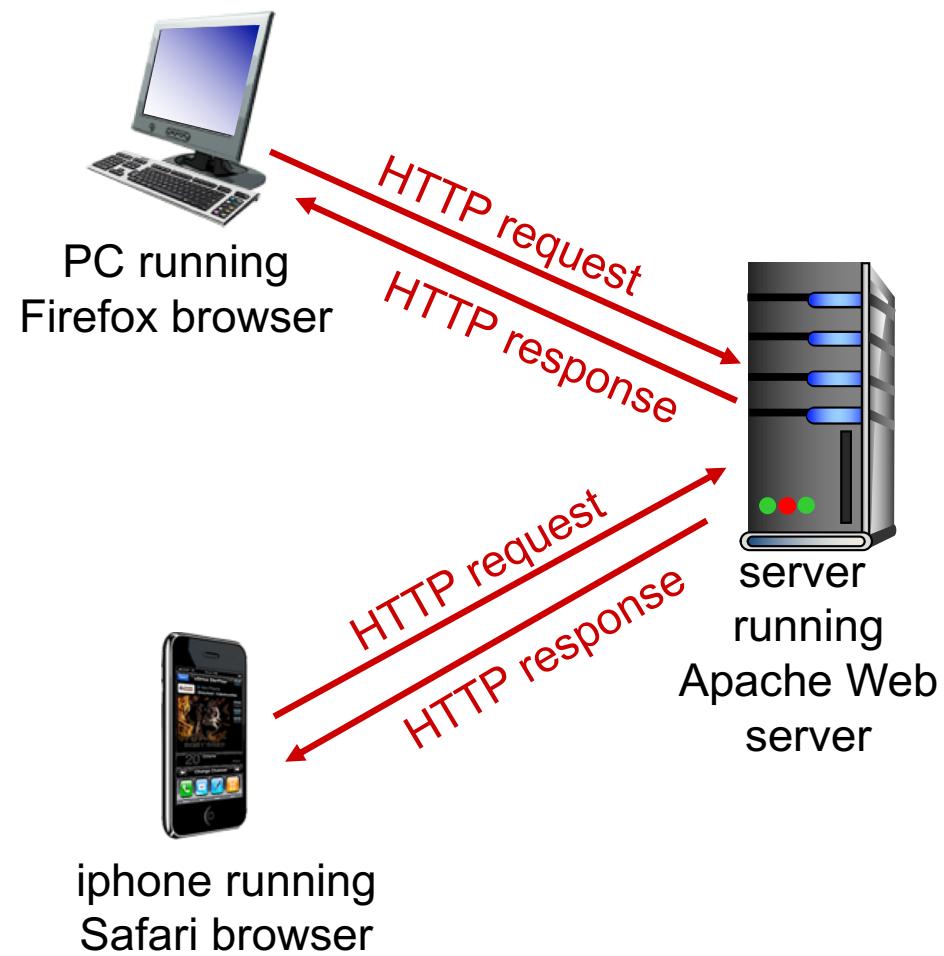


HTTP: Hyper Text Transport Protocol

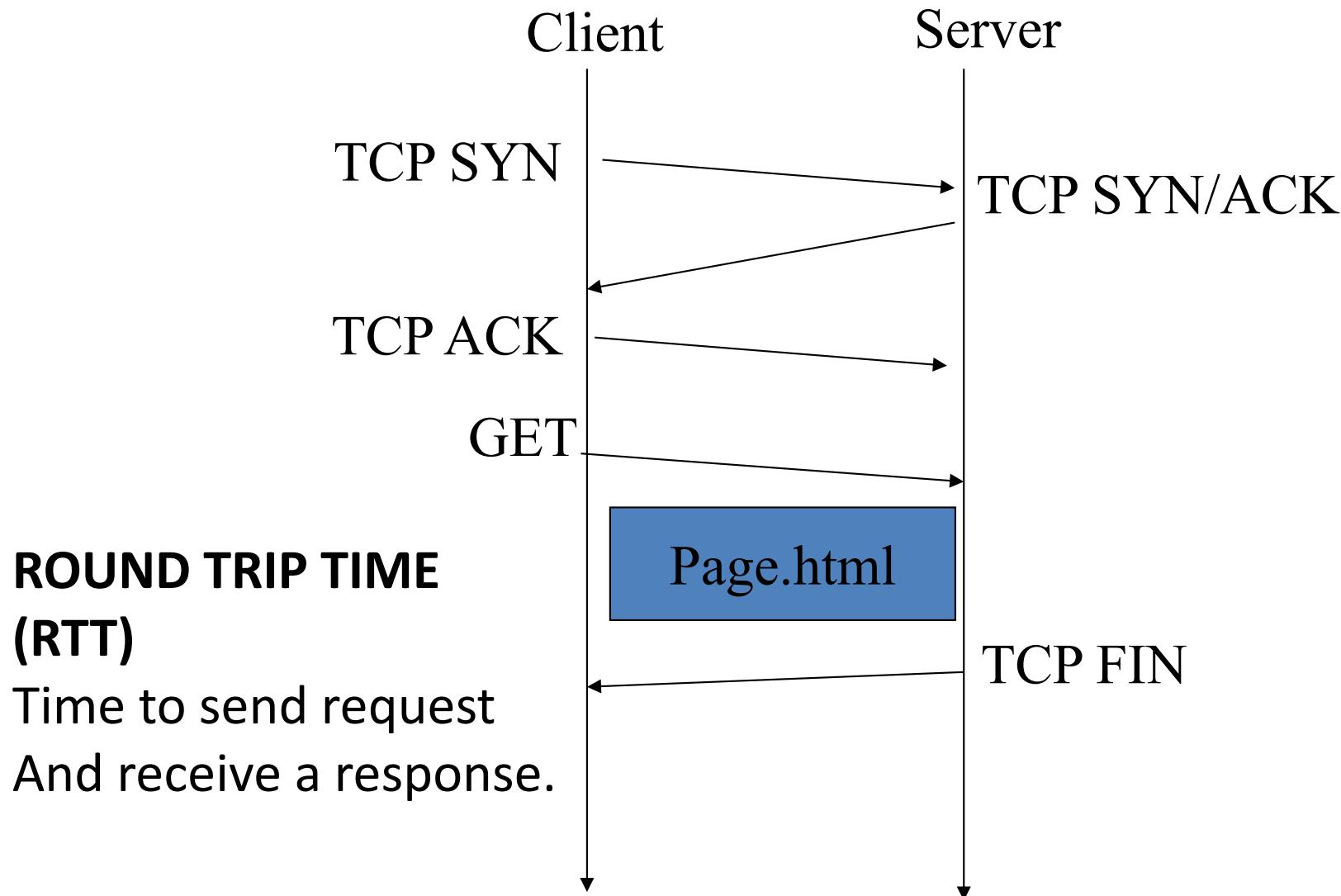
HTTP overview

HTTP: hypertext transfer protocol

- Web's application layer protocol
- client/server model
 - *client*: browser issues GET requests (using HTTP protocol)
 - *server*: Web server sends (using HTTP protocol) objects in response to GET requests



HTTP Request/Response



Response time estimation

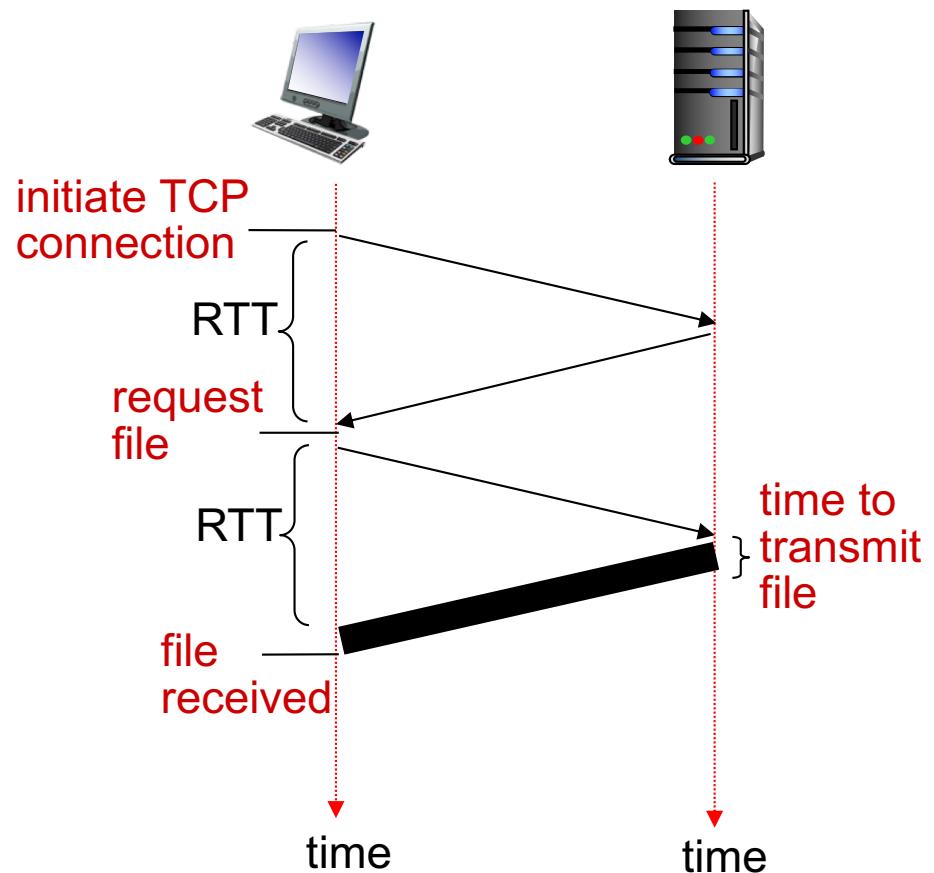
RTT: time for a packet to travel from client to server and back

HTTP response time made up of

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

Example HTTP response time =

$$2\text{RTT} + \text{file transmission time}$$



Lets look at HTTP from the point of view of the Web (WWW)

From Wikipedia: “The World Wide Web is an open source information space where documents and other Web resources are identified by URLs interlinked by hypertext links, and can be accessed from the Internet”.

How does the Web work?

- The Web consists of a set HTML and several links embedded within.
- The browser first obtains the HTML from the server. It then parses the HTML and fetches each embedded link
- To fetch each content (the main HTML and the subsequent embedded content), the Web uses the DNS and then the HTTP protocol

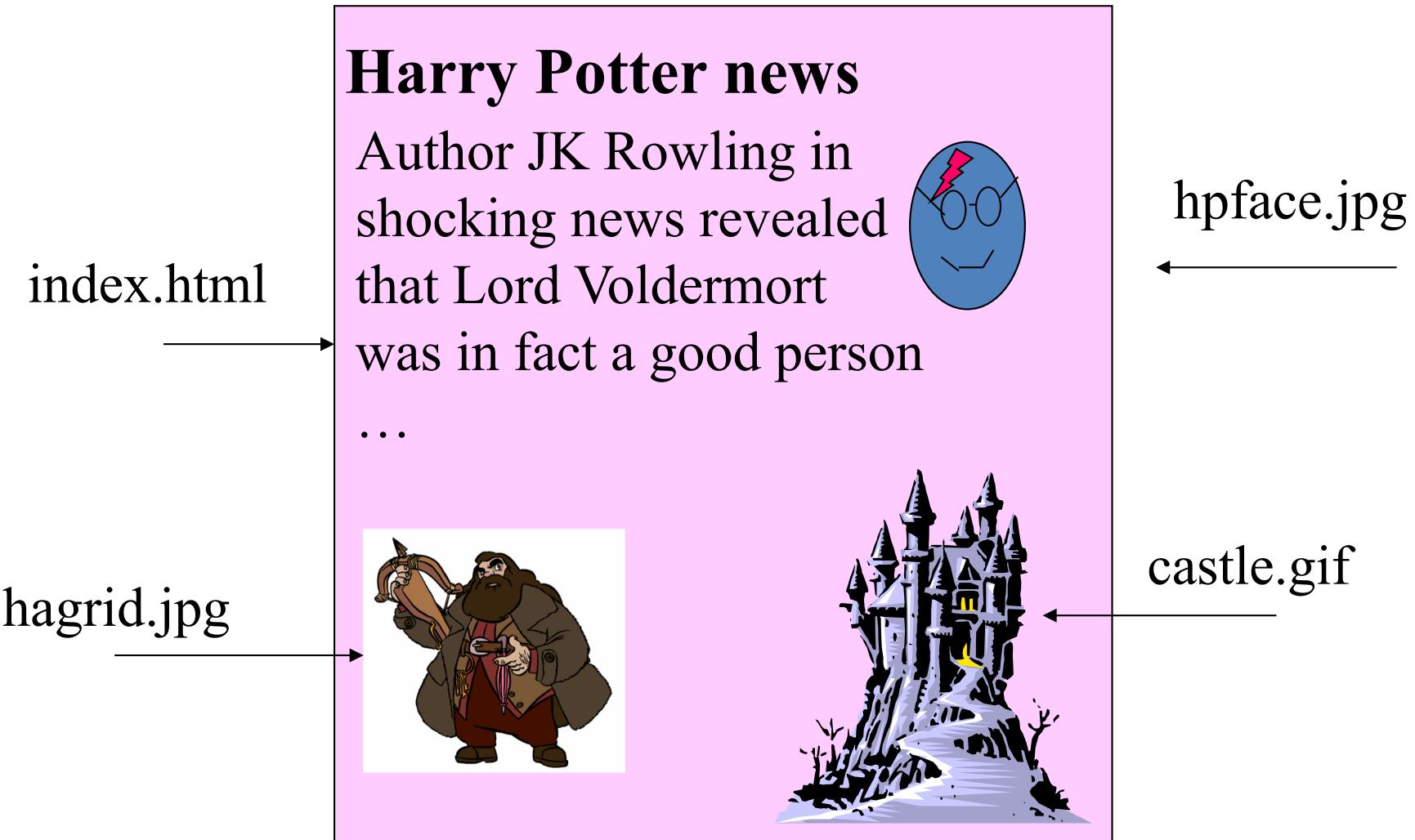
Example Web Page

- www.foo.com

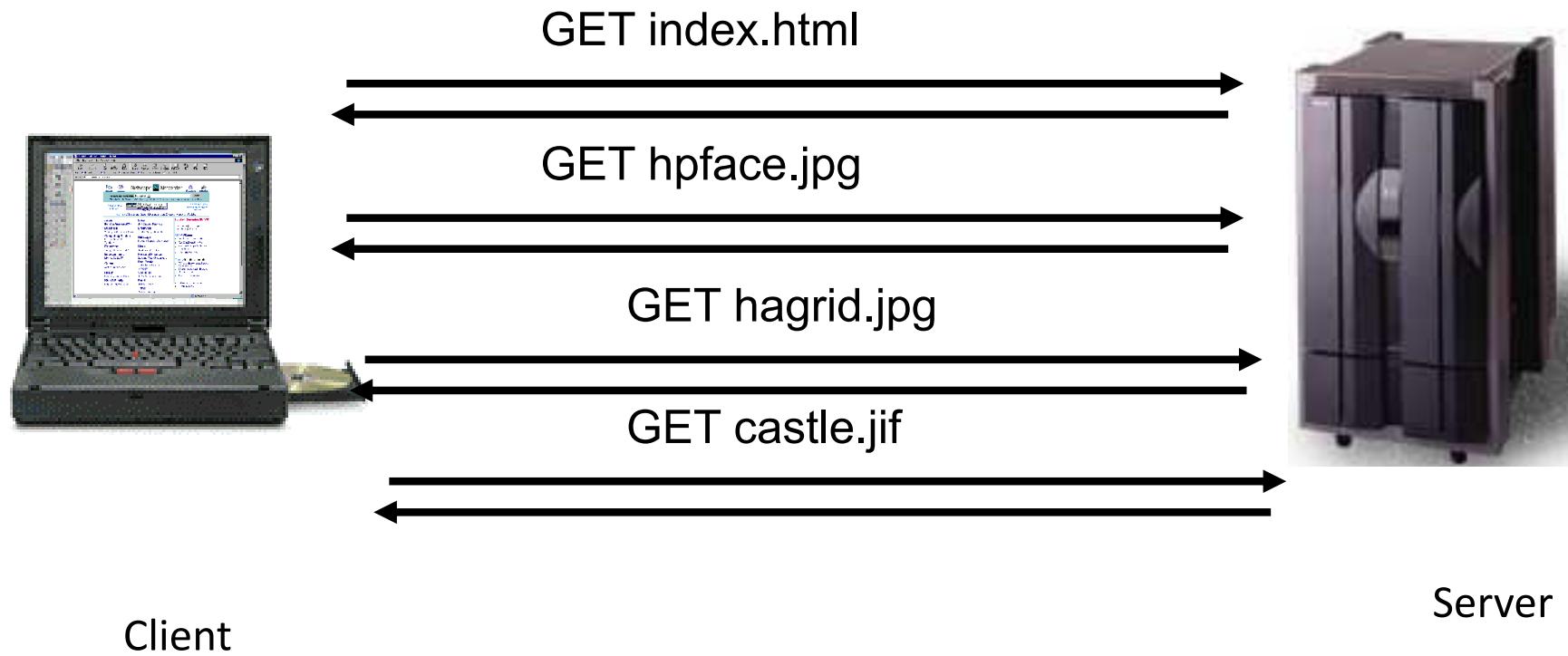
```
<html>
  
  
  
</html>
```

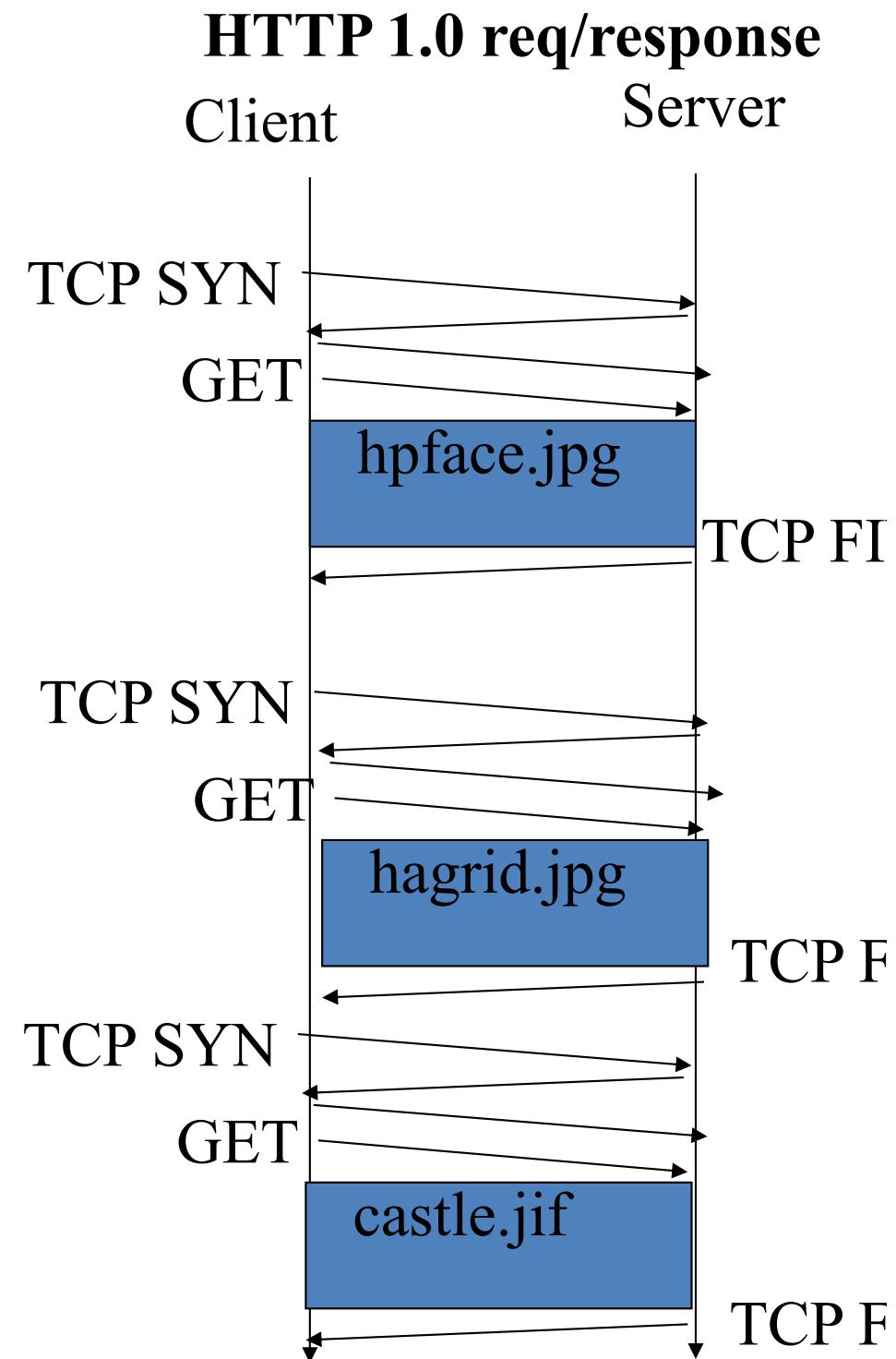
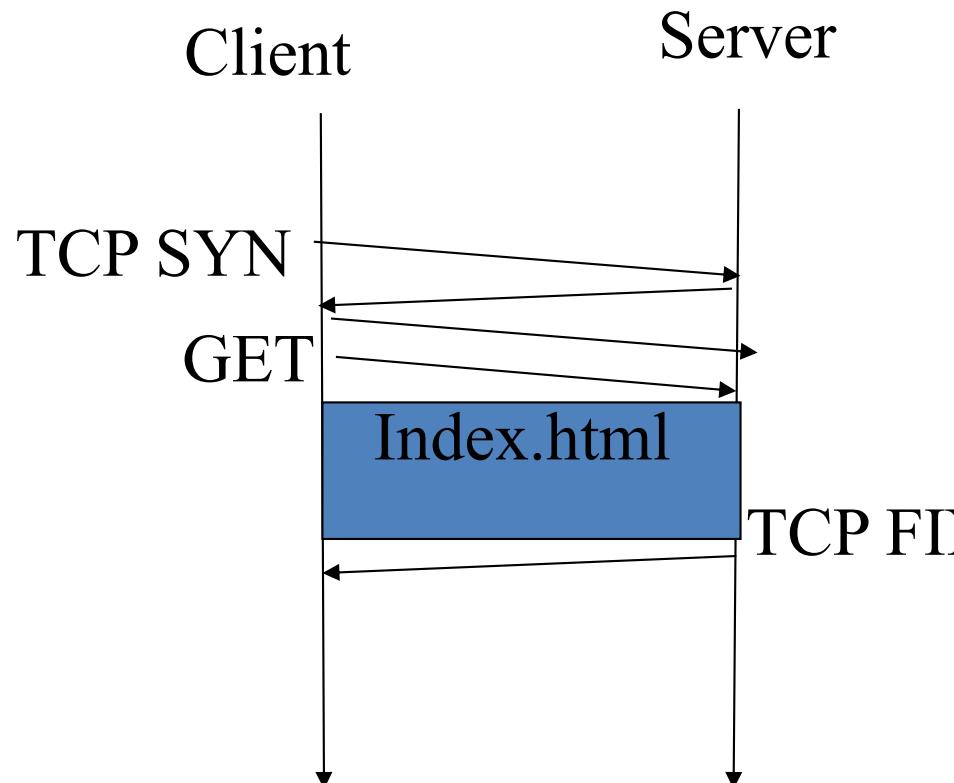
- When you type www.foo.com,
 - Fetch foo.html and start parsing
 - Fetch foo.jpg
 - Fetch bar.jpg
 - Fetch foobar.jpg

Example Web Page



Simple HTTP 1.0 (Non persistent HTTP)





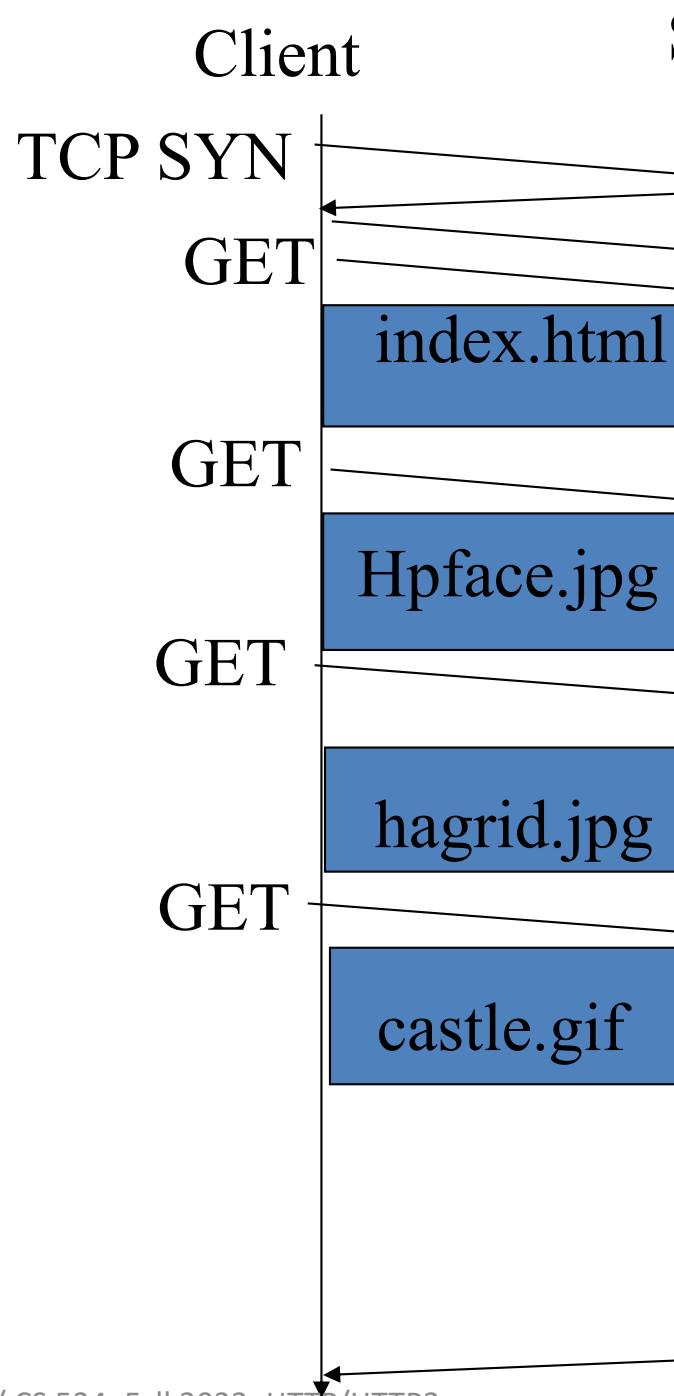
Non Persistent HTTP response time

- requires 2 RTTs + file transfer time per object
- OS overhead for *each* TCP connection

HTTP 1.1 builds on top of HTTP 1.0

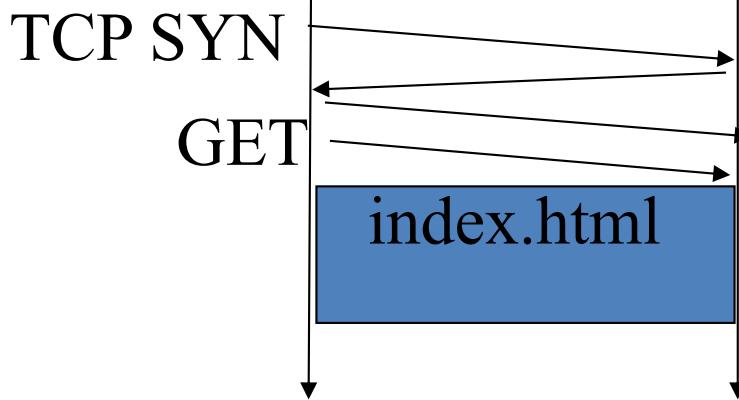
- 3 Ps
 - Persistent
 - Parallelization
 - Pipeling

Persistent HTTP



What is the time taken to transfer all the objects?

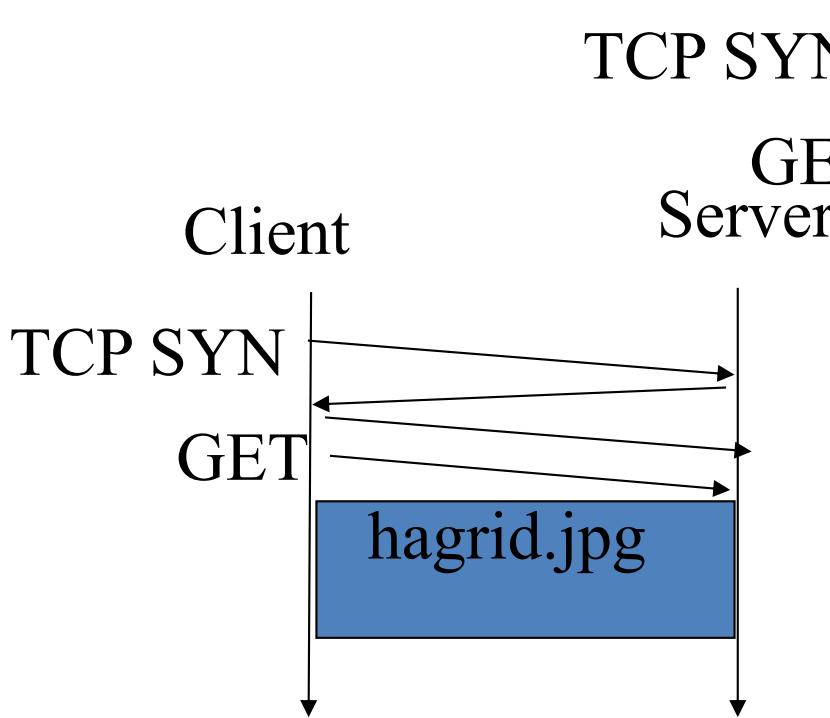
Client Server



Parallel HTTP

Browsers only allow
6 parallel connections per server*

Client Server



Client Server

TCP SYN

GET

Server

hpface.jpg

Client

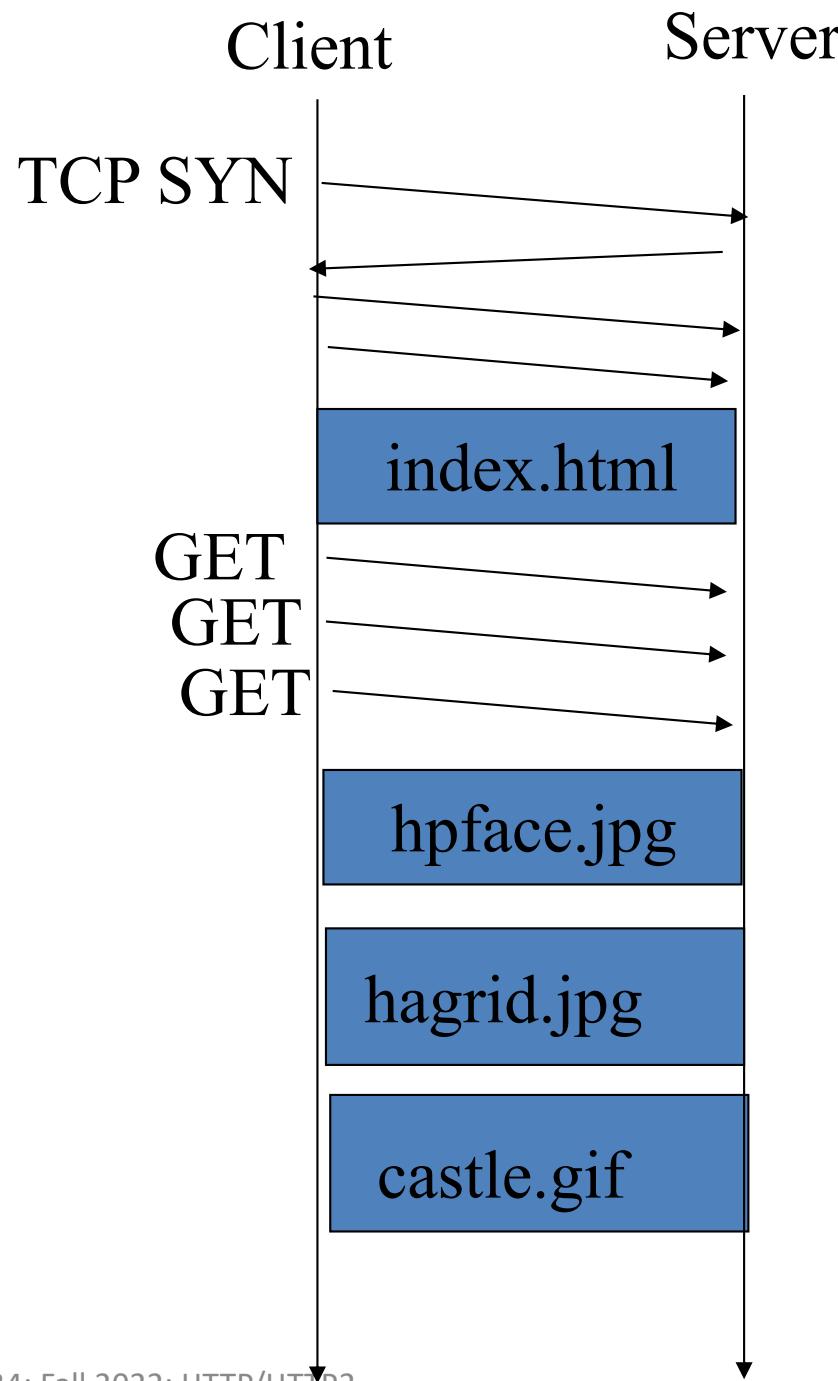
Server

TCP SYN

GET

castle.jif

Pipelined HTTP



What is total time taken?

In practice,

- HTTP 1.1 uses
 - Persistent
 - Parallelization
 - But not pipelining (Why?)

Head-of-line blocking

- In class.

HTTP header formats

HTTP request message

request line
(GET, POST,
HEAD commands)

```
GET /index.html HTTP/1.1\r\n
Host: www.cs.stonybrook.edu\r\n
User-Agent: Firefox/3.6.10\r\n
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
\r\n
```

header
lines

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

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 Sept 2015 20:09:20 GMT\r\nServer: Apache/2.0.52 (CentOS) \r\nLast-Modified: Tue, 20 Sept 2015 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\nCache-control: max-age=0, no-cache, no-store\r\n\r\ndata data data data data ...
```

The Keep-Alive timeout of the server takes preference over the timeout of the client.

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

304 Not modified

- request not modified, use the cached copy

400 Bad Request

- request msg not understood by server

404 Not Found

- requested document not found on this server

505 HTTP Version Not Supported

HTTP2/SPDY

- Fundamental question: Is parallelization better or pipelining better?
 - Depends on number of objects and size of objects
- If pipelining is better in some cases, how to avoid head-of-link block?

Answer: HTTP/2

cat - Google Search

https://www.google.com/search?q=web+page&es_sm=91&source=lnms&tbo=isch&sa=X&ei=n5...

Google cat

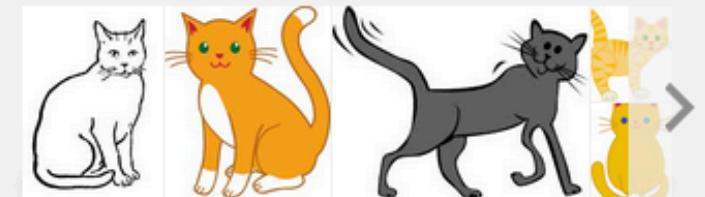
Web Images Videos News Shopping More Search tools SafeSearch



Grumpy Cat



Cute

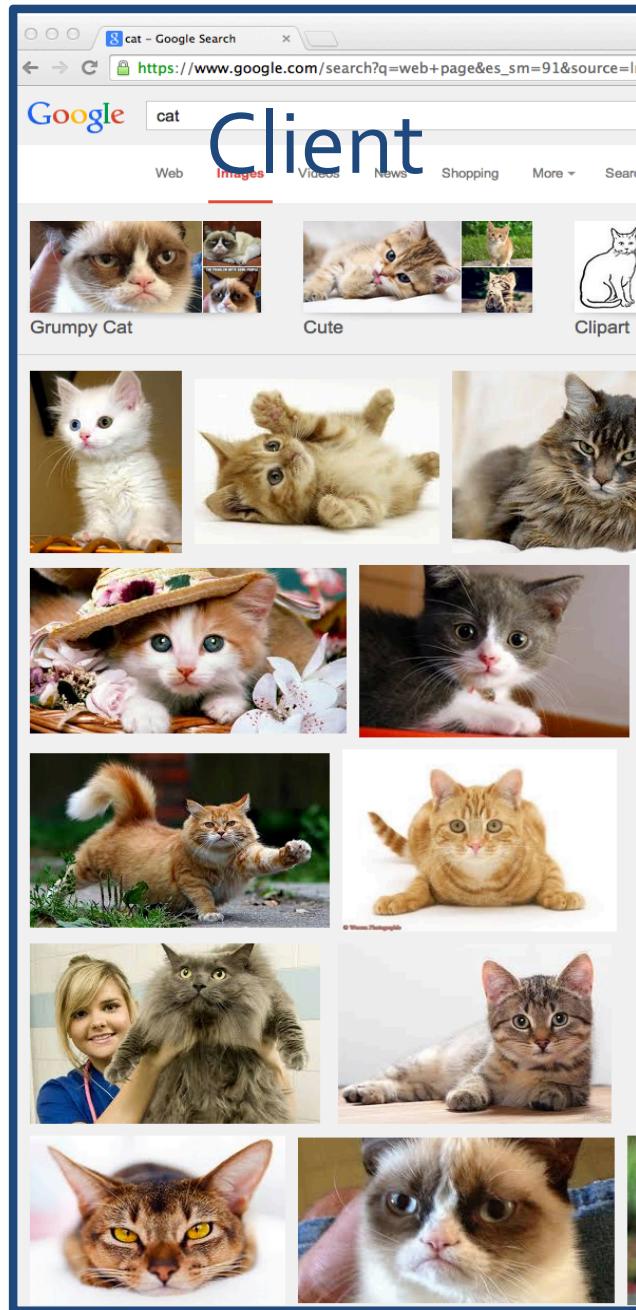


Clipart



arunab, SBU // CS 534: Fall 2022:
HTTP/HTTP2





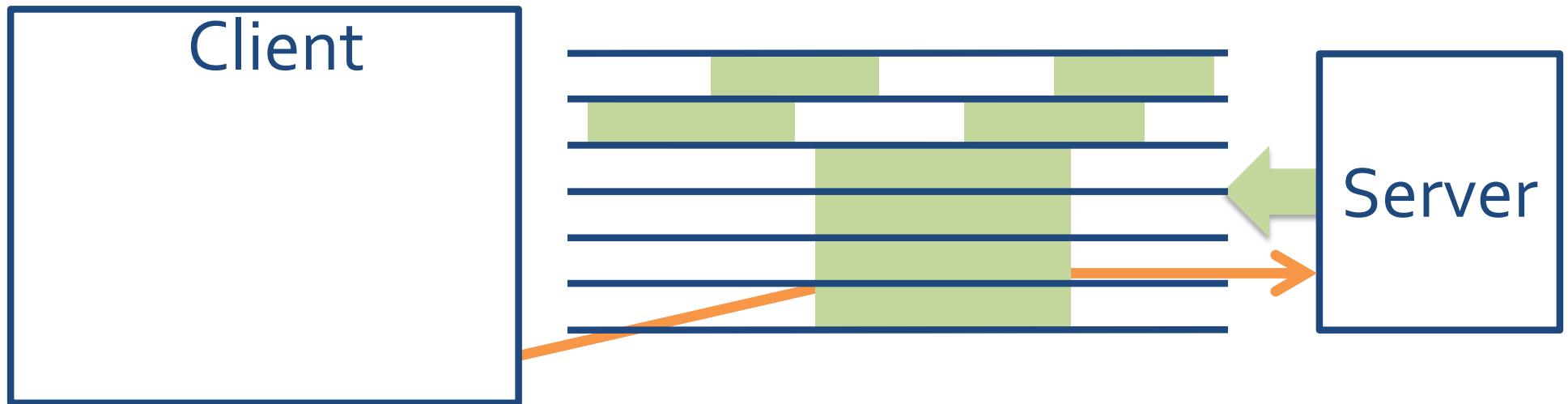
HTTP/1.1 problems (1 of 3)

6 Parallel TCP



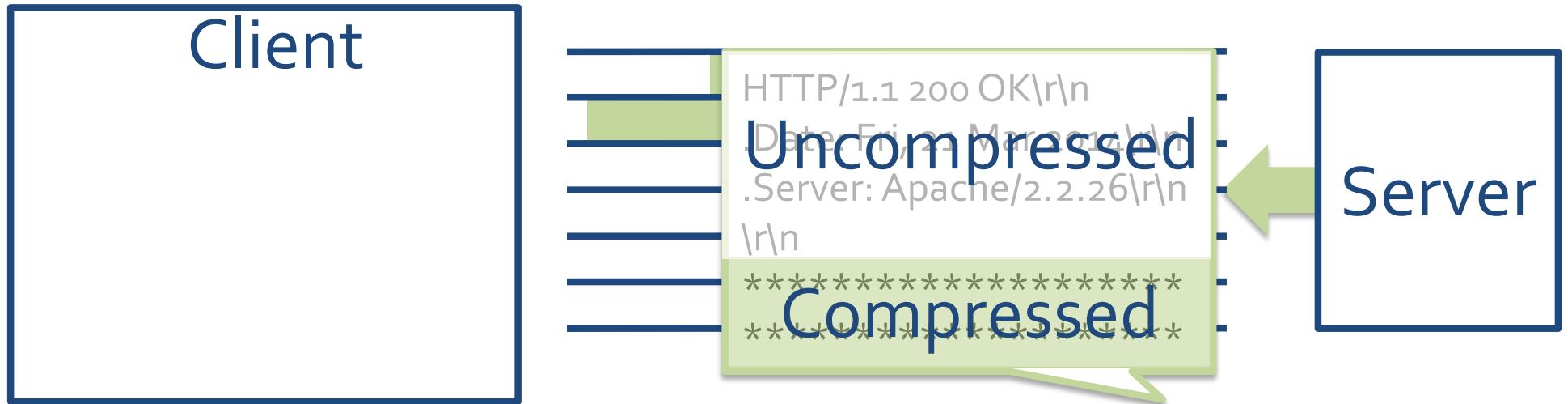
- TCP segment cannot carry more than 1 HTTP request/response
- Open multiple connections that can cause congestion
- Result: 1 round trip per “cat”

HTTP/1.1 problems (2 of 3)



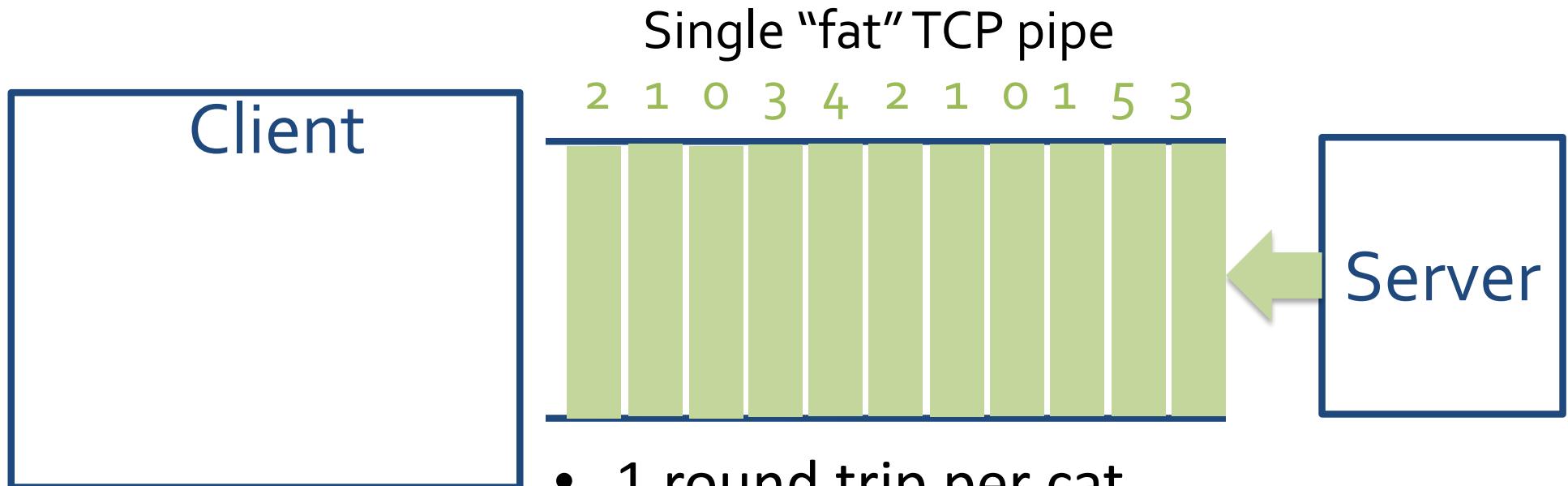
- Object transfers can only be initiated by the client

HTTP 1.1 problems (3 of 3)



- Compresses only HTTP payloads, not headers

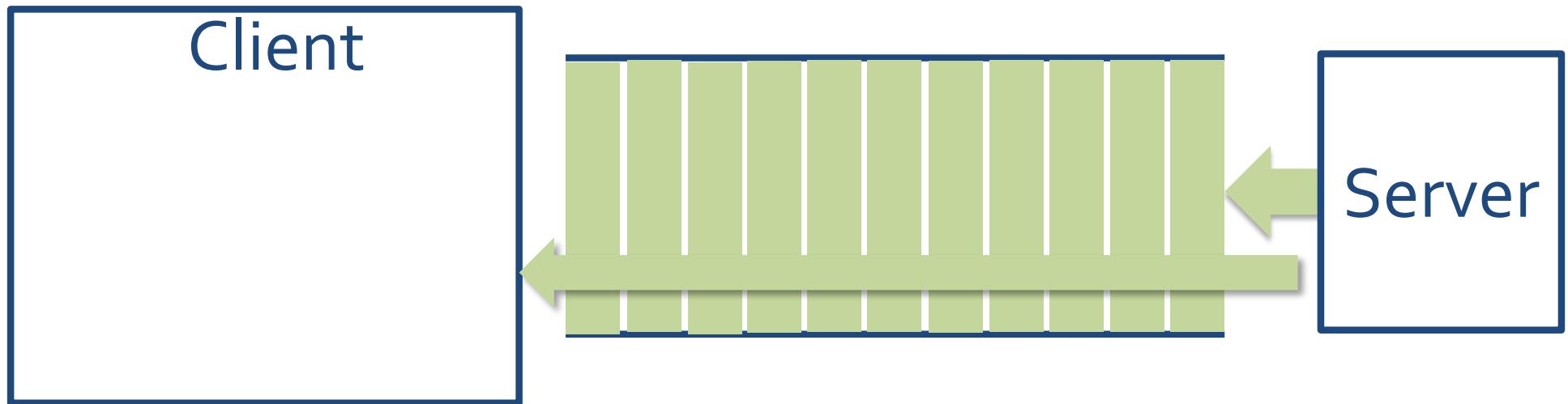
HTTP/2 (1 of 3)



Multiplexes sliced frames into a single TCP connection

- Priorities*
- SSL*

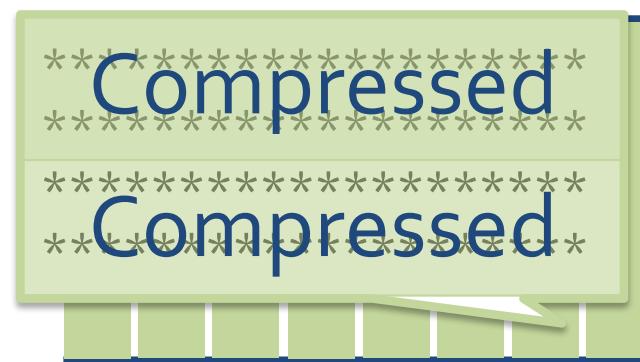
HTTP/2(2 of 3)



- ~~Initiates object transfers strictly by the client~~

Allows servers to initiate Web object transfers (server push)

HTTP/2(3 of 3)



- ~~Compresses only HTTP payloads,
not headers~~
Compresses both HTTP payloads
and headers (over 80%
compression possible)

Other HTTP topics

- Proxy Caching
- HTTPS
- HTTP Cookies