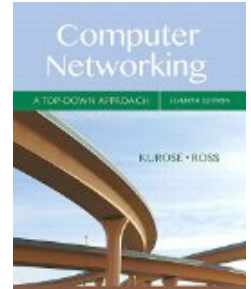


COMP 375: Lecture 06



- **News & Notes:**
 - Project #1 due Monday @ 10PM
 - Project #2 (Web Server) coming soon
- **Reading (Mon, Feb. 12)**
 - Section 2.4 (DNS)

Section 2.2

THE WEB & HTTP

You should be able to differentiate between use cases for GET and POST.

In what scenarios would you use GET? POST?

| | GET | POST |
|----|--|--|
| A. | Piazza post | Search terms, Take-out order |
| B. | Search terms, Take-out order | Piazza post |
| C. | Search terms | Piazza post, Take-out Order |
| D. | Piazza post, Search terms, Take-out Order | |
| E. | | Piazza post, Search terms, Take-out Order |

HTTP may or may not keep a single TCP connection open for multiple requests.

Non-persistent HTTP

- At most one *object* sent over TCP connection
 - connection then closed
- Downloading multiple objects requires multiple connections

Persistent HTTP

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

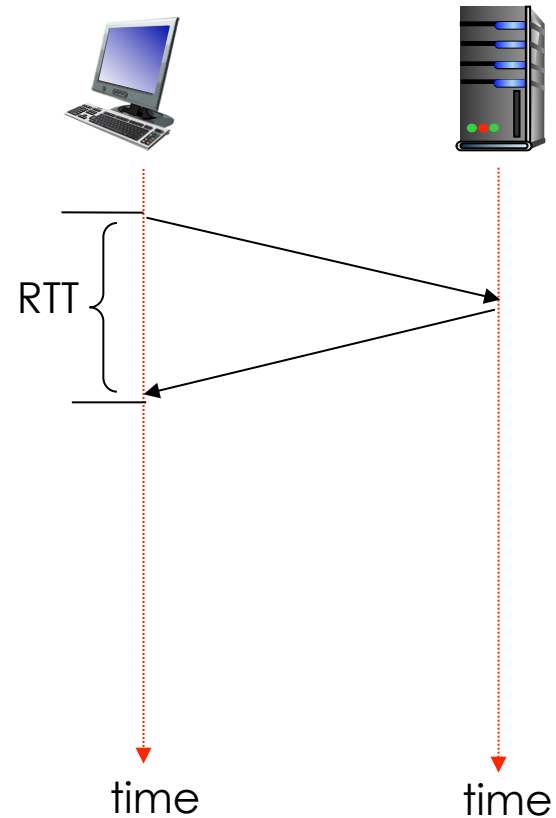
An object may be: image, script, stylesheet, etc.

Persistent and Non-persistent HTTP use the same steps, just in a different order.

Work with your group to order the following pseudo-code steps for cases: persistent HTTP and non-persistent HTTP.

1. foreach object in webpage:
2. Open TCP Connection
3. Close TCP Connection
4. Send request for object
5. Receive response with object

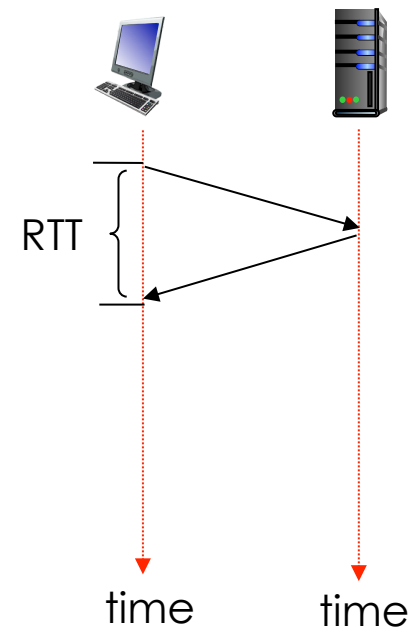
Round Trip Time (RTT) is the time for a small packet to go to/from a server.



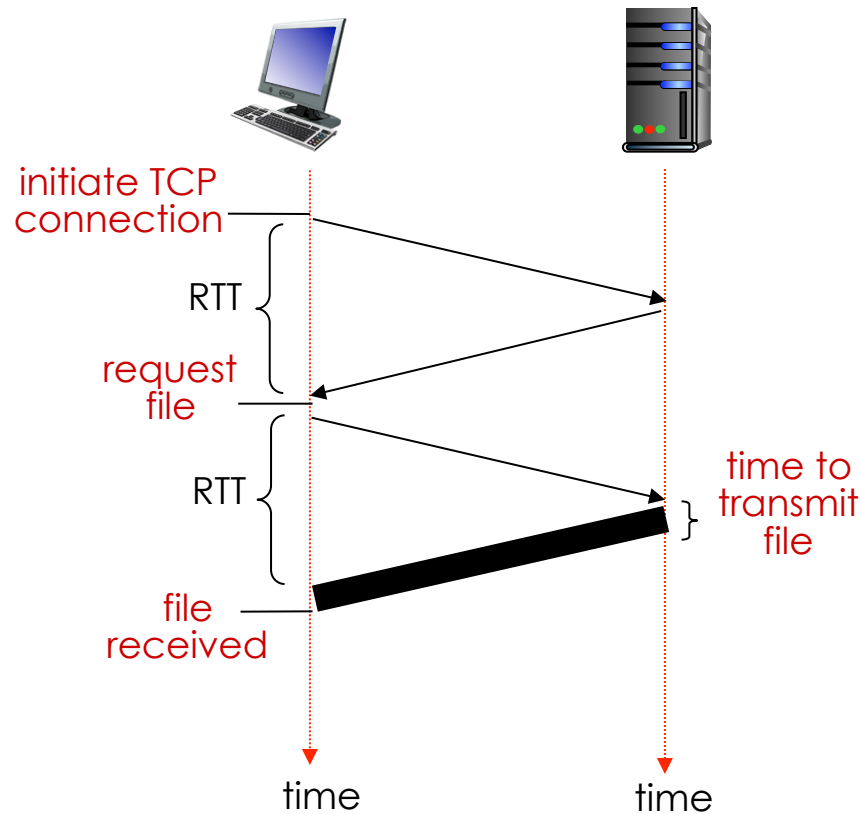
Note: Opening a TCP connection requires 1 RTT.

Non-Persistent HTTP can download a website with several objects in...

| | |
|-----------|--|
| A. | One RTT + (File transfer time per object) |
| B. | (One RTT + File transfer time) per object |
| C. | Two RTTs |
| D. | Two RTTs + (File transfer time per object) |
| E. | (Two RTTs + File transfer time) per object |

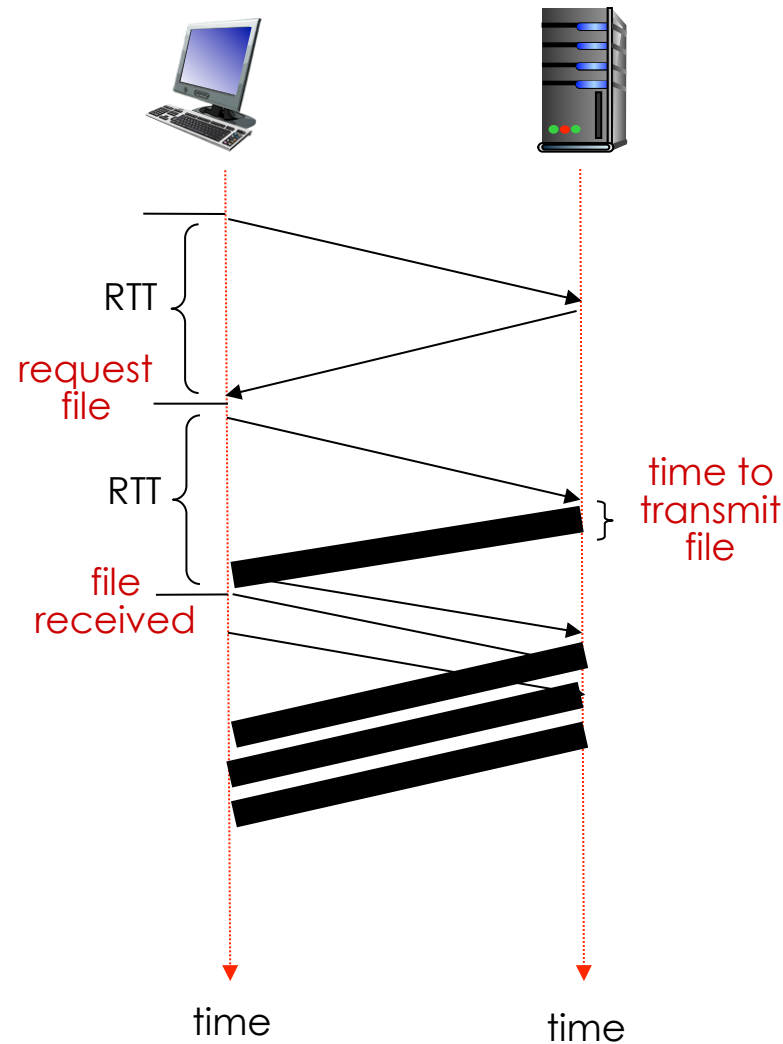


Non-persistent HTTP has high overhead for every object requested.



***NP HTTP response time = 2*RTT + File Transfer Time
... per object!***

Persistent HTTP can reduce overhead by more than 50%.

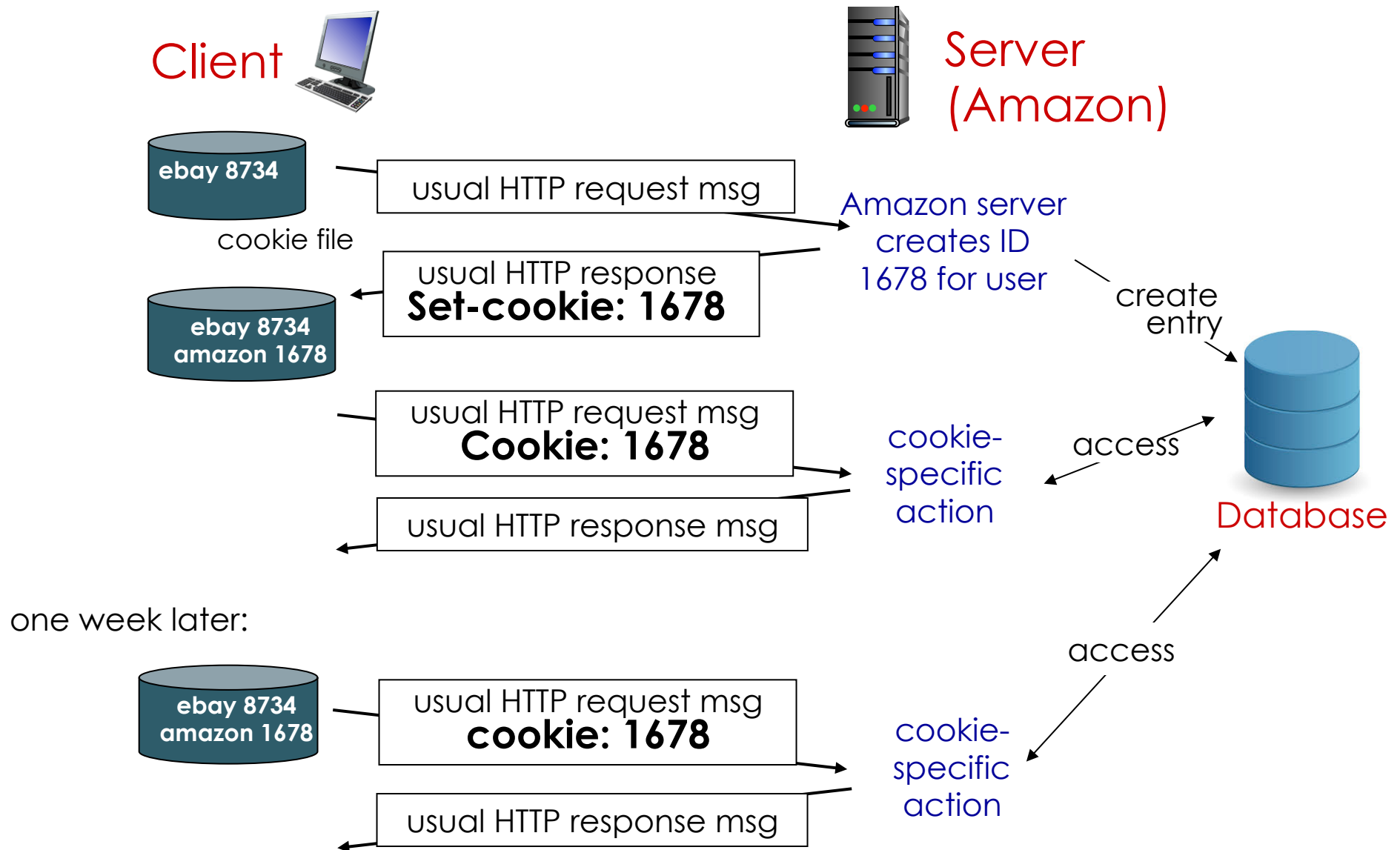


HTTP is a **stateless** protocol.

... but what does that *mean*?

... and is that good or bad?

Cookies allow us to maintain some **state** while using HTTP.



Cookies are an integral part of the modern web.

... so why are websites required by law to tell you they are using cookies?!

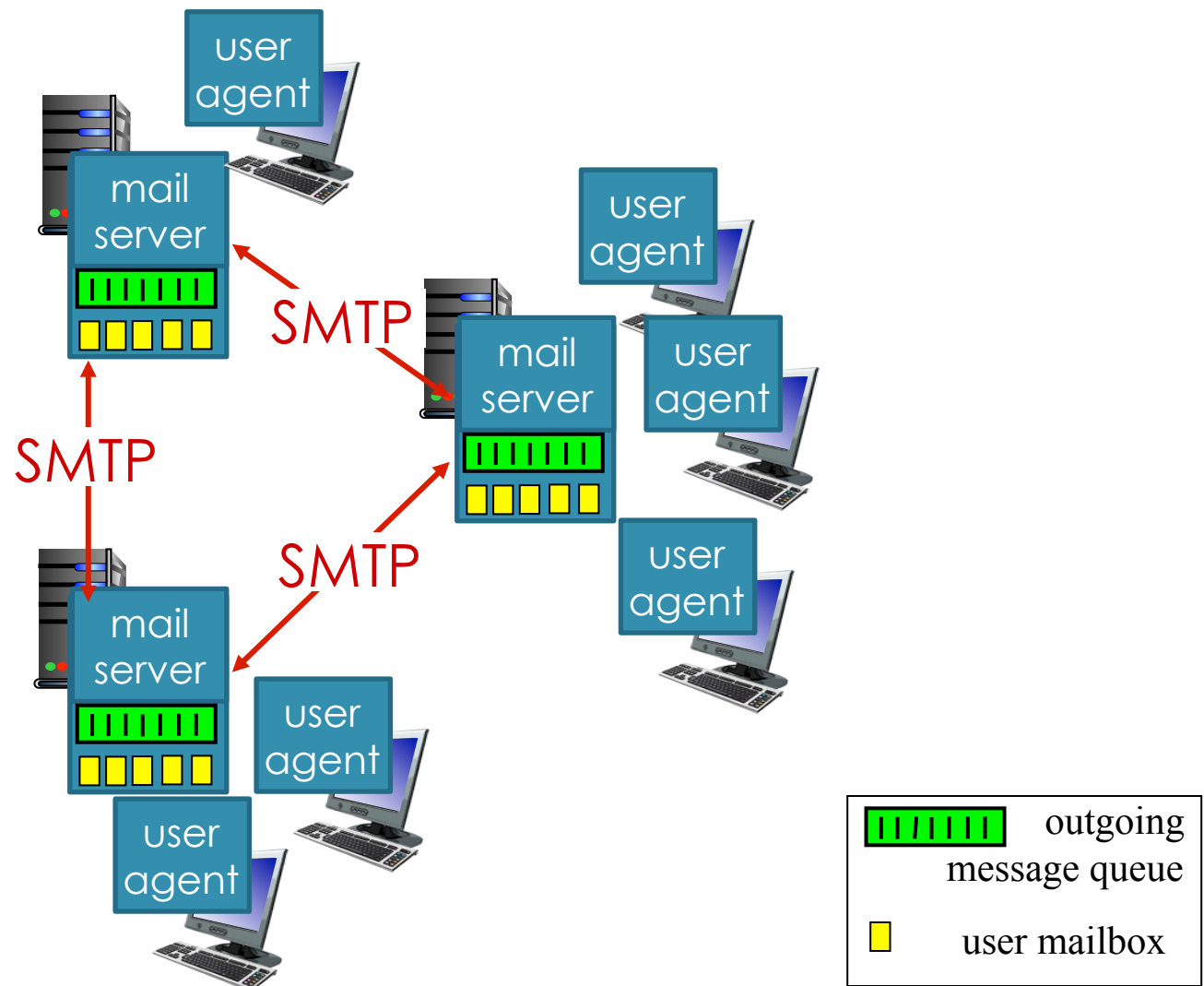
Networks are (relatively) slow and
bandwidth is not free.

What are some things we can do to
reduce latency and/or bandwidth
usage of HTTP?

Section 2.3

EMAIL

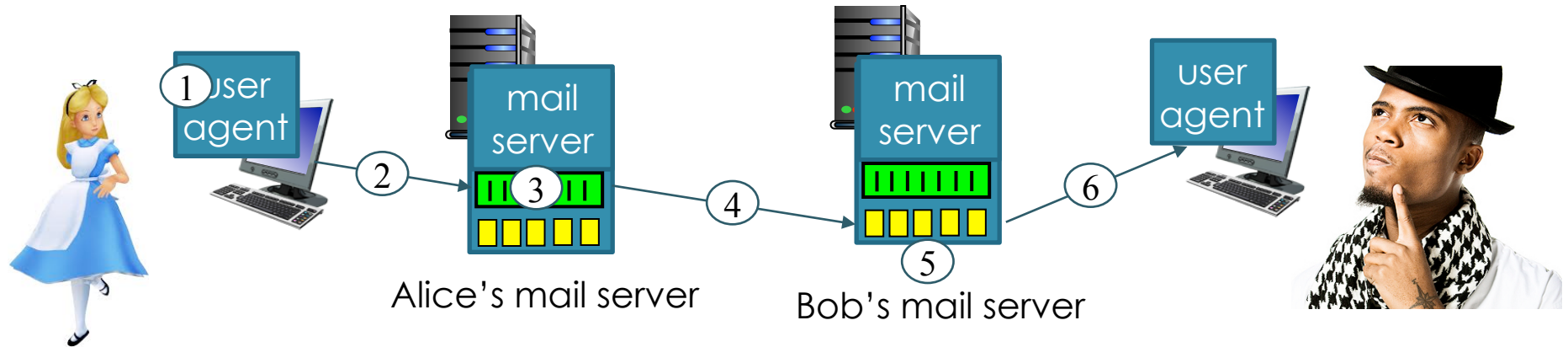
Email consists of three major components:
User agents, Transfer agents, and SMTP.



Which of these best describes what happens when Alice sends an email to Bob?

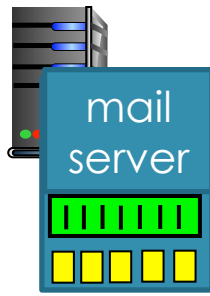
- | | |
|-----------|---|
| A. | Her mail client sends a message to his mail server. |
| B. | Her mail server sends a message to his mail server. |
| C. | Her mail server sends a message to his mail client |
| D. | Her mail client sends a message to his mail client. |

Sending email requires six basic steps and uses multiple protocols.



While SMTP involves only servers, one is a client and one is a server.

**Alice's Mail Server
(mail.alice.com)**



**Bob's Mail Server
(mail.bob.com)**



TCP connect
(port 25)



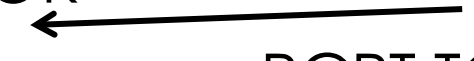
HELO alice.com



MAIL FROM: <me@alice.com>



OK



RCPT TO: <you@bob.com>



OK



DATA



OK



The message (7-bit ASCII)

