CPSC 317 COMPUTER NETWORKING

Module 3: Application Layer Protocols - Day 2 – The Web

READING

- Reading: 2.2

ADMINISTRATION

- Programming Assignment 1 is ongoing
- Quiz 1 starts January 29th
- There are iClicker questions today!

LEARNING GOALS - HTTP

- Describe the web application and the structure of web objects
- Describe the HTTP protocol and format of request/response messages
- Identify common HTTP methods and response codes
- Describe the benefits and basic functionality of HTTP non-persistent connections, persistent connections, and HTTP pipelining
- Explain what HTTP cookies are and what they can be used for
- Describe how Web caches can be used to reduce HTTP data transfer

APPLICATION: THE WORLD WIDE WEB

- One of the first Internet applications that became popular
- Information available on the web in the form of a “web page”
- A web page consists of objects
  - A base HTML file refers to other referenced objects
  - Images, scripts, stylesheets, frames
- Each object (including base file) is addressable by a URL
  - Example: www.students.cs.ubc.ca/~cs-317/2023W2/web/index.html

HYPERTEXT TRANSFER PROTOCOL (HTTP)

- HTTP is the Web's main application layer protocol
- Client/server model
  - Client: browser that requests, receives, displays Web objects
  - Server: sends objects in response to requests
  - uses TCP, typically port 80 (443 for HTTPS)
- For each Web object (URL)
  - Client sends one request message at once
  - Server sends full response message at once
- Stateless: server maintains no information about past requests

HTTP MESSAGE FORMAT

- Request is formatted using text in ASCII
  - First line: method, URL, version
  - Following lines use format “Header-Name: value”
  - Lines end with CR-LF
  - Empty line ends request header
  - If body needed (e.g., form values), it follows the header
    - Size determined by “Content-Length” header
- Response follows same rules, except:
  - First line: version, response code, response text

HTTP MESSAGE EXAMPLE

Request:

```
GET /index.html HTTP/1.1
Host: www.example.com
User-Agent: Firefox/3.6.10
…
Connection: keep-alive
```

Response:

```
HTTP/1.1 200 OK
Date: Sun, 13 Nov 2017 20:09:20 GMT
Content-Length: 2652
…
Content-Type: text/html

<HTML>…
```

REQUEST METHODS

- GET method:
  - Relevant data is in URL
  - Form data, if needed, found in URL itself
  - Often used in forms that search for data (though this varies)
- POST method:
  - Includes form input in message body
  - Often used in forms that submit new data
- HEAD method:
  - Similar to GET, but returns only header
  - Used to check if existing content was modified

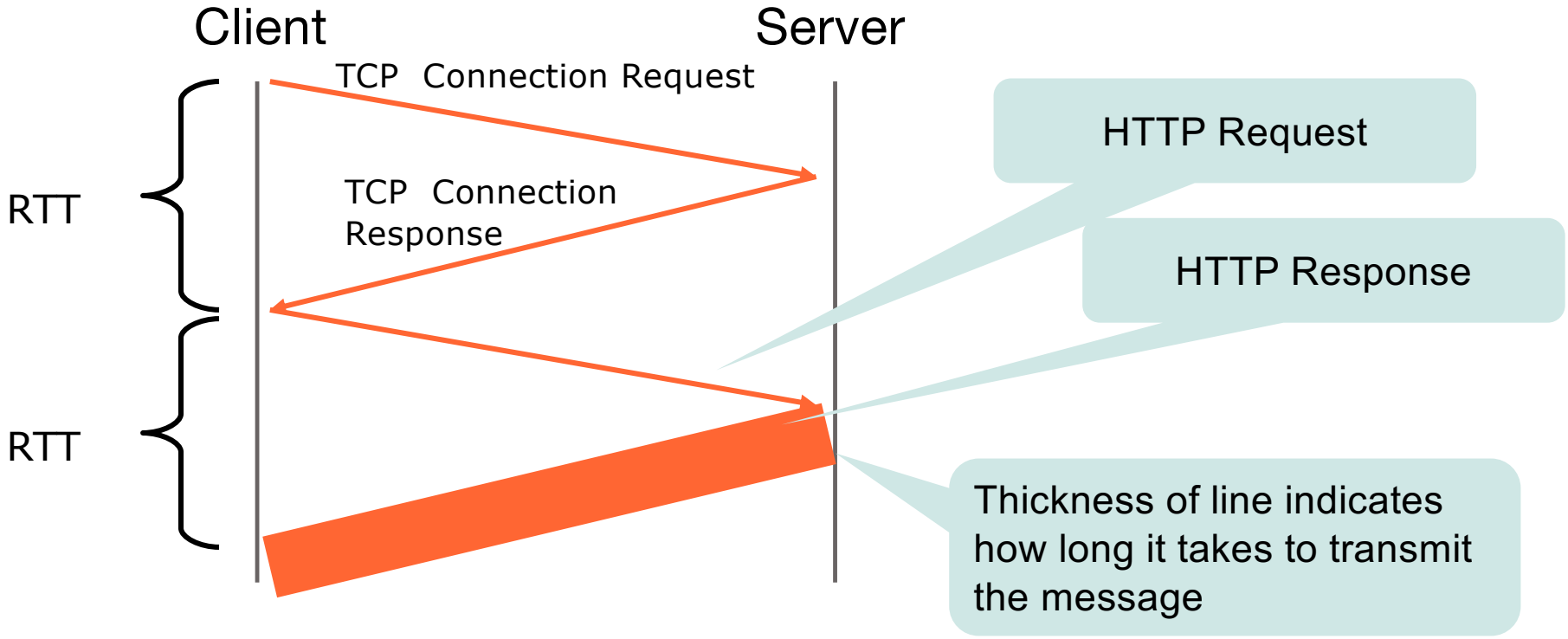
HTTP RESPONSE STATUS CODES

- 2xx: success
  - Example: 200 OK
- 3xx: additional action required
  - Example: 301 Moved (redirection), 304 Not Modified (use cache)
- 4xx: client problem
  - Example: 400 Bad Request, 404 Not Found
- 5xx: server problem
  - Example: 500 Bad Gateway, 505 HTTP Version not supported

HTTP CONNECTIONS (I)

- Non-persistent HTTP (version 1.0)
  - At most one object is sent over each TCP connection (one request, one response)
  - Connection is closed as soon as data is transferred
  - Web page with multiple objects requires multiple connections

REQUEST / RESPONSE DIAGRAM



CLICKER QUESTION

Suppose the web page we are interested in refers to another "object" that we need to fetch as well. How many round trips will it take to retrieve the web page and this additional object?

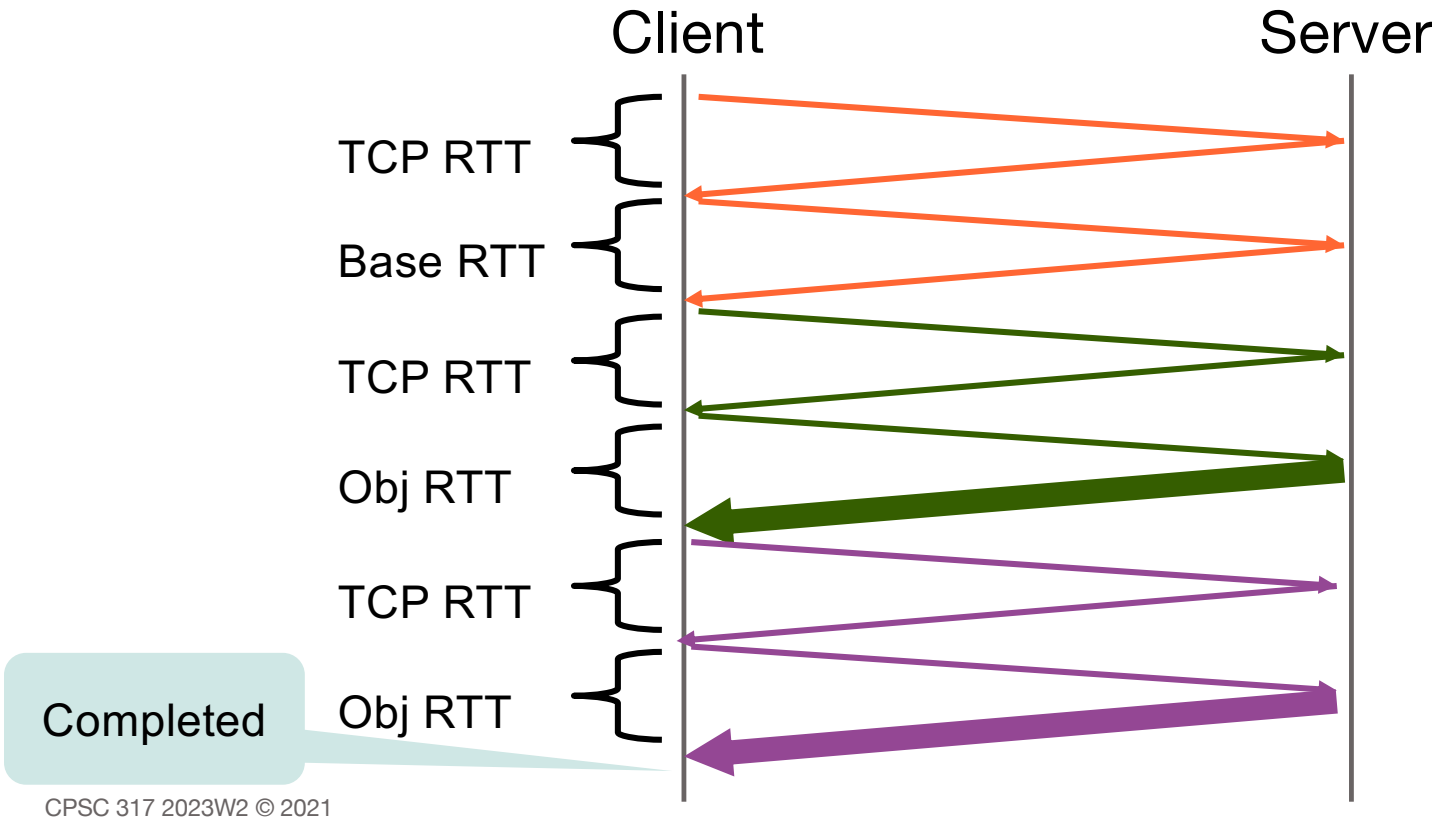
A. 1

B. 2

C. 3

D. 4

REQUESTING MULTIPLE OBJECTS



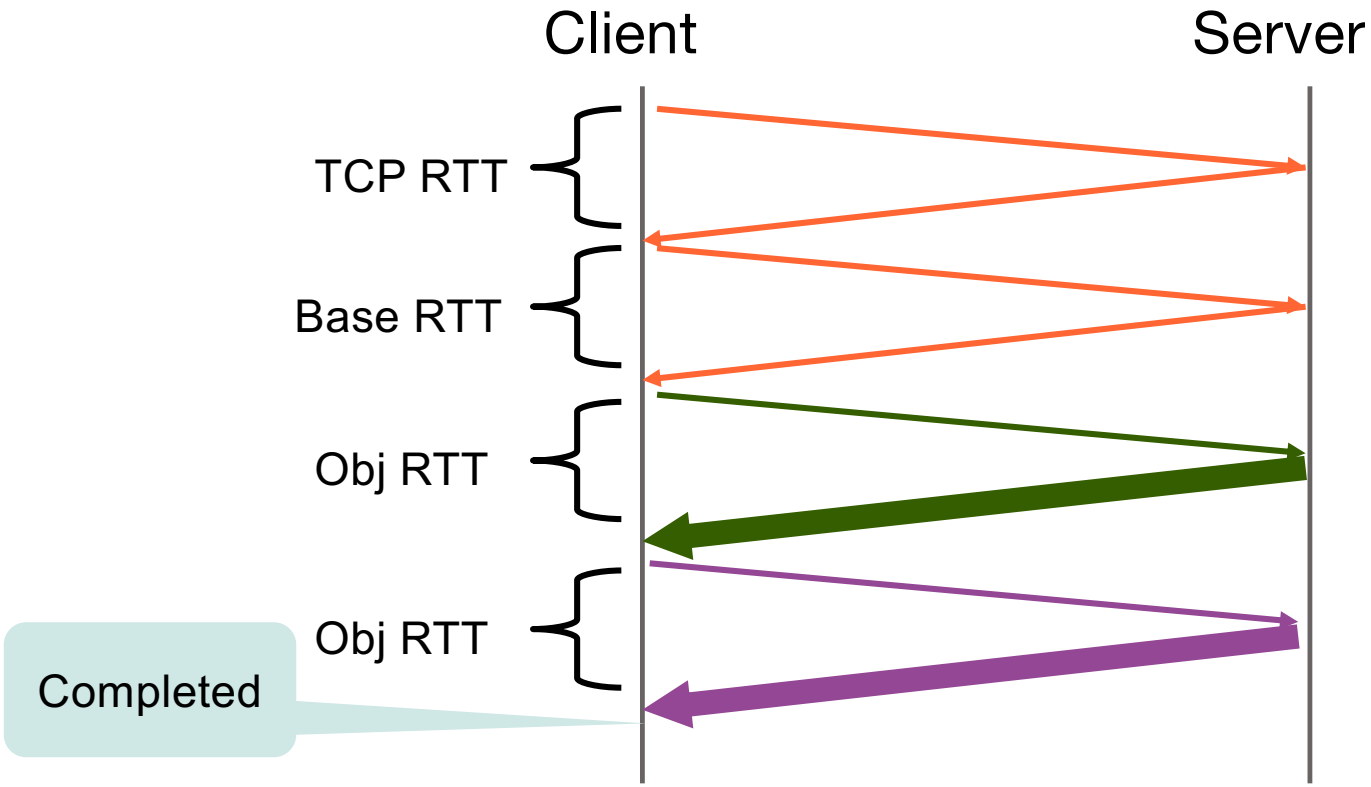
RESPONSE TIME CONSIDERATION

- Imagine a Web page with a main html page and 10 images
- Non-persistent connection:
  - For each object:
    - One RTT to initiate connection
    - One RTT (plus transmission) to request each object
  - 2 RTTs per object: 22 RTTs to retrieve page (plus transmission)
  - Can be improved if several connections are opened in parallel

HTTP CONNECTIONS (II)

- Persistent HTTP (version 1.1)
  - Multiple requests can be sent over single connection
  - Responses are received in the same order as the requests

REQUEST / RESPONSE DIAGRAM



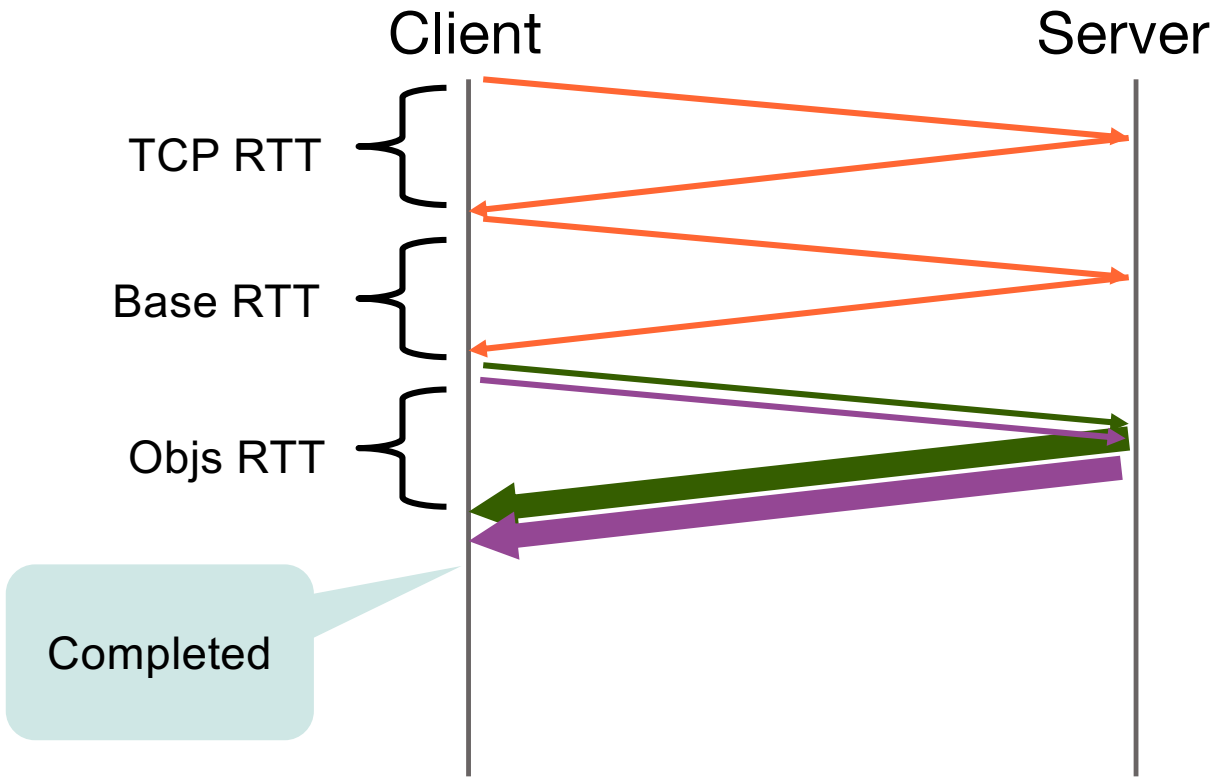
RESPONSE TIME CONSIDERATION (AGAIN)

- Imagine a Web page with a main html page and 10 images
- Persistent connection:
  - RTT to initiate connection only once (assuming all objects in same server)
  - One RTT (plus transmission) to request each object
  - 1 RTT for connection plus 1 RTT per object: 12 RTTs total (plus transmission)

HTTP CONNECTIONS (III)

- Pipelining:
  - Assumes persistent HTTP (version 1.1)
  - Client may send several requests without waiting for response
  - Responses are received in the same order as the requests

HTTP CONNECTIONS - PIPELINING



RESPONSE TIME CONSIDERATION (AGAIN[2])

- Imagine a Web page with a main html page and 10 images
- Persistent connection with pipelining:
  - 1 RTT for connection
  - 1 RTT for first request
  - 1 RTT for all the images (plus transmission)
- Works well only when all requests to same server

COOKIES: PER-USER STATE

- Web sites may use cookies to maintain state
  - "Remember me" authentication
  - Session state
  - Shopping carts, recommendations
- Process:
  - Response includes Set-Cookie header
  - Browser saves cookie associated with the server
  - Next request to the same server includes Cookie header with the same value

TWO COOKIE STYLES

- All the state is in the cookie
  - Cookie has to be one line, and there are limits
    - Total header size limited by servers: ~8Kbytes
    - Individual cookie size limited by browsers: ~4Kbytes
- State (or part of it) may be stored server-side
  - Cookie is used to identify entry in server database
  - Cookie can be just an email address or user identifier

WEB CACHE

- Browser may maintain a cached version of page
  - Reduces network utilization
  - Reduces page load time
- Cache may also be maintained in a separate host (proxy)
  - Browser sends request to proxy, which redirects to server
  - If proxy has page cached, return immediately
  - Typically closer to user than the server (company, ISP)
- Web servers can provide cache policy, like:
  - How long a page should be cached for
  - If cache is private (specific to user) or shared
- Conditional GET: request page, but only if modified

CLICKER QUESTION

Suppose a user is accessing a web page from a server where the RTT is 0.2s. If obtaining the web page takes 10 round trips, how much faster would it be to access the web page through a cache whose RTT is 0.01s and caches all of the objects?

A. 20 times faster
B. 10 times faster
C. 4 times faster
D. 2 times faster

CLICKER QUESTION

Suppose a user is accessing a web page from a server where the RTT is 0.2s. If obtaining the web page takes 10 round trips, how much faster would it be to access the web page through a cache whose RTT is 0.01s and caches 80% of the objects?

A. 20 times faster
B. 10 times faster
C. 4 times faster
D. 2 times faster

CACHE VALIDATION EXAMPLE

Request:

GET /labs/algorithms/img/faculty/PatriceBelleville.jfif HTTP/1.1

Host: www.cs.ubc.ca

User-Agent: aastha

CACHE VALIDATION EXAMPLE

Response:

HTTP/1.1 200 OK
Date: Mon, 22 Jan 2024 12:35:16 GMT
Server: Apache/2.4.51 (Linux/SUSE)
Last-Modified: Thu, 10 Sep 2020 04:48:40 GMT
ETag: "6fa8-5aeee482e5a00"
Accept-Ranges: bytes
Content-Length: 28584
Vary: Accept-Encoding
Content-Type: image/jpeg

Following this is 28584 bytes of JPEG data

CACHE VALIDATION EXAMPLE

Request:

GET /labs/algorithms/img/faculty/PatriceBelleville.jfif HTTP/1.1

Host: www.cs.ubc.ca

User-Agent: aastha

If-Modified-Since: Thu, 10 Sep 2020 04:48:40 GMT

CACHE VALIDATION EXAMPLE

Response:

HTTP/1.1 304 Not Modified

Date: Mon, 22 Jan 2024 12:45:23 GMT

Server: Apache/2.4.51 (Linux/SUSE)

ETag: "6fa8-5aeee482e5a00"

IN-CLASS ACTIVITY

- ICA32
  - Timing of various versions of HTTP
  - Effectiveness of caching