MAIL TRANSFER PROTOCOL

Suzanne Sluizer

and

Jonathan B. Postel

RFC 780

May 1981

Information Sciences Institute
University of Southern California
4676 Admiralty Way
Marina del Rey, California 90291

(213) 822-1511

                         TABLE OF CONTENTS

MAIL TRANSFER PROTOCOL

1. INTRODUCTION

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

    MTP is designed to be independent of the particular transmission
    subsystem and requires only a reliable ordered data stream channel.
    Appendices describe the use of MTP with various transport services.
    A Glossary provides the definitions of terms as used in this
    document.

    An important feature of MTP is its capability to relay mail from one
    transport environment to another. A transport service provides an
    interprocess communication environment (IPCE). An IPCE may cover one
    network, several networks, or a subset of a network. A process can
    communicate directly with another process anywhere in its own IPCE.
    Mail is a special case of interprocess communication. Mail can be
    communicated between proceses in different IPCEs by relaying through
    a process connected to two (or more) IPCEs. More specifically, mail
    can be relayed between hosts on different transport systems by a host
    on both transport systems. It is important to realize that transport
    systems (or IPCEs) are not one-to-one with networks.

2.   THE MTP MODEL

   The MTP design is based on the following model of communication: at
   the initiation of the user, the sender-MTP establishes the
   full-duplex transmission channel. MTP commands are generated by the
   sender-MTP and sent to the receiver-MTP. MTP replies are sent from
   the receiver-MTP to the sender-MTP in response to the commands.

   In the simplest case, once the transmission channel is established
   the MTP-sender sends a MAIL command indicating the sender and
   receiver of the mail. If the MTP-receiver can accept the mail it
   responds with a go ahead reply. Then the MTP-sender sends the mail
   data, terminating with a special sequence. If the MTP-receiver
   successfully processes the mail it responds with an OK reply.
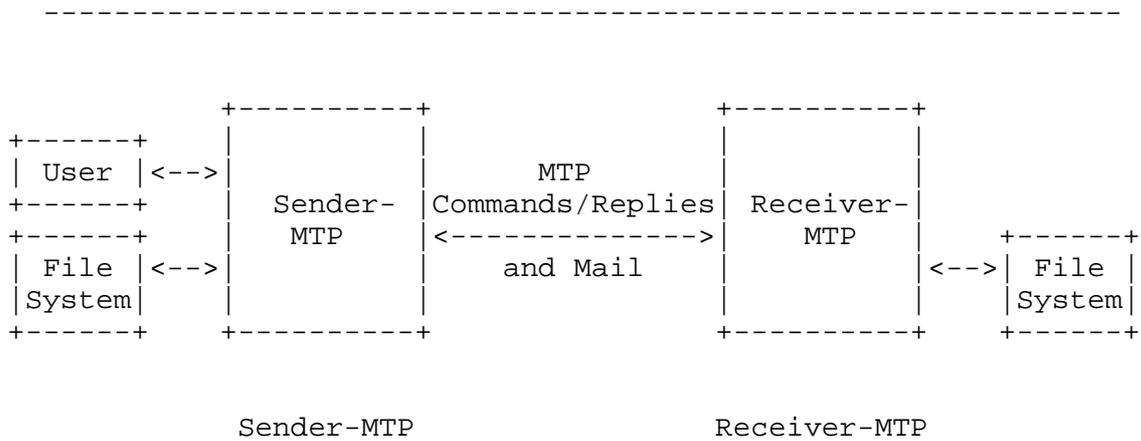
```
        -------------------------------------------------------------


                     +----------+              +----------+
        +------+      |          |              |          |
        | User |<-->|          |       MTP      |          |
        +------+      |  Sender- |Commands/Replies| Receiver-|
        +------+      |   MTP    |<-------------->|   MTP    |      +------+
        | File |<-->|          |     and Mail   |          |<-->| File |
        |System|      |          |              |          |    |System|
        +------+      +----------+              +----------+    +------+


                  Sender-MTP                 Receiver-MTP

                       Model for MTP Use

                            Figure 1

        -------------------------------------------------------------
```

   The MTP provides mechanisms for the transmission of mail; directly
   from the sending user's host to the receiving user's host when the
   two host are connected to the same transport service, or via one or
   more relay MTP-servers when the source and destination hosts are not
   connected to the same transport service.

   To be able to provide the relay capability the MTP-server must be
   supplied with the name of the ultimate destination host as well as
   the destination mailbox name.

The arguments to the MAIL command are a FROM path and a TO path. The
TO path is a source route while the FROM path is a return route
(which may be used to return a message to the sender when an error
occurs with a relayed message).

The preceding discussion has outlined the transmission of one copy of
one message from a source to a destination host and the possibility
of relaying messages between different transport services. The MTP
additionally supports the transmission of one copy of a message
addressed to multiple recipients.

In order for mail to be successfully transmitted the destination
users must be known at the destination receiver-MTP and the mail data
must be correctly received and stored. In the single recipient case
discussed above the positive response to the MAIL command indicated
the recipient was known, and the final OK response indicated the mail
was received and stored.

To support multi-recipient mail, MTP provides two procedures:
Text-First, and Recipients-First. In the text-first scheme the mail
data is sent and acknowledged, then each recipient identification is
sent and acknowledged (or refused) separately. In the
recipients-first scheme the recipients are negotiated first, then the
text is sent and acknowledged (for all recipients at once). The
choice of scheme is up to the MTP-receiver, and depends on the way
mail is handled in the destination host.

The multi-recipient mail procedures are optional and the
determination of which scheme to use is negotiated. The use of the
multi-recipient schemes is strongly encouraged by the economy they
provide in transmission and processing.

The mail commands and replies have a rigid syntax. Replies also have
a numeric code. In the following, examples appear which use actual
commands and replies. The complete lists of commands and replies
appears in Section 5 on specifications.

Commands and replies are not case sensitive. That is, a command or
reply word may be upper case, lower case, or any mixture of upper and
lower case. Note that this is not true of mailbox user names. For
some hosts the user name is case sensitive, and MTP implementations
must take case to preserve the case of user names as they appear in
mailbox arguments.

3.   BASIC MAIL

   The basic command for transmitting mail is MAIL. This command causes
   the transmitted data to be entered into the recipient's mailbox, or
   accepted for relaying to the destination host.

   The mail text is also sent on the transmission channel. This
   requires that the end of the text be signalled so that the command
   and reply dialog can be resumed. MTP signals the end of the mail
   text by sending a line containing only a period. A transparency
   procedure is used to prevent this interfering with the users text
   (see Section 5.5.2).

      MAIL <SP> FROM:<sender-path> <SP> TO:<receiver-path> <CRLF>

         The <sender-path> contains the source mailbox; the
         <receiver-path> contains the destination mailbox. If accepted,
         the receiver-MTP returns a 354 reply and considers all
         succeeding lines to be the message text. The message text is
         terminated by a line containing only a period, upon which a 250
         completion reply is returned. Various errors are possible.

         Actually the <sender-path> and <receiver-path> are more than
         just the mailboxes, they may be source routes. The
         <receiver-path> is a source routing list of hosts and
         destination mailbox; the <sender-path> is a reverse source
         routing list of hosts and source mailbox.

                          Example of MAIL (Basic Mail)

        This MAIL command specifies the mail is sent by Waldo at host A,
        and is to be delivered to Foo at host Y. Here we assume that host
        A contacts host Y directly.

           S: MAIL FROM:<waldo@A> TO:<Foo@Y> <CRLF>
           R: 354 Start mail input; end with <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".

                                    Example 1

     3.1. FORWARDING

        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 host.

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

              This reply indicates that the receiver-MTP knows the user's
              mailbox is on another host and will take responsibility for
              forwarding the mail to that host. This reply is only sent
              when the sender would not expect the mail to be forwarded.
              That is, when <receiver-path> as given in the command
              indicates mail relaying, this reply will not be used. This
              reply could be used for an organization with several hosts
              when each has a list of many of the users on the hosts. A
              host can accept mail for any user on its list and forward it
              to the correct host.

           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
sending-user would prefer to send the mail directly to the
recipient's actual host, the action may be aborted.

The MTP-sender must accept or reject the proposal in the
preliminary reply by sending a continue (CONT) or abort (ABRT)
command. In the case of the continue, the next reply from the
MTP-receiver will be any of the replies expected for the MAIL
command, most likely "354 Start mail input, ...". In the case of
the abort, the next reply from the MTP-receiver will be "201
Command okay, action aborted".

3.2.   SOURCE ROUTING

The receiver-path may be a source route of the form
"@ONE,@TWO,JOE@THREE", where ONE, TWO, and THREE are hosts. This
form is used to emphasize the distinction between an address and a
route.

At some distant future time it might be necessary to expand the
mailbox format to include a region identifier, such as
"user@host@region". If this occured the MTP path convention
could be expanded to
"host@region,host@region,...user@host@region". For example,
"ONE@R1,TWO@R2,JOE@THREE@R3".

The mailbox is an absolute address, and the route is information
about how to get there. The two concepts should not be confused.

The elements of the receiver-path are to be moved to the
sender-path as the message is relayed from one MTP to another. The
sender-path is a reverse source route, that is, a source route to
the originator of the message. When an MTP deletes its identifier
from the receiver-path and inserts it into the sender-path, it
must use the name it is known by in the environment it is sending
into, not the environment the mail came from, in case the MTP is
known be different names in different environments.

When source routing is used the receiver-MTP will receive mail to
be relayed to another MTP. The receiver-MTP may accept the task
of relaying the mail or reject it in the same way it accepts or
reject mail for a local user. It does not use the 151 "User not
local" or 152 "User unknown" preliminary replies. Once the
receiver-MTP accepts the relaying task it receives the mail text
and transforms the command arguments by removing its own
identifier from the receiver-path and inserting it in the

beginning of the sender-path. The receiver-MTP then becomes a
sender-MTP and establishes a transmission channel to the next MTP
in the receiver-path and sends it the mail.

If an MTP has accepted the task of relaying the mail and later
finds that the receiver-path is incorrect or that the mail cannot
be delivered for whatever reason, then it must construct a
notification message and send it to the originator of the
undeliverable mail as indicated by the sender-path. This
notification message must be from the MTP at this host. That is,
the sender-path of the notification message itself will be
"MTP@<host>", and in the notification message header the From
field will be "MTP at <host>". Of course, MTPs should not send
notification messages about problems with notification messages.

4. MULTI-RECIPIENT MAIL

   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 host. In one
   scheme, all recipients are specified first, and then the text is
   sent. In the other scheme, the order is reversed and the text is
   sent first, followed by the recipients. The sender-MTP suggests the
   scheme it would prefer, but receiver-MTP controls which scheme is
   actually used. 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.

   Both schemes are necessary 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. 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.

   4.1. 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, use the specified scheme
           215 <scheme> This is the scheme I prefer
           504 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> <string>] <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 reseting all schemes to their initial state. This
    reset must be done no matter what the reply will be -- 200, 215,
    or 504. The actions necessary for a reset will be explained when
    discussing how each scheme actually works.

    Note that the receiver gets to choose which scheme is used. The
    sender must be prepared to do either.

 4.2. MESSAGE TEXT SPECIFICATION: MAIL

    Regardless of which scheme (if any) has been selected, a MAIL
    command with a non-null receiver-path 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" all schemes to their initial state.

    It is only when the receiver-path argument is null that the
    particular scheme chosen is important.

       MAIL FROM:<sender-path> <CRLF>

Rather than producing an error, the receiver will accept message
text for this "null" recipient specification. What it does with
it depends on which scheme is in effect, and will be described in
the section on Scheme Mechanics.

4.3. 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:<receiver-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
        550 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.

4.4. 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 receiver-path argument; that is, just MAIL <SP>
FROM: <sender-path> <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 (or MAIL with a non-null receiver-path
argument).

A 452 reply (out of storage space) 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
receiver-path 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".

          Example of MRSQ R (Recipients First)

   First the sender must establish that the receiver 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: 550 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 relayed to "fubar@Z"
   through host "X". Now the message text is furnished, by giving
   a MAIL command with no receiver-path argument.

```
        S: MAIL FROM:<waldo@A><CRLF>
        R: 354 Start mail input; end with <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
   relayed to "fubar@Z".

                            Example 2

   4.5.   SCHEME MECHANICS: MRSQ T (TEXT-FIRST)

   In the text-first scheme, MAIL with no receiver-path 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 receiver-path
   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.

                         Example of MRSQ T (Text First)

     First the sender must 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: 504 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 receiver-path argument.

          S: MAIL FROM:<WALDO@A><CRLF>
          R: 354 Start mail input; end with <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: 550 No such user here

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

          S: MRCP TO:<@Y,@X,fubar@Z> <CRLF>
          R: 250 Mail accepted for relaying

The text has now been sent to "Foo" and "bar" at host "Y" and
will be relayed 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.

                            Example 3

  4.6. DISCUSSION

    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
    receiver-path arguments with this multi-recipient extension is the
    ambiguity that would result if a non-null receiver-path 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@Z><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" of all schemes 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. 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.

5. SPECIFICATIONS

    5.1. MTP COMMANDS

        5.1.1. COMMAND SEMANTICS

            The MTP commands define the mail transfer or the mail system
            function requested by the user. MTP commands are character
            strings terminated by <CRLF>. The command codes themselves are
            alphabetic characters terminated by <SP> if parameters follow
            and <CRLF> otherwise. The syntax of mailboxes must conform to
            receiver site conventions. The MTP commands are discussed
            below. MTP replies are discussed in the Section 5.2.

            MAIL (MAIL)

                This command is used to send mail over the transmission
                channel. The argument field contains a sender-path sequence
                and optional receiver-path sequence.

                The sender-path sequence consists of an optional list of
                hosts and the sender mailbox. When the list of hosts is
                present, it is "reverse" source routing information and
                indicates that the mail was relayed through each host on the
                list (the first host in the list was the most recent relay).
                This list is used as source routing to return non-delivery
                notices to the sender. As each relay host adds itself to
                the beginning of the list, it must use its name as known in
                the network to which it is relaying the mail rather than the
                network from which the mail came (if they are different).

                If the receiver-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 relayed to the first
                host on the list.

                The receiver treats the lines following the command as mail
                text from the sender. The mail text is terminated by the
                character sequence "<CRLF>.<CRLF>", (see Section 5.5.2 on
                Transparency).

                As mail is relayed along the receiver-path sequence, each
                relay host must remove itself from the path sequence and put
                itself at the beginning of the sender-path sequence. When
                mail reaches its ultimate destination (the receiver-path

   sequence has only a destination mailbox), the receiver-MTP
   inserts it into the destination mailbox in accordance with
   its host mail conventions. (For example, "MAIL FROM:<X@Y>
   TO:<@A,@B,C@D> <CRLF>" will eventually be relayed as "MAIL
   FROM:<@A,X@Y> TO:<@B,C@D> <CRLF>.)

   If the receiver-path sequence is empty, the mail is destined
   for a printer or other designated place for host general
   delivery mail (if allowed at this host). The mail may be
   marked as sent from the sender as specified in the
   sender-path sequence 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 recipients-first, or text-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 transmission
   channel to the receiver. The command may take an argument
   (e.g., any command name) and return more specific
   information as a response.

QUIT (QUIT)

   This command specifies that the receiver must close the
   transmission channel.

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.

CONTINUE (CONT)

This command specifies that the previously specified action
is to be continued. This is sent only following a
preliminary reply.

ABORT (ABRT)

This command specifies that the previously specified action
is to be aborted. This is sent only following a preliminary
reply. It specifies no further action other than that the
receiver send an OK reply.

5.1.2. COMMAND SYNTAX

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.

But, note that in the sender-path and receiver-path arguments
case is important. In particular, in some hosts the user "foo"
is different from the user "Foo".

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
the line is received.

Square brackets denote an optional argument field. If the
option is not taken, the appropriate default is implied. All
characters are in the ASCII characters set.

          The following are the MTP commands:

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

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

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

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

          QUIT <CRLF>

          NOOP <CRLF>

          CONT <CRLF>

          ABRT <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-path> ::= <path>

    <receiver-path> ::= <path>

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

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

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

    <host> ::= <a> <string> | "#" <number> | "[" <dotnum> "]"

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

    <user> ::= <string>

    <char> ::= <c> | '\' <c> | '\' <s>

    <dotnum> ::= <snum> "." <snum> "." <snum> "." <snum>

    <number> ::= <d> | <d> <number>

    <snum> ::= three digits representing an integer value in the
    range 0 through 255

    <specials> ::= '<', '>', '(', ')', '\', ',', ';', ':', '@',
    '"', and the control characters (ASCII codes 0 through 37
    octal inclusive and 177 octal)

    <a> ::= any one of the 26 letters A through Z in either case

    <c> ::= any one of the 128 ASCII characters except
    <specials>

    <d> ::= any one of the ten digits 0 through 9

    <s> ::= any one of <specials>

    Note that the backslash, '\', is a quote character, which is
    used to indicate that the next character is to be used
    literally instead of with its normal interpretation. For

          example, "Joe\,Smith" could be used to indicate a single
          nine character user field with comma being the fourth
          character of the field.

     Hosts are generally known by names which are translated to
     addresses in each host. Sometimes a host is not known to the
     translation function and communication is blocked. To bypass
     this barrier numeric forms are also allowed for host "names".
     One form is a decimal integer prefixed by a pound sign, "#",
     which indicates the number is the address of the host. Another
     form is four small decimal integers separated by dots and
     enclosed by brackets, e.g., "[123.255.37.321]", which indicates
     a 32 bit ARPA Internet Address in four eight bit fields.

   5.2. MTP REPLIES

      Replies to MTP 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-MTP. Every command must generate exactly one reply.
      Additionally, some commands must occur sequentially, such as
      MRSQ T->MAIL->MRCP or MRSQ R->MRCP->MAIL.

         The details of the command-reply sequence are made explicit in
         the Sections 5.3 and 5.4 on Sequencing and 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. Further
      explanation of the assignment of reply codes is given in the
      Appendix E on the Theory of Reply Codes. Formally, a reply is
      defined to be the sequence: a three-digit code, <SP>, one line of
      text, and <CRLF>.

     5.2.1. REPLY CODES BY FUNCTION GROUPS

        200 Command okay
        201 Command okay, action aborted
        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
        504 Command parameter not implemented

        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 transmission channel
        421 <host> Service not available, closing transmission channel
           [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
        552 Requested mail action aborted: exceeded storage allocation
           [For current mailbox location]
        553 Requested action not taken: mailbox name not allowed
           [E.g., mailbox syntax incorrect]
        354 Start mail input; end with <CRLF>.<CRLF>

     5.2.2. 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
        201 Command okay, action aborted
        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 transmission channel
        250 Requested mail action okay, completed

        354 Start mail input; end with <CRLF>.<CRLF>

        421 <host> Service not available, closing transmission channel
           [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

        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
        504 Command parameter not implemented
        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
           [E.g., mailbox syntax incorrect]

   5.3.   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 response before sending further
      commands.

      The preliminary (1xx) and intermediate (3xx) replies indicate that
      further commands and information are required to complete the
      required action. The preliminary replies require either a
      continue or abort command to proceed; the intermediate replies
      require action dependent further commands.

      One important reply is the connection greetings. Under normal
      circumstances, a receiver will send a 220 "Awaiting input" reply
      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. The sender can then indicate it is
      willing to wait via a continue command, or not via the abort
      command. The receiver will respond to the abort with a 201 reply,
      and to the continue with the 220 reply when ready.

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

          For example,

             220 <SP> USC-ISIF <SP> Service ready <CRLF>

      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

         Each command is listed with its possible replies. Preliminary
         replies are listed first with their succeeding replies indented
         under them, then success and failure completion, and finally
         intermediary replies with the remaining commands from the
         sequence following. The prefixes used before the possible
         replies are "P" for preliminary, "I" for intermediate, "S" for
         success, "F" for failure, and "E" for error. The 421 reply

(service not available, closing transmission channel) may be
given to any command if the MTP-receiver knows it must shut
down. This listing forms the basis for the State Diagrams, in
Section 5.4.

```
CONNECTION ESTABLISHMENT
   P: 120 -> CONT -> S: 220
                     F: 421
            ABRT    S: 201
                     F: 421
   S: 220
   F: 421
MAIL
   P: 151 -> CONT -> I: 354 -> text -> S: 250
      152                             F: 451,552,450,
                                         550,452,553
            ABRT -> S: 201
                     F: 451,552,450,550,452,553
   I: 354 -> text -> S: 250
                     F: 451,552,450,550,452,553
   F: 451, 552, 450, 550, 452, 553
   E: 500, 501, 502, 421
MRSQ
   S: 200, 215
   E: 500, 501, 502, 504, 421
MRCP
   P: 151 -> CONT -> S: 200, 215, 250
      152           F: 451,552,450,550,452,553
            ABRT -> S: 201
                     F: 451,552,450,550,452,553
   S: 200, 215, 250
   F: 451, 552, 450, 550, 452, 553
   E: 500, 501, 502, 503, 421
```

         QUIT
            S: 221
            E: 500, 421
         HELP
            S: 211, 214
            E: 500, 501, 502, 504, 421
         NOOP
            S: 200
            E: 500, 421
         CONT
            S: depends on previous command
            F: depends on previous command
            E: 500, 501, 502, 504, 421
         ABRT
            S: 201,
   E: 500, 501, 502, 504, 421

5.4. STATE DIAGRAMS

   Following are 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".

   First, the diagram that represents most of the MTP commands:

```
                            1,3      +---+
                        ---------->| E |
                        |            +---+
                        |
    +---+    cmd      +---+    2       +---+
    | B |---------->| W |---------->| S |
    +---+            +---+            +---+
                        |
                        |    4,5      +---+
                        ---------->| F |
                                     +---+
```
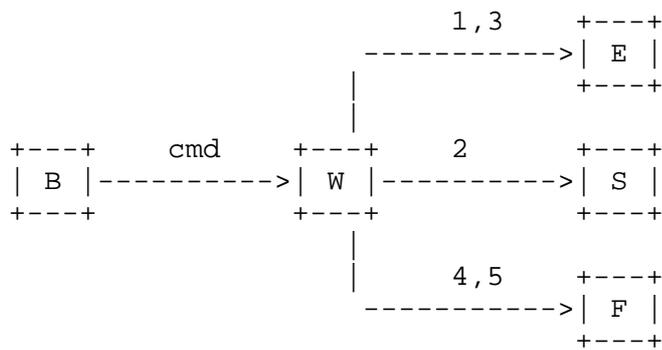
      This diagram models the commands:

         HELP, MRCP, MRSQ, NOOP, QUIT, ABRT.

     A more complex diagram models the MAIL command:

                              ABRT   +---+ 1,3
                     CONT ---- ------------>| W |-------
                          | | +---+ |
                          | |1   4,5| |2   V
           +---+   cmd -->+---+ 2 | | +---+
           | B |--------->| W |-------------------->| E |
           +---+   +---+   ---------->+---+
                         3| |4,5 | | |
                          | | | | |
              ------------- ------ | | |
             | | | | | ---->+---+
             | --------------------------->| S |
             | | | | | | +---+
             | | | -------- |
             | | | | | |
             | | | | | |
             V   2| |1,3 | |
           +---+   text   +---+ | ------->+---+
           | |--------->| W | --------------->| F |
           +---+   +---+-------------------->+---+
                           4,5
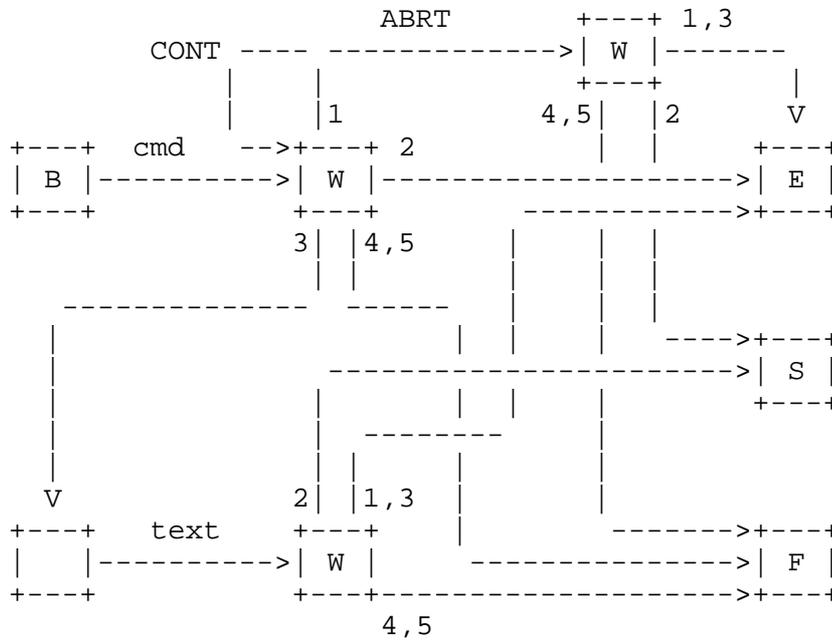
     Note that the "text" here is a series of lines sent from the
     sender to the receiver with no response expected until the last
     line is sent.

   5.5. DETAILS

      5.5.1. MINIMUM IMPLEMENTATION

         In order to make MTP workable, the following minimum
         implementation is required for all receivers:

            COMMANDS -- MAIL
                        QUIT
                        NOOP

      5.5.2. TRANSPARENCY

         Without some provision for data transparency the character
         sequence "<CRLF>.<CRLF>" ends the the mail text and cannot be
         sent by the user. In general, users are not aware of such
         "forbidden" sequences. To allow all user composed text to be
         transmitted transparently the following procedures are used.

         1. Before sending a line of mail text the sender-MTP checks the
         first character of the line. If it is a period, one additional
         period is inserted at the beginning of the line.

         2. When a line of mail text is received by the receiver-MTP it
         checks the the line. If the line is composed of a single
         period it is the end of mail. If the first character is a
         period and there are other characters on the line, the first
         character is deleted.

      5.5.3. SIZES

         There are several objects that ought to have defined maximum
         sizes.

            user

               The maximum total length of a user name is 40 characters.

            host

               The maximum total length of a host name or number is 20
               characters.

path

>   The maximum total length of a sender-path or
>   receiver-path is 100 characters.

command line

>   The maximum total length of a command line including the
>   command word and the <CRLF> is 200 characters.

reply line

>   The maximum total length of a reply line including the
>   reply code and the <CRLF> is 65 characters.

text line

>   The maximum total length of a text line including the the
>   <CRLF> is 1000 characters.

To the maximum extent possible implementation techniques which
impose no limits at all to the length of these objects should
be used.

APPENDIX A

   TCP Transport service

      The Transmission Control Protocol [1] is used in the ARPA
      Internet, and in any network following the US DoD standards for
      internetwork protocols.

      Connection Establishment

         The MTP transmission channel is a TCP connection established
         between the sender process port U and the receiver process port
         L. This single full duplex connection is used as the
         transmission channel. This protocol is assigned the service
         port 57 (71 octal), that is L=57.

      Data Transfer

         The TCP connection supports the transmission of 8-bit bytes.
         The MTP data is 7-bit ASCII characters. Each character is
         transmitted as a 8-bit byte with the high-order bit cleared to
         zero.

APPENDIX B

  NCP Transport service

    The ARPANET Host-to-Host Protocol [2] (implemented by the Network
    Control Program) may be used in the ARPANET.

    Connection Establishment

      The MTP transmission channel is established via NCP between the
      the sender process socket U and receiver process socket L. The
      Initial Connection Protocol [3] is followed resulting in a pair
      of simplex connections. This pair of connections is used as
      the transmission channel. This protocol is assigned the
      contact socket 57 (71 octal), that is L=57.

    Data Transfer

      The NCP data connections are established in 8-bit byte mode.
      The MTP data is 7-bit ASCII characters. Each character is
      transmitted as a 8-bit byte with the high-order bit cleared to
      zero.

APPENDIX C

   NITS

      The Network Independent Transport Service [4] may be used.

      Connection Establishment

         The MTP transmission channel is established via NITS between
         the the sender process and receiver process. The sender
         process executes the CONNECT primitive, and the waiting
         receiver process executes the ACCEPT primitive.

      Data Transfer

         The NITS connection supports the transmission of 8-bit bytes.
         The MTP data is 7-bit ASCII characters. Each character is
         transmitted as a 8-bit byte with the high-order bit cleared to
         zero.

APPENDIX D

   X.25 Transport service

      It may be possible to use the X.25 service [5] as provided by the
      Public Data Networks directly, but there are indications that it
      is too error prone to qualify as a reliable channel. It is
      suggested that a reliable end-to-end protocol such as TCP be used
      on top of X.25 connections.

APPENDIX E

   Theory of Reply Codes

      The three digits of the reply each have a special significance.
      The first digit denotes whether the response is good, bad or
      incomplete. 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 command has been accepted, but the requested action
               is being held in abeyance, pending confirmation of the
               information in this reply. The sender-MTP should send
               another command specifying whether to continue or abort
               the action.

            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 the account 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
        transmission channel.

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
illustrates 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.

For example, 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.

The reply text may be longer than a single line; in these cases
the complete text must be marked so the sender-MTP knows when it
can stop reading the reply. This requires a special format to
indicate a multiple line reply.

The format for multi-line replies requires that every line,
except the last, begin with the reply code, followed
immediately by a hyphen, "-" (also known as minus), followed by
text. The last line will begin with the reply code, followed
immediately by <SP>, optionally some text, and <CRLF>.

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

The sender-MTP then simply needs to search for the reply code
followed by <SP> at the beginning of a line, and ignore all
preceding lines.

GLOSSARY

   ASCII

      American Standard Code for Information Interchange [6].

   command

      A request for a mail service action sent by the sender-MTP to the
      receiver-MTP.

   host

      A computer in the internetwork environment on which mailboxes or
      MTP processes reside.

   line

      A line of text ending with a <CRLF>.

   mail

      A sequence of ASCII characters of arbitrary length, which conforms
      to the standard set in RFC 733 (Standard for the Format of ARPA
      Network Text Messages [7]).

   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.

   receiver-MTP process

      A process which transfers mail in cooperation with a sender-MTP
      process. It waits for a connection to be established via the
      transport service. It receives MTP commands from the sender-MTP,
      sends replies, and governs the transfer of mail.

reply

   A reply is an acknowledgment (positive or negative) sent from
   receiver to sender via the transmission channel 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.

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 transport
   service connection. It initiates MTP commands, receives replies,
   and governs the transfer of mail.

transmission channel

   A full-duplex communication path between a sender-MTP and a
   receiver-MTP for the exchange of commands, replies, and mail text.

transport service

   Any reliable stream-oriented data communication services. For
   example, NCP, TCP, NITS.

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.

word

   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.

<CRLF>

   The characters carriage return and line feed (in that order).

   <SP>

      The space character.

REFERENCES

   [1] TCP

        Postel, J., ed., "DOD Standard Transmission Control Protocol",
        IEN 129, RFC 761, USC/Information Sciences Institute,
        NTIS ADA082609, January 1980. Appears in: Computer Communication
        Review, Special Interest Group on Data Communications, ACM, V.10,
        N.4, October 1980.

   [2] NCP

        McKenzie,A., "Host/Host Protocol for the ARPA Network", NIC 8246,
        January 1972. Also in: Feinler, E. and J. Postel, eds., "ARPANET
        Protocol Handbook", NIC 7104, for the Defense Communications
        Agency by SRI International, Menlo Park, California, Revised
        January 1978.

   [3] Initial Connection Protocol

        Postel, J., "Official Initial Connection Protocol", NIC 7101,
        11 June 1971. Also in: Feinler, E. and J. Postel, eds., "ARPANET
        Protocol Handbook", NIC 7104, for the Defense Communications
        Agency by SRI International, Menlo Park, California, Revised
        January 1978.

   [4] NITS

        PSS/SG3, "A Network Independent Transport Service", Study Group 3,
        The Post Office PSS Users Group, February 1980. Available from
        the DCPU, National Physical Laboratory, Teddington, UK.

   [5] X.25

        CCITT, "Recommendation X.25 - Interface Between Data Terminal
        Equipment (DTE) and Data Circuit-terminating Equipment (DCE) for
        Terminals Operating in the Packet Mode on Public Data Networks,"
        CCITT Orange Book, Vol. VIII.2, International Telephone and
        Telegraph Consultative Committee, Geneva, 1976.

   [6] ASCII

        ASCII, "USA Code for Information Interchange", United States of
        America Standards Institute, X3.4, 1968. Also in: Feinler, E.
        and J. Postel, eds., "ARPANET Protocol Handbook", NIC 7104, for
        the Defense Communications Agency by SRI International, Menlo
        Park, California, Revised January 1978.

   [7] RFC 733

        Crocker, D., J. Vittal, K. Pogran, and D. Henderson, "Standard for
        the Format of ARPA Network Text Messages," RFC 733, NIC 41952,
        November 1977. Also in: Feinler, E. and J. Postel, eds.,
        "ARPANET Protocol Handbook", NIC 7104, for the Defense
        Communications Agency by SRI International, Menlo Park,
        California, Revised January 1978.