CSCE 463/612 Networks and Distributed Processing Spring 2024

Application Layer II

Dmitri Loguinov
Texas A&M University

February 7, 2024

Chapter 2: Roadmap

Web and HTTP

Terminology

- Web page consists of a base HTML-file that may include references to external objects
  - Examples of objects: JPEG image, Java applet, audio file, video stream, or flash animation
- Each object is addressable by a URL (Uniform Resource Locator) with the HTTP scheme

```
http://[user:pass@]host[:port][/path][?query][#fragment]
```

  - Username/password not used often anymore
  - Fragement specifies portion of HTML for browser to jump to
  - Query provides input arguments to scripts

HTTP Overview

- HTTP: HyperText Transfer Protocol
  - HTTP 1.0: RFC 1945 (1996)
  - HTTP 1.1: RFC 2068 (1997), RFC 2616 (1999)
  - HTTP 2: RFC 7540 (2015), binary protocol over TCP
  - HTTP 3: work in progress, QUIC over UDP
- Nonpersistent HTTP
  - At most one object is sent over a TCP connection
  - HTTP/1.0 must use nonpersistent HTTP
- Persistent HTTP
  - Multiple objects sent over single TCP connection
  - HTTP/1.1 uses persistent connections by default
  - Field “Connection: close” overrides this behavior

Nonpersistent HTTP

(contains text, references to 10 jpeg images)

Suppose user enters URL

`www.tamu.edu/someDepartment/home.html`

1a. Client initiates TCP connection to server process at www.tamu.edu using port 80

1b. Server at host www.tamu.edu waiting for TCP connection on port 80 accepts connection, notifies client

2. Client sends HTTP *request message* (containing URL) into TCP socket. Message indicates object /someDepartment/home.html

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

time

Nonpersistent HTTP (Cont.)

4. Server closes TCP connection

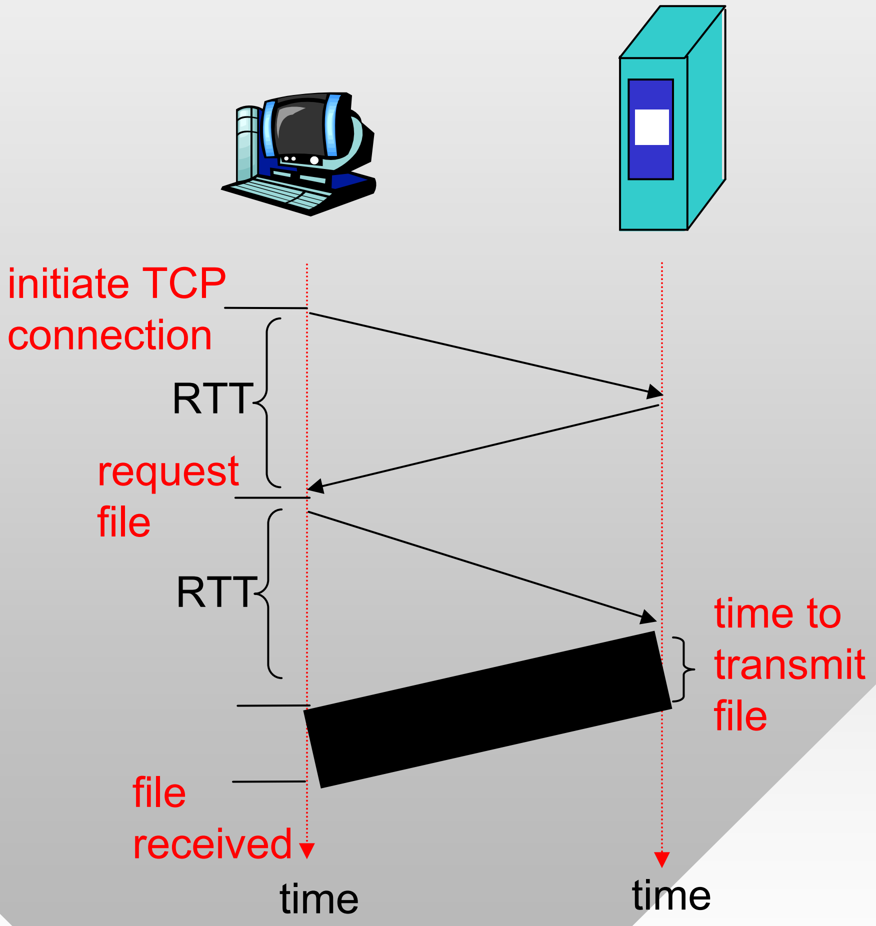
5. Client receives response message containing the html file, displays html. Parsing html file, finds 10 referenced jpeg objects

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

Response Time Modeling

- RTT (Round-Trip Time):
  - Delay for a small packet to travel from client to server and back
- 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 + file load time



Persistent HTTP

HTTP/2 allows out-of-order replies, fragmentation of objects, and prioritization

Nonpersistent HTTP issues:

- Requires two RTTs per object
- Workaround: browsers open parallel TCP connections to fetch referenced objects
- OS must work and allocate host resources for each TCP connection

Persistent HTTP

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

Persistent without pipelining:

- Client issues new request only when previous response has been received
- One RTT for each referenced object + its transmission time

Persistent with pipelining:

- Default in HTTP/1.1
- Client sends requests as soon as it encounters a referenced object
- One RTT for all referenced objects + their transmission times

HTTP Request Message

- Two types of HTTP messages: *request, response*
- HTTP request message:
  - 1.0 and 1.1 use ASCII (human-readable format)

request line (GET, POST, HEAD commands)

header lines

Carriage return, line feed indicates end of header

```
GET /somedir/page.html HTTP/1.1
Host: www.someschool.edu
User-agent: Mozilla/4.0
Connection: close
Accept-language: fr

 Message body (optional)
```

Uploading Form Input

POST method:

- Web page often includes form input
- Input is uploaded to server in message body
- Used for large amounts of data
  - Data is coded using tuples “field=value”, where + stands for space and & for the field separator

```
POST /map.cgi HTTP/1.0
User-Agent: HTTPTool/1.0
Content-Type: application/x-www-form-urlencoded
Content-Length: 30

city=College+Station&zip=77843
```

Uploading Form Input (Cont'd)

URL method:

- Uses the GET command
- Input is encoded in the URL field of request line
  - Append ? to the script path, followed by the URL-coded data
  - GET /path/script.cgi?field1=value1&field2=value2 HTTP/1.0
- For the previous example
  - GET /map.cgi?city=College+Station&zip=77843 HTTP/1.0
- Google example
  - Javascript forces the URL method:
  - www.google.com/search?hl=en&source=hp&q=computer+science&aq=f&aqi=g10&oq=

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 to path specified in URL field
- DELETE
  - Deletes file specified in the URL field

HTTP Response Message

status line (protocol status code status phrase)

```
HTTP/1.1 200 OK
Connection: close
Date: Wed, 07 Feb 2024 12:00:15 GMT
Server: Apache/1.3.0 (Unix)
Last-Modified: Mon, 01 May 2023 ...
Content-Length: 6821
Content-Type: text/html

Message body (optional)
```

header lines

data, e.g., requested HTML file

HTTP Response Status Codes

- Status code is always in the first line of response
  - Followed by a nice textual explanation
- 200 OK
  - Request succeeded, requested object later in this message
- 301 Moved Permanently
  - Requested object moved, new location specified later in this message (see field 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

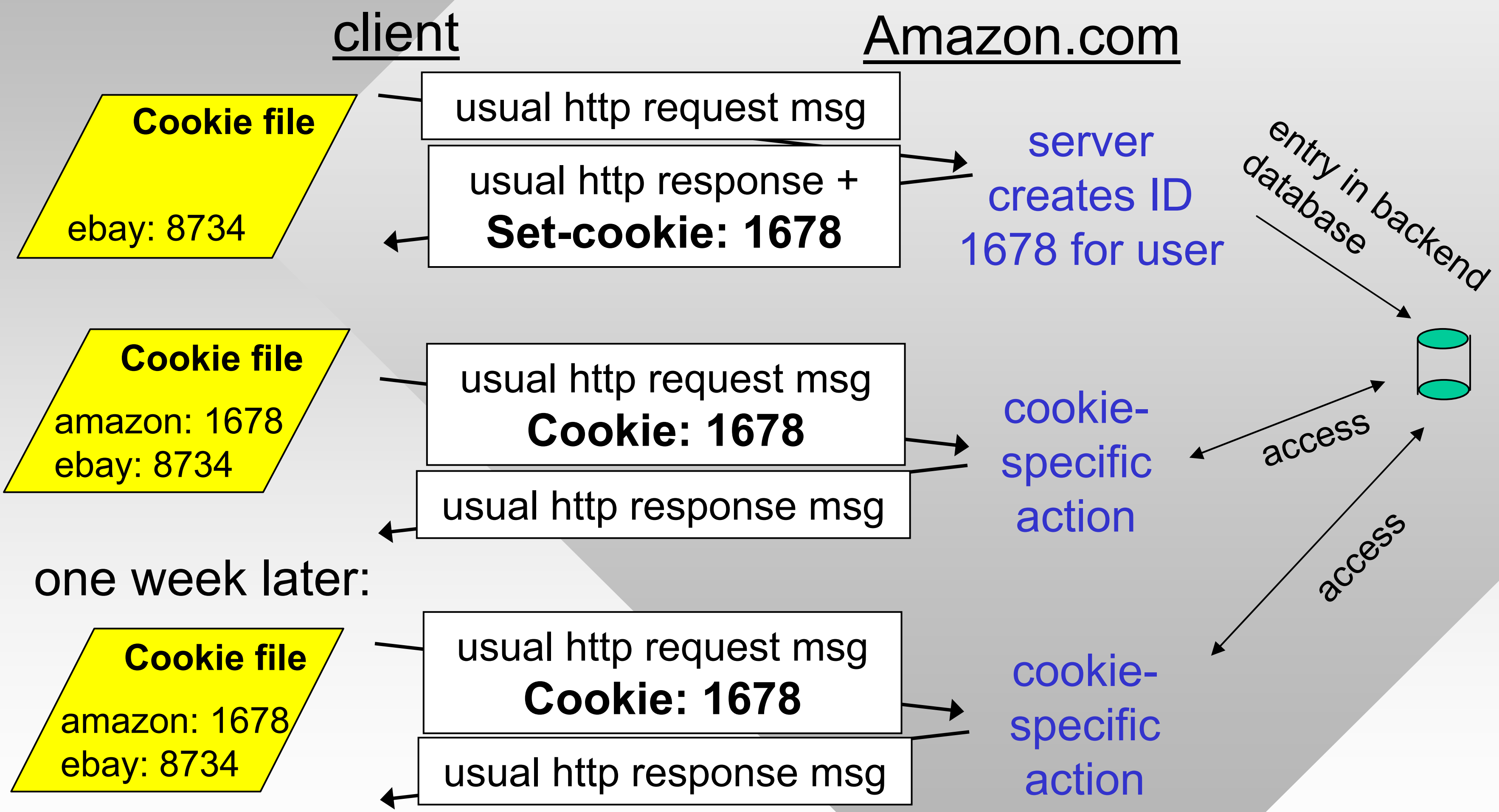
User-Server State: Cookies

- User visits the same web site multiple times
  - Doesn't want to type password or make selections each time
- Website remembers info about the user
  - Amazon shopping cart
  - Pages viewed, items bought, credit cards used
  - Zip code and cable channels (tvguide.com)
  - Weather.com (zip)

Four components:

- Cookie header line in the HTTP response message
- Cookie file kept on user's host and managed by user's browser
- Cookie header line in HTTP request message
- Back-end database at website

Cookies: Keeping State



Cookie Example

```
telnet irl.cs.tamu.edu 80
GET / HTTP/1.0

HTTP/1.1 200 OK
Connection: close
Date: Wed, 02 Feb 2024 18:47:25 GMT
Server: Microsoft-IIS/10.0
MicrosoftOfficeWebServer: 5.0_Pub
X-Powered-By: ASP.NET
Content-Length: 6916
Content-Type: text/html
Set-Cookie: ASPSESSIONIDACSRQCTQ=PIGHLBAAJICJONABJFINMLOA;
  path=/
Cache-control: private
```

Non-persistent cookies expire when browser is closed; *persistent* ones are preserved until a future expiration time ("Expires=" attribute); if multiple cookies provided, each has its own *Set-Cookie* line

path prefix where cookie is valid

cookie value

shared caching not allowed

Cookies (continued)

- Cookie file location is browser-dependent
  - For example, Internet Explorer: C:\Users\<*user*>\AppData\Roaming\Microsoft\Windows\Cookies
  - Impersonation is possible by copying or intercepting user cookies (through sniffing and malicious scripting)
- Other privacy issues
  - Websites accumulate data about users (form input, actions), share this information with others
  - So-called third-party (tracking) cookies
- Incognito browsing mode starts with no cookies
  - New cookies are accepted and kept until browser is closed