M3: Application Layer

Outline

- Principles of network applications
- Web and HTTP
- FTP
- Electronic mail
  - SMTP, POP3, IMAP
- DNS
- P2P applications

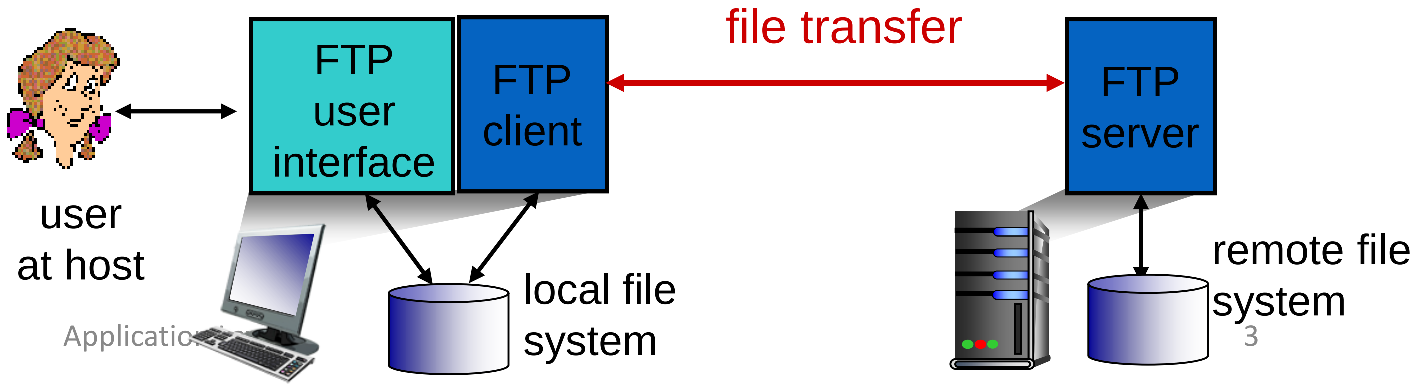
File Transfer: FTP

- Transfer file to/from remote host
- Client/server model
  - client: side that initiates transfer (either to/from remote)
  - server: remote host
- FTP client contacts FTP server at port 21, using TCP
- when server receives file transfer command, *server* opens *2nd* TCP data connection (for file) *to* client
- - server opens another TCP data connection to transfer another file

ftp: RFC 959

ftp server: port 21

FTP client contacts FTP server at port 21, using TCP



- First initiates a control TCP connection with the server side (remote host) on server port number 21.
- Control TCP connection-
  - FTP sends the user identification and password over this control connection.
  - Server side initiates a TCP data connection to the client side
- Data connection
  - Server side initiates a TCP data connection to the client side.
  - Server side initiates a TCP data connection to the client side
  - FTP opens another data connection to transfer another file.

Port 20 for the data port

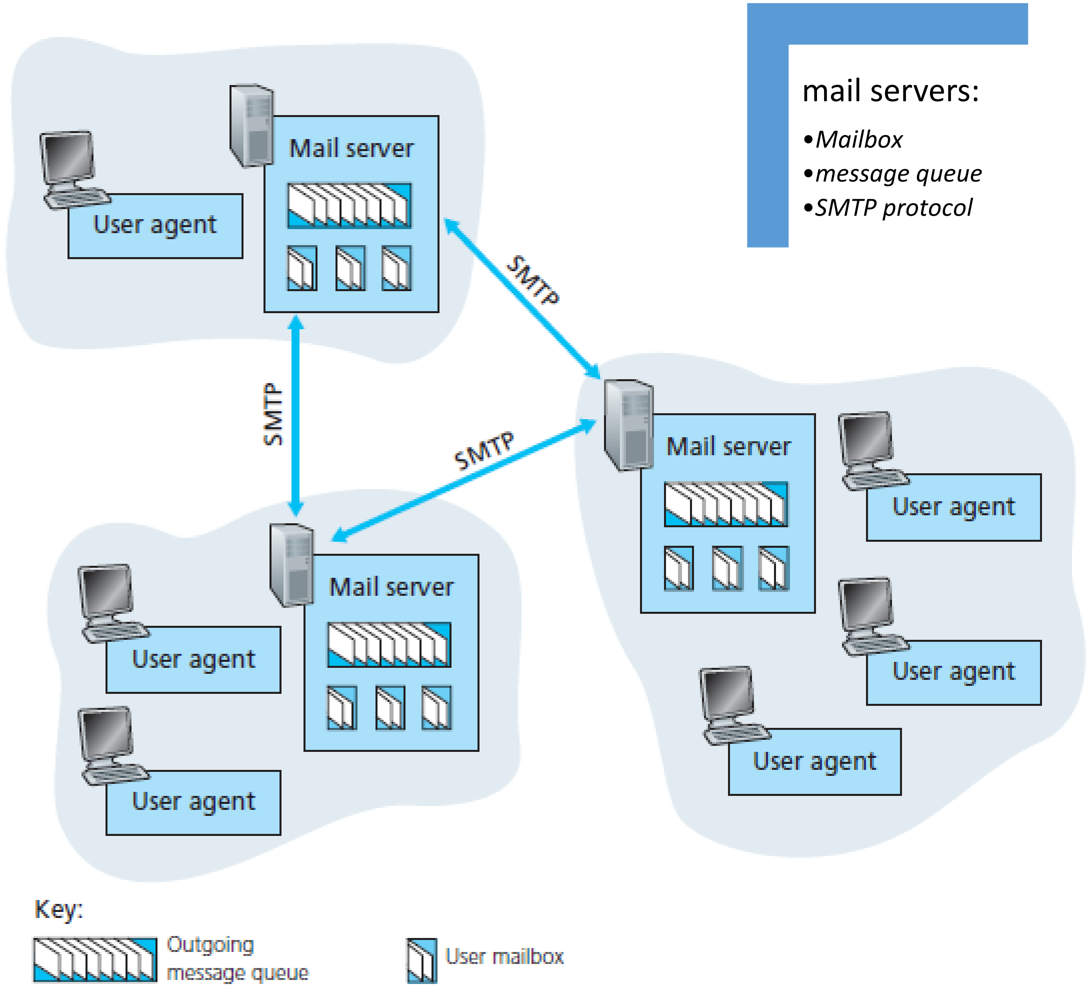
FTP server must maintain **state** about the user.
HTTP ??

Electronic Mail in the Internet

- *Three major components:*
  - user agents
  - mail servers
  - simple mail transfer protocol
- User Agents allow users to
  - Read, reply to, forward, save and compose
    - Microsoft Outlook
    - Apple Mail
- Each recipient, such as Bob, has a mailbox located in one of the mail servers.

<u>Journey</u>

Sender's user agent → sender's mail server, → recipient's mail server



igure 2.16 ♦ A high-level view of the Internet e-mail system

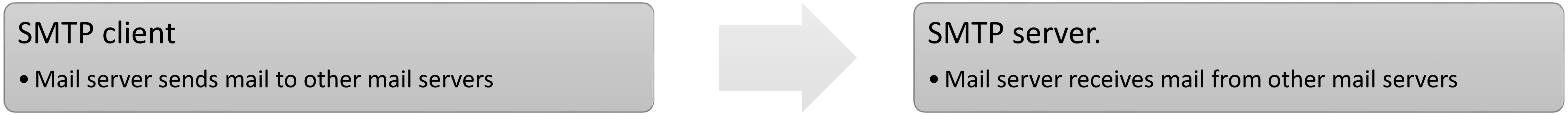
mail servers:

- *Mailbox*
- *message queue*
- *SMTP protocol*

SMTP-RFC 5321

- SMTP is the principal application-layer protocol Internet electronic mail
  - Uses the reliable data transfer service of TCP port 25.
  - SMTP has two sides:
    - **A client side**, which executes on the sender’s mail server,
    - **A server side**, which executes on the recipient’s mail server.

Every mail server



SMTP client

- Mail server sends mail to other mail servers

SMTP server.

- Mail server receives mail from other mail servers

three phases of transfer

- handshaking (greeting)
- transfer of messages
- closure

command/response interaction (like HTTP, FTP)

- commands: ASCII text
- response: status code and phrase

messages must be in 7-bit ASCI

Scenario: Alice sends message to Bob

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

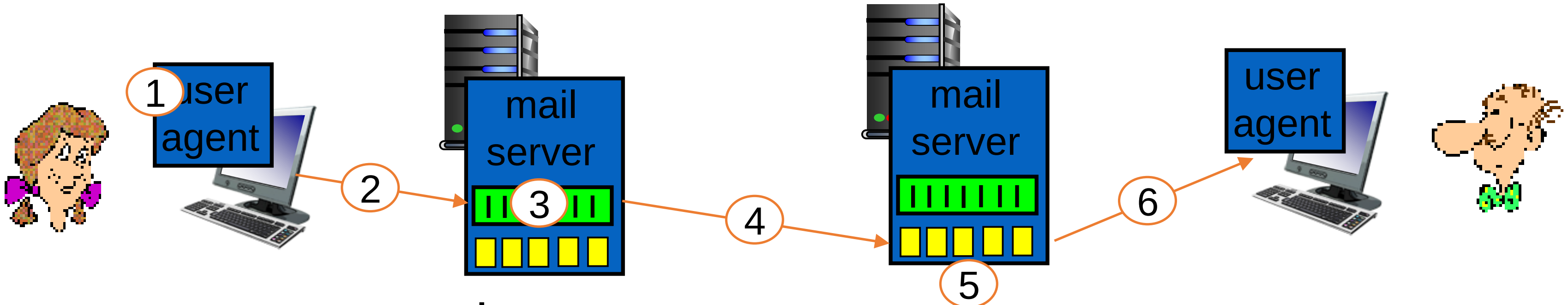
2) Alice's UA sends message to her mail server; message placed in message queue

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



Alice's mail server

Bob's mail server

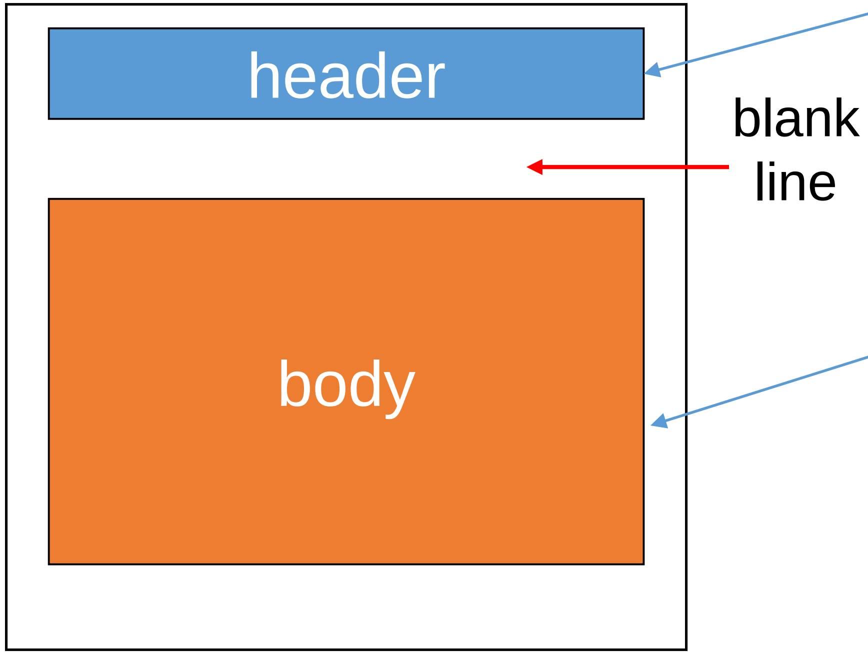
Comparison with HTTP

- HTTP is mainly a **pull protocol**
- HTTP data does not impose this restriction
- HTTP encapsulates each object in its own HTTP response message

- SMTP is primarily a **push protocol**
- SMTP requires each message, including the body of each message, to be in 7-bit ASCII format.
- Internet mail places all of the message’s objects into one message.

Mail Message Formats

- **Mail Message Formats**

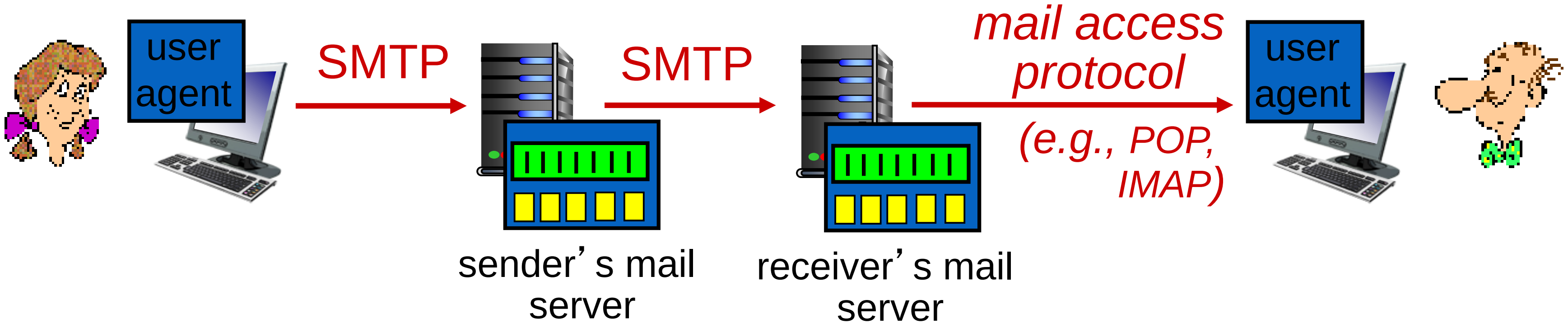


header lines, e.g.,
To:
From:
Subject:

Body: the “message”
ASCII characters only

From: alice@crepes.fr
To: bob@hamburger.edu
Subject: Searching for the meaning of life.

Mail access protocols



- SMTP: delivery/storage to receiver’s server
- mail access protocol: retrieval from 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

transaction phase, client:

- **list**: list message numbers
- **retr**: retrieve message by number
- **dele**: delete
- **quit**

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

```
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
```

POP3 (more) and IMAP

more about POP3

- previous example uses POP3 “download and delete” mode
  - Bob cannot re-read e-mail if he changes client
- POP3 “download-and-keep”: copies of messages on different clients
- POP3 is stateless across sessions

IMAP

- keeps all messages in one place: at server
- allows user to organize messages in folders
- keeps user state across sessions:
  - names of folders and mappings between message IDs and folder name