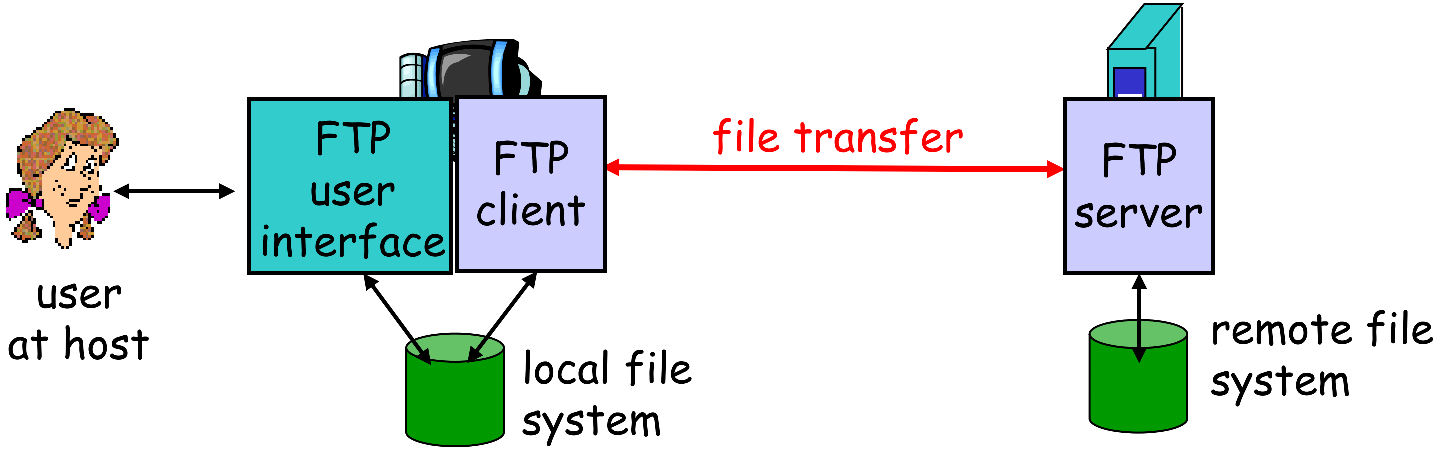
Chapter 2: Application layer

- 2.1 Principles of network applications
- 2.2 Web and HTTP
- 2.3 FTP
- 2.4 Electronic Mail
  - SMTP, POP3, IMAP
- 2.5 DNS
- 2.6 P2P applications
- 2.7 Socket programming with TCP
- 2.8 Socket programming with UDP
- 2.9 Building a Web server

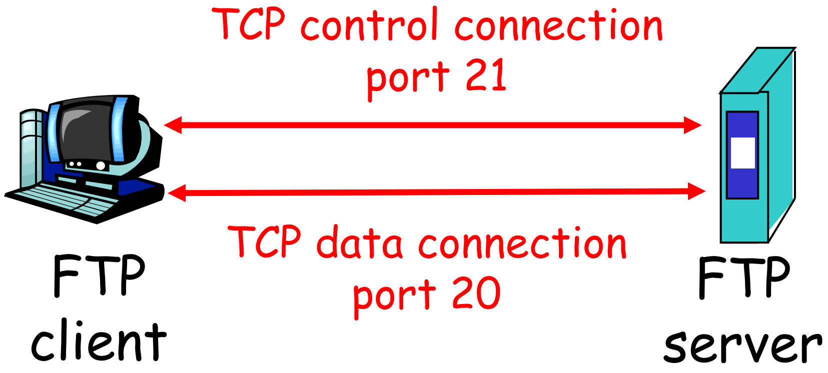
FTP: the file transfer protocol



- transfer file to/from remote host
- client/server model
  - *client:* side that initiates transfer (either to/from remote)
  - *server:* remote host
- ftp: RFC 959
- ftp server: port 21

FTP: separate control, data connections

- FTP client contacts FTP server at port 21, TCP is transport protocol
- client authorized over control connection
- client browses remote directory by sending commands over control connection.
- when server receives file transfer command, server opens 2nd TCP connection (for file) to client
- after transferring one file, server closes data connection.



- server opens another TCP data connection to transfer another file.
- control connection: "out of band"
- FTP server maintains "state": current directory, earlier authentication

FTP commands, responses

Sample commands:

- sent as ASCII text over control channel
- **`USER`** ***`username`***
- **`PASS`** ***`password`***
- **`LIST`** return list of file in current directory
- **`RETR filename`** retrieves (gets) file
- **`STOR filename`** stores (puts) file onto remote host

Sample return codes

- status code and phrase (as in HTTP)
- **`331 Username OK, password required`**
- **`125 data connection already open; transfer starting`**
- **`425 Can't open data connection`**
- **`452 Error writing file`**

Chapter 2: Application layer

- 2.1 Principles of network applications
- 2.2 Web and HTTP
- 2.3 FTP
- 2.4 Electronic Mail
  - SMTP, POP3, IMAP
- 2.5 DNS
- 2.6 P2P applications
- 2.7 Socket programming with TCP
- 2.8 Socket programming with UDP

Electronic Mail

- Most widely used application on the internet
- For sending mails (conjunction):
  - Simple Mail Transfer Protocol (SMTP)
  - Multi-purpose Internet Mail Extension (MIME)
- For receiving mails:
  - Post office protocol version 3 (POP 3) or
  - Internet mail access protocol (IMAP)

Electronic Mail: SMTP [RFC 821]

- All details about SMTP in RFC 821
- Transmits simple text messages only - 7 bit ASCII file
- uses TCP to reliably transfer email message from client to server, port 25
- direct transfer: sending server to receiving server
- three phases of transfer
  - handshaking (greeting)
  - transfer of messages
  - closure
- command/response interaction
  - commands: ASCII text
  - response: status code and phrase

SMTP

- Uses information written on envelope of mail to transfer mail
  - Message header
  - Contains recipient address and other information
- Does not look at contents or message body as long as it is in simple text
  - Only look at the message header

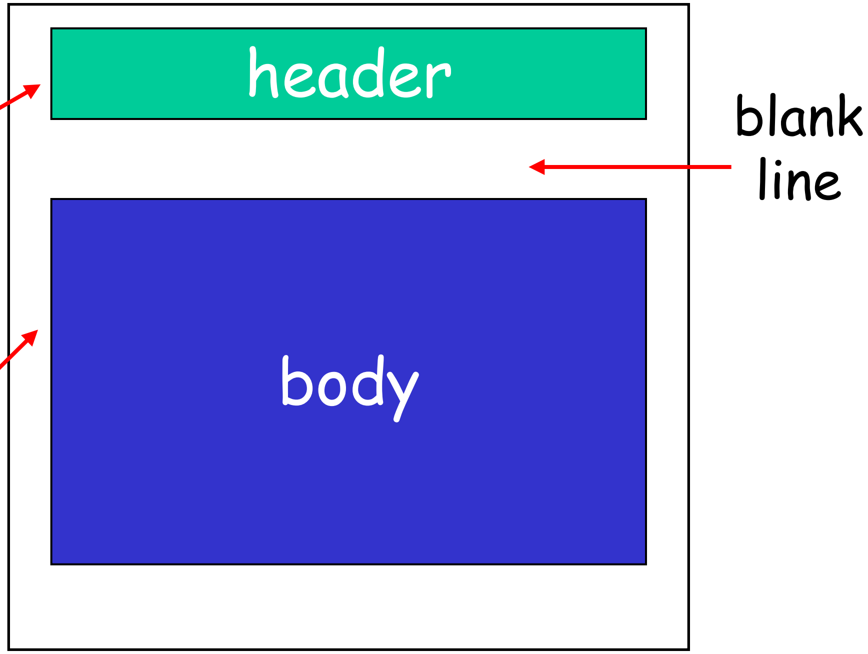
Mail message format

SMTP: protocol for exchanging email msgs

RFC 822: standard for text message format:

- header lines, e.g.,
  - To:
  - From:
  - Subject:

  different from SMTP commands!
- body
  - the "message", ASCII characters only



Basic Operation

- Mail is created by user agent program (mail client)
- Messages are queued and sent as input to SMTP sender program
  - Typically a server process
  - Daemon on UNIX eg. sendmail or qmail

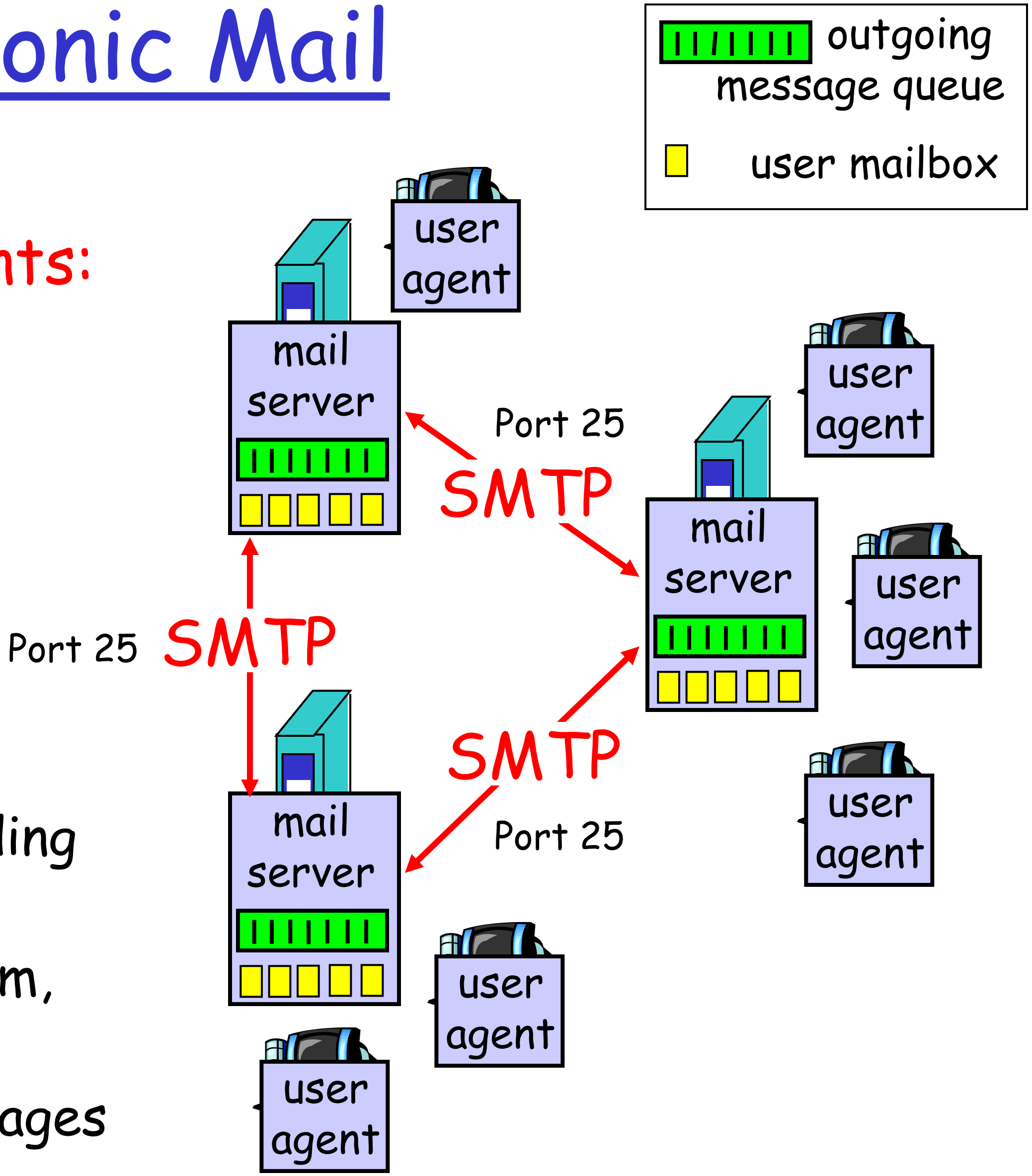
Sending Electronic Mail

Three major components:

- user agents
- mail servers
- simple mail transfer protocol: SMTP

User Agent

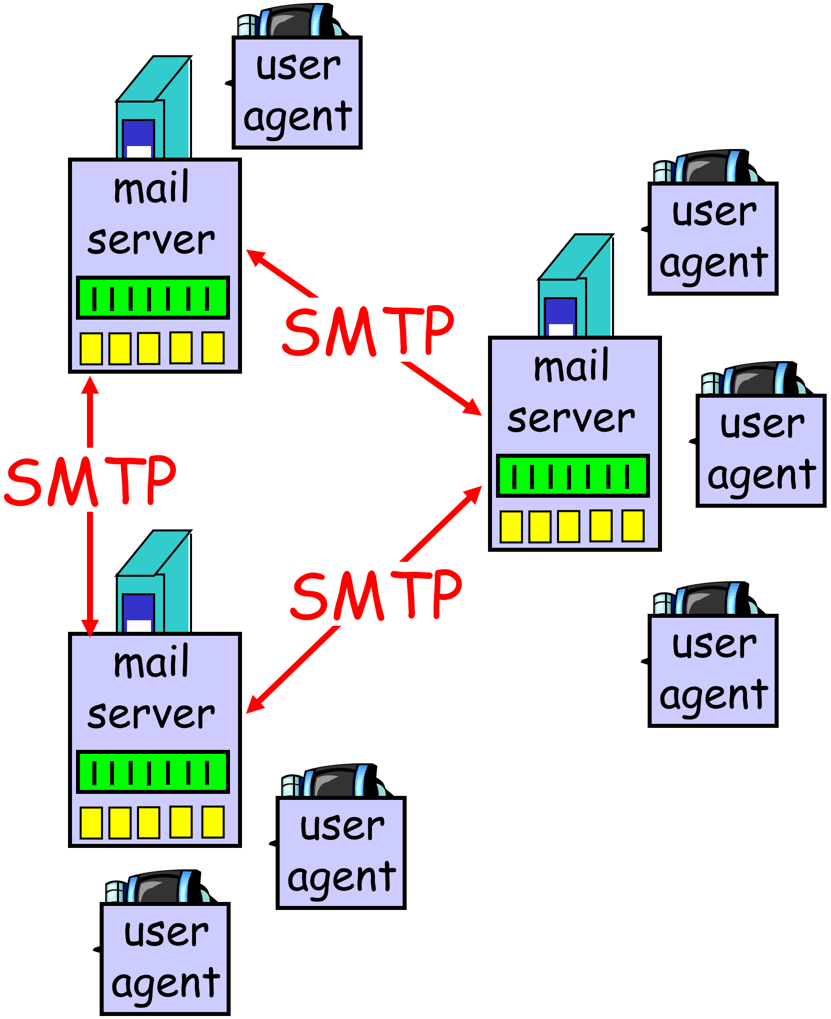
- a.k.a. “mail reader”
- composing, editing, reading mail messages
- e.g., Eudora, Outlook, elm, Mozilla Thunderbird
- outgoing, incoming messages stored on server



Electronic Mail: mail servers

Mail Servers

- mailbox contains incoming messages for user
- message queue of outgoing (to be sent) mail messages
- SMTP protocol between mail servers to send email messages
  - client: sending mail server
  - "server": receiving mail server



Scenario: Alice sends message to Bob

1) Alice uses UA to compose message and "to" `bob@someschool.edu`

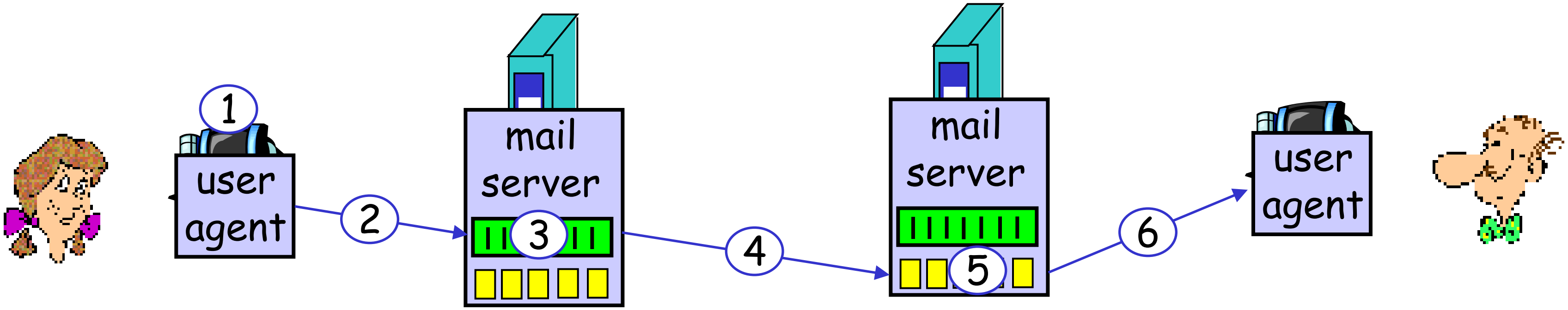
2) Alice's UA sends message to her mail server; message placed in message queue

3) Client side of SMTP opens TCP connection with Bob's mail server

4) SMTP client sends Alice's message over the TCP connection

5) Bob's mail server places the message in Bob's mailbox

6) Bob invokes his user agent to read message



Mail Message Contents

- Each queued message has:
  - Message text
    - RFC 822 header with message envelope and list of recipients
    - Message body, composed by user
  - A list of mail destinations
    - Extracted by user agent/ SMTP server from header for faster access
    - May require expansion of mailing lists (alias)

SMTP Sender

- Takes message from queue
- Transmits to proper destination host
  - Via SMTP transaction
  - Over one or more TCP connections to port 25
- When all destinations are processed (an electronic mail can have more than one destinations), message is deleted

Optimization by Sender

- If same message is sent to multiple users on a given host, it is sent only once
  - Delivery to users handled at destination host

- If multiple messages are ready for given host, a single TCP connection can be used
  - Saves overhead of setting up and dropping connection

Possible Errors

- Host unreachable
- Host out of operation
- TCP connection fail during transfer
- Faulty destination address
  - User error
  - Target user address has changed
  - Redirect if possible
  - Inform user if not
- Sender can re-queue mail
  - Give up after a period

SMTP Protocol - Reliability

- Used to transfer messages from sender to receiver over TCP connection
    - Uses port number 25
- Attempts to provide reliable service
- No guarantee to recover loss messages
- TCP ensures messages arrives at the nearest SMTP server
- No end-to-end ACK to sender
- Error indication report not guaranteed

SMTP Receiver

- Accepts arriving message
- Places in user mailbox or copies to outgoing queue for forwarding
- Receiver must
  - Verify local mail destinations
  - Deal with errors
    - Transmission
    - Lack of disk space

SMTP Forwarding

- Mostly direct transfer from sender host to receiver host
- May go through intermediate mail servers via forwarding capability
  - Sender can specify route

SMTP System Overview

- Commands and responses exchanged between sender and receiver
- Initiative with sender
  - Establishes TCP connection over port 25
- Sender sends commands to receiver
  - E.g. HELO <domain><CRLF>
- Each command generates exactly one reply
  - E.g. 250 requested mail action ok; completed

SMTP Replies

- Starts with 3-digit code
- Leading digit indicates category
  - 2xx: Positive completion reply
  - 3xx: Positive intermediate reply
  - 4xx: Transient negative completion reply
  - 5xx: Permanent negative completion reply

Operation Phases

- Connection setup
- Exchange of command-response pairs
- Connection terminated

Connection Setup - 1

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

Connection Setup - 2

- The MAIL FROM command identifies originator
  - Gives reverse path to be used for error reporting
  - Receiver returns 250 OK or appropriate failure/ error message
- One or more RCPT TO commands identify recipients for the message
- DATA command transfers message text (indicated by . on a single line)

Closing Connection

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

SMTP: final words

- SMTP uses persistent connections
- SMTP requires message (header & body) to be in 7-bit ASCII
- SMTP server uses `CRLF.CRLF` to determine end of message

Comparison with HTTP:

- HTTP: pull
- SMTP: push
- both have ASCII command/response interaction, status codes
- HTTP: each object encapsulated in its own response msg
- SMTP: multiple objects sent in multipart msg