SIMPLE MAIL TRANSFER PROTOCOL

PRADEEP KOLLIPARA

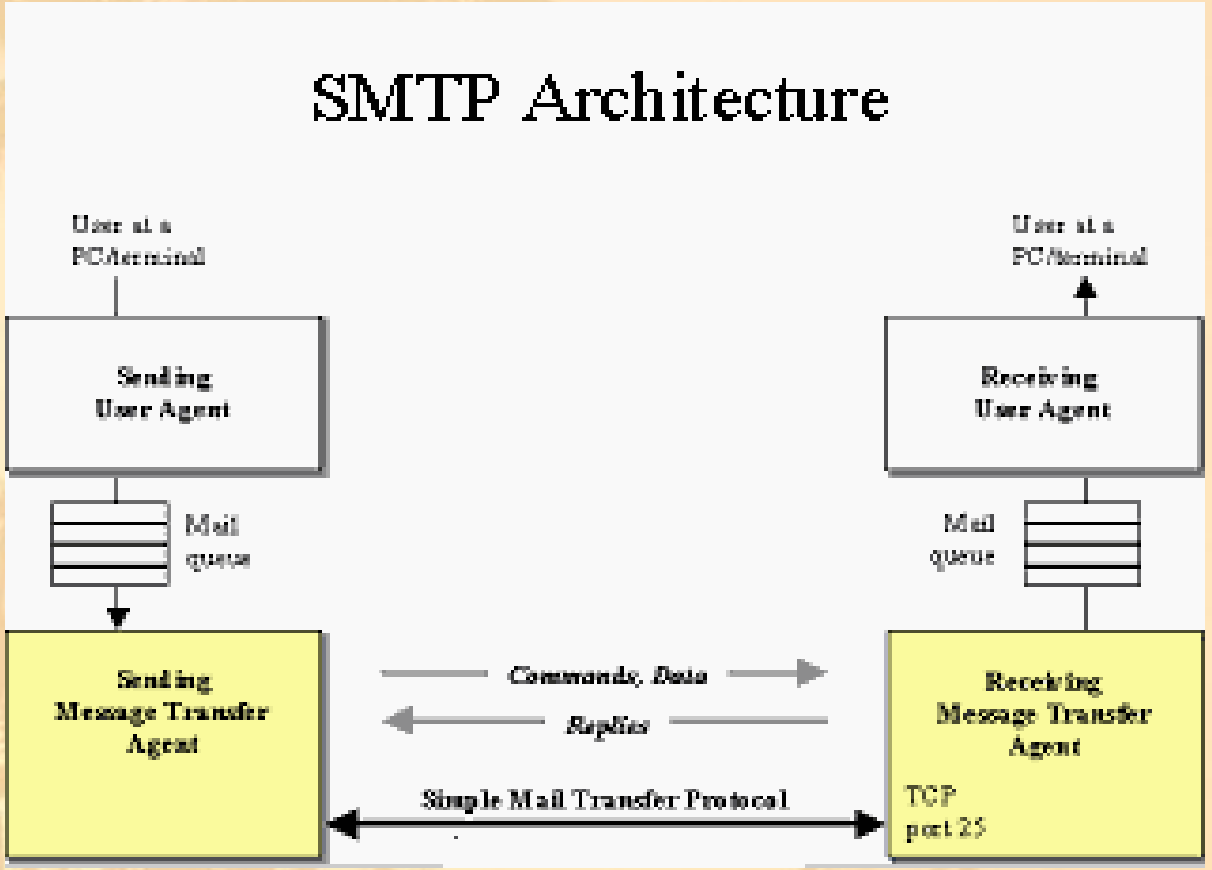
SANDEEP PINNAMANENI

Introduction

- **S**imple **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.

Basic Architecture

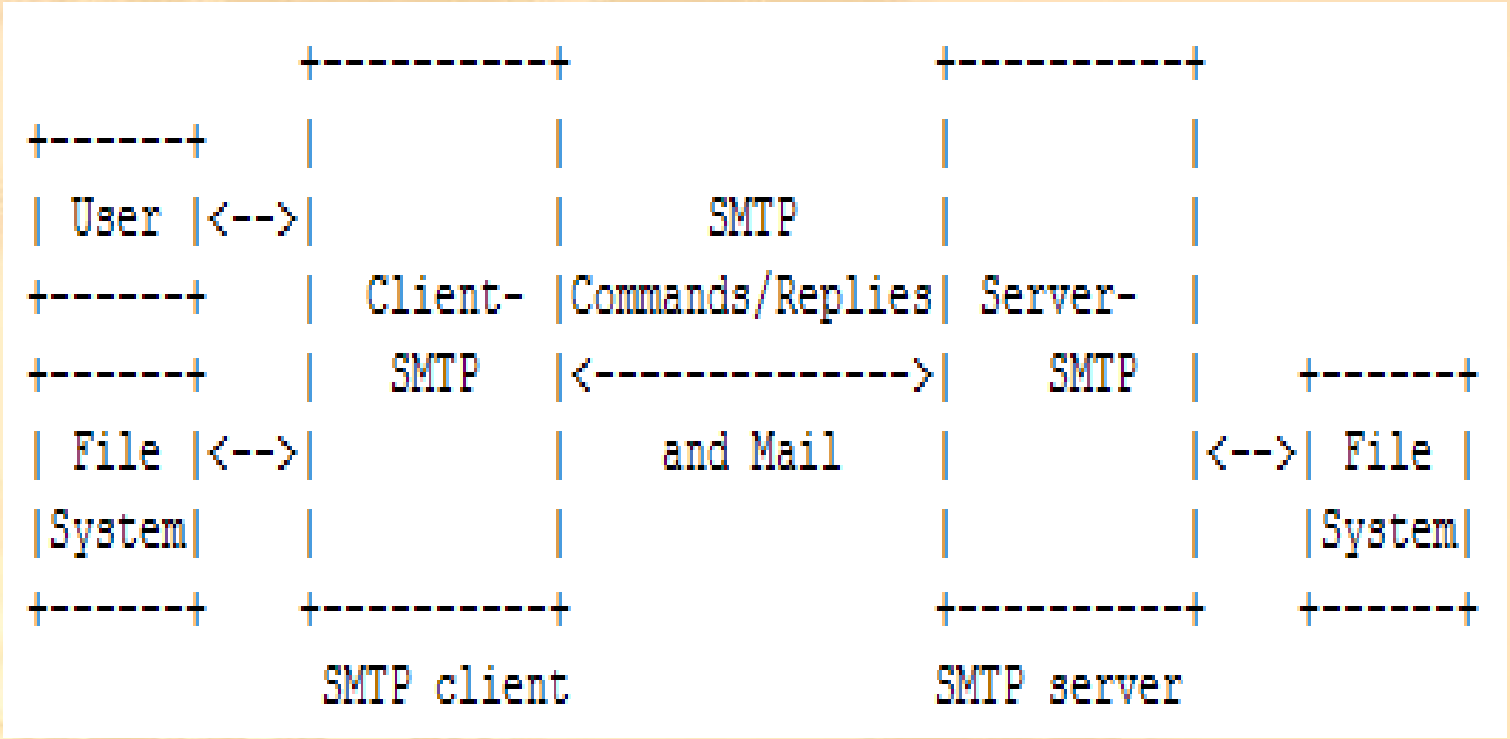


Purpose

- 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
File System
Client-SMTP
SMTP Commands/Replies and Mail
Server-SMTP
File 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

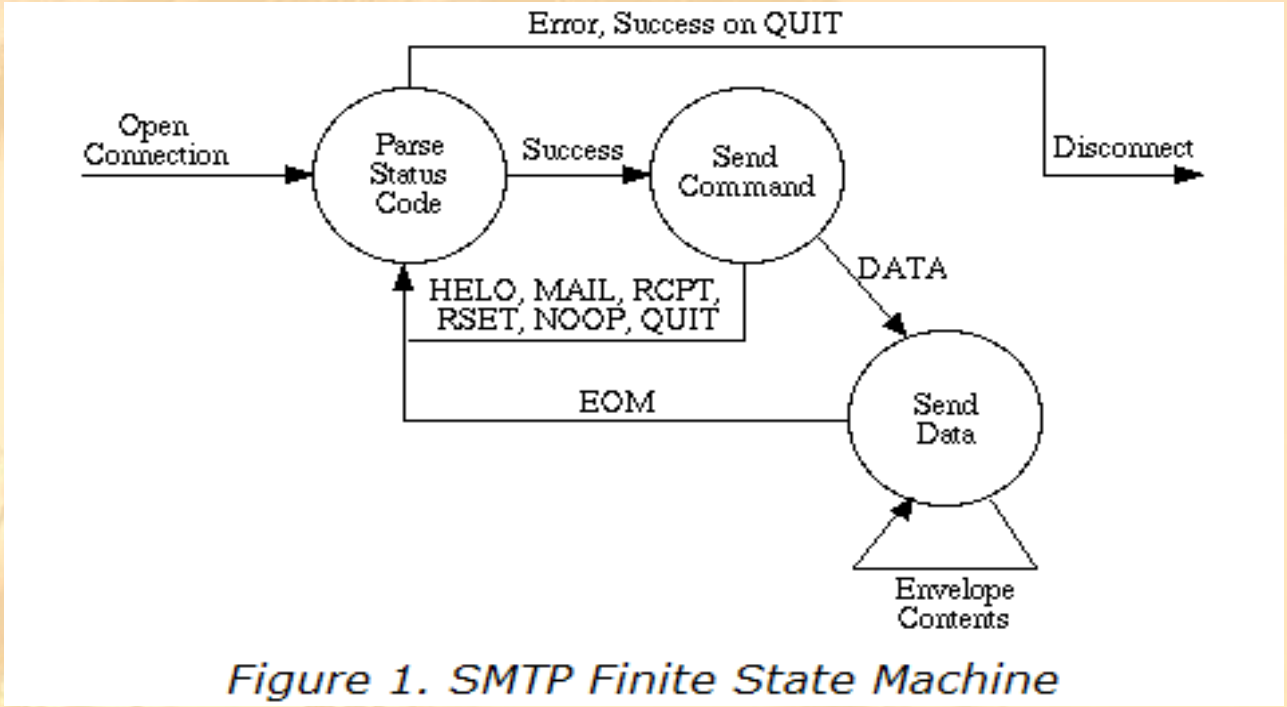


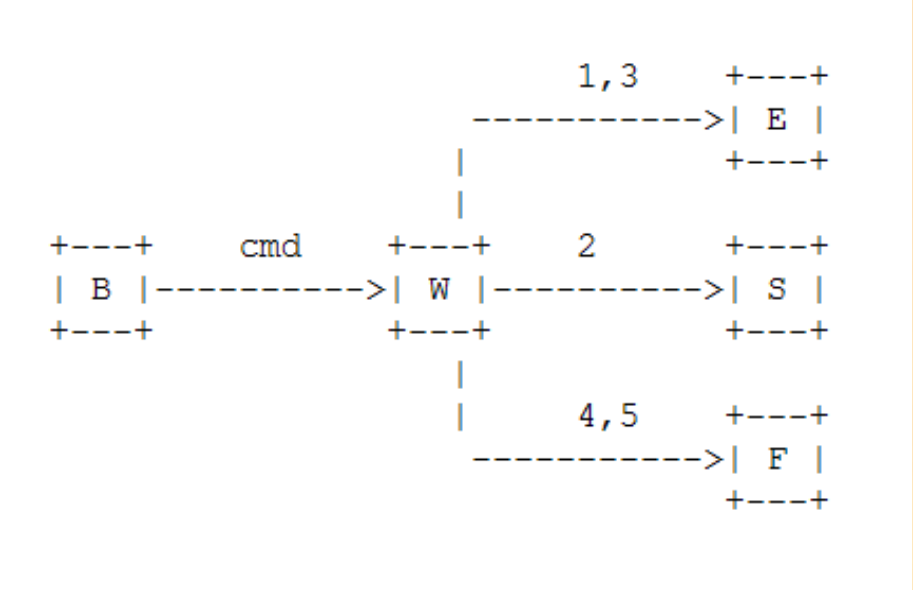
Figure 1. SMTP Finite State Machine

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:

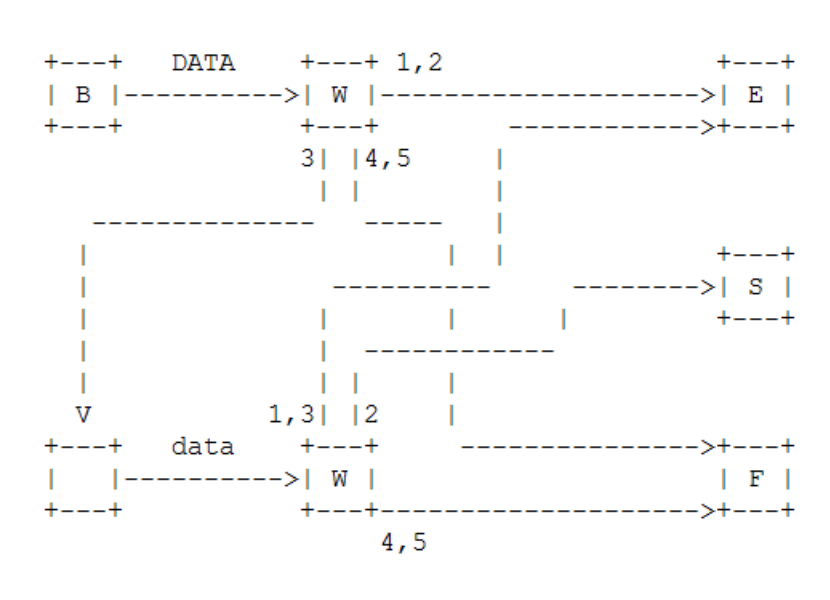
- HELO - Initial State Identification
- MAIL- Mail Sender Reverse Path
- RCPT - One Recipient’s Forward Path
- DATA - Mail Message Text State
- RSET - Abort Transaction and Reset all buffers
- NOOP - No Operation
- QUIT- Commit Message and Close Channel

State Diagram for Commands



For each command there are three possible outcomes: "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".

State Diagram for DATA Command



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.

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

This SMTP example shows mail sent by Smith at host Alpha.ARPA, to Jones, Green, and Brown at host Beta.ARPA.

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

Syntax of the basic commands

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>

NOOP <CRLF>

QUIT <CRLF>

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.

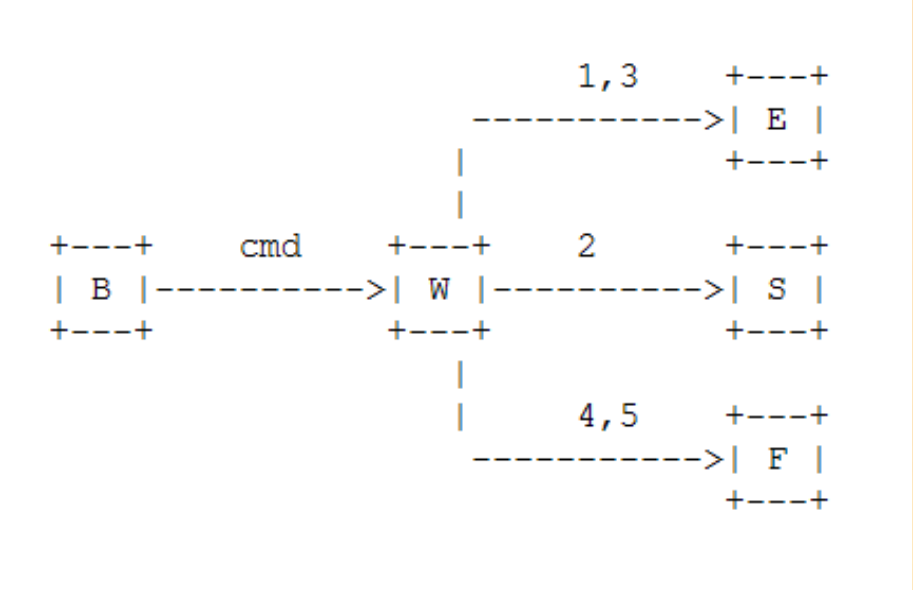
List Of Reply Codes

- 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.

State Diagram for Commands



For each command there are three possible outcomes: "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".

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.
- Infinite mail storms can be triggered. For example, If host 1 holds mailing list A and host 2 holds mailing list B and each list contains an entry for the other one, then a message sent to either list could generate a never ending amount of email traffic unless somebody checks for it.

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.

Extensions

The following are the extensions to SMTP protocol (RFC 821):-

- RFC 2920:-

  SMTP extension to improve SMTP performance by bundling multiple commands within a TCP send operation.

- RFC 3030:-

  This provides two extensions to the SMTP protocol for the transfer of large and binary MIME messages.

- RFC 2487:-

  SMTP extension for transport-layer security during sessions. This adds some security to email while in transit.

RFC 2920

- This is a pipelining service extension.

- When a client SMTP wishes to employ command pipelining, it first issues the EHLO command to the server SMTP. If the server SMTP responds with code 250 to the EHLO command, and the response includes the EHLO keyword value PIPELINING, then the server SMTP has indicated that it can accommodate SMTP command pipelining.

RFC 3030

This defines two extensions to the SMTP service. This provides the BDAT alternative for MIME extensions, as an alternative to the text only DATA command.

- The first extension enables a SMTP client and server to negotiate the use of an alternative to the DATA command, called "BDAT", for efficiently sending large MIME (Multipurpose Internet Mail Extensions) messages.
- The second extension takes advantage of the BDAT command to permit the negotiated sending of MIME messages that employ the binary transfer encoding.

RFC 2487

- This document describes an extension to the SMTP service that allows an SMTP server and client to use transport-layer security to provide private, authenticated communication over the Internet. This gives SMTP agents the ability to protect some or all of their communications from eavesdroppers and attackers.

- TLS [TLS], more commonly known as SSL, is a popular mechanism for enhancing TCP communications with privacy and authentication and is also being used for adding security to many other common protocols that run over TCP.

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.

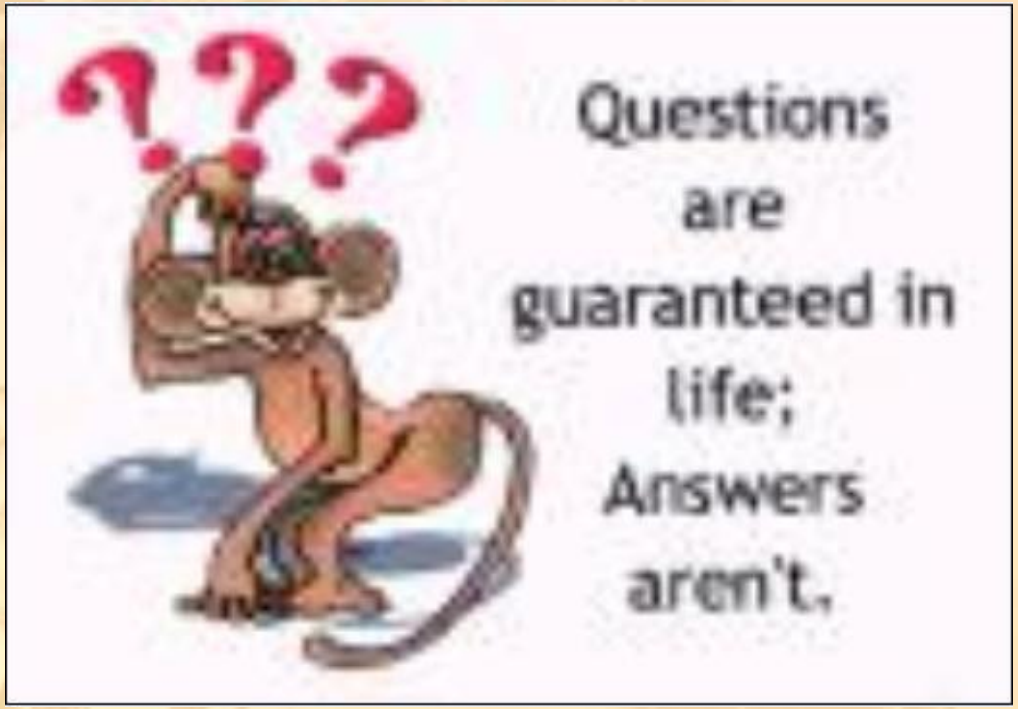
Questions?

- What is the main purpose of SMTP protocol?
- Explain the operation of SMTP with a simple model diagram?
- Explain briefly the basic command set?
- What are various problems with SMTP?
- What are various security considerations in SMTP?

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

Any Queries?



Thank You!