CS 330: Network Applications & Protocols

Application Layer: FTP, SMTP

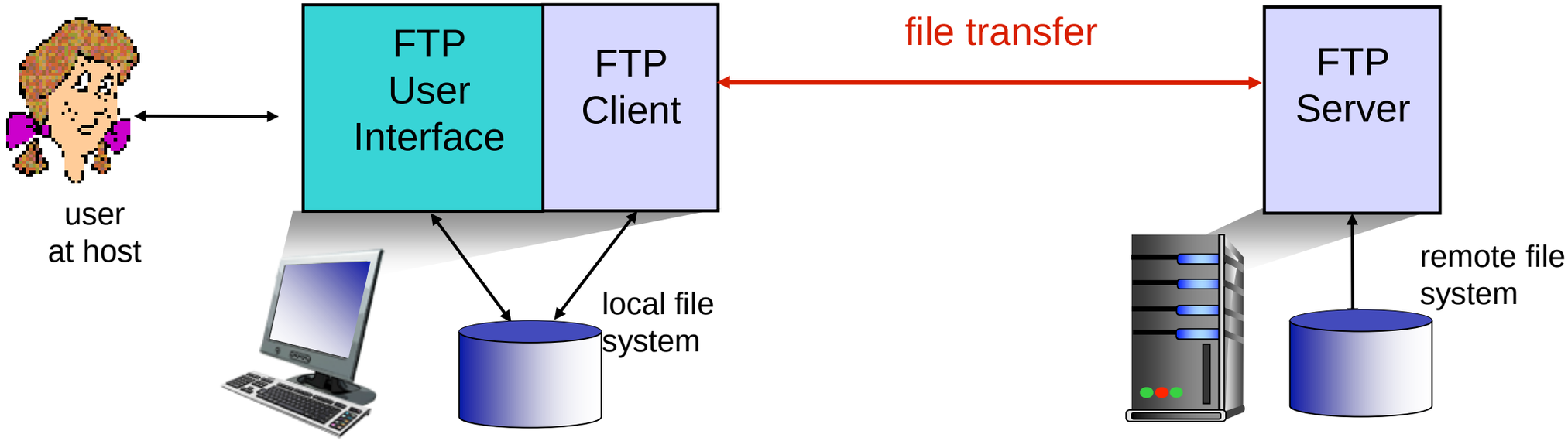
Galin Zhelezov
Department of Physical Sciences
York College of Pennsylvania

Based on Computer Networking: A Top-Down Approach, 7th Edition by Kurose and Ross

Overview of Application Layer

- **Network Application Architectures**
- **HyperText Transfer Protocol (HTTP)**
- **File Transfer and Email protocols (FTP, SMTP)**
  - FTP
  - SMTP, POP3, IMAP
- **Domain Name System (DNS)**
- **Peer-to-Peer Applications (P2P)**

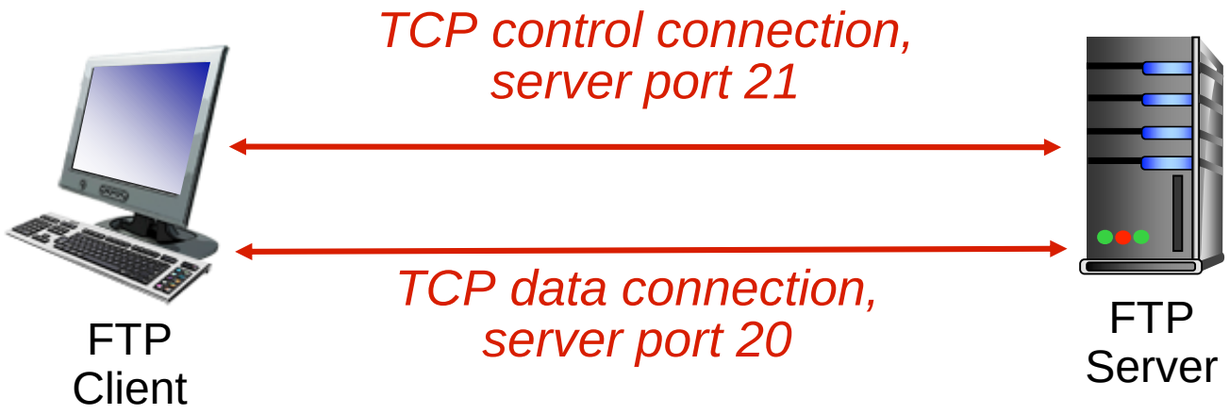
FTP: File Transfer Protocol



- **Used to transfer files to/from a 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 on port 21, using TCP**
- **Uses two parallel TCP connections: Control and Data**
- **Client authorized over control connection**
- **Client browses remote directory, sends commands over control connection**
- **Control connection is persistent**
- **FTP server maintains "state": (i.e. current directory, authentication information)**



- **When server receives file transfer command, server opens 2nd TCP data connection (for file) to client**
- **After transferring one file, server closes data connection**
  - A separate TCP data connection is opened for each transferred file

FTP Commands / Responses

- **Commands are sent as plain ASCII text over the control channel**

  `USER username` : sends username to server

  `PASS password` : sends password to server **in plain text!!**

  `LIST` : returns a list of files in current directory

  `RETR filename` : retrieves (gets) file

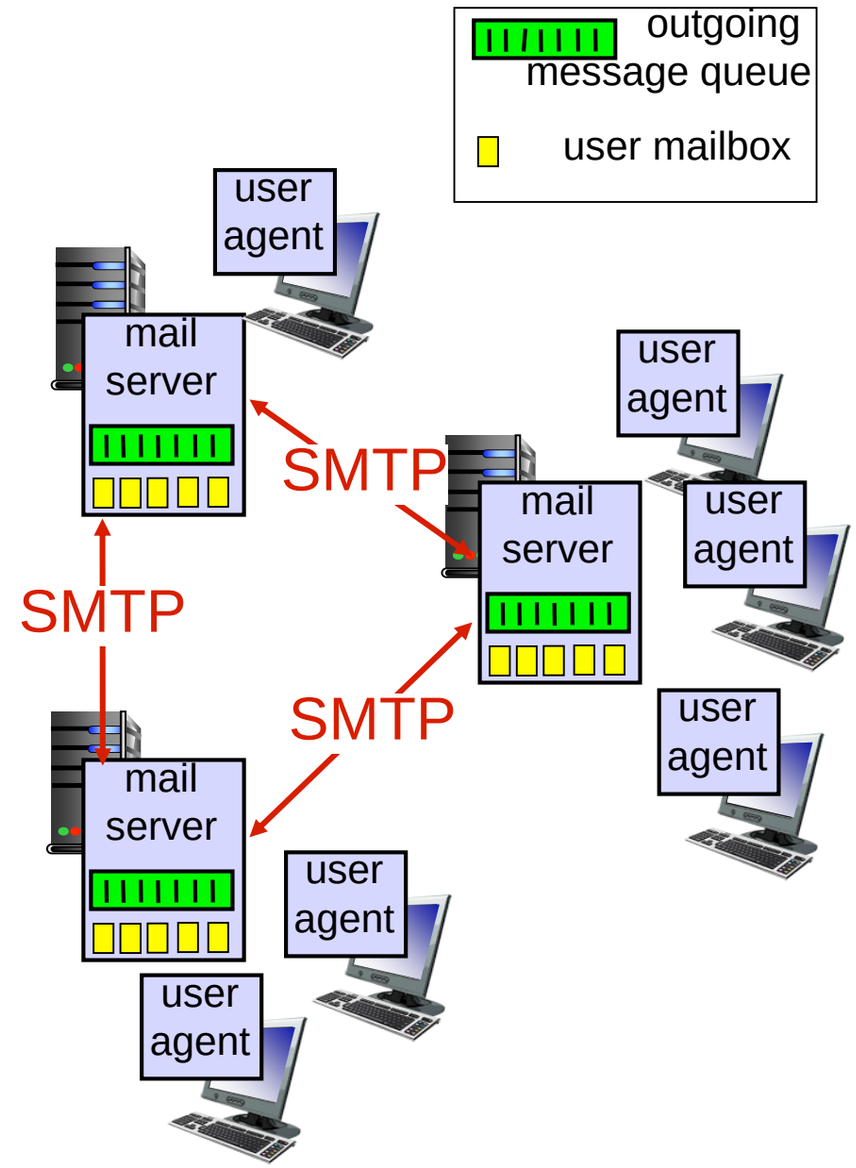
  `STOR filename` : stores (puts) file onto remote host

- **FTP server responds with status codes on the control channel**

```
331 Username OK, password required
125 data connection already open; transfer starting
425 Can't open data connection
452 Error writing file
```

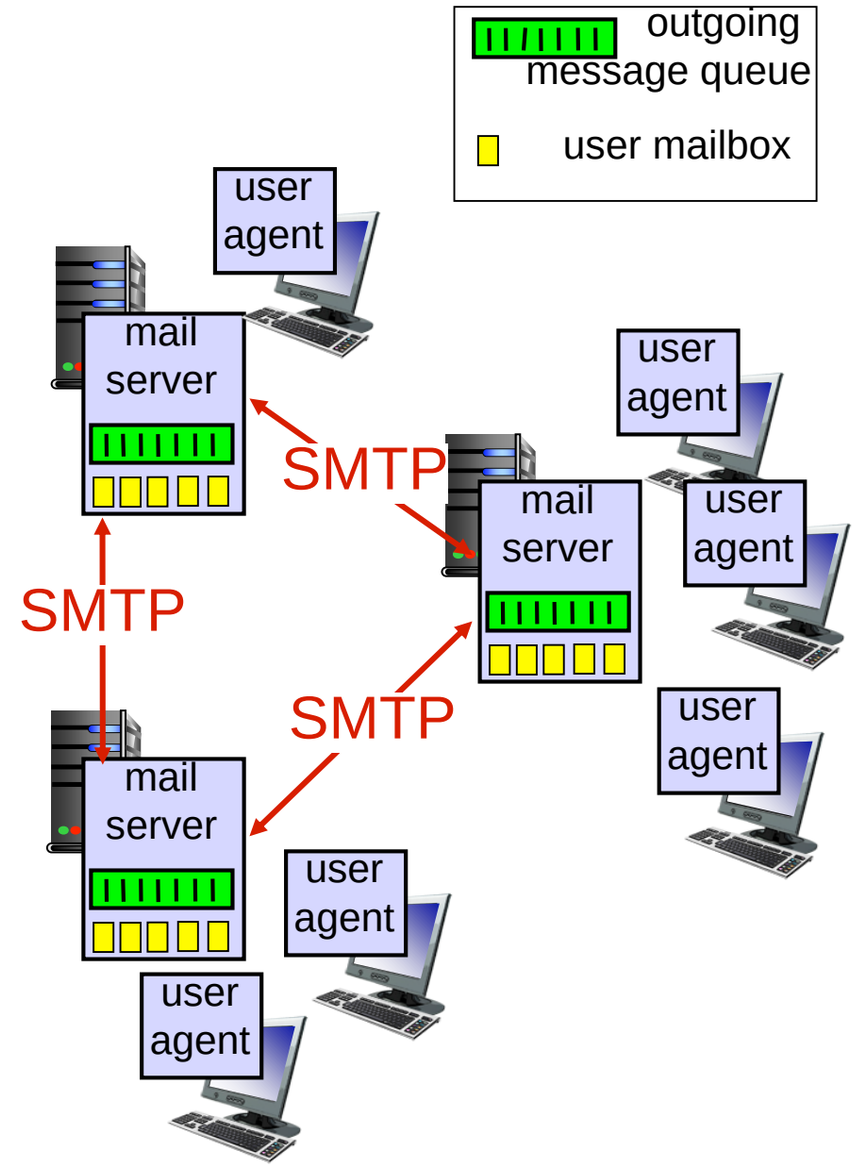
Electronic Mail

- **Three major components:**
  - User agents
  - Mail servers
  - Simple mail transfer protocol: SMTP
- **User Agent**
  - Composing, editing, reading mail messages
  - e.g. Outlook, Thunderbird, iPhone mail client
  - 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
  - Uses SMTP protocol between mail servers to send email messages
    - Client: sending mail server
    - "Server": receiving mail server



Electronic Mail: SMTP

- **Uses TCP to reliably transfer email message from client to server on port 25**
- **Direct transfer: sending server to receiving server**
- **Three phases of transfer**
  - Handshaking (greeting)
  - Transfer of messages
  - Closure
- **Command/response interaction (like HTTP, FTP)**
  - Commands are in ASCII text
  - Server responds with status code and phrase
- **Messages must be in 7-bit ASCII**
  - All binary objects (i.e. attachments MUST be converted to ASCII to send)

Scenario: Alice Sends Message to Bob

(1) Alice uses her mail client to compose message "to" bob@someschool.edu

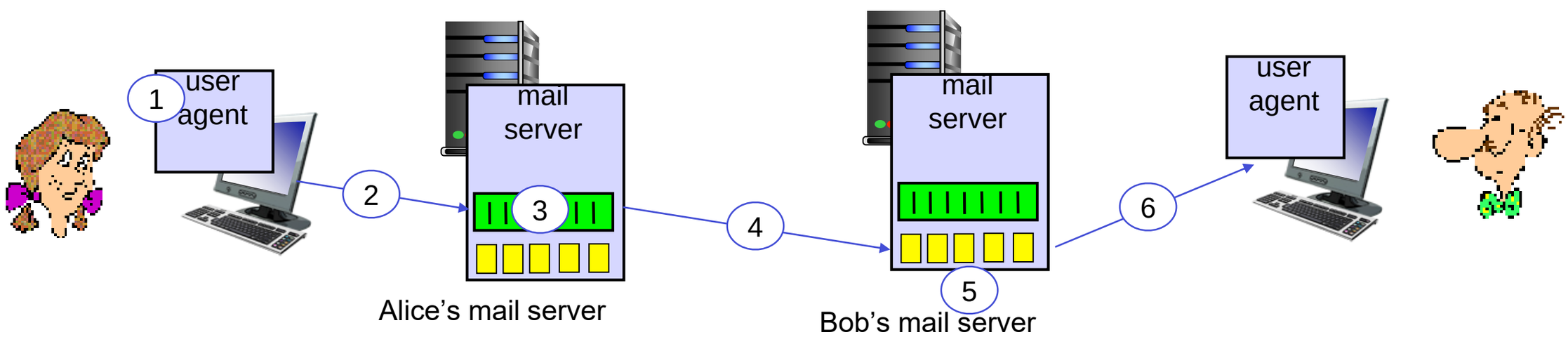
(2) Alice's mail client sends her message to her mail server; the message is placed in a message queue

(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 mail client to read message



Sample SMTP Interaction

```
C: telnet smtp.fakeplace.edu 25

S: 220 fakeplace.edu
C: HELO ycp.edu
S: 250  Hello ycp.edu, pleased to meet you
C: MAIL FROM: alice@ycp.edu
S: 250 alice@ycp.edu... Sender ok
C: RCPT TO: bob@fakeplace.edu
S: 250 bob@fakeplace.edu ... Recipient ok
C: DATA
S: 354 Enter mail, end with "." on a line by itself
C: This is a test email.
   More testing.
   .
S: 250 Message accepted for delivery
C: QUIT
S: 221 fakeplace.edu closing connection
```

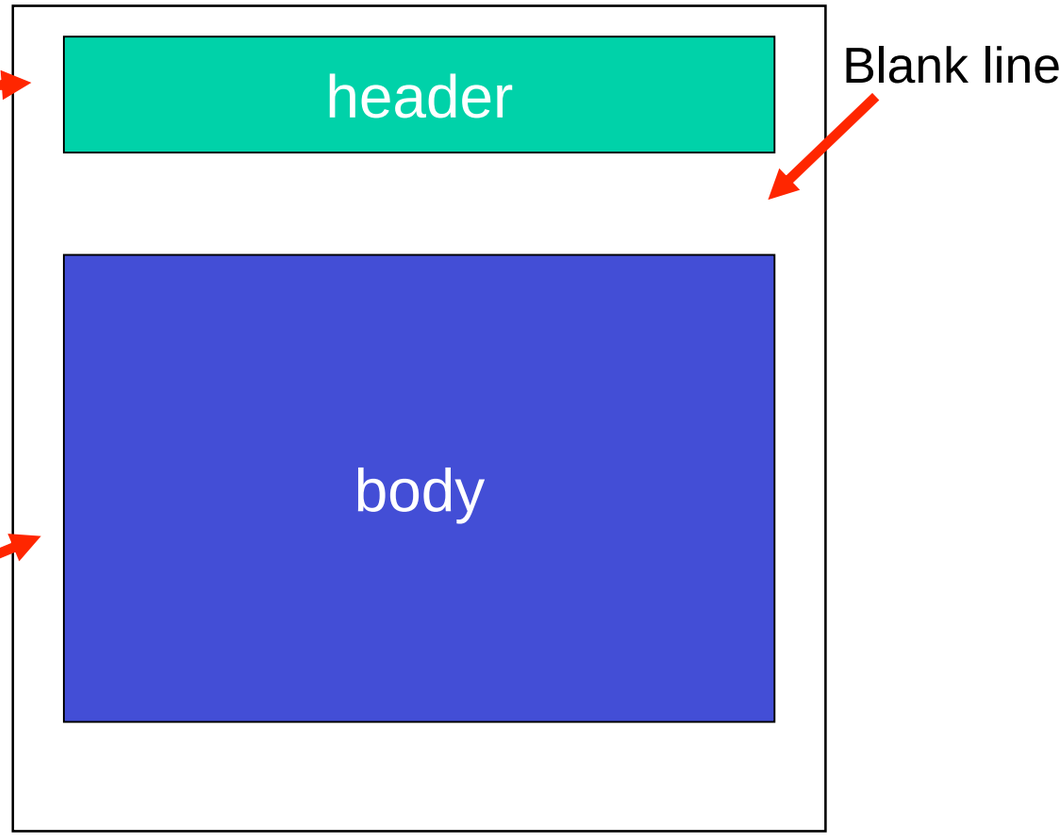
SMTP: Final Words

- **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	SMTP
Persistent/Non-persistent connections	Persistent connections
Pulls data from server	Pushes data to server
Accepts binary objects	Accepts only 7-bit ASCII
Each object in its own response msg	Multiple objects sent in multipart msg

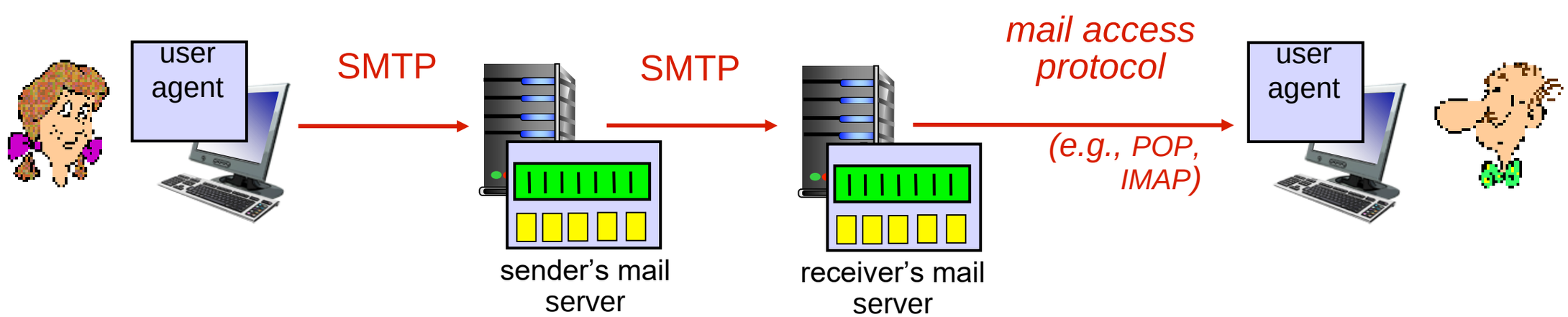
Mail Message Format

- **SMTP: protocol for exchanging email msgs**
- **RFC 822: standard for text message format:**
  - Header lines, e.g.
    To:
    From:
    Subject:
  - Different from SMTP '**`MAIL FROM`**', '**`RCPT TO`**' commands!
- **Body: the "message"**
  - ASCII characters only



Mail Access Protocols

- **SMTP - used for delivery/storage of message to receiver's mail server**
- **Mail access protocols used to retrieve messages from mail server**
  - **POP**: Post Office Protocol [RFC 1939]: authorization, download
  - **IMAP**: Internet Mail Access Protocol [RFC 1730]: more features, including manipulation of stored msgs on server
  - **HTTP**: gmail, Hotmail, Yahoo! Mail, etc.



POP3 Protocol

- **Authorization phase**
  - Client commands:
    - `user` : declare username
    - `pass` : password
  - Server responses
    - `+OK`
    - `-ERR`

```
S: +OK POP3 server ready
C: user bob
S: +OK
C: pass hungry
S: +OK user successfully logged on
```

- **Transaction phase**
  - Client commands:
    - `list` : list message numbers
    - `retr` : retrieve message by number
    - `dele` : delete message
    - `quit`

```
C: list
S: 1 498
S: 2 912
S: .
C: retr 1
S: <message 1 contents>
S: .
C: dele 1
C: retr 2
S: <message 1 contents>
S: .
C: dele 2
C: quit
S: +OK POP3 server signing off
```

IMAP

- **Internet Mail Access Protocol**
- **More sophisticated than POP3 (POP3 is stateless across sessions)**
- **Allows user to organize messages in folders**
- **Messages can be moved from one folder to another**
- **Users can get only headers or other components of the message**