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

Network Working Group
Request for Comments: DRAFT
Replaces: RFC 788, 780, 772

J. Postel
ISI

August 1982

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 |<-->|          |      SMTP      |          |
+------+    |  Sender- |Commands/Replies| Receiver-|
+------+    |   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

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.

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

3.3. 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.

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.

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

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

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

              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

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

3.8. 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.

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.

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).

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

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.

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 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>
```

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.
```

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.

4.2.2. 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
```

```
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
```

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.

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.

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 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 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 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.

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

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

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

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

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

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 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

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

         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

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

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.

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.

[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.