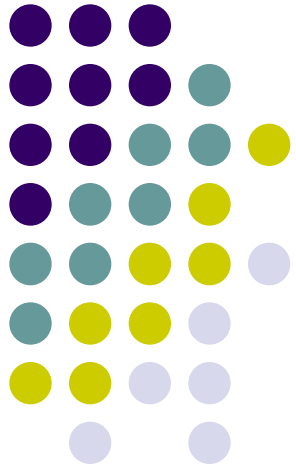
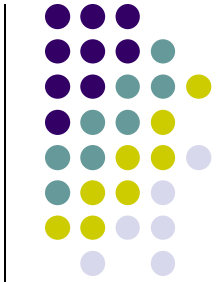
ELEN 4017

Network Fundamentals

Lecture 10 & 11



Purpose of lecture



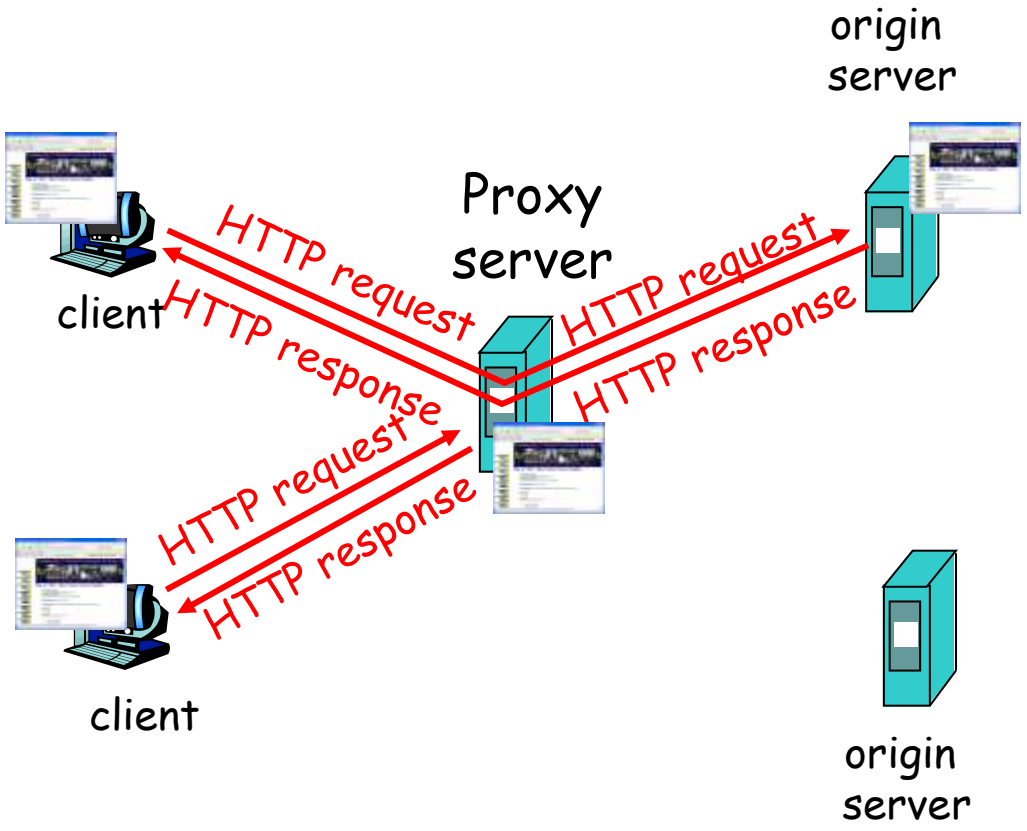
Chapter 2: Application Layer

- Web and HTTP (Caching)

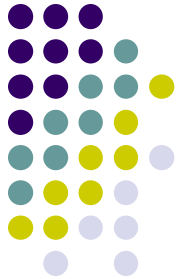
Web caches (proxy server)

Goal: satisfy client request without involving origin server

- user sets browser: Web accesses via cache
- browser sends all HTTP requests to cache
  - object in cache: cache returns object
  - else cache requests object from origin server, then returns object to client



More about Web caching



- cache acts as both client and server
- typically cache is installed by ISP (university, company, residential ISP)

Why Web caching?

- reduce response time for client request
- reduce traffic on an institution's access link.
- Internet dense with caches: enables "poor" content providers to effectively deliver content (but so does P2P file sharing)

Caching example

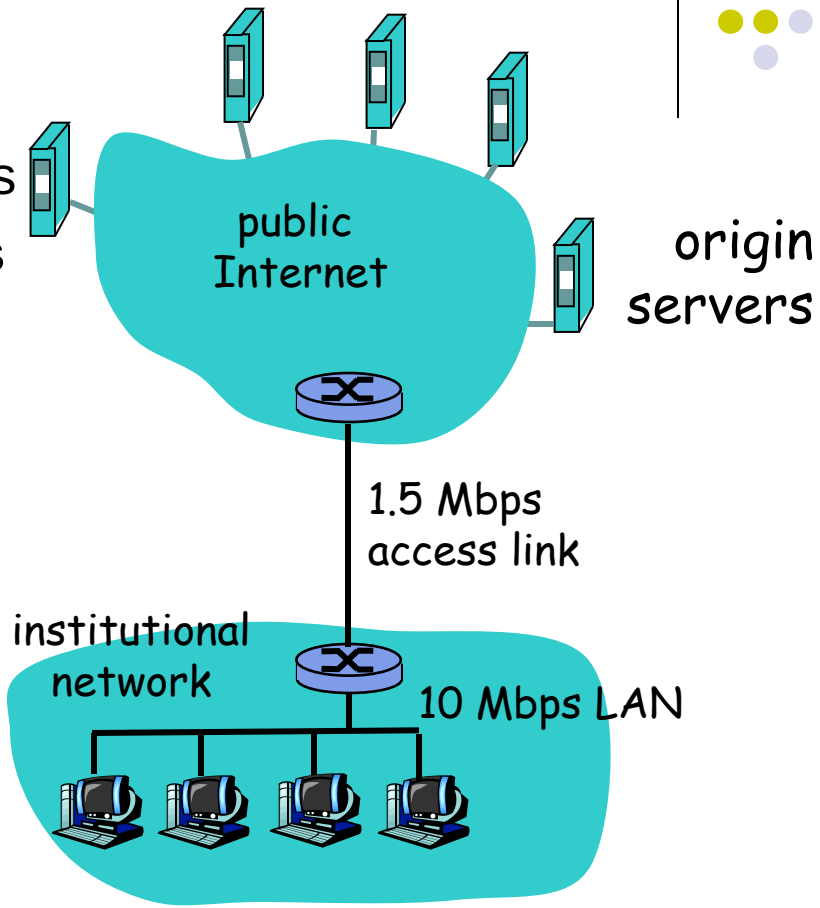
Assumptions

- average object size = 100,000 bits
- avg. request rate from institution's browsers to origin servers = 15/sec
- delay from institutional router to any origin server and back to router = 2 sec

Consequences

- utilization on LAN = 15%
- utilization on access link = 100%
- total delay = Internet delay + access delay + LAN delay

= 2 sec + minutes + milliseconds



Caching example (cont)

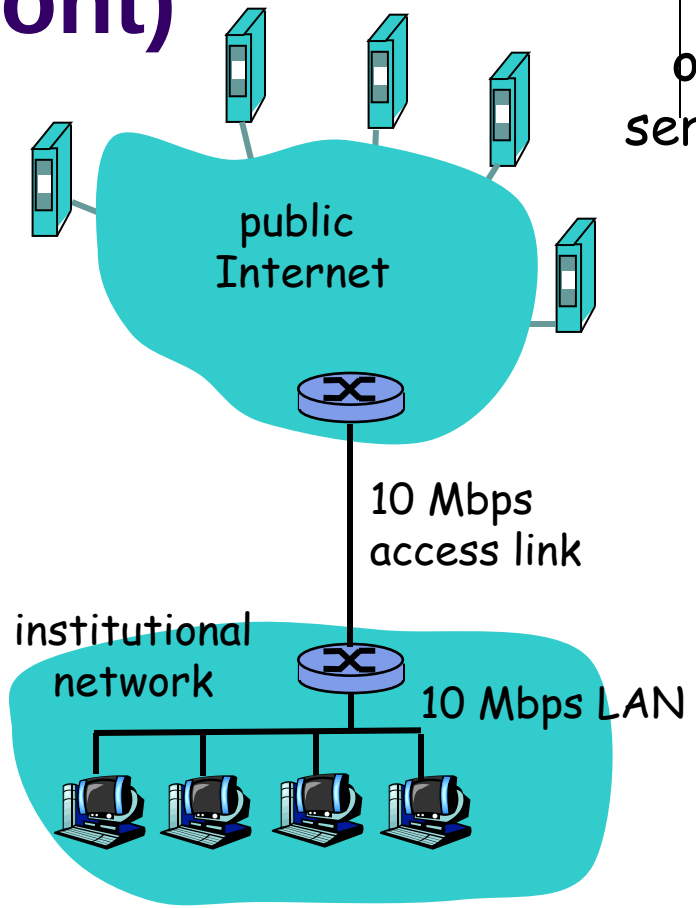
possible solution

- increase bandwidth of access link to, say, 10 Mbps

consequence

- utilization on LAN = 15%
- utilization on access link = 15%
- Total delay = Internet delay + access delay + LAN delay

  = 2 sec + msecs + msecs
- often a costly upgrade



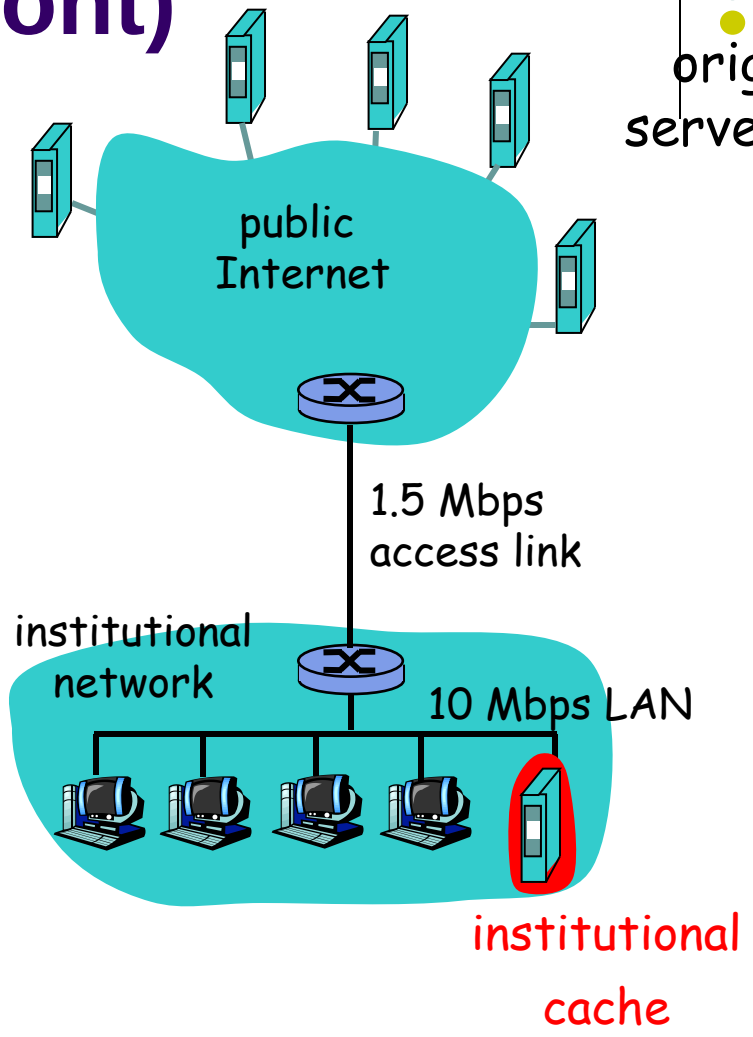
Caching example (cont)

possible solution: install cache

- suppose hit rate is 0.4

consequence

- 40% requests will be satisfied almost immediately
- 60% requests satisfied by origin server
- utilization of access link reduced to 60%, resulting in negligible delays (say 10 msec)
- total avg delay = Internet delay + access delay + LAN delay = .6*(2.01) secs + .4*milliseconds < 1.4 secs

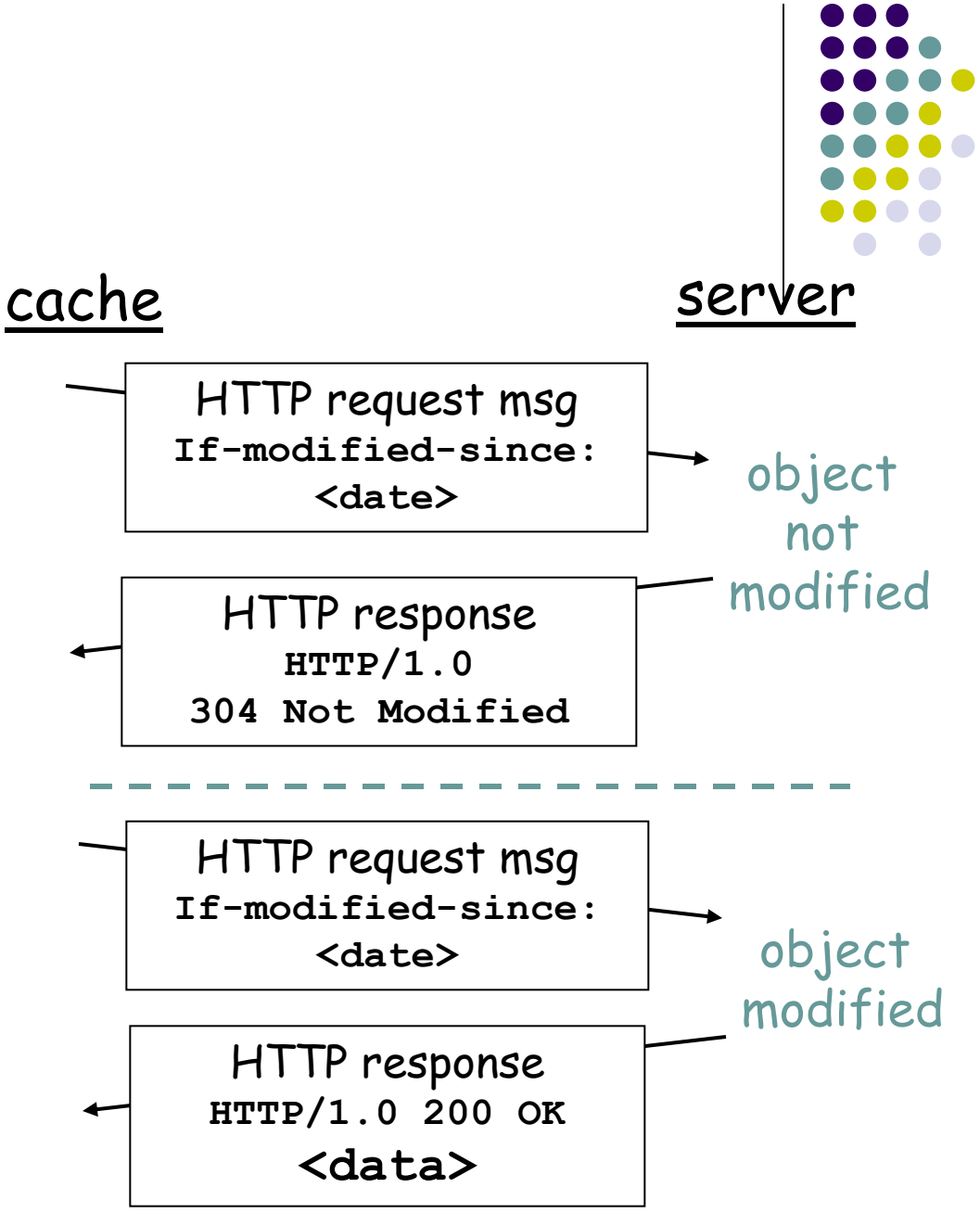


Conditional GET

- Goal: don't send object if cache has up-to-date cached version
- cache: specify date of cached copy in HTTP request

  `If-modified-since: <date>`
- server: response contains no object if cached copy is up-to-date:

  `HTTP/1.0 304 Not Modified`

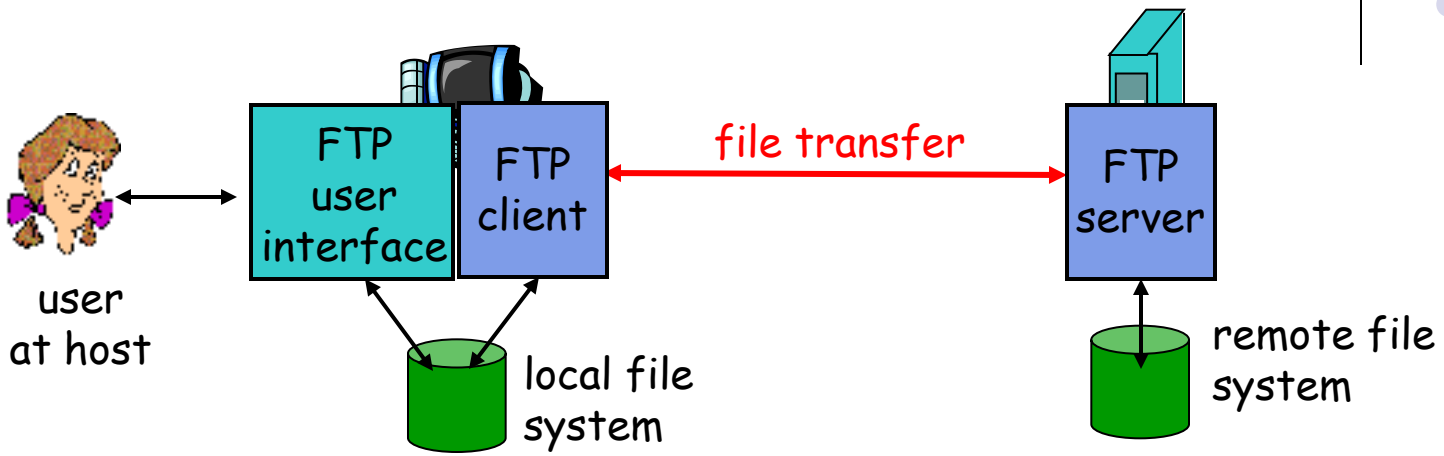


Purpose of lecture

Chapter 2: Application Layer

- FTP and email

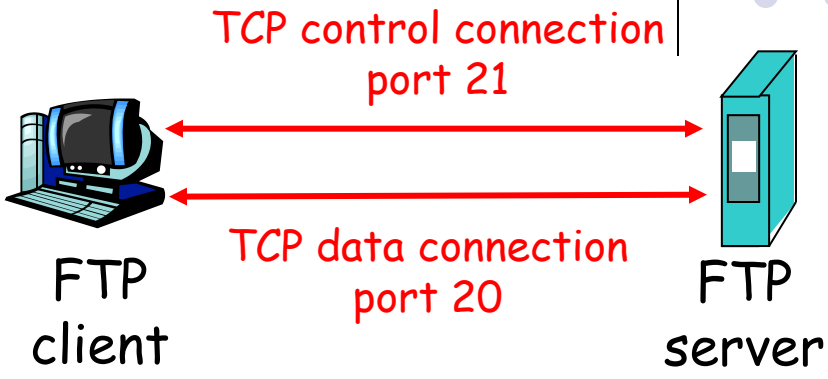
FTP: the file transfer protocol



- 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

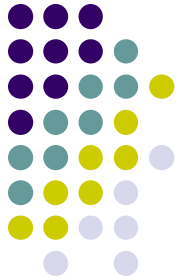
FTP: separate control, data connections

- 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



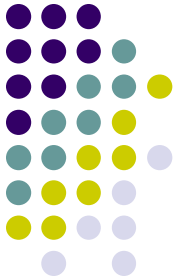
Sample commands:

- 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

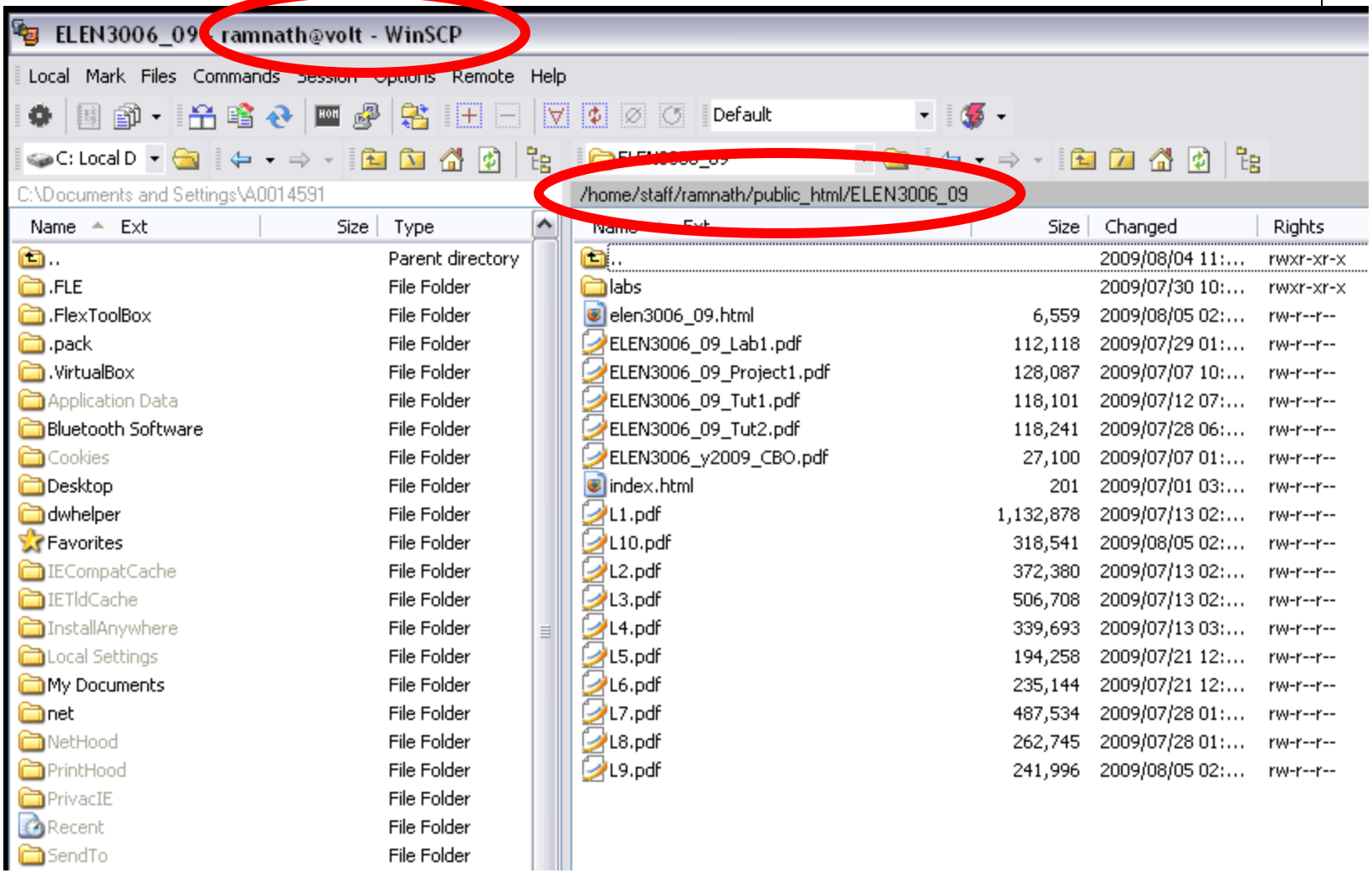
- status code and phrase (as in HTTP)
- **`331 Username OK, password required`**
- **`125 data connection already open; transfer starting`**
- **`425 Can't open data connection`**
- **`452 Error writing file`**

Secure FTP (SCP)

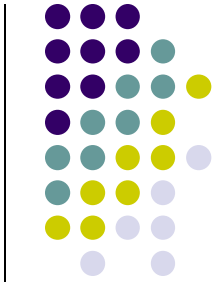


- Command line demo (SCP)

WinSCP


ELEN3006_09 - ramnath@volt - WinSCP
Local Mark Files Commands Session Options Remote Help
Default
C: Local D
C:\Documents and Settings\A0014591
/home/staff/ramnath/public_html/ELEN3006_09
Name Ext Size Type
.. Parent directory
.FLE File Folder
.FlexToolBox File Folder
.pack File Folder
.VirtualBox File Folder
Application Data File Folder
Bluetooth Software File Folder
Cookies File Folder
Desktop File Folder
dwhelper File Folder
Favorites File Folder
IECompatCache File Folder
IETldCache File Folder
InstallAnywhere File Folder
Local Settings File Folder
My Documents File Folder
net File Folder
NetHood File Folder
PrintHood File Folder
PrivacIE File Folder
Recent File Folder
SendTo File Folder
Name Ext Size Changed Rights
.. 2009/08/04 11:... rwxr-xr-x
labs 2009/07/30 10:... rwxr-xr-x
elen3006_09.html 6,559 2009/08/05 02:... rw-r--r--
ELEN3006_09_Lab1.pdf 112,118 2009/07/29 01:... rw-r--r--
ELEN3006_09_Project1.pdf 128,087 2009/07/07 10:... rw-r--r--
ELEN3006_09_Tut1.pdf 118,101 2009/07/12 07:... rw-r--r--
ELEN3006_09_Tut2.pdf 118,241 2009/07/28 06:... rw-r--r--
ELEN3006_y2009_CBO.pdf 27,100 2009/07/07 01:... rw-r--r--
index.html 201 2009/07/01 03:... rw-r--r--
L1.pdf 1,132,878 2009/07/13 02:... rw-r--r--
L10.pdf 318,541 2009/08/05 02:... rw-r--r--
L2.pdf 372,380 2009/07/13 02:... rw-r--r--
L3.pdf 506,708 2009/07/13 02:... rw-r--r--
L4.pdf 339,693 2009/07/13 03:... rw-r--r--
L5.pdf 194,258 2009/07/21 12:... rw-r--r--
L6.pdf 235,144 2009/07/21 12:... rw-r--r--
L7.pdf 487,534 2009/07/28 01:... rw-r--r--
L8.pdf 262,745 2009/07/28 01:... rw-r--r--
L9.pdf 241,996 2009/08/05 02:... rw-r--r--

Purpose of lecture



Chapter 2: Application Layer

- FTP
- Email

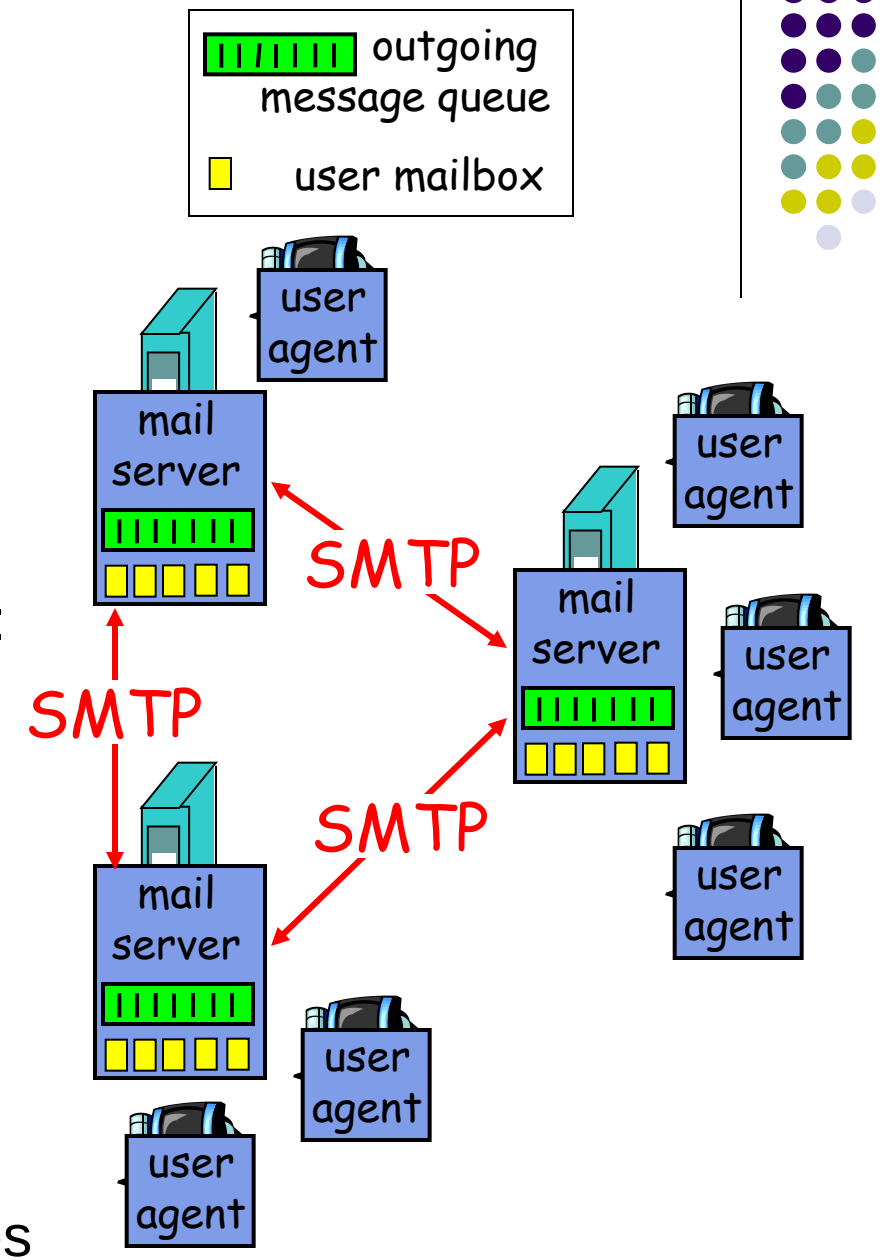
Electronic Mail

Three major components:

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

User Agent

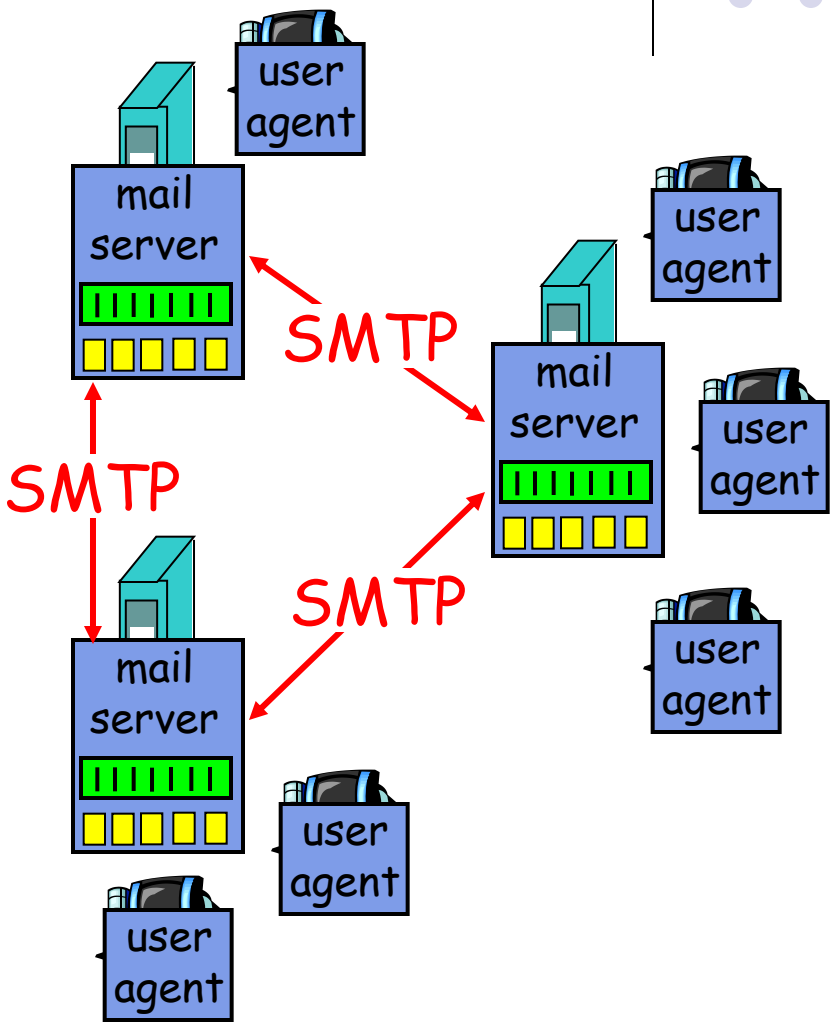
- a.k.a. “mail reader”
- composing, editing, reading mail messages
- e.g., Outlook, Thunderbird
- outgoing, incoming messages stored on server



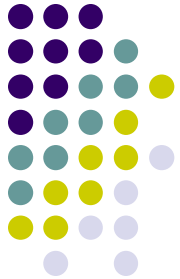
Electronic Mail: mail servers

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



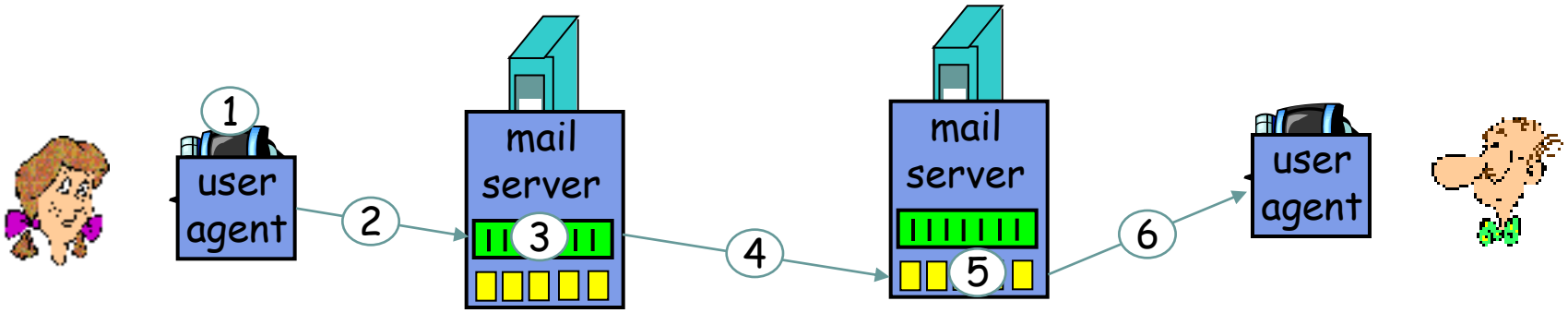
Electronic Mail: SMTP [RFC 2821]



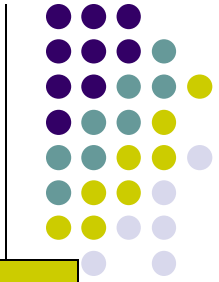
- uses TCP to reliably transfer email message from client to server, port 25
- direct transfer: sending server to receiving server
- three phases of transfer
  - handshaking (greeting)
  - transfer of messages
  - closure
- command/response interaction
  - commands: ASCII text
  - response: status code and phrase
- messages must be in 7-bit ASCII

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



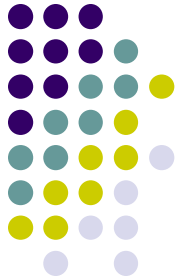
Sample SMTP interaction



```
S: 220 hamburger.edu
C: HELO crepes.fr
S: 250  Hello crepes.fr, pleased to meet you
C: MAIL FROM: <alice@crepes.fr>
S: 250 alice@crepes.fr... Sender ok
C: RCPT TO: <bob@hamburger.edu>
S: 250 bob@hamburger.edu ... Recipient ok
C: DATA
S: 354 Enter mail, end with "." on a line by itself
C: Do you like ketchup?
C: How about pickles?
C: .
S: 250 Message accepted for delivery
C: QUIT
S: 221 hamburger.edu closing connection
```

Server response after TCP conn opened

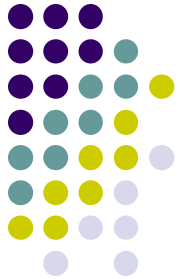
Try SMTP interaction for yourself:



- **`telnet servername 25`**
- see 220 reply from server
- enter HELO, MAIL FROM, RCPT TO, DATA, QUIT commands

above lets you send email without using email client (reader)

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

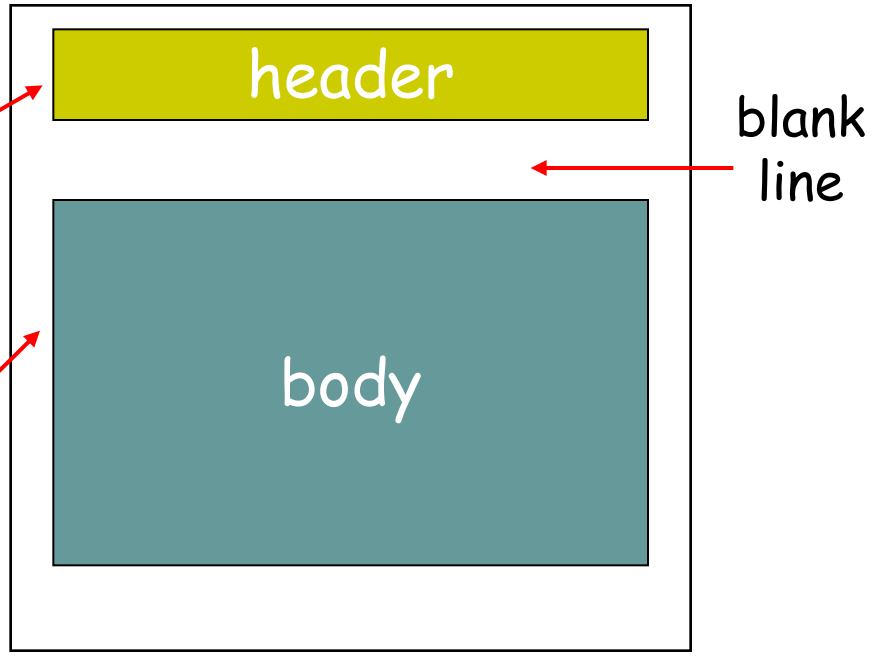
Mail message format

SMTP: protocol for exchanging email msgs

RFC 822: standard for text message format:

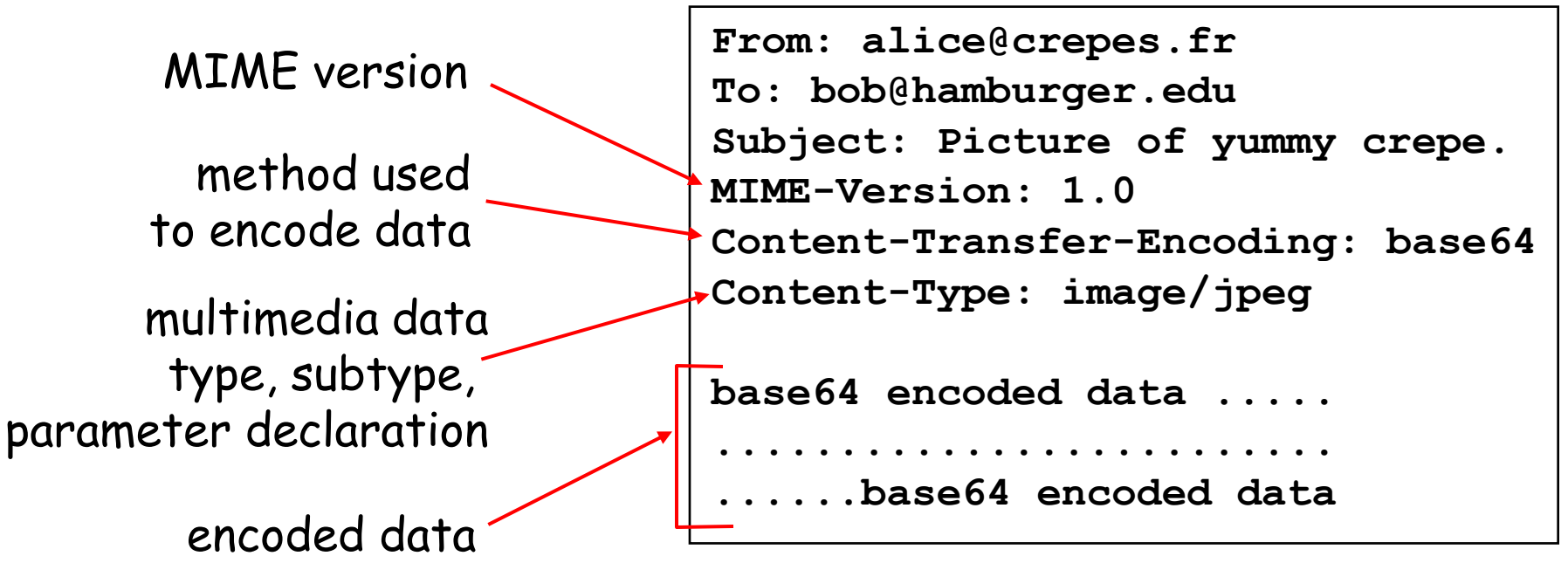
- header lines, e.g.,
  - To:
  - From:
  - Subject:

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

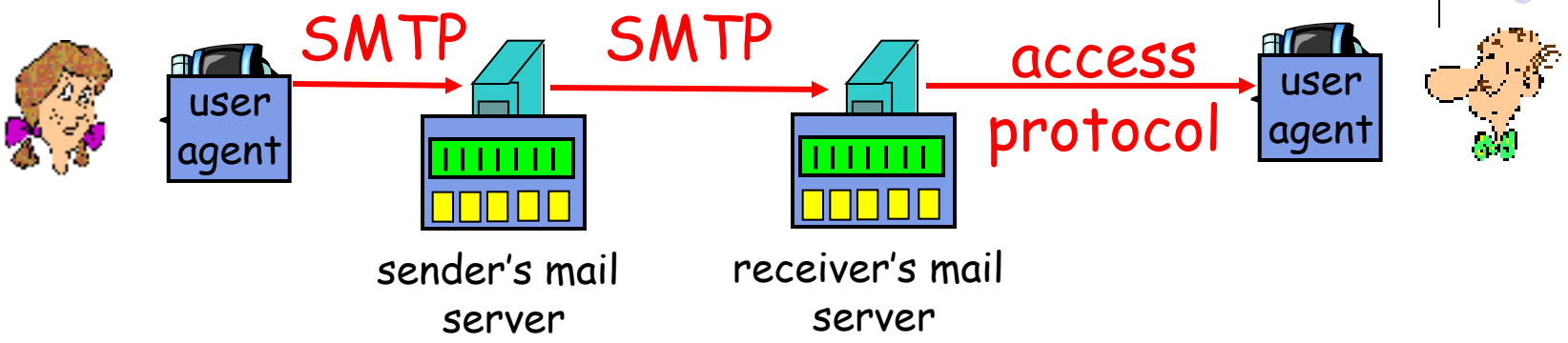


Message format: multimedia extensions

- MIME: multimedia mail extension, RFC 2045, 2056
- additional lines in msg header declare MIME content type



Mail access protocols



- SMTP: delivery/storage to receiver's server
- Mail access protocol: retrieval from server
  - POP: Post Office Protocol [RFC 1939]
    - authorization (agent <-->server) and download
  - IMAP: Internet Mail Access Protocol [RFC 1730]
    - more features (more complex)
    - 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

```
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

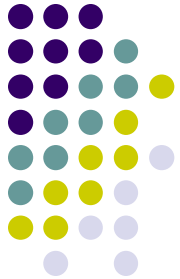
More about POP3

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

IMAP

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

Hotmail - history



- Dec 1995 founders approached venture capitalist, pitched web email.
- 3 full time, 12-14 part time, launched in July 1996
- After 1 month had 100 000 users
- 18 months – 12 million subscribers
- Sold to Microsoft for $400 million.
- First mover advantage & viral marketing → killer application.