

Welcome to

CS 3516:

Advanced Computer Networks

Prof. Yanhua Li

Time: 9:00am –9:50am M, T, R, and F

Location: AK219

Fall 2018 A-term

Web changes how people receive/publish information



One Way



On demand

Application Layer 2-2

Chapter 2: outline

2.1 principles of network applications

- app architectures
- app requirements

2.2 Web and HTTP

- Overview
- Persistent vs non-persistent
- HTTP message formats
- Project I demo
- Web cookies
- Web Proxy



Web and HTTP

Worcester Polytechnic Institute [US] <https://www.wpi.edu>

Industria... News Web

MENU ☰



Worcester Polytechnic Institute

SEARCH



More Than "Wicked Smaht"

At WPI, we are scientists, engineers, and future business leaders. We are also athletes, artists, thespians, entrepreneurs, musicians, and spirited fans. (And you'll hear more than just a Boston accent.) Welcome home.

Web and HTTP

First, a review...

- ❖ *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*, e.g.,

www.someschool.edu/someDept/pic.gif

host name

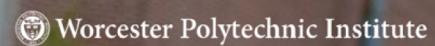
path name

Web and HTTP

Worcester Polytechnic Institute [US] <https://www.wpi.edu>

Industria... News Web

MENU ☰



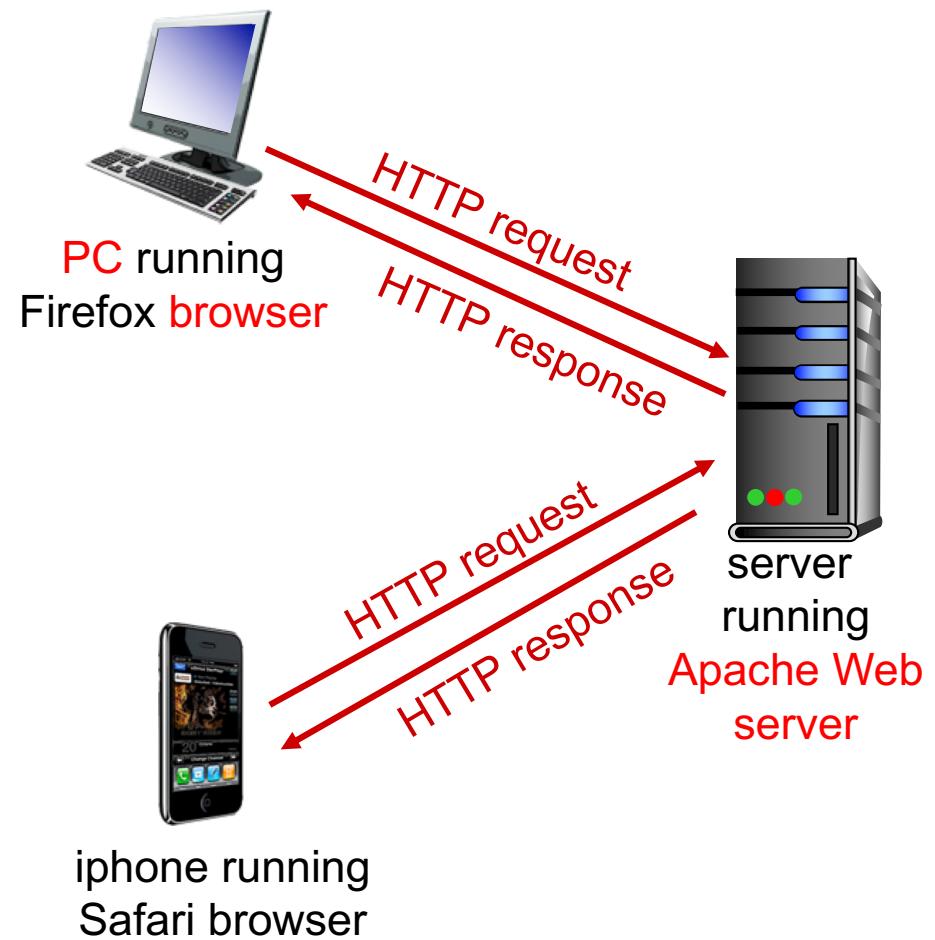
SEARCH



HTTP overview

HTTP: hypertext transfer protocol

- ❖ Web's application layer protocol
- ❖ client/server model
 - **client:** browser that requests, receives, (using HTTP protocol) and "displays" Web objects
 - **server:** Web server sends (using HTTP protocol) objects in response to requests





HTTP overview (continued)

uses TCP:

- ❖ 1. client initiates TCP connection (creates socket) to server, port 80
- ❖ 2. server accepts TCP connection from client
- ❖ 3. HTTP messages (application-layer protocol messages) exchanged between browser (HTTP client) and Web server (HTTP server)
- ❖ 4. TCP connection closed

HTTP is “stateless”

- ❖ server maintains no information about past client requests

protocols that maintain “state” are complex!

- ❖ past history (state) must be maintained
- ❖ if server/client crashes, their views of “state” may be inconsistent, must be reconciled

aside

Chapter 2: outline

2.1 principles of network applications

- app architectures
- app requirements

2.2 Web and HTTP

- Overview
- **Persistent vs non-persistent**
- HTTP message formats
- Project I demo
- Web cookies
- Web Proxy

HTTP connections

non-persistent HTTP

- ❖ at most **one object** sent over TCP connection
 - connection then closed
- ❖ downloading multiple objects required multiple connections

persistent HTTP

- ❖ multiple objects can be sent over single TCP connection between client, server

Non-persistent HTTP

suppose user enters URL:

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

(contains text,
references to 10
jpeg images)

1a. HTTP client initiates TCP connection to HTTP server (process) at `www.someSchool.edu` on port 80

1b. HTTP server at host `www.someSchool.edu` waiting for TCP connection at port 80. “accepts” connection, notifying client

2. HTTP client sends HTTP *request message* (containing URL) into TCP connection socket. Message indicates that client wants object `someDepartment/home.index`

3. HTTP server receives request message, forms *response message* containing requested object, and sends message into its socket

time
↓

Non-persistent HTTP (cont.)

time
↓

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



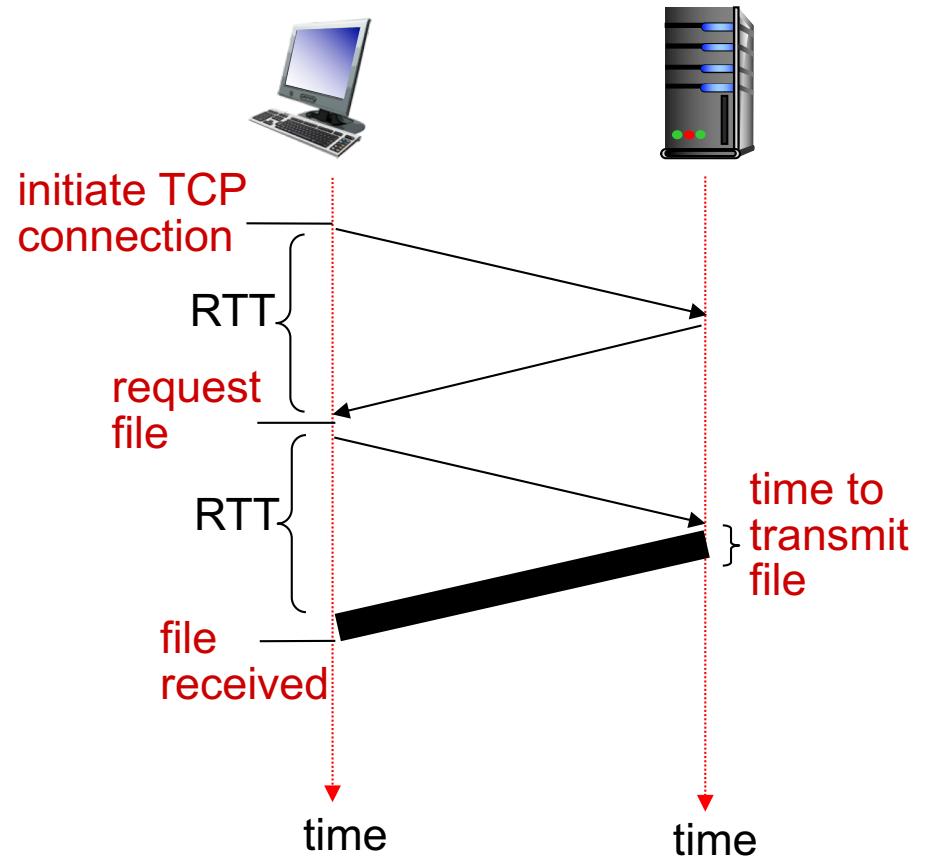
6. Steps 1-5 repeated for each of 10 jpeg objects

Non-persistent HTTP: response time

RTT (definition): time for a small packet to travel from client to server and back

HTTP 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
- ❖ non-persistent HTTP response time = $2\text{RTT} + \text{file transmission time}$



Persistent HTTP (Default)

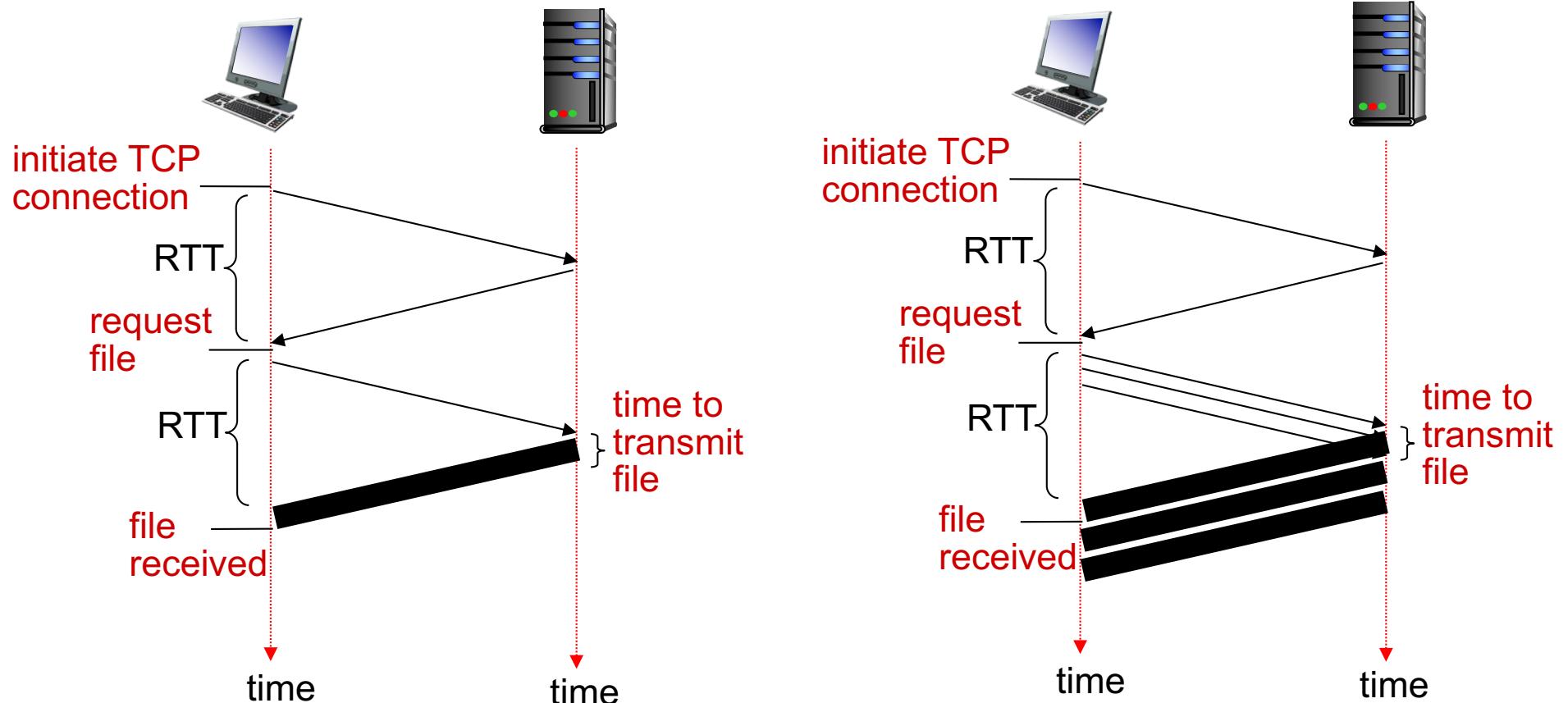
non-persistent HTTP issues:

- ❖ requires 2 RTTs per object
- ❖ **OS overhead** for each TCP connection
- ❖ **browsers often open parallel TCP connections** to fetch referenced objects
 - 5-10
 - Adjustable

persistent HTTP:

- ❖ **server leaves connection open** after sending response
- ❖ **subsequent HTTP messages** between same client/server sent over open connection
- ❖ **Back-to-back requests/responses:** client sends requests as soon as it encounters a referenced object
 - as little as one RTT for all the referenced objects

Non-persistent vs persistent



Chapter 2: outline

2.1 principles of network applications

- app architectures
- app requirements

2.2 Web and HTTP

- Overview
- Persistent vs non-persistent
- **HTTP message formats**
- **Demo of Project I**
- **Web cookies**
- **Web Proxy**

HTTP request message

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

request line
(GET, POST,
HEAD commands)

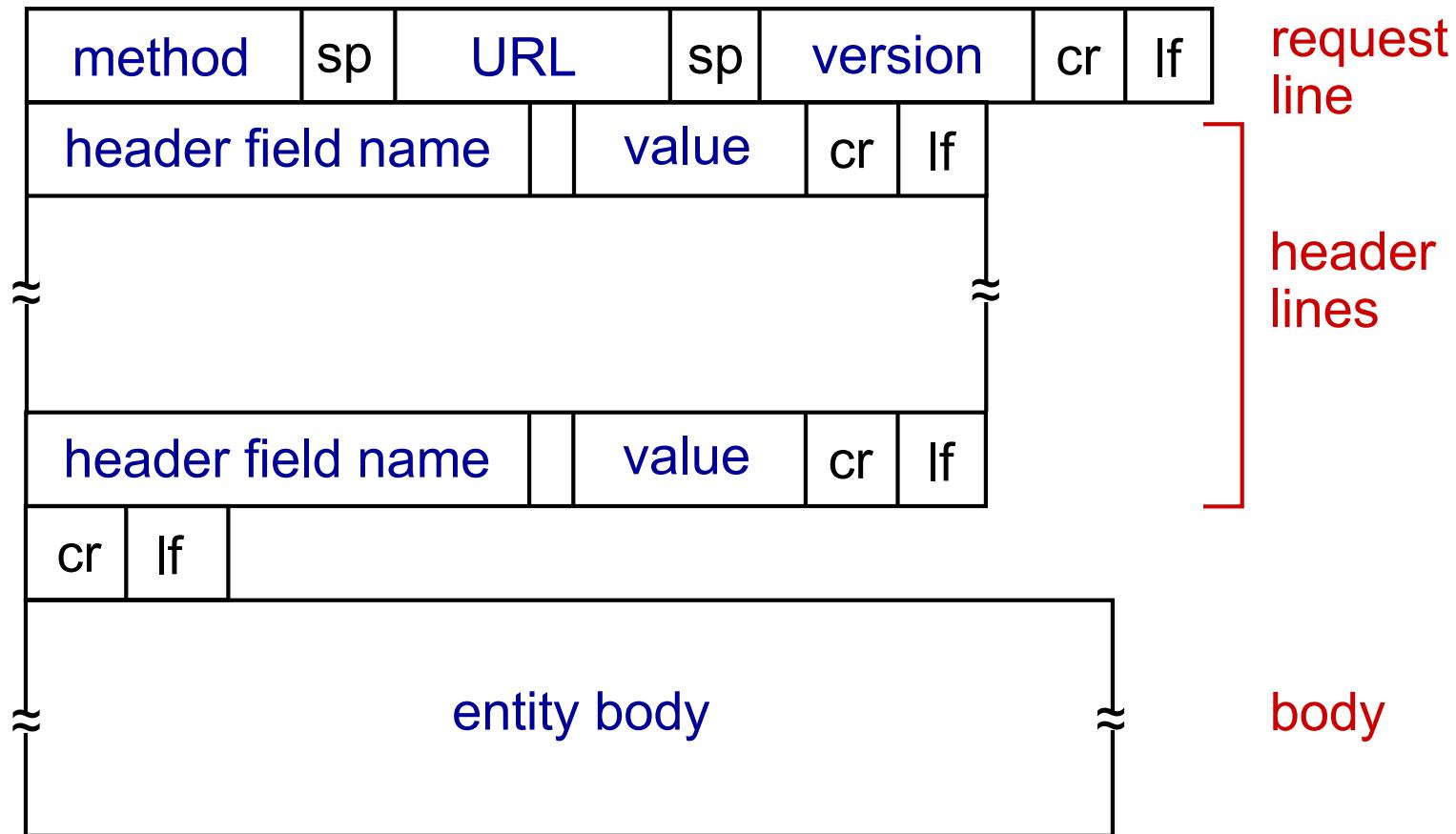
header
lines

carriage return,
line feed at start
of line indicates
end of header lines

```
GET /index.html HTTP/1.1\r\n
Host: www-net.cs.umass.edu\r\n
User-Agent: Firefox/3.6.10\r\n%Client Type
Accept: text/html,application/xhtml+xml\r\n
Accept-Language: en-us,en;q=0.5\r\n
Accept-Encoding: gzip,deflate\r\n
Accept-Charset: ISO-8859-1,utf-8;q=0.7\r\n
Keep-Alive: 115\r\n
Connection: keep-alive\r\n%Persistent
\r\n
```

carriage return character
line-feed character

HTTP request message: general format



Request from input (POST & GET)

POST method:

- ❖ web page often includes input
- ❖ input is uploaded to server in entity body



GET+URL method:

- ❖ uses GET method
- ❖ input is uploaded in URL field of request line:

`www.somesite.com/animalsearch?monkeys&banana`

Method types

HTTP/1.0:

- ❖ GET
- ❖ POST
- ❖ HEAD
 - asks server to leave requested object out of response

HTTP/1.1:

- ❖ GET, POST, HEAD
- ❖ PUT
 - uploads file in entity body to path specified in URL field
- ❖ DELETE
 - deletes file specified in the URL field

HTTP response message

status line

(protocol

status code

status phrase)

header
lines

data, e.g.,
requested
HTML file

```
HTTP/1.1 200 OK\r\nDate: Sun, 26 Sep 2010 20:09:20 GMT\r\nServer: Apache/2.0.52 (CentOS) \r\nLast-Modified: Tue, 30 Oct 2007 17:00:02  
GMT\r\nETag: "17dc6-a5c-bf716880"\r\nAccept-Ranges: bytes\r\nContent-Length: 2652\r\nKeep-Alive: timeout=10, max=100\r\nConnection: Keep-Alive\r\nContent-Type: text/html; charset=ISO-8859-  
1\r\n\r\n
```

data data data data data ...

HTTP response status codes

- ❖ status code appears in 1st line in server-to-client response message.
- ❖ some sample codes:

200 OK

- request succeeded, requested object later in this msg

301 Moved Permanently

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

400 Bad Request

- request msg not understood by server

404 Not Found

- requested document not found on this server

505 HTTP Version Not Supported

Trying out HTTP (client side) for yourself

I. Telnet to your favorite Web server:

```
telnet gaia.cs.umass.edu 80
```

opens TCP connection to port 80
(default HTTP server port) at cis.poly.edu.
anything typed in sent
to port 80 at cis.poly.edu

2. type in a GET HTTP request:

```
GET /kurose_ross/interactive/index.php HTTP/1.1
```

```
Host: gaia.cs.umass.edu
```

by typing this in (hit carriage
return twice), you send
this minimal (but complete)
GET request to HTTP server

3. look at response message sent by HTTP server!

(or use Wireshark to look at captured HTTP request/response)

Project I Demo

Chapter 2: outline

2.1 principles of network applications

- app architectures
- app requirements

2.2 Web and HTTP

- Overview
- Persistent vs non-persistent
- HTTP message formats
- **Web cookies**
- **Web Proxy**

User-server state: cookies

many Web sites use cookies

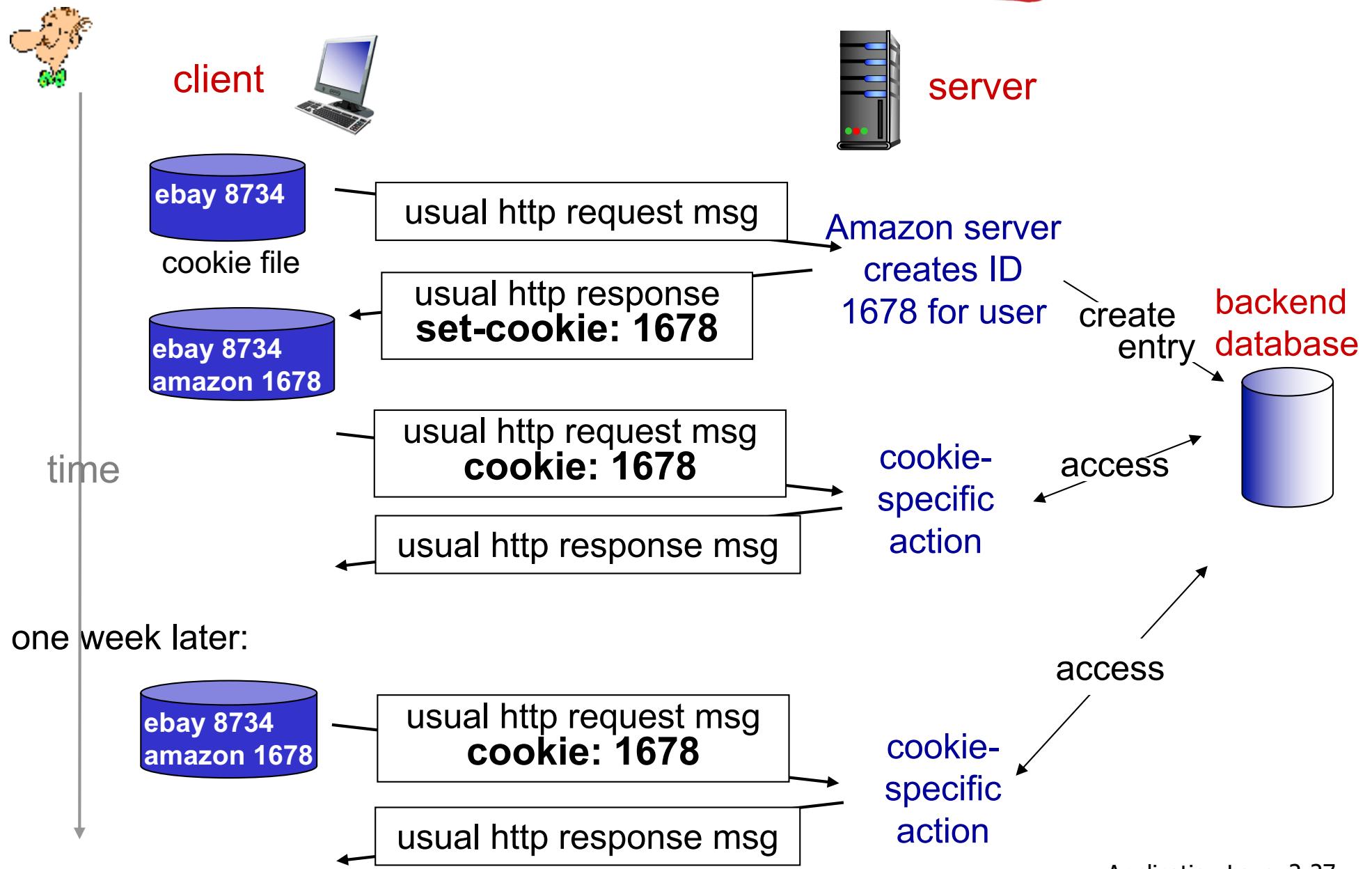
four components:

- 1) cookie header line of *HTTP response message*
- 2) cookie header line in next *HTTP request message*
- 3) **cookie file** kept on user's host, managed by user's browser
- 4) **back-end database** at Web site

example:

- ❖ Susan always access Internet from PC
- ❖ visits specific e-commerce site for first time
- ❖ when initial HTTP requests arrives at site, site creates:
 - unique ID
 - entry in backend database for ID

Cookies: keeping “state” (cont.)



Cookies (continued)



*what cookies can be used
for:*

- ❖ authorization
- ❖ shopping carts
- ❖ recommendations
- ❖ user session state (Web e-mail)

aside
cookies and privacy:

- ❖ cookies permit sites to learn a lot about you
- ❖ you may supply name and e-mail to sites

Chapter 2: outline

2.1 principles of network applications

- app architectures
- app requirements

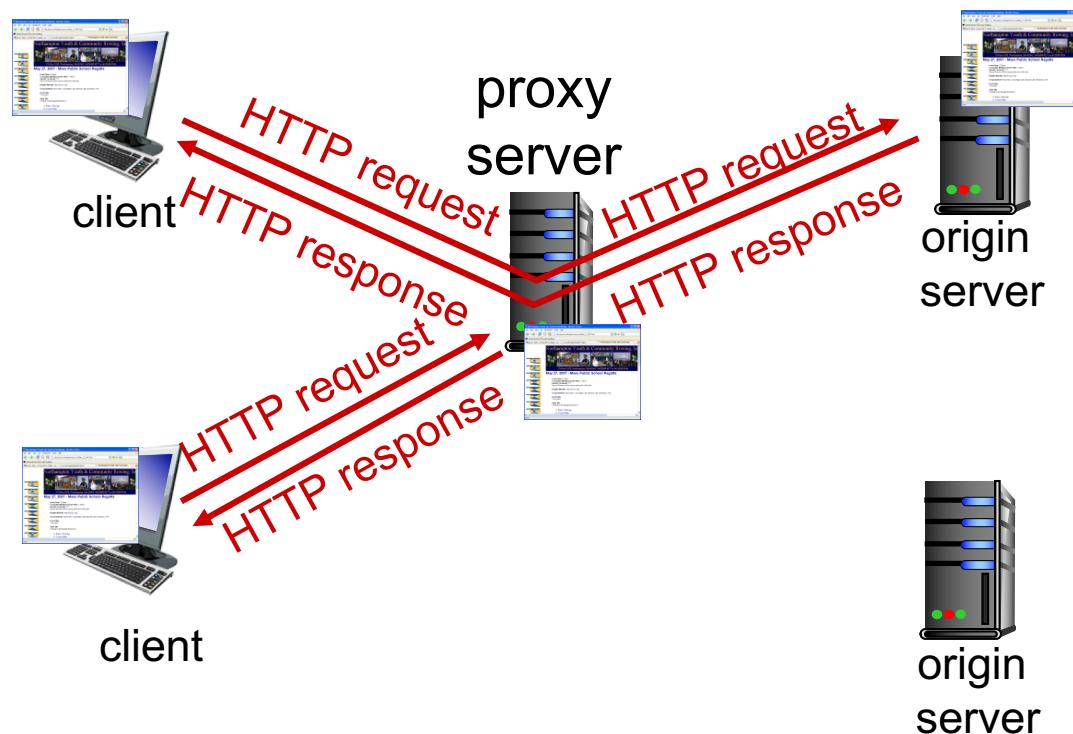
2.2 Web and HTTP

- Overview
- Persistent vs non-persistent
- HTTP message formats
- Web cookies
- **Web Proxy**

Web caches (proxy server)

goal: satisfy client request without involving origin server

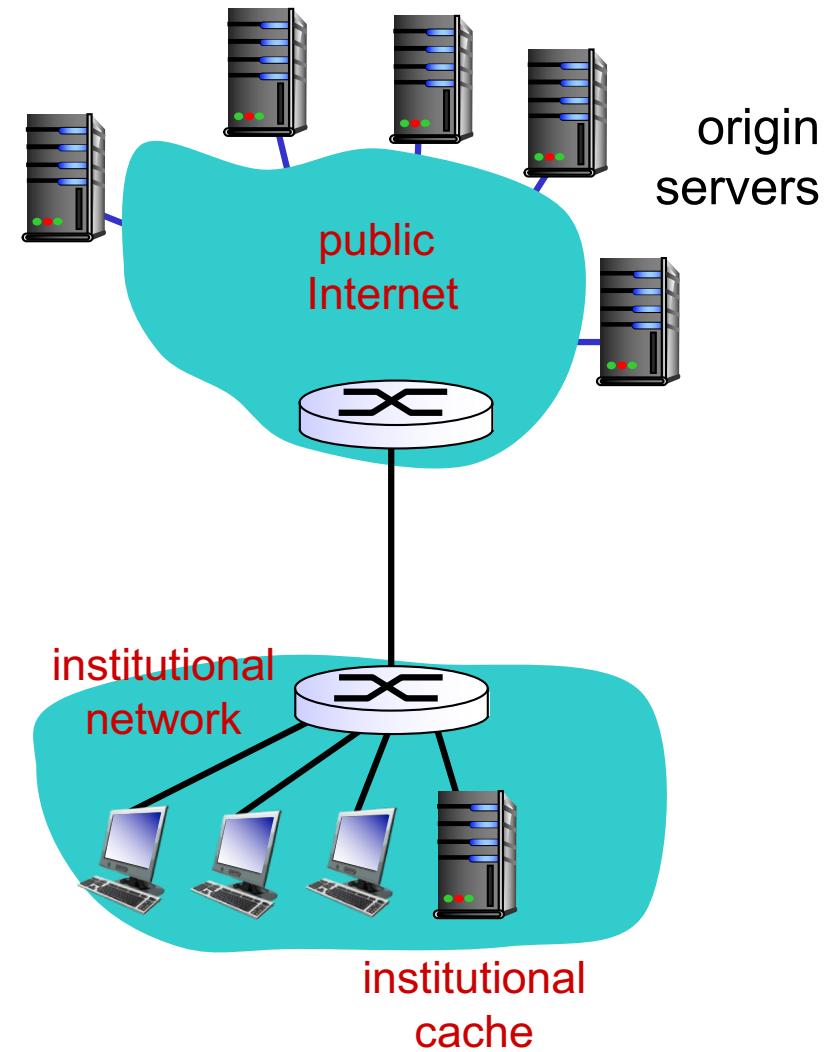
- ❖ user sets browser: Web accesses via cache
- ❖ browser sends all HTTP requests to cache
 - object in cache: cache returns object
 - Else: cache requests object from origin server, then returns object to client



Caching example:

- ❖ total delay = Internet delay + access delay + LAN delay

- ❖ cache acts as **both client and server**
 - server for original requesting client
 - client to origin server
- ❖ typically cache is **installed by access ISP** (university, company, residential ISP)

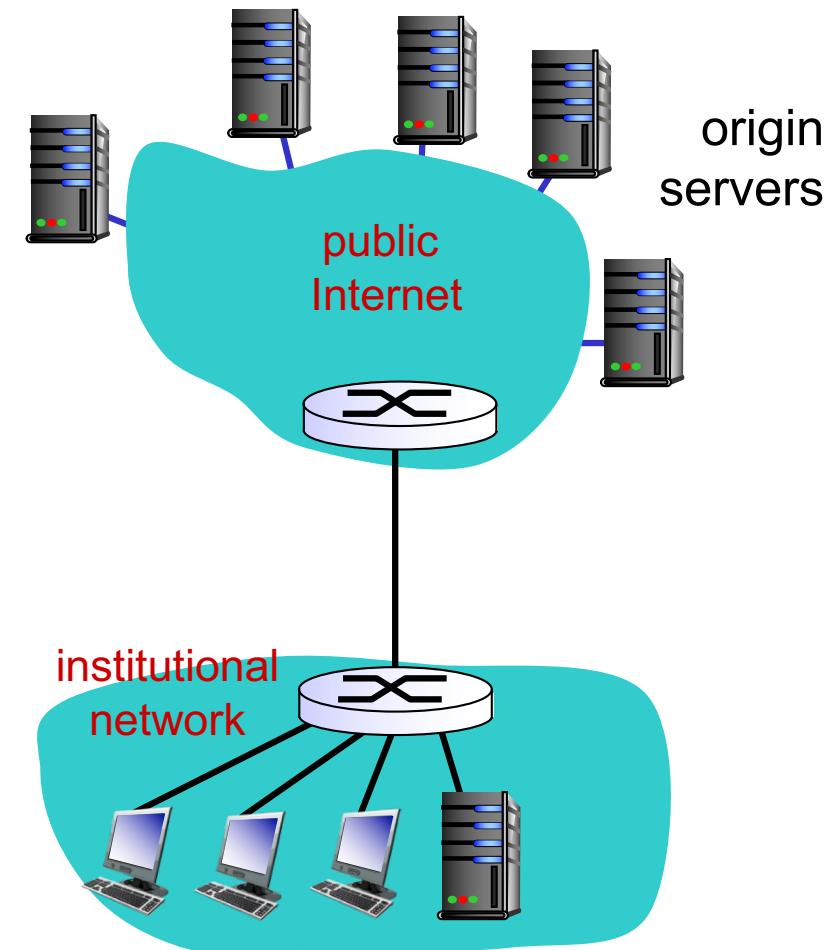


More about Web caching



why Web caching?

- ❖ **Client:** reduce response time for client request
- ❖ **Access:** reduce traffic on an institution's access link
- ❖ **Internet:** Internet dense with caches: enables “poor” content providers to effectively deliver content (so too does P2P file sharing)



Concepts

- ❖ HTTP: HyperText Transfer Protocol
- ❖ HTML: HyperText Makeup Language
- ❖ URL: Uniform Resource Locator
- ❖ Web browser = Web Client
- ❖ RTT: Round-Trip Time
- ❖ TCP Three-way handshake:

Quiz 3 this Friday

On HTTP protocol with a bonus question

Questions?