RFC 821

SIMPLE MAIL TRANSFER PROTOCOL

Jonathan B. Postel

August 1982

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

(213) 822-1511

TABLE OF CONTENTS

SIMPLE MAIL TRANSFER PROTOCOL

1. INTRODUCTION

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

   SMTP is independent of the particular transmission subsystem and
   requires only a reliable ordered data stream channel. Appendices A,
   B, C, and D describe the use of SMTP with various transport services.
   A Glossary provides the definitions of terms as used in this
   document.

   An important feature of SMTP is its capability to relay mail across
   transport service environments. A transport service provides an
   interprocess communication environment (IPCE). An IPCE may cover one
   network, several networks, or a subset of a network. It is important
   to realize that transport systems (or IPCEs) are not one-to-one with
   networks. A process can communicate directly with another process
   through any mutually known IPCE. Mail is an application or use of
   interprocess communication. Mail can be communicated between
   processes 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.

2. THE SMTP MODEL

   The SMTP design is based on the following model of communication: as
   the result of a user mail request, the sender-SMTP establishes a
   two-way transmission channel to a receiver-SMTP. The receiver-SMTP
   may be either the ultimate destination or an intermediate. SMTP
   commands are generated by the sender-SMTP and sent to the
   receiver-SMTP. SMTP replies are sent from the receiver-SMTP to the
   sender-SMTP in response to the commands.

   Once the transmission channel is established, the SMTP-sender sends a
   MAIL command indicating the sender of the mail. If the SMTP-receiver
   can accept mail it responds with an OK reply. The SMTP-sender then
   sends a RCPT command identifying a recipient of the mail. If the
   SMTP-receiver can accept mail for that recipient it responds with an
   OK reply; if not, it responds with a reply rejecting that recipient
   (but not the whole mail transaction). The SMTP-sender and
   SMTP-receiver may negotiate several recipients. When the recipients
   have been negotiated the SMTP-sender sends the mail data, terminating
   with a special sequence. If the SMTP-receiver successfully processes
   the mail data it responds with an OK reply. The dialog is purposely
   lock-step, one-at-a-time.
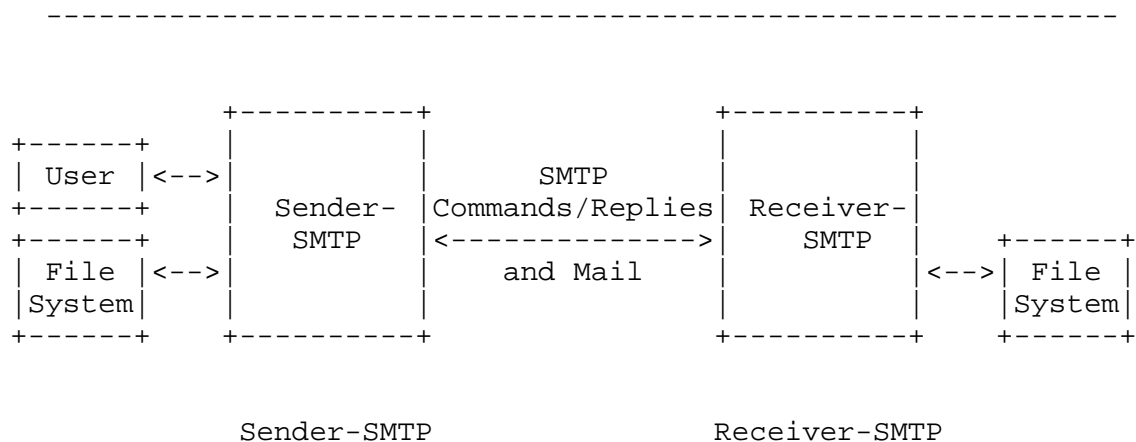
```
      -------------------------------------------------------------


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


              Sender-SMTP                  Receiver-SMTP

                        Model for SMTP Use

                             Figure 1

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

   The SMTP 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 SMTP-servers when the source and destination hosts are not
connected to the same transport service.

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

The argument to the MAIL command is a reverse-path, which specifies
who the mail is from. The argument to the RCPT command is a
forward-path, which specifies who the mail is to. The forward-path
is a source route, while the reverse-path is a return route (which
may be used to return a message to the sender when an error occurs
with a relayed message).

When the same message is sent to multiple recipients the SMTP
encourages the transmission of only one copy of the data for all the
recipients at the same destination host.

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 4 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 SMTP implementations
must take case to preserve the case of user names as they appear in
mailbox arguments. Host names are not case sensitive.

Commands and replies are composed of characters from the ASCII
character set [1]. When the transport service provides an 8-bit byte
(octet) transmission channel, each 7-bit character is transmitted
right justified in an octet with the high order bit cleared to zero.

When specifying the general form of a command or reply, an argument
(or special symbol) will be denoted by a meta-linguistic variable (or
constant), for example, "<string>" or "<reverse-path>". Here the
angle brackets indicate these are meta-linguistic variables.
However, some arguments use the angle brackets literally. For
example, an actual reverse-path is enclosed in angle brackets, i.e.,
"<John.Smith@USC-ISI.ARPA>" is an instance of <reverse-path> (the
angle brackets are actually transmitted in the command or reply).

3. THE SMTP PROCEDURES

   This section presents the procedures used in SMTP in several parts.
   First comes the basic mail procedure defined as a mail transaction.
   Following this are descriptions of forwarding mail, verifying mailbox
   names and expanding mailing lists, sending to terminals instead of or
   in combination with mailboxes, and the opening and closing exchanges.
   At the end of this section are comments on relaying, a note on mail
   domains, and a discussion of changing roles. Throughout this section
   are examples of partial command and reply sequences, several complete
   scenarios are presented in Appendix F.

   3.1. MAIL

      There are three steps to SMTP mail transactions. The transaction
      is started with a MAIL command which gives the sender
      identification. A series of one or more RCPT commands follows
      giving the receiver information. Then a DATA command gives the
      mail data. And finally, the end of mail data indicator confirms
      the transaction.

         The first step in the procedure is the MAIL command. The
         <reverse-path> contains the source mailbox.

            MAIL <SP> FROM:<reverse-path> <CRLF>

         This command tells the SMTP-receiver that a new mail
         transaction is starting and to reset all its state tables and
         buffers, including any recipients or mail data. It gives the
         reverse-path which can be used to report errors. If accepted,
         the receiver-SMTP returns a 250 OK reply.

         The <reverse-path> can contain more than just a mailbox. The
         <reverse-path> is a reverse source routing list of hosts and
         source mailbox. The first host in the <reverse-path> should be
         the host sending this command.

         The second step in the procedure is the RCPT command.

            RCPT <SP> TO:<forward-path> <CRLF>

         This command gives a forward-path identifying one recipient.
         If accepted, the receiver-SMTP returns a 250 OK reply, and
         stores the forward-path. If the recipient is unknown the
         receiver-SMTP returns a 550 Failure reply. This second step of
         the procedure can be repeated any number of times.

The <forward-path> can contain more than just a mailbox. The
<forward-path> is a source routing list of hosts and the
destination mailbox. The first host in the <forward-path>
should be the host receiving this command.

The third step in the procedure is the DATA command.

    DATA <CRLF>

If accepted, the receiver-SMTP returns a 354 Intermediate reply
and considers all succeeding lines to be the message text.
When the end of text is received and stored the SMTP-receiver
sends a 250 OK reply.

Since the mail data is sent on the transmission channel the end
of the mail data must be indicated so that the command and
reply dialog can be resumed. SMTP indicates the end of the
mail data by sending a line containing only a period. A
transparency procedure is used to prevent this from interfering
with the user's text (see Section 4.5.2).

    Please note that the mail data includes the memo header
    items such as Date, Subject, To, Cc, From [2].

The end of mail data indicator also confirms the mail
transaction and tells the receiver-SMTP to now process the
stored recipients and mail data. If accepted, the
receiver-SMTP returns a 250 OK reply. The DATA command should
fail only if the mail transaction was incomplete (for example,
no recipients), or if resources are not available.

The above procedure is an example of a mail transaction. These
commands must be used only in the order discussed above.
Example 1 (below) illustrates the use of these commands in a mail
transaction.

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

                      Example of the SMTP Procedure

   This SMTP example shows mail sent by Smith at host Alpha.ARPA,
   to Jones, Green, and Brown at host Beta.ARPA.  Here we assume
   that host Alpha contacts host Beta directly.

      S: MAIL FROM:<Smith@Alpha.ARPA>
      R: 250 OK

      S: RCPT TO:<Jones@Beta.ARPA>
      R: 250 OK

      S: RCPT TO:<Green@Beta.ARPA>
      R: 550 No such user here

      S: RCPT TO:<Brown@Beta.ARPA>
      R: 250 OK

      S: DATA
      R: 354 Start mail input; end with <CRLF>.<CRLF>
      S: Blah blah blah...
      S: ...etc. etc. etc.
      S: <CRLF>.<CRLF>
      R: 250 OK

   The mail has now been accepted for Jones and Brown.  Green did
   not have a mailbox at host Beta.

                              Example 1

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

     3.2. FORWARDING

        There are some cases where the destination information in the
        <forward-path> is incorrect, but the receiver-SMTP knows the
        correct destination. In such cases, one of the following replies
        should be used to allow the sender to contact the correct
        destination.

           251 User not local; will forward to <forward-path>

              This reply indicates that the receiver-SMTP knows the user's
              mailbox is on another host and indicates the correct
              forward-path to use in the future. Note that either the
              host or user or both may be different. The receiver takes
              responsibility for delivering the message.

           551 User not local; please try <forward-path>

              This reply indicates that the receiver-SMTP knows the user's
              mailbox is on another host and indicates the correct
              forward-path to use. Note that either the host or user or
              both may be different. The receiver refuses to accept mail
              for this user, and the sender must either redirect the mail
              according to the information provided or return an error
              response to the originating user.

        Example 2 illustrates the use of these responses.

                          Example of Forwarding

        Either

        S: RCPT TO:<Postel@USC-ISI.ARPA>
        R: 251 User not local; will forward to <Postel@USC-ISIF.ARPA>

        Or

        S: RCPT TO:<Paul@USC-ISIB.ARPA>
        R: 551 User not local; please try <Mockapetris@USC-ISIF.ARPA>

                               Example 2

   3.3. VERIFYING AND EXPANDING

      SMTP provides as additional features, commands to verify a user
      name or expand a mailing list. This is done with the VRFY and
      EXPN commands, which have character string arguments. For the
      VRFY command, the string is a user name, and the response may
      include the full name of the user and must include the mailbox of
      the user. For the EXPN command, the string identifies a mailing
      list, and the multiline response may include the full name of the
      users and must give the mailboxes on the mailing list.

      "User name" is a fuzzy term and used purposely. If a host
      implements the VRFY or EXPN commands then at least local mailboxes
      must be recognized as "user names". If a host chooses to
      recognize other strings as "user names" that is allowed.

      In some hosts the distinction between a mailing list and an alias
      for a single mailbox is a bit fuzzy, since a common data structure
      may hold both types of entries, and it is possible to have mailing
      lists of one mailbox. If a request is made to verify a mailing
      list a positive response can be given if on receipt of a message
      so addressed it will be delivered to everyone on the list,
      otherwise an error should be reported (e.g., "550 That is a
      mailing list, not a user"). If a request is made to expand a user
      name a positive response can be formed by returning a list
      containing one name, or an error can be reported (e.g., "550 That
      is a user name, not a mailing list").

      In the case of a multiline reply (normal for EXPN) exactly one
      mailbox is to be specified on each line of the reply. In the case
      of an ambiguous request, for example, "VRFY Smith", where there
      are two Smith's the response must be "553 User ambiguous".

      The case of verifying a user name is straightforward as shown in
      example 3.

                      Example of Verifying a User Name

            Either

               S: VRFY Smith
               R: 250 Fred Smith <Smith@USC-ISIF.ARPA>

            Or

               S: VRFY Smith
               R: 251 User not local; will forward to <Smith@USC-ISIQ.ARPA>

            Or

               S: VRFY Jones
               R: 550 String does not match anything.

            Or

               S: VRFY Jones
               R: 551 User not local; please try <Jones@USC-ISIQ.ARPA>

            Or

               S: VRFY Gourzenkyinplatz
               R: 553 User ambiguous.

                                 Example 3

The case of expanding a mailbox list requires a multiline reply as
shown in example 4.

                   Example of Expanding a Mailing List

       Either

          S: EXPN Example-People
          R: 250-Jon Postel <Postel@USC-ISIF.ARPA>
          R: 250-Fred Fonebone <Fonebone@USC-ISIQ.ARPA>
          R: 250-Sam Q. Smith <SQSmith@USC-ISIQ.ARPA>
          R: 250-Quincy Smith <@USC-ISIF.ARPA:Q-Smith@ISI-VAXA.ARPA>
          R: 250-<joe@foo-unix.ARPA>
          R: 250 <xyz@bar-unix.ARPA>

       Or

          S: EXPN Executive-Washroom-List
          R: 550 Access Denied to You.

                             Example 4

The character string arguments of the VRFY and EXPN commands
cannot be further restricted due to the variety of implementations
of the user name and mailbox list concepts. On some systems it
may be appropriate for the argument of the EXPN command to be a
file name for a file containing a mailing list, but again there is
a variety of file naming conventions in the Internet.

The VRFY and EXPN commands are not included in the minimum
implementation (Section 4.5.1), and are not required to work
across relays when they are implemented.

3.4. SENDING AND MAILING

   The main purpose of SMTP is to deliver messages to user's
   mailboxes. A very similar service provided by some hosts is to
   deliver messages to user's terminals (provided the user is active
   on the host). The delivery to the user's mailbox is called
   "mailing", the delivery to the user's terminal is called
   "sending". Because in many hosts the implementation of sending is
   nearly identical to the implementation of mailing these two
   functions are combined in SMTP. However the sending commands are
   not included in the required minimum implementation
   (Section 4.5.1). Users should have the ability to control the
   writing of messages on their terminals. Most hosts permit the
   users to accept or refuse such messages.

   The following three command are defined to support the sending
   options. These are used in the mail transaction instead of the
   MAIL command and inform the receiver-SMTP of the special semantics
   of this transaction:

      SEND <SP> FROM:<reverse-path> <CRLF>

         The SEND command requires that the mail data be delivered to
         the user's terminal. If the user is not active (or not
         accepting terminal messages) on the host a 450 reply may
         returned to a RCPT command. The mail transaction is
         successful if the message is delivered the terminal.

      SOML <SP> FROM:<reverse-path> <CRLF>

         The Send Or MaiL command requires that the mail data be
         delivered to the user's terminal if the user is active (and
         accepting terminal messages) on the host. If the user is
         not active (or not accepting terminal messages) then the
         mail data is entered into the user's mailbox. The mail
         transaction is successful if the message is delivered either
         to the terminal or the mailbox.

      SAML <SP> FROM:<reverse-path> <CRLF>

         The Send And MaiL command requires that the mail data be
         delivered to the user's terminal if the user is active (and
         accepting terminal messages) on the host. In any case the
         mail data is entered into the user's mailbox. The mail
         transaction is successful if the message is delivered the
         mailbox.

The same reply codes that are used for the MAIL commands are used
for these commands.

     3.5. OPENING AND CLOSING

        At the time the transmission channel is opened there is an
        exchange to ensure that the hosts are communicating with the hosts
        they think they are.

        The following two commands are used in transmission channel
        opening and closing:

           HELO <SP> <domain> <CRLF>

           QUIT <CRLF>

        In the HELO command the host sending the command identifies
        itself; the command may be interpreted as saying "Hello, I am
        <domain>".

                       Example of Connection Opening

           R: 220 BBN-UNIX.ARPA Simple Mail Transfer Service Ready
           S: HELO USC-ISIF.ARPA
           R: 250 BBN-UNIX.ARPA

                                 Example 5

                       Example of Connection Closing

           S: QUIT
           R: 221 BBN-UNIX.ARPA Service closing transmission channel

                                 Example 6

     3.6. RELAYING

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

        Conceptually the elements of the forward-path are moved to the
        reverse-path as the message is relayed from one server-SMTP to
        another. The reverse-path is a reverse source route, (i.e., a
        source route from the current location of the message to the
        originator of the message). When a server-SMTP deletes its
        identifier from the forward-path and inserts it into the
        reverse-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 server-SMTP is known by different names in
        different environments.

        If when the message arrives at an SMTP the first element of the
        forward-path is not the identifier of that SMTP the element is not
        deleted from the forward-path and is used to determine the next
        SMTP to send the message to. In any case, the SMTP adds its own
        identifier to the reverse-path.

        Using source routing the receiver-SMTP receives mail to be relayed
        to another server-SMTP The receiver-SMTP may accept or reject the
        task of relaying the mail in the same way it accepts or rejects
        mail for a local user. The receiver-SMTP transforms the command
        arguments by moving its own identifier from the forward-path to
        the beginning of the reverse-path. The receiver-SMTP then becomes
        a sender-SMTP, establishes a transmission channel to the next SMTP
        in the forward-path, and sends it the mail.

        The first host in the reverse-path should be the host sending the
        SMTP commands, and the first host in the forward-path should be
        the host receiving the SMTP commands.

        Notice that the forward-path and reverse-path appear in the SMTP
        commands and replies, but not necessarily in the message. That
        is, there is no need for these paths and especially this syntax to
        appear in the "To:" , "From:", "CC:", etc. fields of the message
        header.

        If a server-SMTP has accepted the task of relaying the mail and

later finds that the forward-path is incorrect or that the mail
cannot be delivered for whatever reason, then it must construct an
"undeliverable mail" notification message and send it to the
originator of the undeliverable mail (as indicated by the
reverse-path).

This notification message must be from the server-SMTP at this
host. Of course, server-SMTPs should not send notification
messages about problems with notification messages. One way to
prevent loops in error reporting is to specify a null reverse-path
in the MAIL command of a notification message. When such a
message is relayed it is permissible to leave the reverse-path
null. A MAIL command with a null reverse-path appears as follows:

    MAIL FROM:<>

An undeliverable mail notification message is shown in example 7.
This notification is in response to a message originated by JOE at
HOSTW and sent via HOSTX to HOSTY with instructions to relay it on
to HOSTZ. What we see in the example is the transaction between
HOSTY and HOSTX, which is the first step in the return of the
notification message.

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


              Example Undeliverable Mail Notification Message

       S: MAIL FROM:<>
       R: 250 ok
       S: RCPT TO:<@HOSTX.ARPA:JOE@HOSTW.ARPA>
       R: 250 ok
       S: DATA
       R: 354 send the mail data, end with .
       S: Date: 23 Oct 81 11:22:33
       S: From: SMTP@HOSTY.ARPA
       S: To: JOE@HOSTW.ARPA
       S: Subject: Mail System Problem
       S:
       S:    Sorry JOE, your message to SAM@HOSTZ.ARPA lost.
       S:   HOSTZ.ARPA said this:
       S:     "550 No Such User"
       S: .
       R: 250 ok


                                Example 7

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

   3.7. DOMAINS

      Domains are a recently introduced concept in the ARPA Internet
      mail system. The use of domains changes the address space from a
      flat global space of simple character string host names to a
      hierarchically structured rooted tree of global addresses. The
      host name is replaced by a domain and host designator which is a
      sequence of domain element strings separated by periods with the
      understanding that the domain elements are ordered from the most
      specific to the most general.

      For example, "USC-ISIF.ARPA", "Fred.Cambridge.UK", and
      "PC7.LCS.MIT.ARPA" might be host-and-domain identifiers.

      Whenever domain names are used in SMTP only the official names are
      used, the use of nicknames or aliases is not allowed.

    3.8. CHANGING ROLES

        The TURN command may be used to reverse the roles of the two
        programs communicating over the transmission channel.

        If program-A is currently the sender-SMTP and it sends the TURN
        command and receives an ok reply (250) then program-A becomes the
        receiver-SMTP.

        If program-B is currently the receiver-SMTP and it receives the
        TURN command and sends an ok reply (250) then program-B becomes
        the sender-SMTP.

        To refuse to change roles the receiver sends the 502 reply.

        Please note that this command is optional. It would not normally
        be used in situations where the transmission channel is TCP.
        However, when the cost of establishing the transmission channel is
        high, this command may be quite useful. For example, this command
        may be useful in supporting be mail exchange using the public
        switched telephone system as a transmission channel, especially if
        some hosts poll other hosts for mail exchanges.

   4. THE SMTP SPECIFICATIONS

      4.1. SMTP COMMANDS

         4.1.1. COMMAND SEMANTICS

            The SMTP commands define the mail transfer or the mail system
            function requested by the user. SMTP 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 SMTP commands are discussed
            below. The SMTP replies are discussed in the Section 4.2.

            A mail transaction involves several data objects which are
            communicated as arguments to different commands. The
            reverse-path is the argument of the MAIL command, the
            forward-path is the argument of the RCPT command, and the mail
            data is the argument of the DATA command. These arguments or
            data objects must be transmitted and held pending the
            confirmation communicated by the end of mail data indication
            which finalizes the transaction. The model for this is that
            distinct buffers are provided to hold the types of data
            objects, that is, there is a reverse-path buffer, a
            forward-path buffer, and a mail data buffer. Specific commands
            cause information to be appended to a specific buffer, or cause
            one or more buffers to be cleared.

            HELLO (HELO)

               This command is used to identify the sender-SMTP to the
               receiver-SMTP. The argument field contains the host name of
               the sender-SMTP.

               The receiver-SMTP identifies itself to the sender-SMTP in
               the connection greeting reply, and in the response to this
               command.

               This command and an OK reply to it confirm that both the
               sender-SMTP and the receiver-SMTP are in the initial state,
               that is, there is no transaction in progress and all state
               tables and buffers are cleared.

MAIL (MAIL)

This command is used to initiate a mail transaction in which
the mail data is delivered to one or more mailboxes. The
argument field contains a reverse-path.

The reverse-path consists of an optional list of hosts and
the sender mailbox. When the list of hosts is present, it
is a "reverse" source route 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 a
source route 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 IPCE to which it is
relaying the mail rather than the IPCE from which the mail
came (if they are different). In some types of error
reporting messages (for example, undeliverable mail
notifications) the reverse-path may be null (see Example 7).

This command clears the reverse-path buffer, the
forward-path buffer, and the mail data buffer; and inserts
the reverse-path information from this command into the
reverse-path buffer.

RECIPIENT (RCPT)

This command is used to identify an individual recipient of
the mail data; multiple recipients are specified by multiple
use of this command.

The forward-path consists of an optional list of hosts and a
required destination mailbox. When the list of hosts is
present, it is a source route and indicates that the mail
must be relayed to the next host on the list. If the
receiver-SMTP does not implement the relay function it may
user the same reply it would for an unknown local user
(550).

When mail is relayed, the relay host must remove itself from
the beginning forward-path and put itself at the beginning
of the reverse-path. When mail reaches its ultimate
destination (the forward-path contains only a destination
mailbox), the receiver-SMTP inserts it into the destination
mailbox in accordance with its host mail conventions.

For example, mail received at relay host A with arguments

        FROM:<USERX@HOSTY.ARPA>
        TO:<@HOSTA.ARPA,@HOSTB.ARPA:USERC@HOSTD.ARPA>

will be relayed on to host B with arguments

        FROM:<@HOSTA.ARPA:USERX@HOSTY.ARPA>
        TO:<@HOSTB.ARPA:USERC@HOSTD.ARPA>.

This command causes its forward-path argument to be appended to the forward-path buffer.

DATA (DATA)

The receiver treats the lines following the command as mail data from the sender. This command causes the mail data from this command to be appended to the mail data buffer. The mail data may contain any of the 128 ASCII character codes.

The mail data is terminated by a line containing only a period, that is the character sequence "<CRLF>.<CRLF>" (see Section 4.5.2 on Transparency). This is the end of mail data indication.

The end of mail data indication requires that the receiver must now process the stored mail transaction information. This processing consumes the information in the reverse-path buffer, the forward-path buffer, and the mail data buffer, and on the completion of this command these buffers are cleared. If the processing is successful the receiver must send an OK reply. If the processing fails completely the receiver must send a failure reply.

When the receiver-SMTP accepts a message either for relaying or for final delivery it inserts at the beginning of the mail data a time stamp line. The time stamp line indicates the identity of the host that sent the message, and the identity of the host that received the message (and is inserting this time stamp), and the date and time the message was received. Relayed messages will have multiple time stamp lines.

When the receiver-SMTP makes the "final delivery" of a message it inserts at the beginning of the mail data a

return path line. The return path line preserves the
information in the <reverse-path> from the MAIL command.
Here, final delivery means the message leaves the SMTP
world. Normally, this would mean it has been delivered to
the destination user, but in some cases it may be further
processed and transmitted by another mail system.

   It is possible for the mailbox in the return path be
   different from the actual sender's mailbox, for example,
   if error responses are to be delivered a special error
   handling mailbox rather than the message senders.

The preceding two paragraphs imply that the final mail data
will begin with a return path line, followed by one or more
time stamp lines. These lines will be followed by the mail
data header and body [2]. See Example 8.

Special mention is needed of the response and further action
required when the processing following the end of mail data
indication is partially successful. This could arise if
after accepting several recipients and the mail data, the
receiver-SMTP finds that the mail data can be successfully
delivered to some of the recipients, but it cannot be to
others (for example, due to mailbox space allocation
problems). In such a situation, the response to the DATA
command must be an OK reply. But, the receiver-SMTP must
compose and send an "undeliverable mail" notification
message to the originator of the message. Either a single
notification which lists all of the recipients that failed
to get the message, or separate notification messages must
be sent for each failed recipient (see Example 7). All
undeliverable mail notification messages are sent using the
MAIL command (even if they result from processing a SEND,
SOML, or SAML command).

             Example of Return Path and Received Time Stamps

      Return-Path: <@GHI.ARPA,@DEF.ARPA,@ABC.ARPA:JOE@ABC.ARPA>
      Received: from GHI.ARPA by JKL.ARPA ; 27 Oct 81 15:27:39 PST
      Received: from DEF.ARPA by GHI.ARPA ; 27 Oct 81 15:15:13 PST
      Received: from ABC.ARPA by DEF.ARPA ; 27 Oct 81 15:01:59 PST
      Date: 27 Oct 81 15:01:01 PST
      From: JOE@ABC.ARPA
      Subject: Improved Mailing System Installed
      To: SAM@JKL.ARPA

      This is to inform you that ...

                              Example 8

          SEND (SEND)

             This command is used to initiate a mail transaction in which
             the mail data is delivered to one or more terminals. The
             argument field contains a reverse-path. This command is
             successful if the message is delivered to a terminal.

             The reverse-path consists of an optional list of hosts and
             the sender mailbox. When the list of hosts is present, it
             is a "reverse" source route 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 a
             source route 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 IPCE to which it is
             relaying the mail rather than the IPCE from which the mail
             came (if they are different).

             This command clears the reverse-path buffer, the
             forward-path buffer, and the mail data buffer; and inserts
             the reverse-path information from this command into the
             reverse-path buffer.

          SEND OR MAIL (SOML)

             This command is used to initiate a mail transaction in which
             the mail data is delivered to one or more terminals or

mailboxes. For each recipient the mail data is delivered to
the recipient's terminal if the recipient is active on the
host (and accepting terminal messages), otherwise to the
recipient's mailbox. The argument field contains a
reverse-path. This command is successful if the message is
delivered to a terminal or the mailbox.

The reverse-path consists of an optional list of hosts and
the sender mailbox. When the list of hosts is present, it
is a "reverse" source route 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 a
source route 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 IPCE to which it is
relaying the mail rather than the IPCE from which the mail
came (if they are different).

This command clears the reverse-path buffer, the
forward-path buffer, and the mail data buffer; and inserts
the reverse-path information from this command into the
reverse-path buffer.

   SEND AND MAIL (SAML)

This command is used to initiate a mail transaction in which
the mail data is delivered to one or more terminals and
mailboxes. For each recipient the mail data is delivered to
the recipient's terminal if the recipient is active on the
host (and accepting terminal messages), and for all
recipients to the recipient's mailbox. The argument field
contains a reverse-path. This command is successful if the
message is delivered to the mailbox.

The reverse-path consists of an optional list of hosts and
the sender mailbox. When the list of hosts is present, it
is a "reverse" source route 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 a
source route 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 IPCE to which it is
relaying the mail rather than the IPCE from which the mail
came (if they are different).

This command clears the reverse-path buffer, the

forward-path buffer, and the mail data buffer; and inserts
the reverse-path information from this command into the
reverse-path buffer.

RESET (RSET)

This command specifies that the current mail transaction is
to be aborted. Any stored sender, recipients, and mail data
must be discarded, and all buffers and state tables cleared.
The receiver must send an OK reply.

VERIFY (VRFY)

This command asks the receiver to confirm that the argument
identifies a user. If it is a user name, the full name of
the user (if known) and the fully specified mailbox are
returned.

This command has no effect on any of the reverse-path
buffer, the forward-path buffer, or the mail data buffer.

EXPAND (EXPN)

This command asks the receiver to confirm that the argument
identifies a mailing list, and if so, to return the
membership of that list. The full name of the users (if
known) and the fully specified mailboxes are returned in a
multiline reply.

This command has no effect on any of the reverse-path
buffer, the forward-path buffer, or the mail data buffer.

HELP (HELP)

This command causes the receiver to send helpful information
to the sender of the HELP command. The command may take an
argument (e.g., any command name) and return more specific
information as a response.

This command has no effect on any of the reverse-path
buffer, the forward-path buffer, or the mail data buffer.

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.

   This command has no effect on any of the reverse-path
   buffer, the forward-path buffer, or the mail data buffer.

QUIT (QUIT)

   This command specifies that the receiver must send an OK
   reply, and then close the transmission channel.

   The receiver should not close the transmission channel until
   it receives and replies to a QUIT command (even if there was
   an error). The sender should not close the transmission
   channel until it send a QUIT command and receives the reply
   (even if there was an error response to a previous command).
   If the connection is closed prematurely the receiver should
   act as if a RSET command had been received (canceling any
   pending transaction, but not undoing any previously
   completed transaction), the sender should act as if the
   command or transaction in progress had received a temporary
   error (4xx).

TURN (TURN)

   This command specifies that the receiver must either (1)
   send an OK reply and then take on the role of the
   sender-SMTP, or (2) send a refusal reply and retain the role
   of the receiver-SMTP.

   If program-A is currently the sender-SMTP and it sends the
   TURN command and receives an OK reply (250) then program-A
   becomes the receiver-SMTP. Program-A is then in the initial
   state as if the transmission channel just opened, and it
   then sends the 220 service ready greeting.

   If program-B is currently the receiver-SMTP and it receives
   the TURN command and sends an OK reply (250) then program-B
   becomes the sender-SMTP. Program-B is then in the initial
   state as if the transmission channel just opened, and it
   then expects to receive the 220 service ready greeting.

   To refuse to change roles the receiver sends the 502 reply.

There are restrictions on the order in which these command may
be used.

   The first command in a session must be the HELO command.
   The HELO command may be used later in a session as well. If
   the HELO command argument is not acceptable a 501 failure
   reply must be returned and the receiver-SMTP must stay in
   the same state.

   The NOOP, HELP, EXPN, and VRFY commands can be used at any
   time during a session.

   The MAIL, SEND, SOML, or SAML commands begin a mail
   transaction. Once started a mail transaction consists of
   one of the transaction beginning commands, one or more RCPT
   commands, and a DATA command, in that order. A mail
   transaction may be aborted by the RSET command. There may
   be zero or more transactions in a session.

   If the transaction beginning command argument is not
   acceptable a 501 failure reply must be returned and the
   receiver-SMTP must stay in the same state. If the commands
   in a transaction are out of order a 503 failure reply must
   be returned and the receiver-SMTP must stay in the same
   state.

   The last command in a session must be the QUIT command. The
   QUIT command can not be used at any other time in a session.

4.1.2. COMMAND SYNTAX

   The commands consist of a command code followed by an argument
   field. 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 "TO" or "to" for the forward-path. Command codes and
   the argument fields are separated by one or more spaces.
   However, within the reverse-path and forward-path arguments
   case is important. In particular, in some hosts the user
   "smith" is different from the user "Smith".

The argument field consists of a variable length character
string ending with the character sequence <CRLF>. The receiver
is to take no action until this sequence is received.

Square brackets denote an optional argument field. If the
option is not taken, the appropriate default is implied.

          The following are the SMTP commands:

               HELO <SP> <domain> <CRLF>

               MAIL <SP> FROM:<reverse-path> <CRLF>

               RCPT <SP> TO:<forward-path> <CRLF>

               DATA <CRLF>

               RSET <CRLF>

               SEND <SP> FROM:<reverse-path> <CRLF>

               SOML <SP> FROM:<reverse-path> <CRLF>

               SAML <SP> FROM:<reverse-path> <CRLF>

               VRFY <SP> <string> <CRLF>

               EXPN <SP> <string> <CRLF>

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

               NOOP <CRLF>

               QUIT <CRLF>

               TURN <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.

     <reverse-path> ::= <path>

     <forward-path> ::= <path>

     <path> ::= "<" [<a-d-l> ":"] <mailbox> ">"

     <a-d-l> ::= <at-domain> | <at-domain> "," <a-d-l>

     <at-domain> ::= "@" <domain>

     <domain> ::= <element> | <element> "." <domain>

     <element> ::= <name> | "#" <number> | "[" <dotnum> "]"

     <mailbox> ::= <local-part> "@" <domain>

     <local-part> ::= <dot-string> | <quoted-string>

     <name> ::= <a> <ldh-str> <let-dig>

     <ldh-str> ::= <let-dig-hyp> | <let-dig-hyp> <ldh-str>

     <let-dig> ::= <a> | <d>

     <let-dig-hyp> ::= <a> | <d> | "-"

     <dot-string> ::= <string> | <string> "." <dot-string>

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

     <quoted-string> ::= """ <qtext> """

     <qtext> ::= "\" <x> | "\" <x> <qtext> | <q> | <q> <qtext>

     <char> ::= <c> | "\" <x>

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

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

     <CRLF> ::= <CR> <LF>

```
         <CR> ::= the carriage return character (ASCII code 13)

         <LF> ::= the line feed character (ASCII code 10)

         <SP> ::= the space character (ASCII code 32)

         <snum> ::= one, two, or three digits representing a decimal
                    integer value in the range 0 through 255

         <a> ::= any one of the 52 alphabetic characters A through Z
                 in upper case and a through z in lower case

         <c> ::= any one of the 128 ASCII characters, but not any
                 <special> or <SP>

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

         <q> ::= any one of the 128 ASCII characters except <CR>,
                 <LF>, quote ("), or backslash (\)

         <x> ::= any one of the 128 ASCII characters (no exceptions)

         <special> ::= "<" | ">" | "(" | ")" | "[" | "]" | "\" | "."
                   | "," | ";" | ":" | "@"  """ | the control
                   characters (ASCII codes 0 through 31 inclusive and
                   127)
```

   Note that the backslash, "\", is a quote character, which is
   used to indicate that the next character is to be used
   literally (instead of 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. Note that the name elements of domains
   are the official names -- no use of nicknames or aliases is
   allowed.

   Sometimes a host is not known to the translation function and
   communication is blocked. To bypass this barrier two 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., "[123.255.37.2]", which indicates a 32-bit ARPA Internet
   Address in four 8-bit fields.

The time stamp line and the return path line are formally
defined as follows:

<return-path-line> ::= "Return-Path:" <SP><reverse-path><CRLF>

<time-stamp-line> ::= "Received:" <SP> <stamp> <CRLF>

    <stamp> ::= <from-domain> <by-domain> <opt-info> ";"
                <daytime>

    <from-domain> ::= "FROM" <SP> <domain> <SP>

    <by-domain> ::= "BY" <SP> <domain> <SP>

    <opt-info> ::= [<via>] [<with>] [<id>] [<for>]

    <via> ::= "VIA" <SP> <link> <SP>

    <with> ::= "WITH" <SP> <protocol> <SP>

    <id> ::= "ID" <SP> <string> <SP>

    <for> ::= "FOR" <SP> <path> <SP>

    <link> ::= The standard names for links are registered with
               the Network Information Center.

    <protocol> ::= The standard names for protocols are
                   registered with the Network Information Center.

    <daytime> ::= <SP> <date> <SP> <time>

    <date> ::= <dd> <SP> <mon> <SP> <yy>

    <time> ::= <hh> ":" <mm> ":" <ss> <SP> <zone>

    <dd> ::= the one or two decimal integer day of the month in
             the range 1 to 31.

    <mon> ::= "JAN" | "FEB" | "MAR" | "APR" | "MAY" | "JUN" |
              "JUL" | "AUG" | "SEP" | "OCT" | "NOV" | "DEC"

    <yy> ::= the two decimal integer year of the century in the
             range 00 to 99.

        <hh> ::= the two decimal integer hour of the day in the
                 range 00 to 24.

        <mm> ::= the two decimal integer minute of the hour in the
                 range 00 to 59.

        <ss> ::= the two decimal integer second of the minute in the
                 range 00 to 59.

        <zone> ::= "UT" for Universal Time (the default) or other
                   time zone designator (as in [2]).

                          Return Path Example

           Return-Path: <@CHARLIE.ARPA,@BAKER.ARPA:JOE@ABLE.ARPA>

                              Example 9

                         Time Stamp Line Example

         Received: FROM ABC.ARPA BY XYZ.ARPA ; 22 OCT 81 09:23:59 PDT

            Received: from ABC.ARPA by XYZ.ARPA via TELENET with X25
                     id M12345 for Smith@PDQ.ARPA ; 22 OCT 81 09:23:59 PDT

                              Example 10

   4.2. SMTP REPLIES

      Replies to SMTP commands are devised to ensure the synchronization
      of requests and actions in the process of mail transfer, and to
      guarantee that the sender-SMTP always knows the state of the
      receiver-SMTP. Every command must generate exactly one reply.

         The details of the command-reply sequence are made explicit in
         Section 5.3 on Sequencing and Section 5.4 State Diagrams.

      An SMTP 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-SMTP need not 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 and context dependent, so there are
      likely to be varying texts for each reply code. A discussion of
      the theory of reply codes is given in Appendix E. Formally, a
      reply is defined to be the sequence: a three-digit code, <SP>,
      one line of text, and <CRLF>, or a multiline reply (as defined in
      Appendix E). Only the EXPN and HELP commands are expected to
      result in multiline replies in normal circumstances, however
      multiline replies are allowed for any command.

       4.2.1. REPLY CODES BY FUNCTION GROUPS

          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]

          220 <domain> Service ready
          221 <domain> Service closing transmission channel
          421 <domain> Service not available,
              closing transmission channel
             [This may be a reply to any command if the service knows it
             must shut down]

          250 Requested mail action okay, completed
          251 User not local; will forward to <forward-path>
          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: error in processing
          551 User not local; please try <forward-path>
          452 Requested action not taken: insufficient system storage
          552 Requested mail action aborted: exceeded storage allocation
          553 Requested action not taken: mailbox name not allowed
             [E.g., mailbox syntax incorrect]
          354 Start mail input; end with <CRLF>.<CRLF>
          554 Transaction failed

         4.2.2. NUMERIC ORDER LIST OF REPLY CODES

            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]
            220 <domain> Service ready
            221 <domain> Service closing transmission channel
            250 Requested mail action okay, completed
            251 User not local; will forward to <forward-path>

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

            421 <domain> 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]
            551 User not local; please try <forward-path>
            552 Requested mail action aborted: exceeded storage allocation
            553 Requested action not taken: mailbox name not allowed
               [E.g., mailbox syntax incorrect]
            554 Transaction failed

   4.3.   SEQUENCING OF COMMANDS AND REPLIES

      The communication between the sender and receiver is intended to
      be an alternating dialogue, controlled by the sender. As such,
      the sender issues a command and the receiver responds with a
      reply. The sender must wait for this response before sending
      further commands.

      One important reply is the connection greeting. Normally, a
      receiver will send a 220 "Service ready" reply when the connection
      is completed. The sender should wait for this greeting message
      before sending any commands.

         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.ARPA <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. The prefixes
         used before the possible replies are "P" for preliminary (not
         used in SMTP), "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 SMTP-receiver knows it must shut down. This
         listing forms the basis for the State Diagrams in Section 4.4.

            CONNECTION ESTABLISHMENT
               S: 220
               F: 421
            HELO
               S: 250
               E: 500, 501, 504, 421
            MAIL
               S: 250
               F: 552, 451, 452
               E: 500, 501, 421

                  RCPT
                     S: 250, 251
                     F: 550, 551, 552, 553, 450, 451, 452
                     E: 500, 501, 503, 421
                  DATA
                     I: 354 -> data -> S: 250
                                        F: 552, 554, 451, 452
                     F: 451, 554
                     E: 500, 501, 503, 421
                  RSET
                     S: 250
                     E: 500, 501, 504, 421
                  SEND
                     S: 250
                     F: 552, 451, 452
                     E: 500, 501, 502, 421
                  SOML
                     S: 250
                     F: 552, 451, 452
                     E: 500, 501, 502, 421
                  SAML
                     S: 250
                     F: 552, 451, 452
                     E: 500, 501, 502, 421
                  VRFY
                     S: 250, 251
                     F: 550, 551, 553
                     E: 500, 501, 502, 504, 421
                  EXPN
                     S: 250
                     F: 550
                     E: 500, 501, 502, 504, 421
                  HELP
                     S: 211, 214
                     E: 500, 501, 502, 504, 421
                  NOOP
                     S: 250
                     E: 500, 421
                  QUIT
                     S: 221
                     E: 500
                  TURN
                     S: 250
                     F: 502
                     E: 500, 503

    4.4. STATE DIAGRAMS

       Following are state diagrams for a simple-minded SMTP
       implementation. Only the first digit of the reply codes is used.
       There is one state diagram for each group of SMTP 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 SMTP commands:

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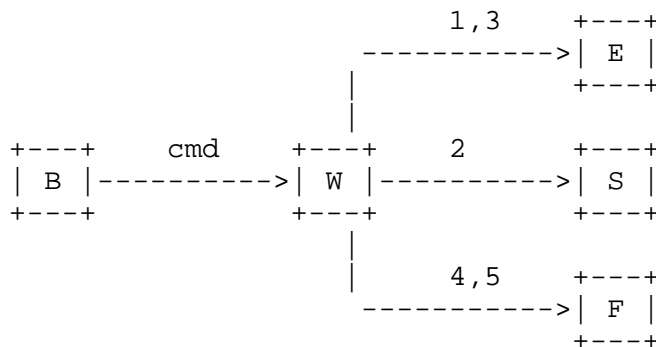
          This diagram models the commands:

             HELO, MAIL, RCPT, RSET, SEND, SOML, SAML, VRFY, EXPN, HELP,
             NOOP, QUIT, TURN.

A more complex diagram models the DATA command:

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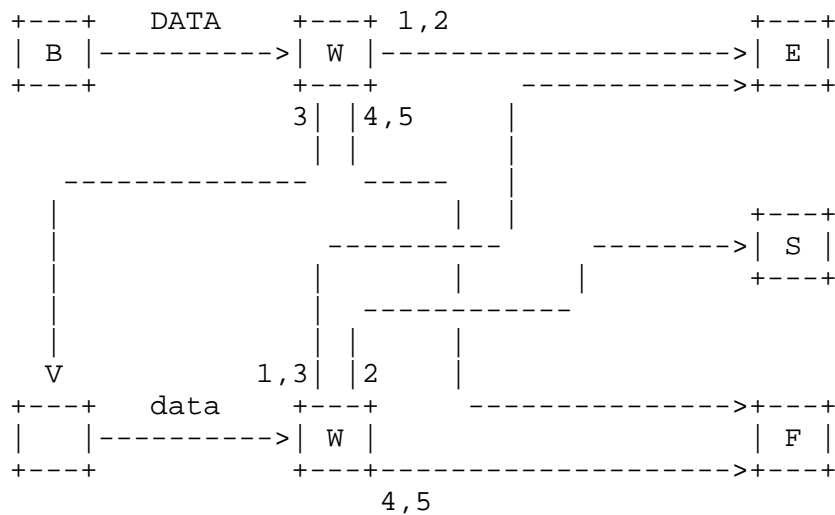
Note that the "data" here is a series of lines sent from the
sender to the receiver with no response expected until the last
line is sent.

     4.5. DETAILS

       4.5.1. MINIMUM IMPLEMENTATION

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

             COMMANDS -- HELO
                         MAIL
                         RCPT
                         DATA
                         RSET
                         NOOP
                         QUIT

       4.5.2. TRANSPARENCY

          Without some provision for data transparency the character
          sequence "<CRLF>.<CRLF>" ends 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-SMTP 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-SMTP
             it checks 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.

          The mail data may contain any of the 128 ASCII characters. All
          characters are to be delivered to the recipient's mailbox
          including format effectors and other control characters. If
          the transmission channel provides an 8-bit byte (octets) data
          stream, the 7-bit ASCII codes are transmitted right justified
          in the octets with the high order bits cleared to zero.

             In some systems it may be necessary to transform the data as
             it is received and stored. This may be necessary for hosts
             that use a different character set than ASCII as their local
             character set, or that store data in records rather than

strings. If such transforms are necessary, they must be
reversible -- especially if such transforms are applied to
mail being relayed.

4.5.3. SIZES

There are several objects that have required minimum maximum
sizes. That is, every implementation must be able to receive
objects of at least these sizes, but must not send objects
larger than these sizes.

```
**************************************************
*                                                *
*  TO THE MAXIMUM EXTENT POSSIBLE, IMPLEMENTATION *
*  TECHNIQUES WHICH IMPOSE NO LIMITS ON THE LENGTH *
*  OF THESE OBJECTS SHOULD BE USED.               *
*                                                *
**************************************************
```

user

The maximum total length of a user name is 64 characters.

domain

The maximum total length of a domain name or number is 64
characters.

path

The maximum total length of a reverse-path or
forward-path is 256 characters (including the punctuation
and element separators).

command line

The maximum total length of a command line including the
command word and the <CRLF> is 512 characters.

reply line

The maximum total length of a reply line including the
reply code and the <CRLF> is 512 characters.

         text line

            The maximum total length of a text line including the
            <CRLF> is 1000 characters (but not counting the leading
            dot duplicated for transparency).

         recipients buffer

            The maximum total number of recipients that must be
            buffered is 100 recipients.

         ****
         *   *
         *   TO THE MAXIMUM EXTENT POSSIBLE, IMPLEMENTATION *
         *   TECHNIQUES WHICH IMPOSE NO LIMITS ON THE LENGTH *
         *   OF THESE OBJECTS SHOULD BE USED.   *
         *   *
         ****

      Errors due to exceeding these limits may be reported by using
      the reply codes, for example:

         500 Line too long.

         501 Path too long

         552 Too many recipients.

         552 Too much mail data.

APPENDIX A

   TCP Transport service

      The Transmission Control Protocol [3] is used in the ARPA
      Internet, and in any network following the US DoD standards for
      internetwork protocols.

      Connection Establishment

         The SMTP 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 25 (31 octal), that is L=25.

      Data Transfer

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

APPENDIX B

  NCP Transport service

      The ARPANET Host-to-Host Protocol [4] (implemented by the Network
      Control Program) may be used in the ARPANET.

      Connection Establishment

          The SMTP transmission channel is established via NCP between
          the sender process socket U and receiver process socket L. The
          Initial Connection Protocol [5] 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 25 (31 octal), that is L=25.

      Data Transfer

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

APPENDIX C

   NITS

      The Network Independent Transport Service [6] may be used.

      Connection Establishment

         The SMTP transmission channel is established via NITS between
         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 SMTP data is 7-bit ASCII characters. Each character is
         transmitted as an 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 [7] as provided by the
      Public Data Networks directly, however, 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-SMTP will be able to
      determine its next action (proceed as planned, redo, retrench,
      etc.) by simply examining this first digit. A sender-SMTP 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-SMTP should send
               another command specifying whether to continue or abort
               the action.

                  [Note: SMTP does not have any commands that allow this
                  type of reply, and so does not have the continue or
                  abort commands.]

            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-SMTP 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- SMTPs) must
agree on the interpretation. Each reply in this category
might have a different time value, but the sender-SMTP 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
and 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-SMTP 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-SMTP 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-SMTP any new information will return
         a 250 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-SMTP knows when it
   can stop reading the reply. This requires a special format to
   indicate a multiple line reply.

         The format for multiline 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

         In many cases the sender-SMTP then simply needs to search for
         the reply code followed by <SP> at the beginning of a line, and
         ignore all preceding lines. In a few cases, there is important
         data for the sender in the reply "text". The sender will know
         these cases from the current context.

APPENDIX F

   Scenarios

      This section presents complete scenarios of several types of SMTP
      sessions.

   A Typical SMTP Transaction Scenario

      This SMTP example shows mail sent by Smith at host USC-ISIF, to
      Jones, Green, and Brown at host BBN-UNIX. Here we assume that
      host USC-ISIF contacts host BBN-UNIX directly. The mail is
      accepted for Jones and Brown. Green does not have a mailbox at
      host BBN-UNIX.

         R: 220 BBN-UNIX.ARPA Simple Mail Transfer Service Ready
         S: HELO USC-ISIF.ARPA
         R: 250 BBN-UNIX.ARPA

         S: MAIL FROM:<Smith@USC-ISIF.ARPA>
         R: 250 OK

         S: RCPT TO:<Jones@BBN-UNIX.ARPA>
         R: 250 OK

         S: RCPT TO:<Green@BBN-UNIX.ARPA>
         R: 550 No such user here

         S: RCPT TO:<Brown@BBN-UNIX.ARPA>
         R: 250 OK

         S: DATA
         R: 354 Start mail input; end with <CRLF>.<CRLF>
         S: Blah blah blah...
         S: ...etc. etc. etc.
         S: .
         R: 250 OK

         S: QUIT
         R: 221 BBN-UNIX.ARPA Service closing transmission channel

                           Scenario 1

     Aborted SMTP Transaction Scenario

            R: 220 MIT-Multics.ARPA Simple Mail Transfer Service Ready
            S: HELO ISI-VAXA.ARPA
            R: 250 MIT-Multics.ARPA

            S: MAIL FROM:<Smith@ISI-VAXA.ARPA>
            R: 250 OK

            S: RCPT TO:<Jones@MIT-Multics.ARPA>
            R: 250 OK

            S: RCPT TO:<Green@MIT-Multics.ARPA>
            R: 550 No such user here

            S: RSET
            R: 250 OK

            S: QUIT
            R: 221 MIT-Multics.ARPA Service closing transmission channel

                                 Scenario 2

     Relayed Mail Scenario

          Step 1 -- Source Host to Relay Host

             R: 220 USC-ISIE.ARPA Simple Mail Transfer Service Ready
             S: HELO MIT-AI.ARPA
             R: 250 USC-ISIE.ARPA

             S: MAIL FROM:<JQP@MIT-AI.ARPA>
             R: 250 OK

             S: RCPT TO:<@USC-ISIE.ARPA:Jones@BBN-VAX.ARPA>
             R: 250 OK

             S: DATA
             R: 354 Start mail input; end with <CRLF>.<CRLF>
             S: Date: 2 Nov 81 22:33:44
             S: From: John Q. Public <JQP@MIT-AI.ARPA>
             S: Subject: The Next Meeting of the Board
             S: To: Jones@BBN-Vax.ARPA
             S:
             S: Bill:
             S: The next meeting of the board of directors will be
             S: on Tuesday.
             S: John.
             S: .
             R: 250 OK

             S: QUIT
             R: 221 USC-ISIE.ARPA Service closing transmission channel

        Step 2 -- Relay Host to Destination Host

          R: 220 BBN-VAX.ARPA Simple Mail Transfer Service Ready
          S: HELO USC-ISIE.ARPA
          R: 250 BBN-VAX.ARPA

          S: MAIL FROM:<@USC-ISIE.ARPA:JQP@MIT-AI.ARPA>
          R: 250 OK

          S: RCPT TO:<Jones@BBN-VAX.ARPA>
          R: 250 OK

          S: DATA
          R: 354 Start mail input; end with <CRLF>.<CRLF>
          S: Received: from MIT-AI.ARPA by USC-ISIE.ARPA ;
             2 Nov 81 22:40:10 UT
          S: Date: 2 Nov 81 22:33:44
          S: From: John Q. Public <JQP@MIT-AI.ARPA>
          S: Subject: The Next Meeting of the Board
          S: To: Jones@BBN-Vax.ARPA
          S:
          S: Bill:
          S: The next meeting of the board of directors will be
          S: on Tuesday.
          S: John.
          S: .
          R: 250 OK

          S: QUIT
          R: 221 USC-ISIE.ARPA Service closing transmission channel

                              Scenario 3

     Verifying and Sending Scenario

          R: 220 SU-SCORE.ARPA Simple Mail Transfer Service Ready
          S: HELO MIT-MC.ARPA
          R: 250 SU-SCORE.ARPA

          S: VRFY Crispin
          R: 250 Mark Crispin <Admin.MRC@SU-SCORE.ARPA>

          S: SEND FROM:<EAK@MIT-MC.ARPA>
          R: 250 OK

          S: RCPT TO:<Admin.MRC@SU-SCORE.ARPA>
          R: 250 OK

          S: DATA
          R: 354 Start mail input; end with <CRLF>.<CRLF>
          S: Blah blah blah...
          S: ...etc. etc. etc.
          S: .
          R: 250 OK

          S: QUIT
          R: 221 SU-SCORE.ARPA Service closing transmission channel

                              Scenario 4

   Sending and Mailing Scenarios

      First the user's name is verified, then an attempt is made to
      send to the user's terminal. When that fails, the messages is
      mailed to the user's mailbox.

         R: 220 SU-SCORE.ARPA Simple Mail Transfer Service Ready
         S: HELO MIT-MC.ARPA
         R: 250 SU-SCORE.ARPA

         S: VRFY Crispin
         R: 250 Mark Crispin <Admin.MRC@SU-SCORE.ARPA>

         S: SEND FROM:<EAK@MIT-MC.ARPA>
         R: 250 OK

         S: RCPT TO:<Admin.MRC@SU-SCORE.ARPA>
         R: 450 User not active now

         S: RSET
         R: 250 OK

         S: MAIL FROM:<EAK@MIT-MC.ARPA>
         R: 250 OK

         S: RCPT TO:<Admin.MRC@SU-SCORE.ARPA>
         R: 250 OK

         S: DATA
         R: 354 Start mail input; end with <CRLF>.<CRLF>
         S: Blah blah blah...
         S: ...etc. etc. etc.
         S: .
         R: 250 OK

         S: QUIT
         R: 221 SU-SCORE.ARPA Service closing transmission channel

                              Scenario 5

     Doing the preceding scenario more efficiently.

        R: 220 SU-SCORE.ARPA Simple Mail Transfer Service Ready
        S: HELO MIT-MC.ARPA
        R: 250 SU-SCORE.ARPA

        S: VRFY Crispin
        R: 250 Mark Crispin <Admin.MRC@SU-SCORE.ARPA>

        S: SOML FROM:<EAK@MIT-MC.ARPA>
        R: 250 OK

        S: RCPT TO:<Admin.MRC@SU-SCORE.ARPA>
        R: 250 User not active now, so will do mail.

        S: DATA
        R: 354 Start mail input; end with <CRLF>.<CRLF>
        S: Blah blah blah...
        S: ...etc. etc. etc.
        S: .
        R: 250 OK

        S: QUIT
        R: 221 SU-SCORE.ARPA Service closing transmission channel

                            Scenario 6

     Mailing List Scenario

        First each of two mailing lists are expanded in separate sessions
        with different hosts. Then the message is sent to everyone that
        appeared on either list (but no duplicates) via a relay host.

           Step 1 -- Expanding the First List

              R: 220 MIT-AI.ARPA Simple Mail Transfer Service Ready
              S: HELO SU-SCORE.ARPA
              R: 250 MIT-AI.ARPA

              S: EXPN Example-People
              R: 250-<ABC@MIT-MC.ARPA>
              R: 250-Fred Fonebone <Fonebone@USC-ISIQ.ARPA>
              R: 250-Xenon Y. Zither <XYZ@MIT-AI.ARPA>
              R: 250-Quincy Smith <@USC-ISIF.ARPA:Q-Smith@ISI-VAXA.ARPA>
              R: 250-<joe@foo-unix.ARPA>
              R: 250 <xyz@bar-unix.ARPA>

              S: QUIT
              R: 221 MIT-AI.ARPA Service closing transmission channel

          Step 2 -- Expanding the Second List

             R: 220 MIT-MC.ARPA Simple Mail Transfer Service Ready
             S: HELO SU-SCORE.ARPA
             R: 250 MIT-MC.ARPA

             S: EXPN Interested-Parties
             R: 250-Al Calico <ABC@MIT-MC.ARPA>
             R: 250-<XYZ@MIT-AI.ARPA>
             R: 250-Quincy Smith <@USC-ISIF.ARPA:Q-Smith@ISI-VAXA.ARPA>
             R: 250-<fred@BBN-UNIX.ARPA>
             R: 250 <xyz@bar-unix.ARPA>

             S: QUIT
             R: 221 MIT-MC.ARPA Service closing transmission channel

          Step 3 -- Mailing to All via a Relay Host

             R: 220 USC-ISIE.ARPA Simple Mail Transfer Service Ready
             S: HELO SU-SCORE.ARPA
             R: 250 USC-ISIE.ARPA

             S: MAIL FROM:<Account.Person@SU-SCORE.ARPA>
             R: 250 OK
             S: RCPT TO:<@USC-ISIE.ARPA:ABC@MIT-MC.ARPA>
             R: 250 OK
             S: RCPT TO:<@USC-ISIE.ARPA:Fonebone@USC-ISIQA.ARPA>
             R: 250 OK
             S: RCPT TO:<@USC-ISIE.ARPA:XYZ@MIT-AI.ARPA>
             R: 250 OK
             S: RCPT
                 TO:<@USC-ISIE.ARPA,@USC-ISIF.ARPA:Q-Smith@ISI-VAXA.ARPA>
             R: 250 OK
             S: RCPT TO:<@USC-ISIE.ARPA:joe@FOO-UNIX.ARPA>
             R: 250 OK
             S: RCPT TO:<@USC-ISIE.ARPA:xyz@BAR-UNIX.ARPA>
             R: 250 OK
             S: RCPT TO:<@USC-ISIE.ARPA:fred@BBN-UNIX.ARPA>
             R: 250 OK

             S: DATA
             R: 354 Start mail input; end with <CRLF>.<CRLF>
             S: Blah blah blah...
             S: ...etc. etc. etc.
             S: .
             R: 250 OK

             S: QUIT
             R: 221 USC-ISIE.ARPA Service closing transmission channel

                              Scenario 7

     Forwarding Scenarios

           R: 220 USC-ISIF.ARPA Simple Mail Transfer Service Ready
           S: HELO LBL-UNIX.ARPA
           R: 250 USC-ISIF.ARPA

           S: MAIL FROM:<mo@LBL-UNIX.ARPA>
           R: 250 OK

           S: RCPT TO:<fred@USC-ISIF.ARPA>
           R: 251 User not local; will forward to <Jones@USC-ISI.ARPA>

           S: DATA
           R: 354 Start mail input; end with <CRLF>.<CRLF>
           S: Blah blah blah...
           S: ...etc. etc. etc.
           S: .
           R: 250 OK

           S: QUIT
           R: 221 USC-ISIF.ARPA Service closing transmission channel

                                Scenario 8

        Step 1 -- Trying the Mailbox at the First Host

           R: 220 USC-ISIF.ARPA Simple Mail Transfer Service Ready
           S: HELO LBL-UNIX.ARPA
           R: 250 USC-ISIF.ARPA

           S: MAIL FROM:<mo@LBL-UNIX.ARPA>
           R: 250 OK

           S: RCPT TO:<fred@USC-ISIF.ARPA>
           R: 251 User not local; will forward to <Jones@USC-ISI.ARPA>

           S: RSET
           R: 250 OK

           S: QUIT
           R: 221 USC-ISIF.ARPA Service closing transmission channel

        Step 2 -- Delivering the Mail at the Second Host

           R: 220 USC-ISI.ARPA Simple Mail Transfer Service Ready
           S: HELO LBL-UNIX.ARPA
           R: 250 USC-ISI.ARPA

           S: MAIL FROM:<mo@LBL-UNIX.ARPA>
           R: 250 OK

           S: RCPT TO:<Jones@USC-ISI.ARPA>
           R: OK

           S: DATA
           R: 354 Start mail input; end with <CRLF>.<CRLF>
           S: Blah blah blah...
           S: ...etc. etc. etc.
           S: .
           R: 250 OK

           S: QUIT
           R: 221 USC-ISI.ARPA Service closing transmission channel

                           Scenario 9

    Too Many Recipients Scenario

          R: 220 BERKELEY.ARPA Simple Mail Transfer Service Ready
          S: HELO USC-ISIF.ARPA
          R: 250 BERKELEY.ARPA

          S: MAIL FROM:<Postel@USC-ISIF.ARPA>
          R: 250 OK

          S: RCPT TO:<fabry@BERKELEY.ARPA>
          R: 250 OK

          S: RCPT TO:<eric@BERKELEY.ARPA>
          R: 552 Recipient storage full, try again in another transaction

          S: DATA
          R: 354 Start mail input; end with <CRLF>.<CRLF>
          S: Blah blah blah...
          S: ...etc. etc. etc.
          S: .
          R: 250 OK

          S: MAIL FROM:<Postel@USC-ISIF.ARPA>
          R: 250 OK

          S: RCPT TO:<eric@BERKELEY.ARPA>
          R: 250 OK

          S: DATA
          R: 354 Start mail input; end with <CRLF>.<CRLF>
          S: Blah blah blah...
          S: ...etc. etc. etc.
          S: .
          R: 250 OK

          S: QUIT
          R: 221 BERKELEY.ARPA Service closing transmission channel

                              Scenario 10

    Note that a real implementation must handle many recipients as
    specified in Section 4.5.3.

GLOSSARY

   ASCII

      American Standard Code for Information Interchange [1].

   command

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

   domain

      The hierarchially structured global character string address of a
      host computer in the mail system.

   end of mail data indication

      A special sequence of characters that indicates the end of the
      mail data. In particular, the five characters carriage return,
      line feed, period, carriage return, line feed, in that order.

   host

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

   line

      A a sequence of ASCII characters ending with a <CRLF>.

   mail data

      A sequence of ASCII characters of arbitrary length, which conforms
      to the standard set in the Standard for the Format of ARPA
      Internet Text Messages (RFC 822 [2]).

   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@domain". Additionally, the "container" in which mail
      is stored.

   receiver-SMTP process

      A process which transfers mail in cooperation with a sender-SMTP
      process. It waits for a connection to be established via the
      transport service. It receives SMTP commands from the
      sender-SMTP, sends replies, and performs the specified operations.

   reply

      A reply is an acknowledgment (positive or negative) sent from
      receiver to sender via the transmission channel in response to a
      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-SMTP process

      A process which transfers mail in cooperation with a receiver-SMTP
      process. A local language may be used in the user interface
      command/reply dialogue. The sender-SMTP initiates the transport
      service connection. It initiates SMTP commands, receives replies,
      and governs the transfer of mail.

   session

      The set of exchanges that occur while the transmission channel is
      open.

   transaction

      The set of exchanges required for one message to be transmitted
      for one or more recipients.

   transmission channel

      A full-duplex communication path between a sender-SMTP and a
      receiver-SMTP 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 sequence of printing characters.

   <CRLF>

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

   <SP>

      The space character.

REFERENCES

    [1] 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.

    [2] RFC 822

         Crocker, D., "Standard for the Format of ARPA Internet Text
         Messages," RFC 822, Department of Electrical Engineering,
         University of Delaware, August 1982.

    [3] TCP

         Postel, J., ed., "Transmission Control Protocol - DARPA Internet
         Program Protocol Specification", RFC 793, USC/Information Sciences
         Institute, NTIS AD Number A111091, September 1981. Also in:
         Feinler, E. and J. Postel, eds., "Internet Protocol Transition
         Workbook", SRI International, Menlo Park, California, March 1982.

    [4] 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.

    [5] 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.

    [6] 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.

   [7] 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.