SIMPLE MAIL TRANSFER PROTOCOL

PRADEEP KOLLIPARA SANDEEP
PINNAMANENI

Introduction

Simple **M**ail **T**ransfer **P**rotocol is the standard e-mail protocol on the Internet and part of the TCP/IP protocol suite. SMTP defines the message format and the message transfer agent (MTA), which stores and forwards the mail. SMTP was originally designed for only plain text (ASCII text), but MIME and other encoding methods enable executable programs and multimedia files to be attached to and transported with the e-mail message.

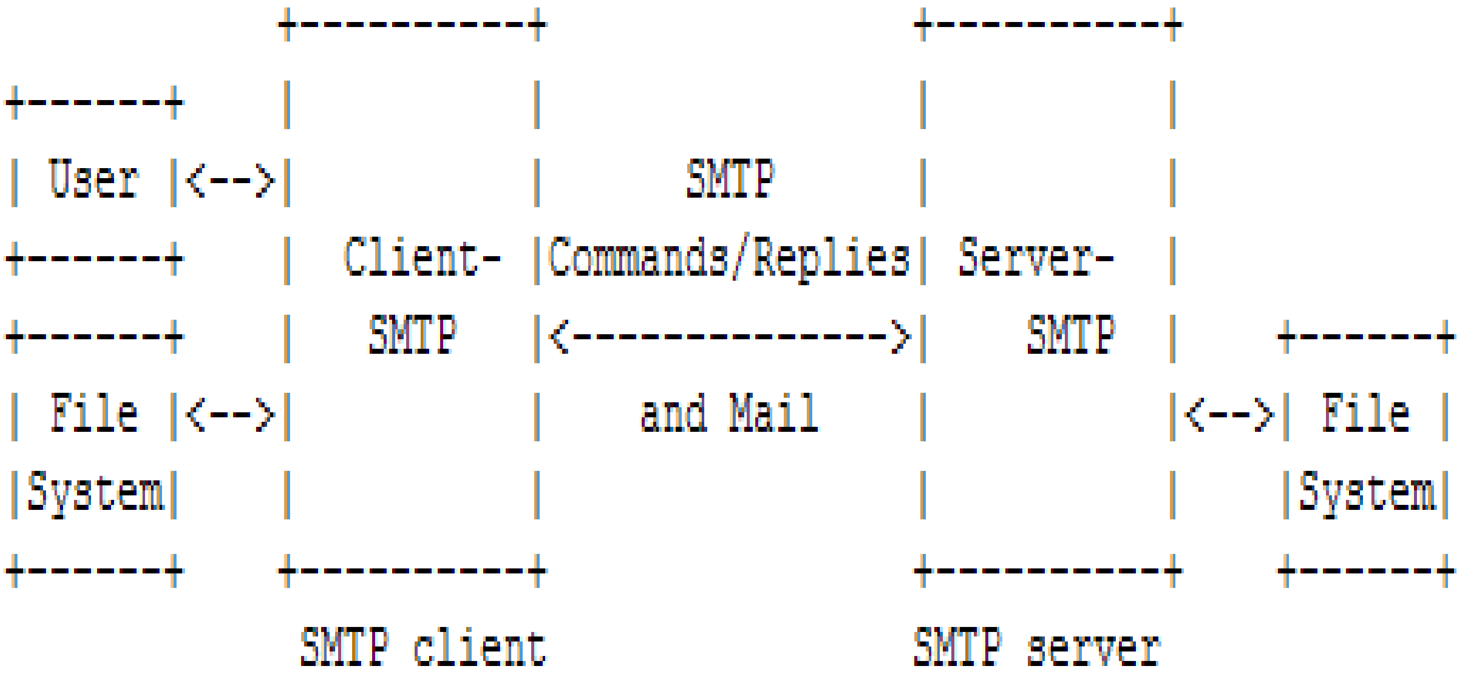
SMTP is a relatively simple, text-based protocol, where one or more recipients of a message are specified and then the message text is transferred. SMTP uses TCP port 25.

Purpose

The primary purpose of SMTP is to transfer email between mail servers. However, it is critical for email clients as well. In order to send email, the client sends the message to an outgoing mail server, which in turn contacts the destination mail server for delivery. For this reason, it is necessary to specify an SMTP server when configuring an email client.

One important point to make about the SMTP protocol is that it does not require authentication. This allows anyone on the Internet to send email to anyone else or even to large groups of people. It is this characteristic of SMTP that makes junk email or *spam* possible.

SMTP Model



```
            +----------+                +----------+
+------+    |          |                |          |
| User |<-->|          |      SMTP      |          |
+------+    |  Client- |Commands/Replies| Server-  |
+------+    |   SMTP   |<-------------->|    SMTP  |    +------+
| File |<-->|          |    and Mail    |          |<-->| File |
|System|    |          |                |          |    |System|
+------+    +----------+                +----------+    +------+
             SMTP client                SMTP server
```

Operation

When an SMTP client has a message to transmit, it establishes a two- way transmission channel to an SMTP server. The responsibility of an SMTP client is to transfer mail messages to one or more SMTP servers.

Once the transmission channel is established and initial handshaking completed, the SMTP client normally initiates a mail transaction. Such a transaction consists of a series of commands to specify the originator and destination of the mail and transmission of the message content (including any headers or other structure) itself.

Operation (contd..)

The server responds to each command with a reply; replies may indicate that the command was accepted, that additional commands are expected, or that a temporary or permanent error condition exists.

Once a given mail message has been transmitted, the client may either request that the connection be shut down or may initiate other mail transactions.

State Machine

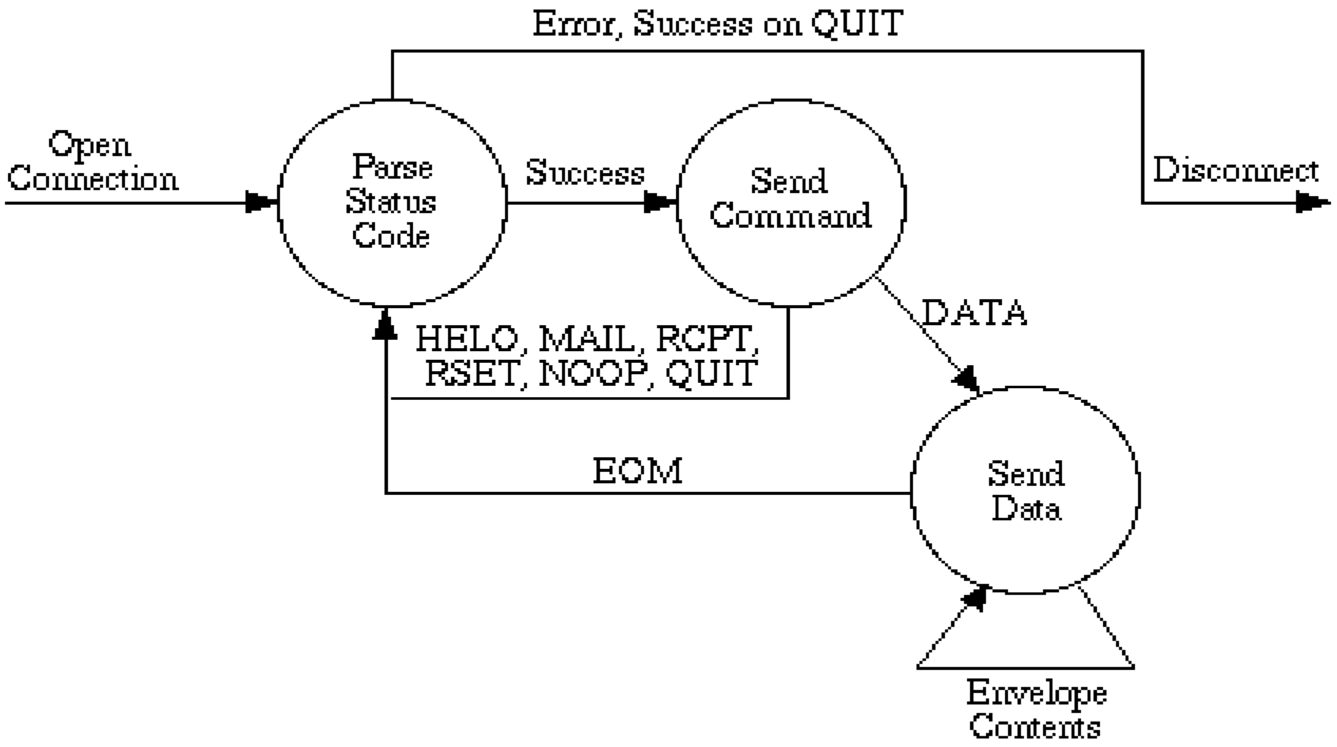


Figure 1. SMTP Finite State Machine

Basic Commands

SMTP defines a small required command set, with several optional commands included for convenience purposes. The minimal set required for an SMTP sending client are:

HELO - Initial State Identification

MAIL- Mail Sender Reverse Path

RCPT - One Recipient’s Forward Path

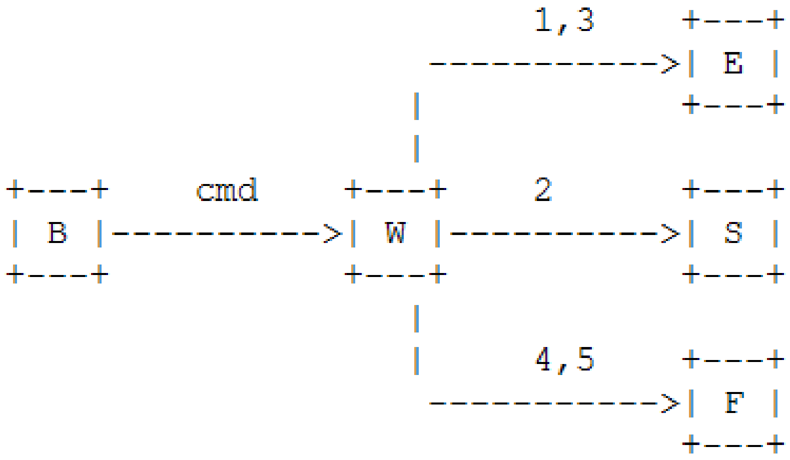
DATA - Mail Message Text State

RSET - Abort Transaction and Reset all buffers

NOOP - No Operation

QUIT- Commit Message and Close Channel

State Diagram for Commands



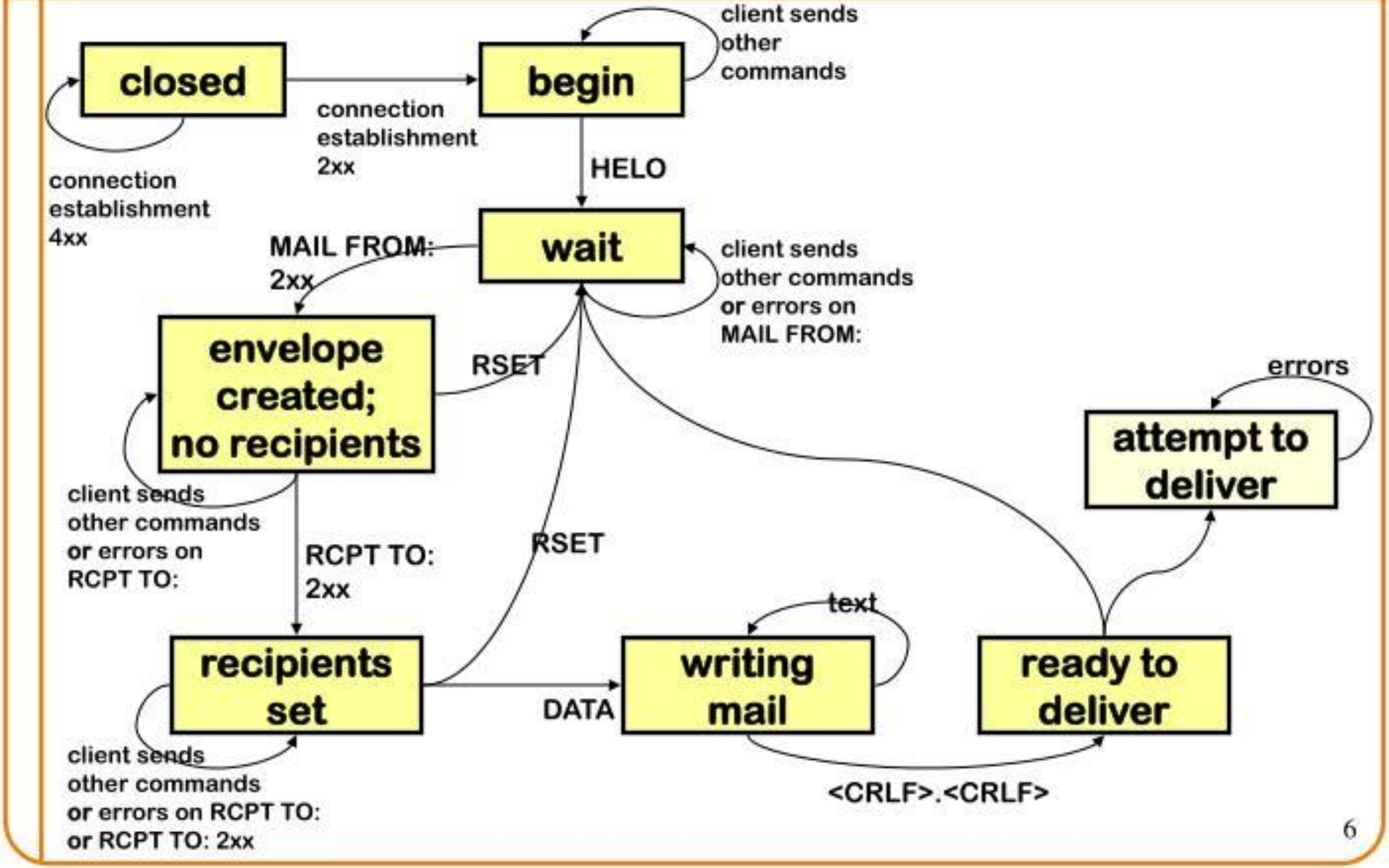
```
                                 1,3    +---+
                            ----------->| E |
                           |            +---+
                           |
+---+    cmd    +---+      2            +---+
| B |---------->| W |------------------>| S |
+---+           +---+                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

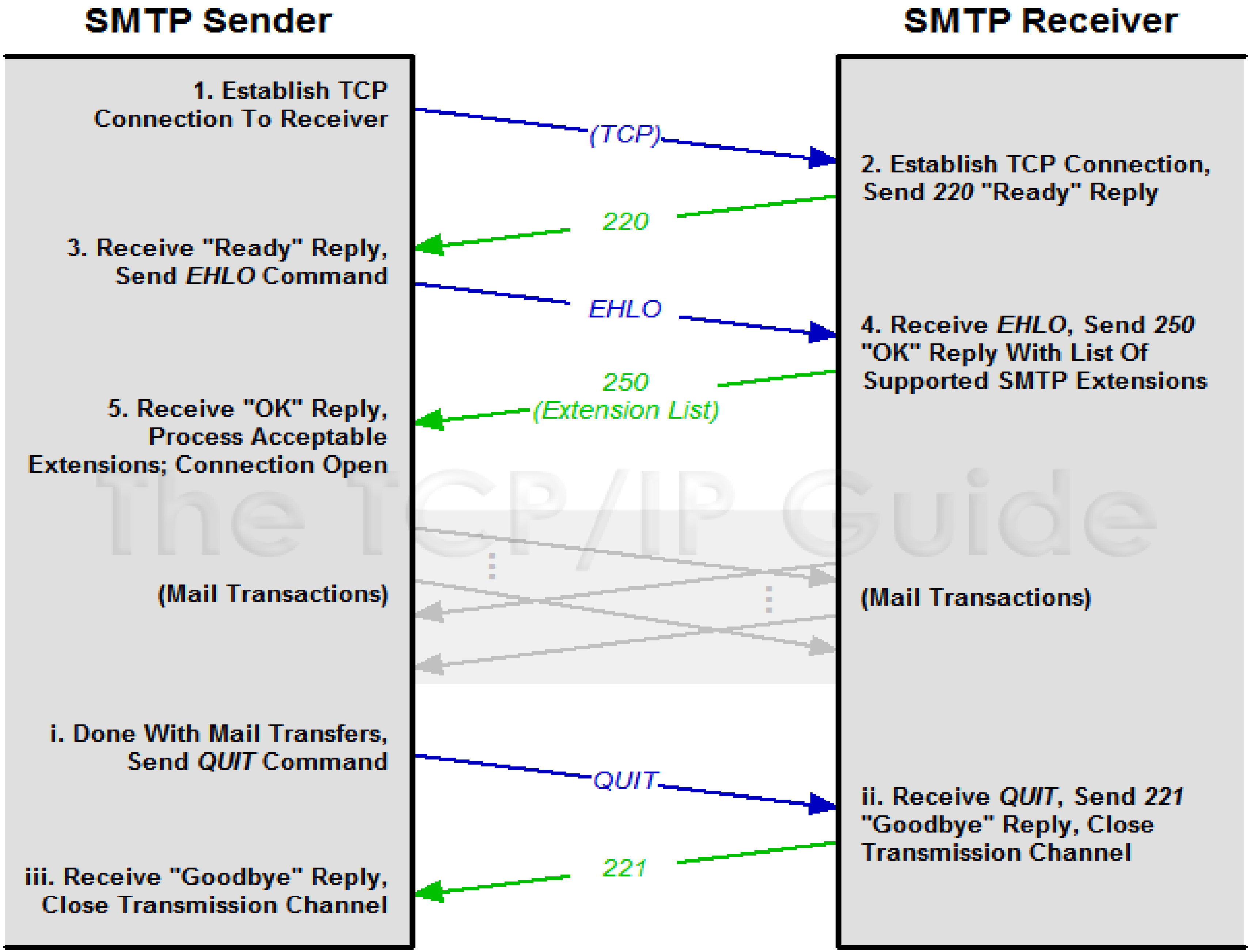
For each comma

"success"(S), "failure" (F), and "error" (E). In the state diagram above we use the symbol B for "begin", and the symbol W for "wait for reply".

SMTP State Diagram

Server States


closed
begin
connection establishment 2xx
client sends other commands
connection establishment 4xx
HELO
wait
MAIL FROM: 2xx
client sends other commands or errors on MAIL FROM:
envelope created; no recipients
RSET
errors
attempt to deliver
client sends other commands or errors on RCPT TO:
RCPT TO: 2xx
RSET
text
recipients set
DATA
writing mail
ready to deliver
client sends other commands or errors on RCPT TO: or RCPT TO: 2xx
<CRLF>.<CRLF>



SMTP Sender

SMTP Receiver

1. Establish TCP Connection To Receiver

(TCP)

2. Establish TCP Connection, Send *220* "Ready" Reply

220

3. Receive "Ready" Reply, Send *EHLO* Command

EHLO

4. Receive *EHLO*, Send *250* "OK" Reply With List Of Supported SMTP Extensions

250 (Extension List)

5. Receive "OK" Reply, Process Acceptable Extensions; Connection Open

(Mail Transactions)

(Mail Transactions)

i. Done With Mail Transfers, Send *QUIT* Command

QUIT

ii. Receive *QUIT*, Send *221* "Goodbye" Reply, Close Transmission Channel

221

iii. Receive "Goodbye" Reply, Close Transmission Channel

SMTP PROCEDURE

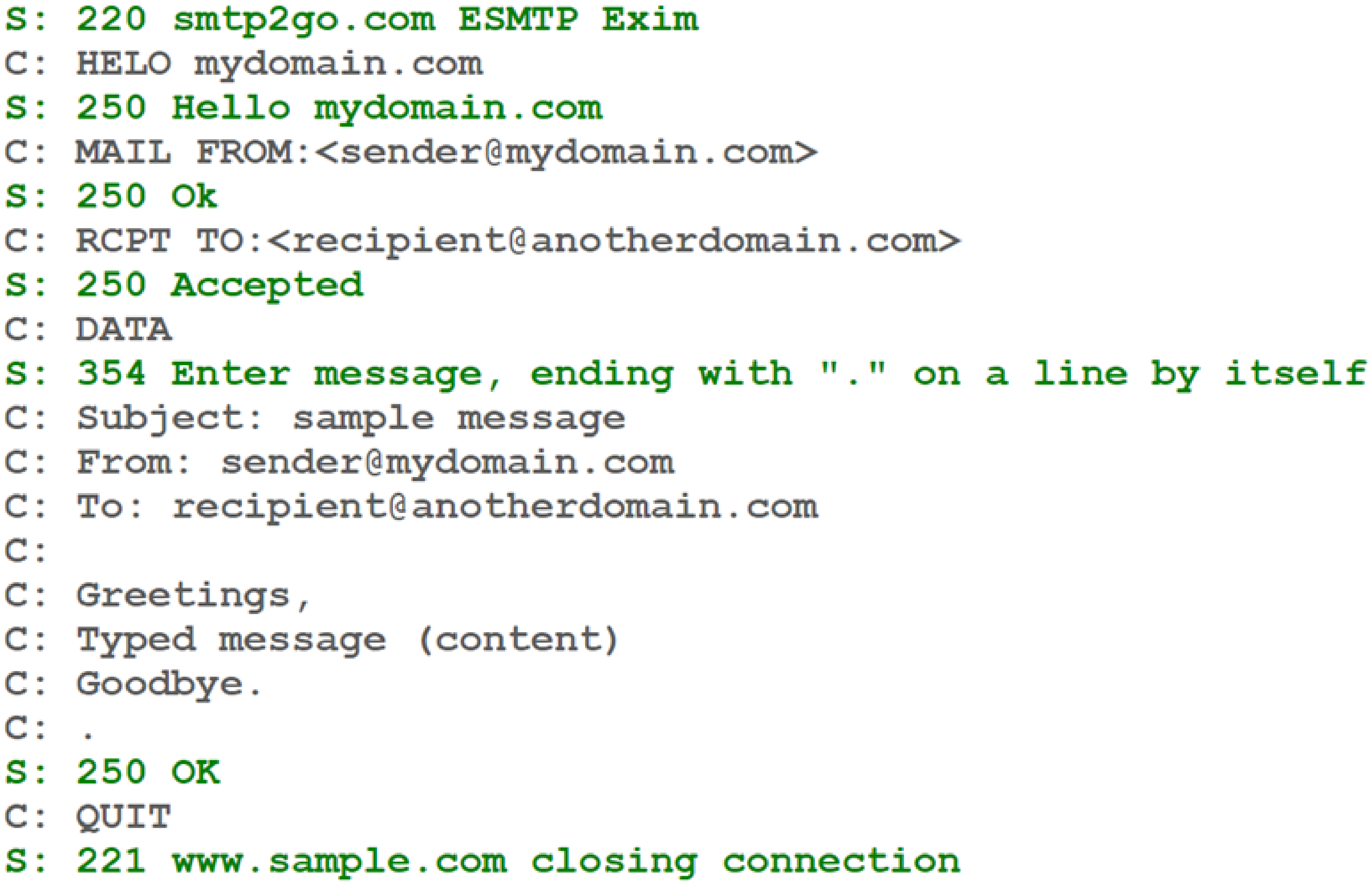
There are three steps in SMTP mail transactions.

- The transaction is started with a MAIL command which gives the sender identification. If accepted the receiver-SMTP returns a 250 OK reply.

- A series of one or more RCPT commands follows giving the receiver information. 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.

- Then a DATA command gives the mail data. If accepted, the receiver-SMTP returns a 354 Intermediate reply and considers all succeeding lines to be the message text. And finally, the end of mail data indicator confirms the transaction. When the end of text is received and stored the SMTP-receiver sends a 250 OK reply.

Example of SMTP Procedure



```
S: 220 smtp2go.com ESMTP Exim
C: HELO mydomain.com
S: 250 Hello mydomain.com
C: MAIL FROM:<sender@mydomain.com>
S: 250 Ok
C: RCPT TO:<recipient@anotherdomain.com>
S: 250 Accepted
C: DATA
S: 354 Enter message, ending with "." on a line by itself
C: Subject: sample message
C: From: sender@mydomain.com
C: To: recipient@anotherdomain.com
C:
C: Greetings,
C: Typed message (content)
C: Goodbye.
C: .
S: 250 OK
C: QUIT
S: 221 www.sample.com closing connection
```

SMTP Replies

Replies to SMTP commands serve to ensure the synchronization of requests and actions in the process of mail transfer and to guarantee that the SMTP client always knows the state of the SMTP server.

Every command MUST generate exactly one reply.

An SMTP reply consists of a three digit number followed by some text. The number is for use by automata to determine what state to enter next; the text is for the human user.

Formally, a reply is defined to be the sequence: a three-digit code, <SP>, one line of text, and <CRLF>, or a multi-line reply.

List Of Reply Codes

211 System status, or system help reply .

214 Help message.

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.

451 Requested action aborted: local error in processing

452 Requested action not taken: insufficient system storage.

List Of Reply Codes (cont…)

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.

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.

Problems with simple SMTP

The first one relates to message length. Some older implementations cannot handle messages exceeding 64KB.

Another problem relates to timeouts. If the Client and server have different timeouts, one of them may give up while the other is still busy, unexpectedly terminating the connection.

Security issue

ESMTP (RFC 2821)

To get around the problems with simple SMTP, extended SMTP has been defined in RFC 2821. Clients wanting to use it should send an EHLO message instead of HELO initially. If this is rejected, then the server is a regular SMTP server, and the client should proceed in the usual way. If the EHLO is accepted, then new commands and parameters are allowed.

Time-out in ESMTP

An SMTP client MUST provide a timeout mechanism. To implement this, a timer is set for each SMTP command and for each buffer of the data transfer.

The minimum per-command timeout values SHOULD be as follows:

Initial 220 Message: 5 minutes.

MAIL Command: 5 minutes.

RCPT Command: 5 minutes.

DATA Initiation: 2 minutes.

Data Block: 3 minutes.

DATA Termination: 10 minutes.

Reliable Delivery and Replies by E-mail

When the receiver-SMTP accepts a piece of mail (by sending a "250 OK" message in response to DATA), it is accepting responsibility for delivering or relaying the message.

If there is a delivery failure after acceptance of a message, the receiver-SMTP MUST formulate and mail a notification message. This notification MUST be sent using a null ("<>") reverse path in the envelope. The recipient of this notification MUST be the address from the envelope return path. However, if this address is null ("<>"), the receiver-SMTP MUST NOT send a notification.

SMTP Security and Spamming

One of the limitations of the original SMTP is that it has no facility for authentication of senders. Therefore the *SMTP-AUTH* extension was defined. In spite of this, E-mail spamming is still a major problem. Modifying SMTP extensively, or replacing it completely, is not believed to be practical, due to the network effects of the huge installed base of SMTP. *INTERNET MAIL 2000* is one such proposal for replacement.

SMTP mail is inherently insecure in that it is feasible for even fairly casual users to negotiate directly with receiving and relaying SMTP servers and create messages that will trick a naive recipient into believing that they came from somewhere else.

References

http://www.faqs.org/rfcs/rfc821.html

www.ncsl.org/programs/lis/cip/ppt/sjohnson03/sld009.html

www.faqs.org/rfcs/rfc2821.html

http://afrodita.rcub.bg.ac.yu/redhat/docs80/rhl-rg-en-8.0/ch-email.html

www.postech.ac.kr/cse/hpc/research/webcache/book/apps/ftp.htm

www.faqs.org/rfcs/rfc2920.html

www.faqs.org/rfcs/rfc3030.html

www.faqs.org/rfcs/rfc2487.html