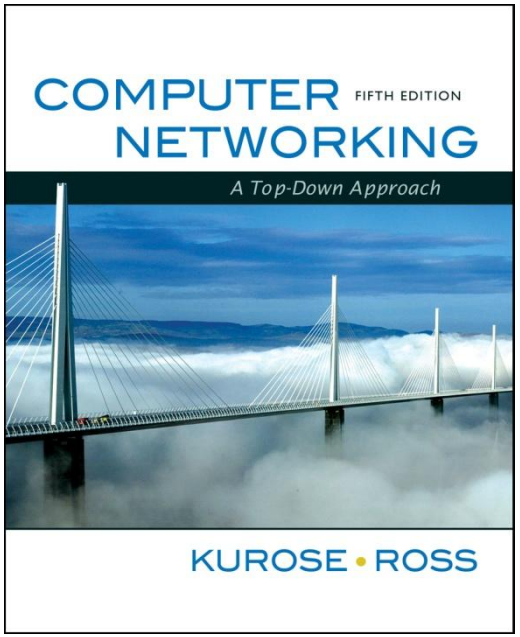
EE450 Disscussion#4
Feb 08, 2013



A Review of HTTP/FTP Protocols

Reference:
J.F Kurose and K.W. Ross

Computer Networking: A Top Down Approach,

Jim Kurose, Keith Ross
Addison-Wesley

Application layer and its protocols

- **Application layer:** The application layer is responsible for supporting network applications.
- The application layer includes many protocols, including :

  - HTTP to support the Web,
  - SMTP to support electronic mail,
  - and FTP to support file transfer.

...

Web and HTTP

First some jargons

- Web page consists of objects
- Object can be HTML file, JPEG image, Java applet, audio file,…
- Web page consists of base HTML-file which includes several referenced objects
- Each object is addressable by a URL
- Example URL:

`www.someschool.edu/someDept/pic.gif`

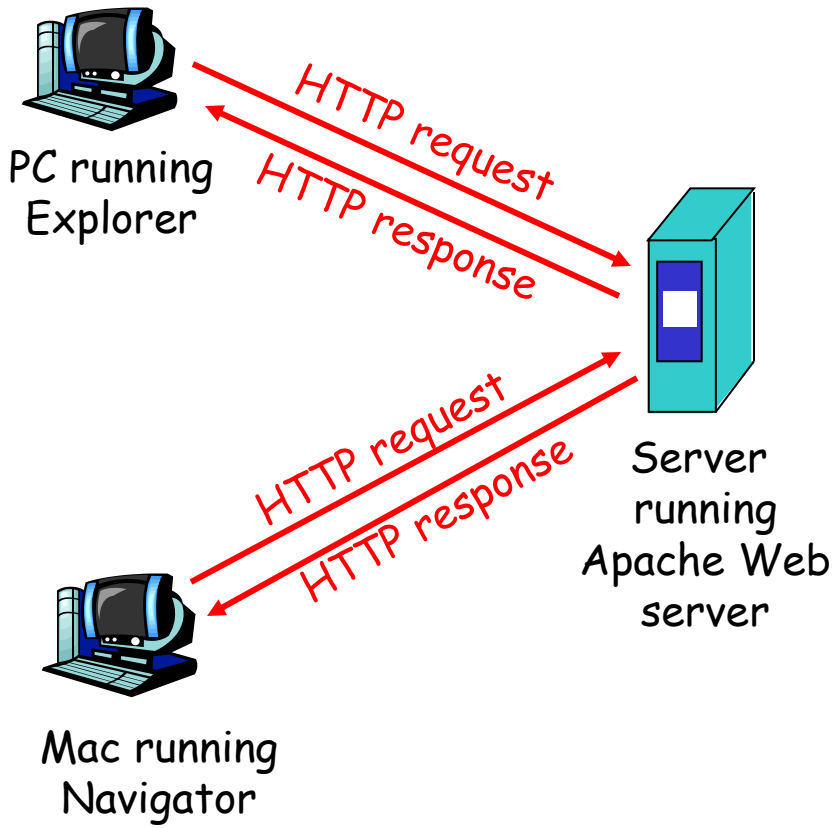
host name | path name

HTTP overview

HTTP: HyperText Transfer Protocol

- Web's application layer protocol
- client/server model:
  - *client:* browser

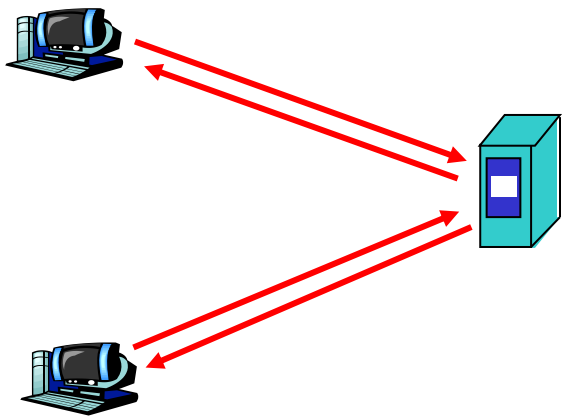
    Browser is a user agent for Web. It requests, receives, "displays" Web objects
  - *server:* Web server sends objects in response to requests



HTTP overview (continued)

Uses TCP:

1. client **initiates TCP connection** with the server
2. client **sends a HTTP request** message into the socket associated with the TCP connection
3. The HTTP server receives the request message; **encapsulates the object in a HTTP response message**, and sends.



4. client receives the response message.
5. **The TCP connection terminates.**

HTTP is "stateless"

- server maintains no information about past client requests

HTTP connections

Nonpersistent HTTP

- At most one object is sent over a TCP connection.
- HTTP/1.0 uses nonpersistent HTTP

Persistent HTTP

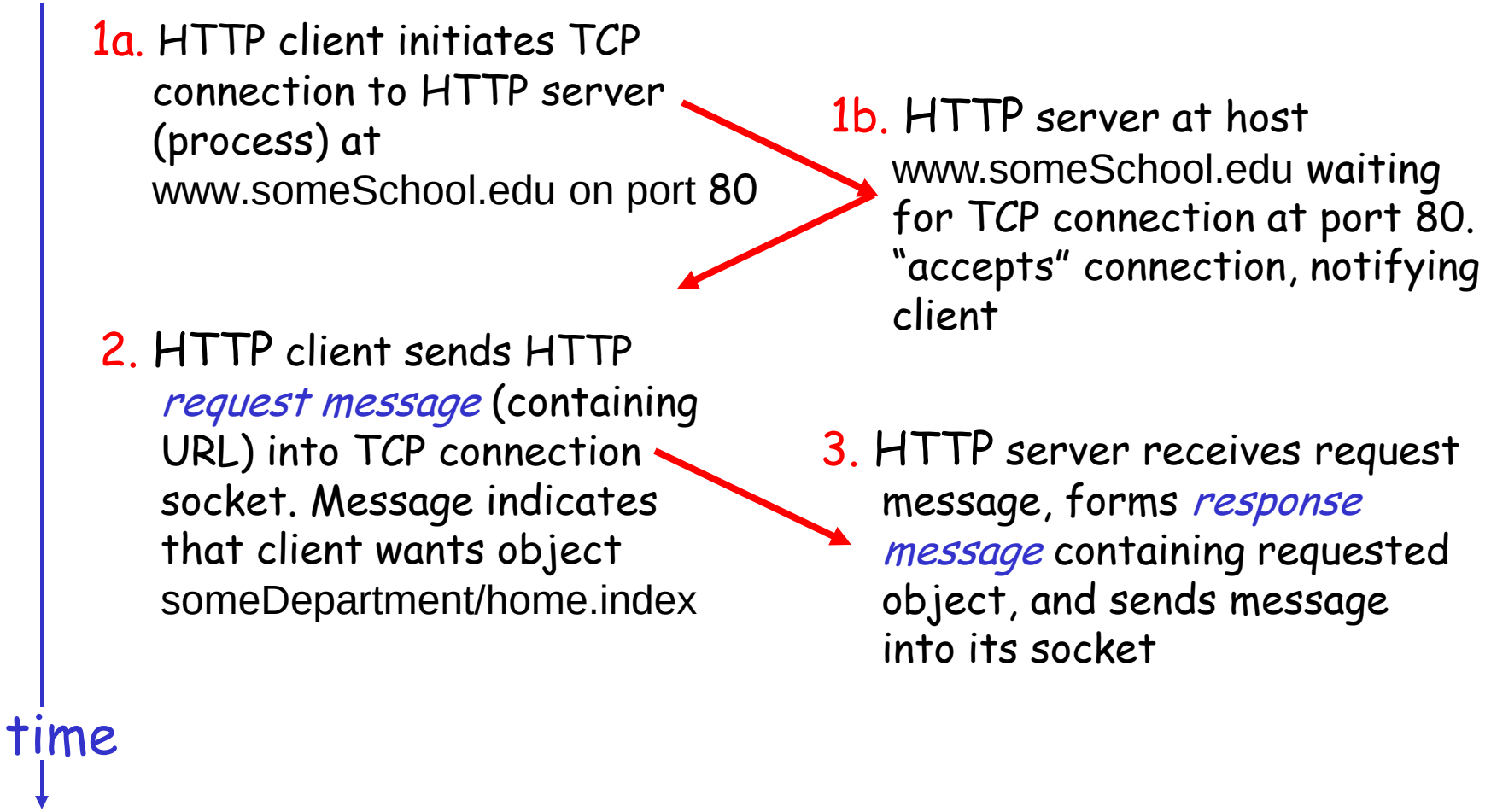
- Multiple objects can be sent over single TCP connection
- HTTP/1.1 uses persistent connections in default mode

Nonpersistent HTTP

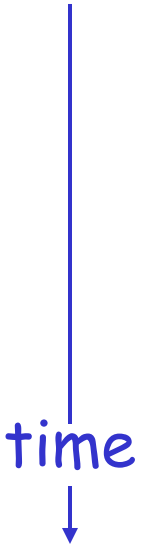
Suppose user enters URL

`www.someSchool.edu/someDepartment/home.index`

(contains text, references to 10 jpeg images)



Nonpersistent HTTP (cont.)



4. HTTP server closes TCP connection.

5. HTTP client receives response message containing html file, displays html. Parsing html file, finds 10 referenced jpeg objects

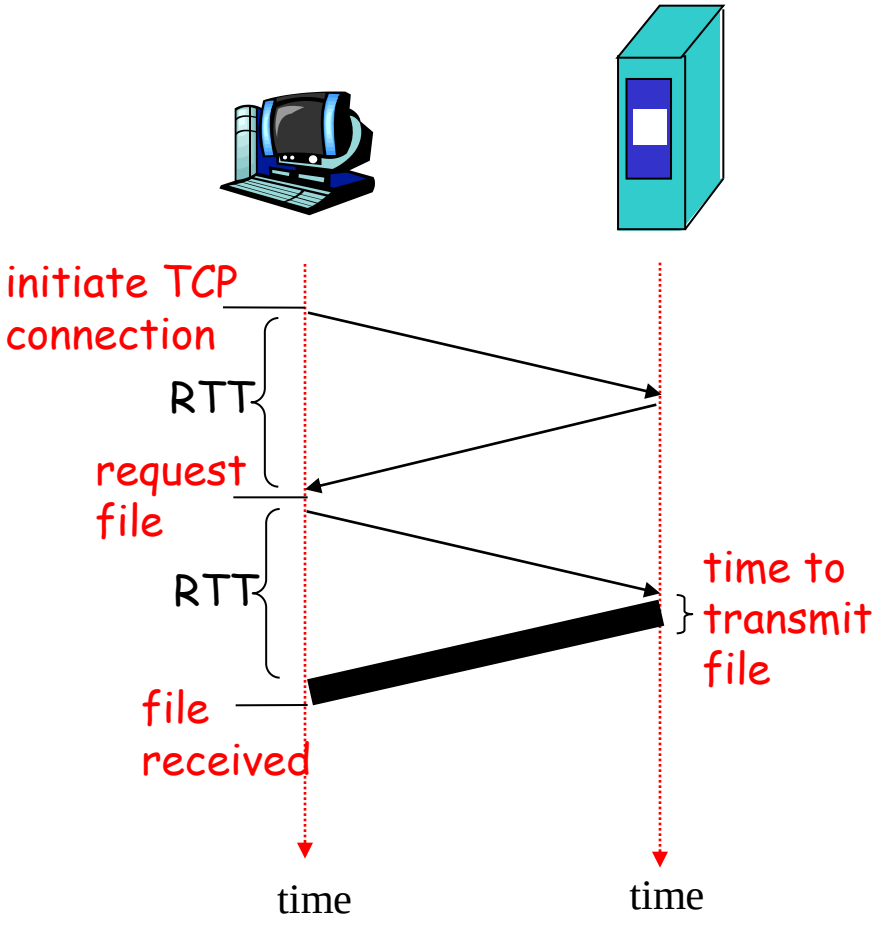
6. Steps 1-5 repeated for each of 10 jpeg objects

Non-Persistent HTTP: Response time

Response time:

- one RTT to initiate TCP connection
- one RTT for HTTP request and first few bytes of HTTP response to return
- file transmission time

total = 2RTT+transmit time



Persistent HTTP

Nonpersistent HTTP issues:

- requires 2 RTTs per object
- OS overhead for *each* TCP connection
- To alleviate: browsers often open parallel TCP connections

Persistent HTTP

- server leaves connection open after sending response
- subsequent HTTP messages between same client/server sent over open connection

Persistent ***without*** pipelining:

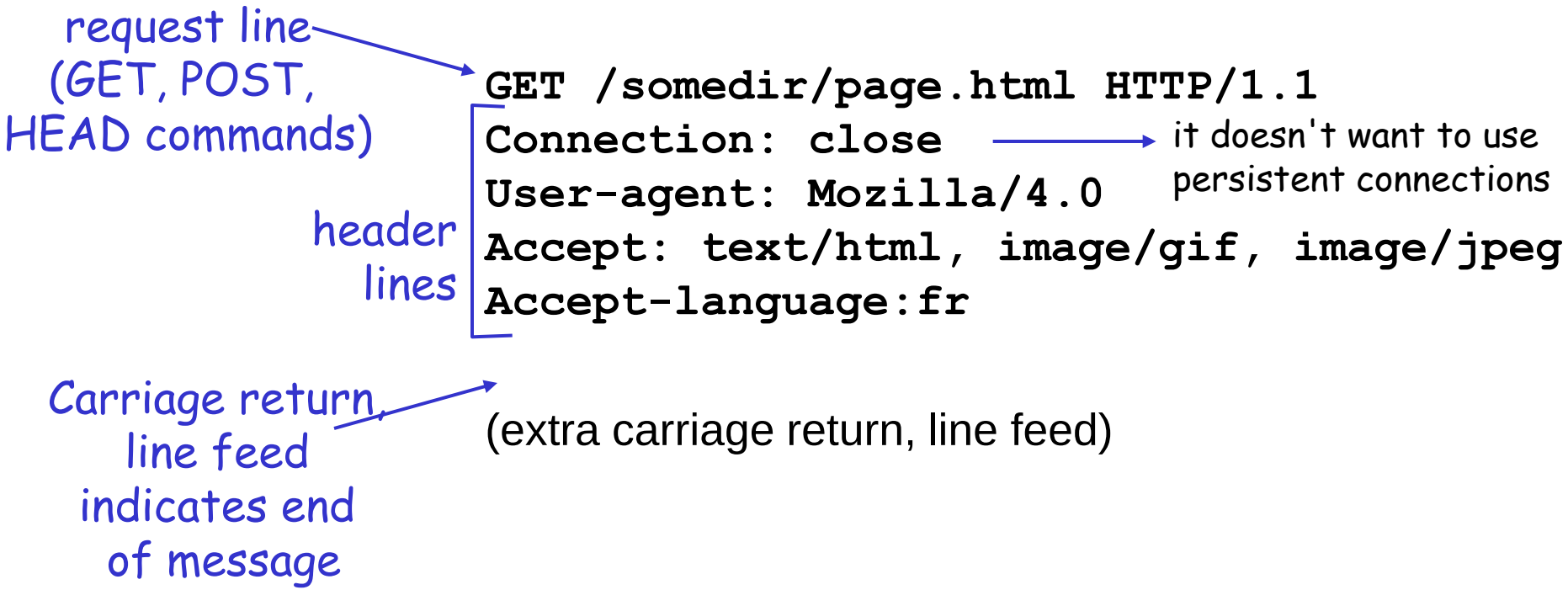
- client issues new request only when previous response has been received
- one RTT for each referenced object

Persistent ***with*** pipelining:

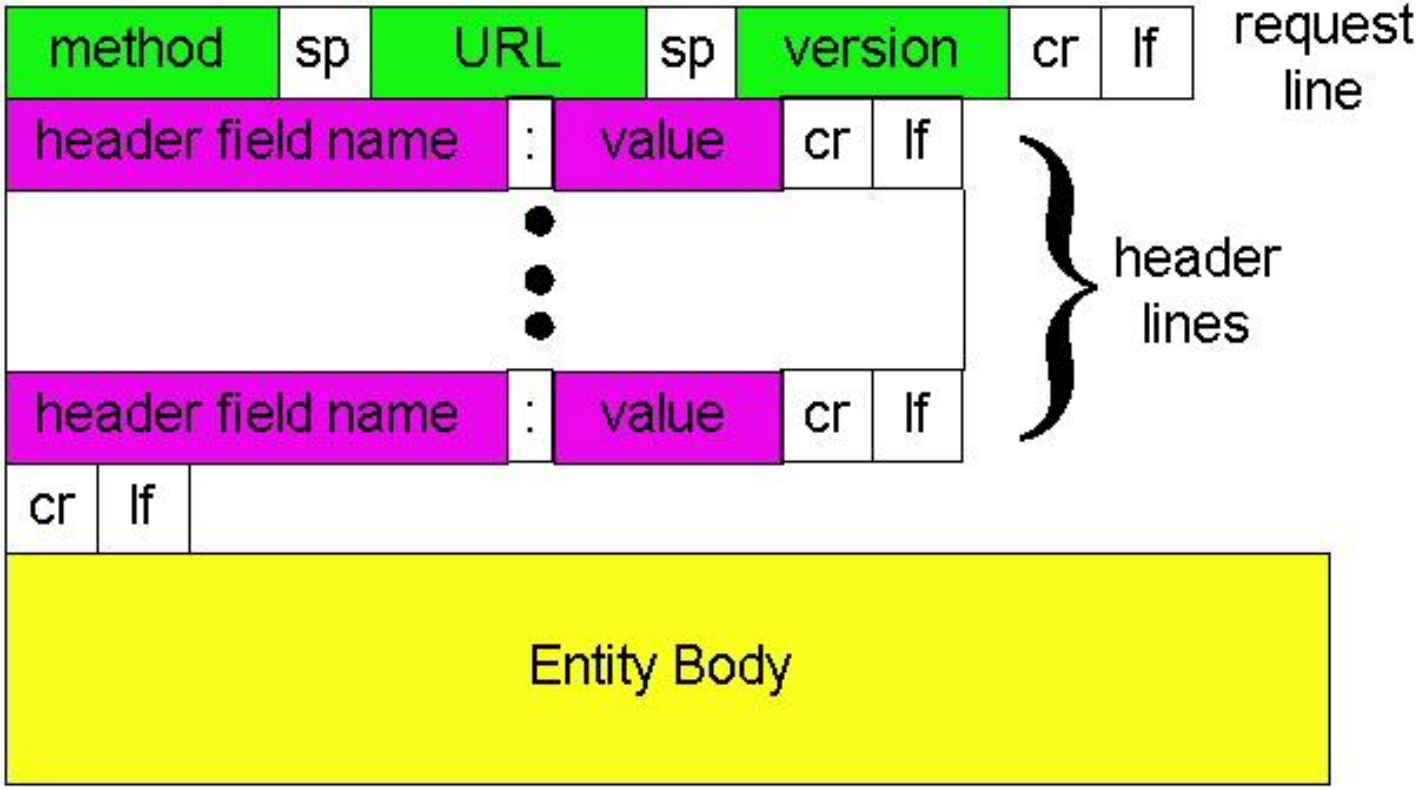
- default in HTTP/1.1
- client sends requests as soon as it encounters a referenced object
- as little as one RTT for all the referenced objects

HTTP request message

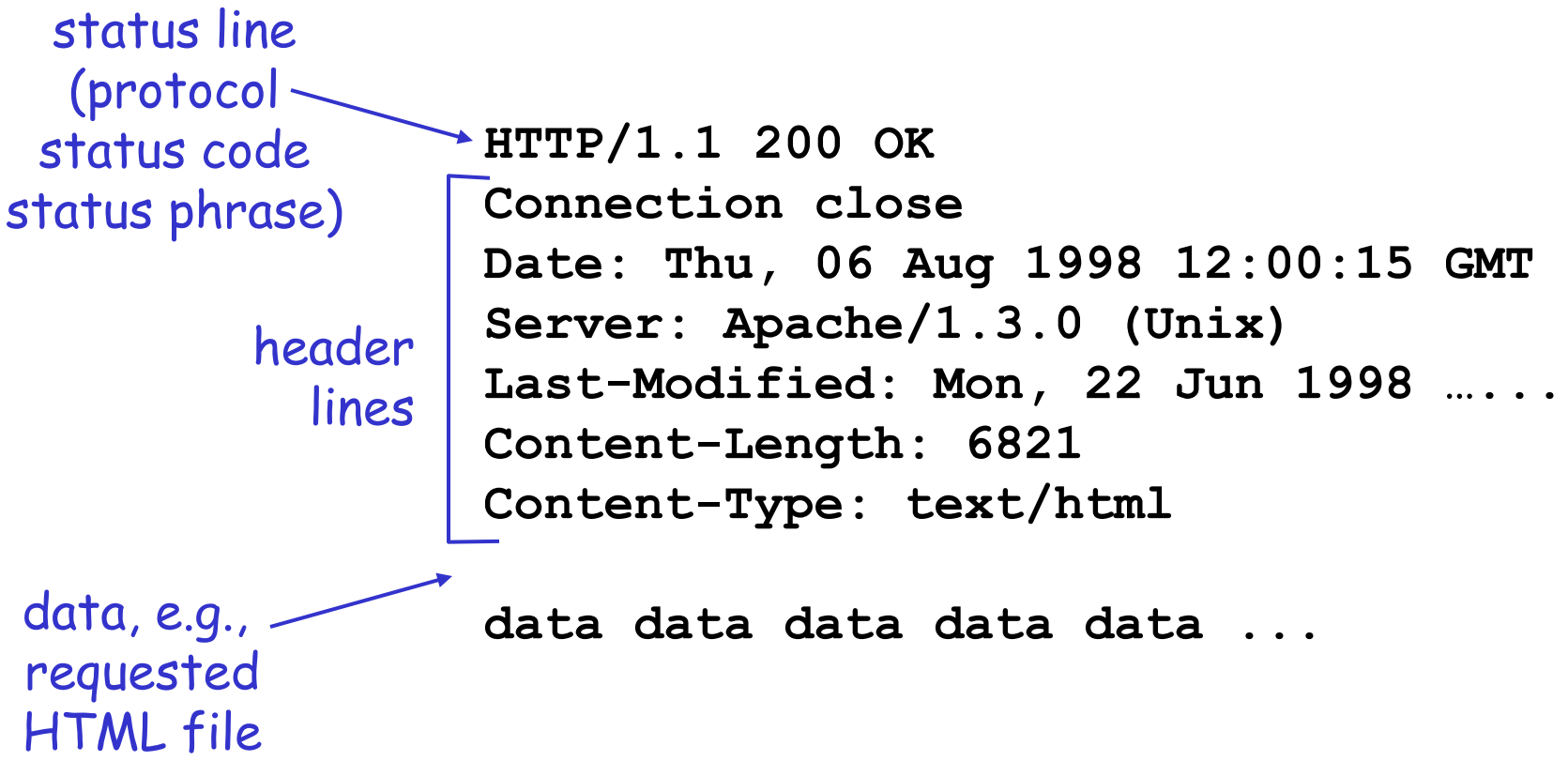
- two types of HTTP messages: *request, response*
- HTTP request message:
  - ASCII (human-readable format)



HTTP request message: general format



HTTP response message



HTTP response status codes

In first line in server->client response message.

A few sample codes:

200 OK

- request succeeded, requested object later in this message

301 Moved Permanently

- requested object moved, new location specified later in this message (Location:)

400 Bad Request

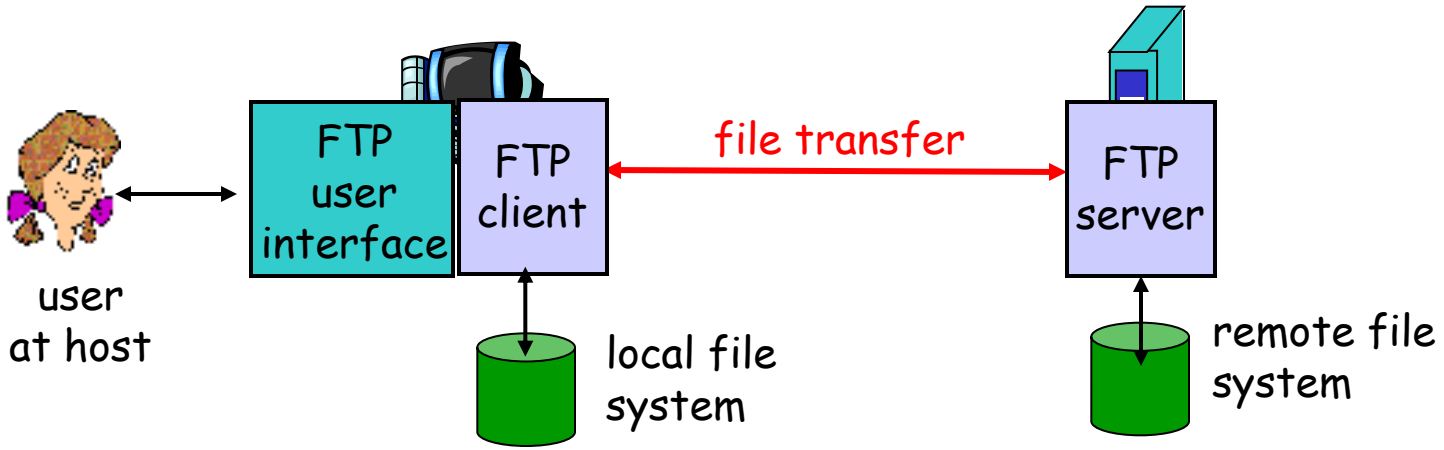
- request message not understood by server

404 Not Found

- requested document not found on this server

505 HTTP Version Not Supported

FTP: file transfer protocol



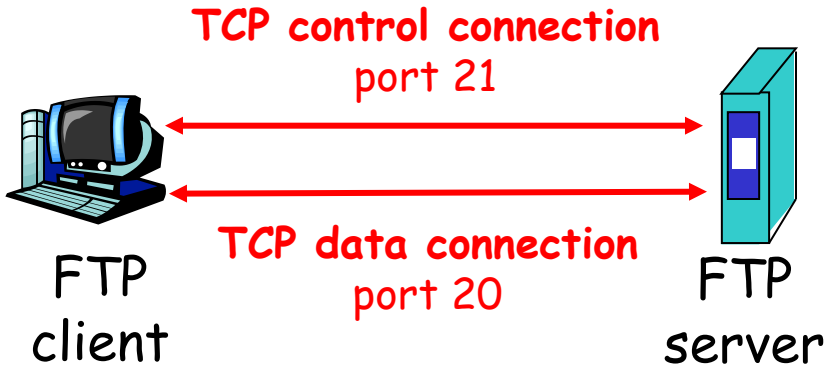
- transfer file to/from remote host
- client/server model:
  - *client:* side that initiates transfer (either to/from remote)
  - *server:* remote host

FTP: file transfer protocol

- The user first provides the hostname of the remote host→
- The FTP client establish a TCP connection with the FTP server.
- The user provides the user identification and password, which get sent over the TCP connection as part of FTP commands.
- the user copies one or more files stored in the local file system into the remote file system (or vice versa).

FTP: separate control, data connections

- FTP client contacts FTP server at port 21
- 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.
- the control connection remains open throughout the duration of the user session, but a new data connection is created for each file

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**