



## Simple Mail Transfer Protocol (SMTP)



## SMTP

- Protocol for mail transfer between mail servers
- RFC 821
- Runs on port 25 over TCP
- Simple text message transfer



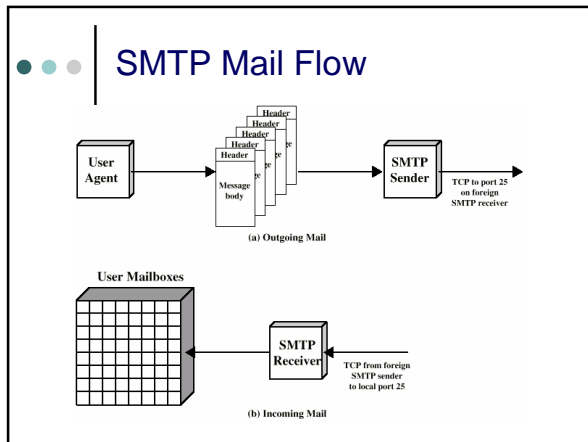
## Basic Steps

- Mail message created by user agent/mail client (for ex. outlook, pine etc.)
  - Mail header: recipient's address etc.
  - Mail Body: actual content
- Message sent to SMTP sender program (added to mail queue)
- SMTP sender looks at header, creates list of destinations from header
- Sender program sends message to all destinations one by one
- When delivery is complete for a destination, that destination is deleted from the list of destinations
- When all destinations are over, message is deleted



## SMTP Receiver

- Accept incoming messages
- Place in user mailbox, or to mail queue for forwarding
- Must be able to verify local destinations (user known?)
- Must be able to deal with errors
- Receiver acknowledges complete transfer to sender – only indicates that message has arrived at host, not necessarily delivered to destinations (users)



- ## SMTP Protocol
- Sender initiates TCP connection (Connection Setup)
  - Commands and responses between sender/receiver
  - Sender sends commands, one response to each command
  - After transfer, sender terminates connection (Connection Termination)

- ## Connection Setup
- Sender opens TCP connection with receiver
  - Once connected, receiver identifies itself
    - 220 service ready
  - Sender identifies itself
    - HELO <domain>
  - Receiver accepts sender's identification
    - 250 OK
  - If mail service not available, step 2 above becomes:
    - 421 service not available

- ## Message Transfer
- Sender may send one or more messages to receiver
  - Each message transfer has the following phases:
    - One MAIL command, identifies originator
      - Gives reverse path to use for error reporting
      - Receiver returns 250 OK or appropriate fail/error message
    - One or more RCPT commands, identifies recipients for the message
      - Each recipient identified by a separate RCPT
      - Separate reply for each recipient (250 OK etc.)
    - One DATA command transfers message text
      - End of message indicated by line containing just period (.)



## Connection Termination

- Sender sends QUIT and waits for reply
- Then initiate TCP close operation
- Receiver initiates TCP close after sending reply to QUIT



## Example SMTP Session

- C: <client connects to service port 25>
- C: HELO cse.iitkgp.edu *sending host identifies self*
- S: 250 OK Hello cse.iitkgp.edu *receiver acknowledges*
- C: MAIL FROM: <ag@iitkgp.edu> *identify sending user*
- S: 250 <ag@iitkgp.edu>... Sender ok *receiver acknowledges*
- C: RCPT TO: mk@microsoft.com *identify target user*
- S: 250 root... Recipient ok *receiver acknowledges*
- C: DATA
- S: 354 Enter mail, end with "." on a line by itself
- C: This is a test mail.
- C: How are you doing?
- C: . *end of multiline send*
- S: 250 WAA01865 Message accepted for delivery
- C: QUIT *sender signs off*
- S: 221 microsoft.com closing connection *receiver disconnects*
- C: <client hangs up>



## Sending to Multiple Users

- C: RCPT TO: mk@microsoft.com *identify target user*
- S: 250 root... Recipient ok *receiver acknowledges*
- C: DATA
- S: 354 Enter mail, end with "." on a line by itself
- C: How are you doing mk?
- C: . *end of send*
- S: 250 WAA01865 Message accepted for delivery
- RCPT TO: user1@microsoft.com *identify next target user*
- S: 250 root... Recipient ok *receiver acknowledges*
- C: DATA
- S: 354 Enter mail, end with "." on a line by itself
- C: How are you doing user1?
- C: . *end of send*
- S: 250 WAA01865 Message accepted for delivery
- C: QUIT *sender signs off*
- S: 221 microsoft.com closing connection *receiver disconnects*
- C: <client hangs up>



## RFC 822 Text Message Format

- Message viewed as having envelope (header) and contents (body)
- Envelope contains information required to transmit and deliver message
- Message is sequence of lines of text
  - Header usually keyword followed by colon followed by arguments



## Example

Date: Tue, 16 Jan 1996 10:37:17 (EST)  
From: "William Stallings" <ws@host.com>  
Subject: The syntax of RFC 822  
To: Smith@otherhost.com  
Cc: Jones@Yet-another\_host.com

This is the main text, delimited from the header by a blank line.



## MIME

- Extension to RFC822
- Main motivation
  - SMTP can not transmit executables, images, audio/video clips, International characters (e.g. å, ä, ã, è, é, ê, ë) that require 8-bit ASCII etc.
- MIME (Multipurpose Internet Mail Extensions)
  - Allows other types of non-text data to be carried by SMTP
  - Encodes image, video clip, voice data as text data to be transmitted over SMTP
  - RFC 2045-2049
- Defines new header fields, standardized content formats, and encodings to transfer them over mail



- Defines five new message header fields:
  - MIME-Version: version no.
  - Content-Type: type of data in body
  - Content-transfer-encoding: type of encoding used
  - Content-ID: uniquely identify MIME entity
  - Content Description: plain text description of body



- Text body
- Multipart
  - Mixed, Parallel, Alternative, Digest
- Message
  - RFC 822, Partial, External-body
- Image
  - jpeg, gif
- Video
  - mpeg
- Audio
  - Basic
- Application
  - Postscript, octet stream



## Content Transfer Encodings

- Content transfer encoding field
  - Six values
  - Three (7bit, 8bit, binary) no encoding done
    - Provide info about nature of data. 7 bit fine for simple text over SMTP, others may be fine for other mail transport systems
  - Base64
    - Maps arbitrary binary input onto printable output
  - Few other encodings not of interest to us



## POP3 (Post Office Protocol Version 3)

- Used to access mailboxes on remote servers
- Server listens on TCP port 110
- Client connects to server
- Command-response exchanged to download mail
- Client closes connection
- RFC 1939



## Basic POP3 Commands/Replies

- Commands
  - USER <name> - identifies the user
  - PASS <password> - authentication for user
  - STAT - lists all messages in the mailbox of user
  - LIST <msg no.> - lists the content of a message
  - RETR <msg no.> - retrieves a particular message
  - DELE <msg no.> - Deletes a particular message
  - NOOP
  - RSET
  - QUIT
- Replies
  - +OK
  - -ERR



## Example POP3 Session

- <client connects to service port 110>
- S: +OK POP3 server ready <1896.6971@cse.iitkgp.edu>
- C: USER agupta
- S: +OK agupta
- C: PASS mypassword
- S: +OK agupta's mailbox has 2 messages (320 octets)
- C: STAT
- S: +OK 2 320
- C: LIST
- S: +OK 2 messages (320 octets)
- S: 1 120
- S: 2 200
- S: .



- C: RETR 1
- S: +OK 120 octets
- S: <the POP3 server sends the text of message 1>
- S: .
- C: DELE 1
- S: +OK message 1 deleted
- C: RETR 2
- S: +OK 200 octets
- S: <the POP3 server sends the text of message 2>
- S: .
- C: DELE 2
- S: +OK message 2 deleted
- C: QUIT
- S: +OK cse POP3 server signing off (maildrop empty)
- C: <client hangs up>