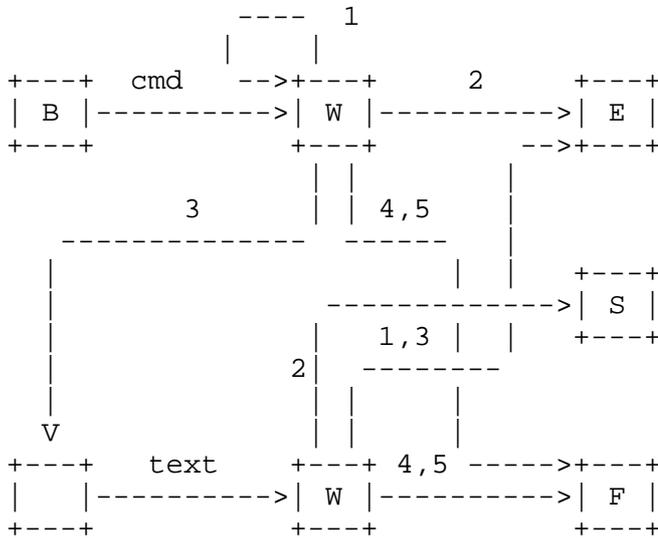
Note that the "text" here is a series of lines sent from the
sender to the receiver with no response expected until the last
line is sent. (The last line must consist of only a single
period.)

Finally we present a generalized diagram that could be used to model
the command and reply interchange:

```
             ------------------------------------
             |                                  |
  Begin      |                                  |
     |       V                                  |
     |     +---+   cmd    +---+ 2       +---+    |
    -->|   |------->|   |---------->|   |    |
     |     |   |          | W |          | S |-----|
    -->|   |      -->|   |-----      |   |    |
     |     +---+     |   +---+ 4,5 |    +---+    |
     |       |       |   | |       |             |
     |       |       |  1| |3      |     +---+    |
     |       |       |   | |       |     |   |    |
     |       |      ----  |  ---->|   | F |-----
     |       |              |           |   |    |
     |       |              |           +---+    |
     ------------------------
             |
             |
             V
            End
```
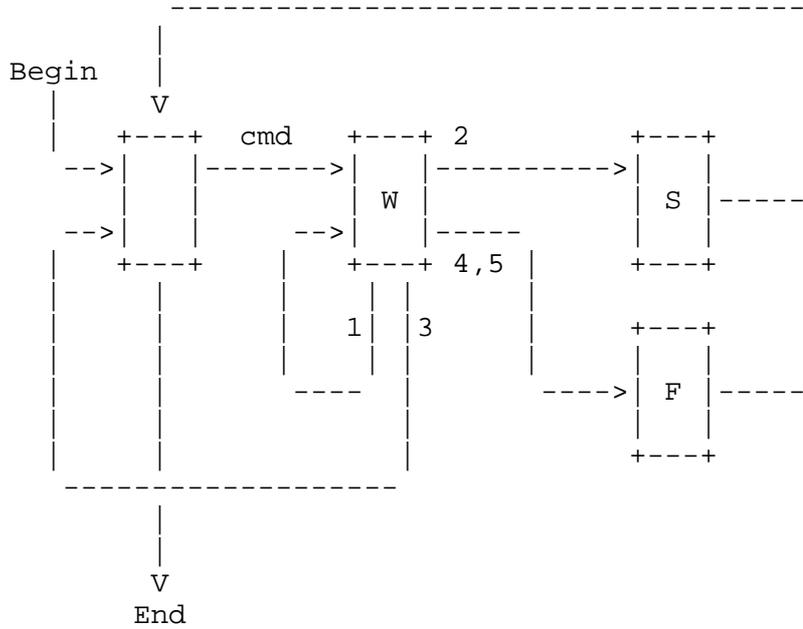
CONNECTION ESTABLISHMENT

   The MTP control connection is established via TCP/NCP between the
   receiver process port/socket L and the sender process port/socket U.
   This protocol is assigned the service port/socket 57 (71 octal), that
   is L=57.

APPENDIX A

                    Example of MRSQ R (Recipients-first)

  This is an example of how MRSQ R is used. First the sender must
  establish that the receiver in fact implements MRSQ.

      S: MRSQ <CRLF>
      R: 200 OK, no scheme selected

  An MRSQ with a null argument always returns a 200 if implemented,
  selecting the default "scheme", i.e., none of them. If MRSQ were not
  implemented, a code of 4xx or 5xx would be returned.

      S: MRSQ R <CRLF>
      R: 200 OK, using that scheme

  All is well; now the recipients can be specified.

      S: MRCP TO:<Foo@Y> <CRLF>
      R: 200 OK

      S: MRCP TO:<Raboof@Y> <CRLF>
      R: 553 No such user here

      S: MRCP TO:<bar@Y> <CRLF>
      R: 200 OK

      S: MRCP TO:<@Y,@X,fubar@Z> <CRLF>
      R: 200 OK

  Note that the failure of "Raboof" has no effect on the storage of
  mail for "Foo", "bar" or the mail to be forwarded to "fubar@Z"
  through host "X". Now the message text is furnished, by giving a
  MAIL command with no "TO" argument.

      S: MAIL FROM:<waldo@A><CRLF>
      R: 354 Type mail, ended by <CRLF>.<CRLF>
      S: Blah blah blah blah....etc. etc. etc.
      S: <CRLF>.<CRLF>
      R: 250 Mail sent

  The mail text has now been sent to "Foo" and "bar" as well as
  forwarded to "fubar@Z".

                                     29

APPENDIX B

                    Example of MRSQ T (Text-first)

   Using the same message as the previous example to establish that the
   receiver implements MRSQ.

      S: MRSQ ? <CRLF>
      R: 215 T Text first, please

   MRSQ is indeed implemented, and the receiver says that it prefers
   "T", but that needn't stop the sender from trying something else.

      S: MRSQ R <CRLF>
      R: 501 Sorry, I really can't do that

   It's possible that it could have understood "R" also, but in general
   it's best to use the "preferred" scheme, since the receiver knows
   which is most efficient for its particular site.

      S: MRSQ T <CRLF>
      R: 200 OK, using that scheme

   Scheme "T" is now selected, and the message text is sent by giving a
   mail command with no "TO" argument.

      S: MAIL FROM:<WALDO@A><CRLF>
      R: 354 Type mail, ended by <CRLF>.<CRLF>
      S: Blah blah blah blah....etc. etc. etc.
      S: <CRLF>.<CRLF>
      R: 250 Mail stored

   Now recipients can be specified.

      S: MRCP TO:<Foo@Y> <CRLF>
      R: 250 Stored mail sent

      S: MRCP TO:<Raboof@Y> <CRLF>
      R: 553 No such user here

      S: MRCP TO:<bar@Y> <CRLF>
      R: 250 Stored mail sent

      S: MRCP TO:<@Y,@X,fubar@Z> <CRLF>
      R: 200 OK

The text has now been sent to "Foo" and "bar" at host "Y" and will be
forwarded to "fubar@Z" through host "X", and still remains stored. A
new message can be sent with another MAIL/MRCP ... sequence, but a
careful sender would reset the state using the exchange below.

    S: MRSQ ? <CRLF>
    R: 215 T Text first, please

Which resets the state without altering the scheme in effect.