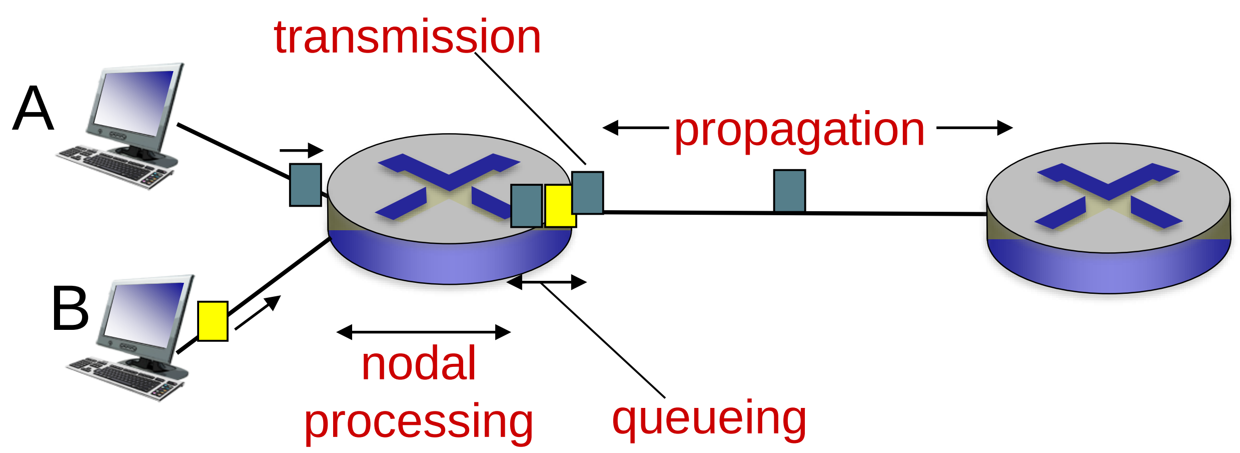
CS118 Discussion 1B, Week 2

Zhehui Zhang
HAINES A2 / Friday / 12:00pm-1:50pm

Outline

- Lecture Review
  - Packet delay, loss, throughput
    - Questions for review
  - Applications
    - HTTP: three factors
    - SMTP, DNS, P2P

Packet delay review



$$\boxed{d_{nodal} = d_{proc} + d_{queue} + d_{trans} + d_{prop}}$$

d_{proc}: nodal processing
- check bit errors
- determine output link

d_{queue}: queueing delay [DEMO]
- time waiting at output link for transmission
- depends on congestion level of router

d_{trans}: transmission delay:
- *L*: packet length (bits)
- *R*: link *bandwidth (bps)*
- $d_{trans} = L/R$

d_{prop}: propagation delay:
- *d*: length of physical link
- *s*: propagation speed (~$2\text{x}10^8$ m/sec)
- $d_{prop} = d/s$

Question 1

- Suppose Host A wants to send a large file to Host B. The path from Host A to Host B has three links, of rates R_1=500kbps, R_2=2Mbps, and R_3=1Mbps.

  a. Assuming no other traffic in the network, what is the throughput for the file transfer?

  b. Suppose the file is 4 million bytes. Dividing the file size by the throughput, roughly how long will it take to transfer the file to Host B?

  c. Repeat (a) and (b), but now with R_2 reduce to 100kpbs.

Solution

a. Assuming no other traffic in the network, what is the throughput for the file transfer? **Ans: 500 kbps**

b. Suppose the file is 4 million bytes. Dividing the file size by the throughput, roughly how long will it take to transfer the file to Host B? **Ans: (4*10^6)*8/(500*10^3)= 64 seconds**

c. Repeat (a) and (b), but now with R2 reduce to 100kpbs. **Ans: 320 seconds**

Question 1 cont’d

- Suppose Host A wants to send a large file to Host B. The path from Host A to Host B has three links, of rates R_1=500kbps, R_2=100kbps, and R_3=1Mbps.

  d. Suppose the file is 200 bytes and it is segmented into two 100 bytes packets. What is the queuing delay for the second packet at host A, the first node, the second node?

Question 2

a. How long does it take a packet of length 1000 bytes to propagate over a link of distance 2500km, propagation speed 2.5x10^8 m/s, and transmission rate 2 Mbps?

b. More generally, how long does it take a packet of length L to propagate over a link of distance d, propagation speed s, and transmission rate R bps?

c. Dose this delay depend on packet length?

d. Does this delay depend on transmission rate?

Solution

a. How long does it take a packet of length 1000 bytes to propagate over a link of distance 2500km, propagation speed 2.5x10^8 m/s, and transmission rate 2 Mbps?

   - **Ans: (2500*10^3)/(2.5*10^8)=0.01s = 10ms**

b. More generally, how long does it take a packet of length L to propagate over a link of distance d, propagation speed s, and transmission rate R bps? **Ans: d/s**

c. Dose this delay depend on packet length? Ans: No

d. Does this delay depend on transmission rate? Ans: No

Question 3

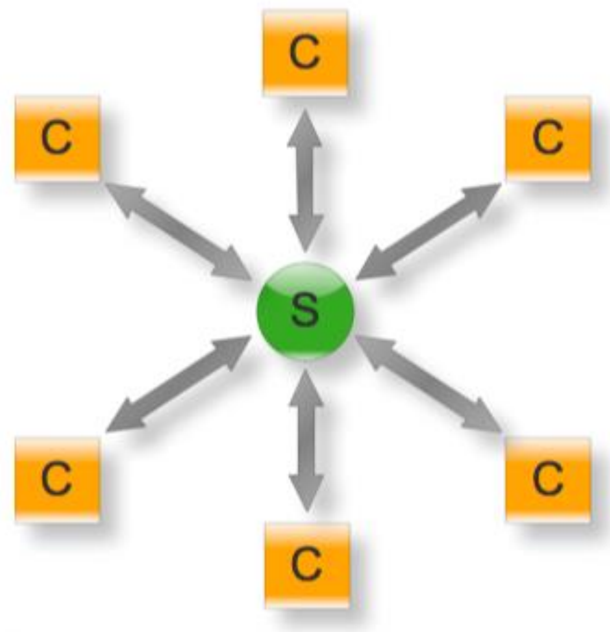
- Consider an HTTP client that wants to retrieve a Web document at a given URL. The IP address of the HTTP server is initially unknown. What transport and application layer protocols besides HTTP are needed in this scenario?

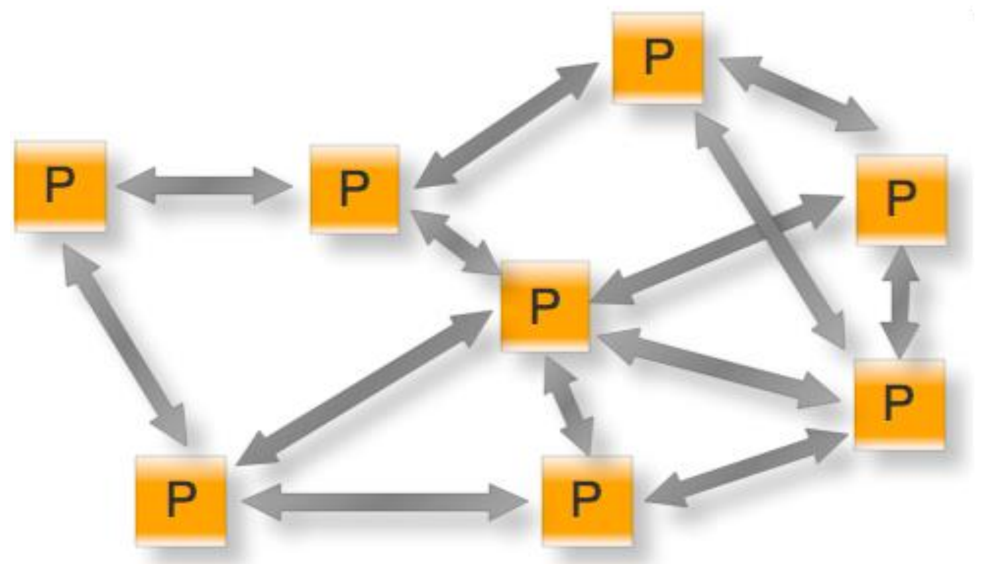
Solution

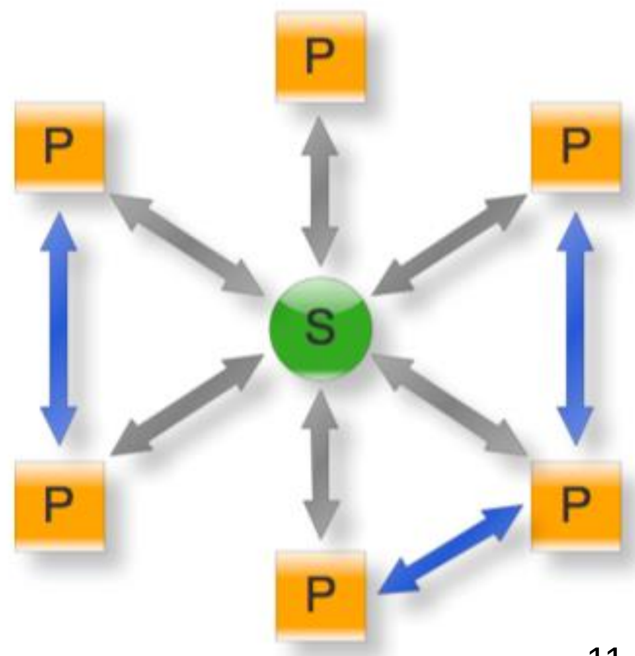
- Application layer protocols: DNS and HTTP
- Transport layer protocols: UDP for DNS; TCP for HTTP

Application Layer: Models

- Application Architectures
    - Client-server model: Web (TCP), FTP (TCP), E-mail (TCP), DNS (UDP/TCP), RTP
    - Peer-to-Peer (P2P): BitTorrent (TCP), Tor (aka Onion Routing, TCP)
    - Hybrid model: Skype (TCP&UDP), GTalk (TCP&UDP)



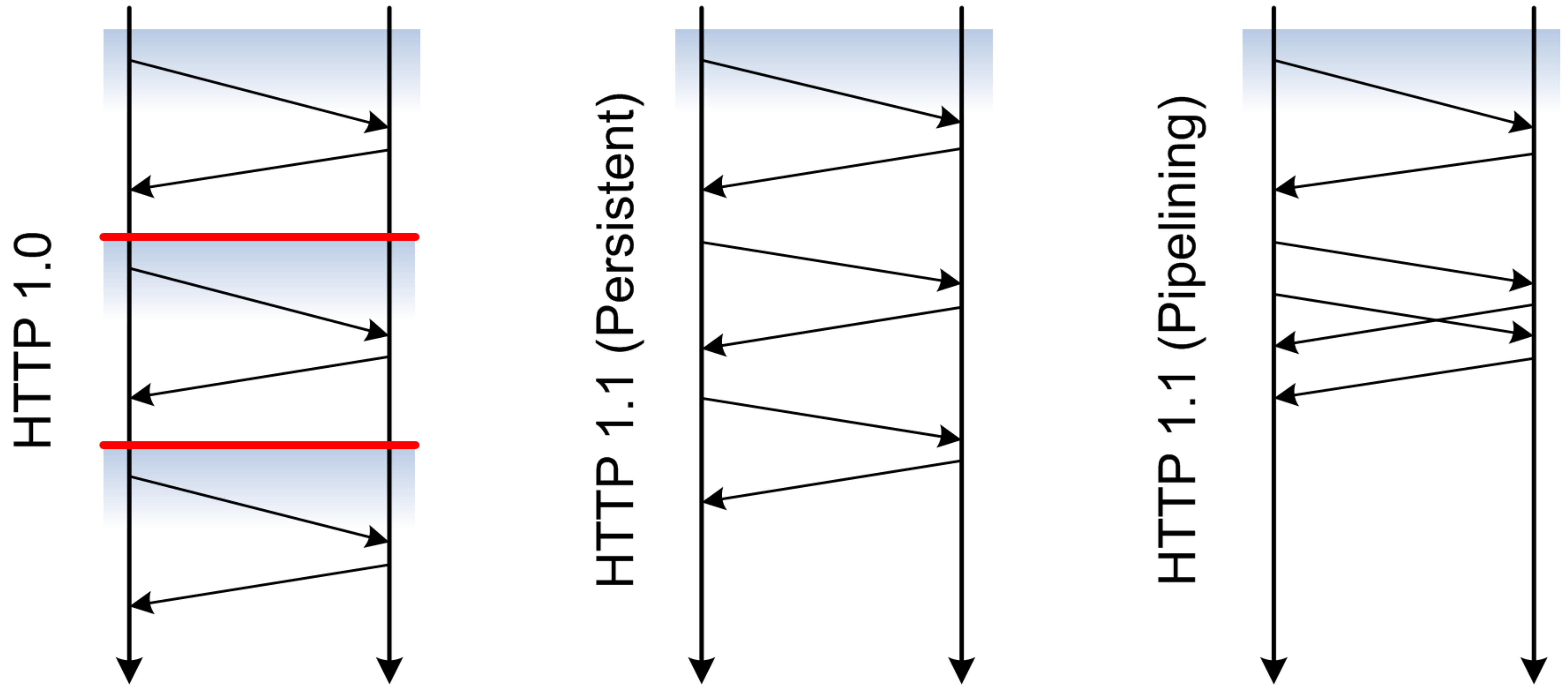




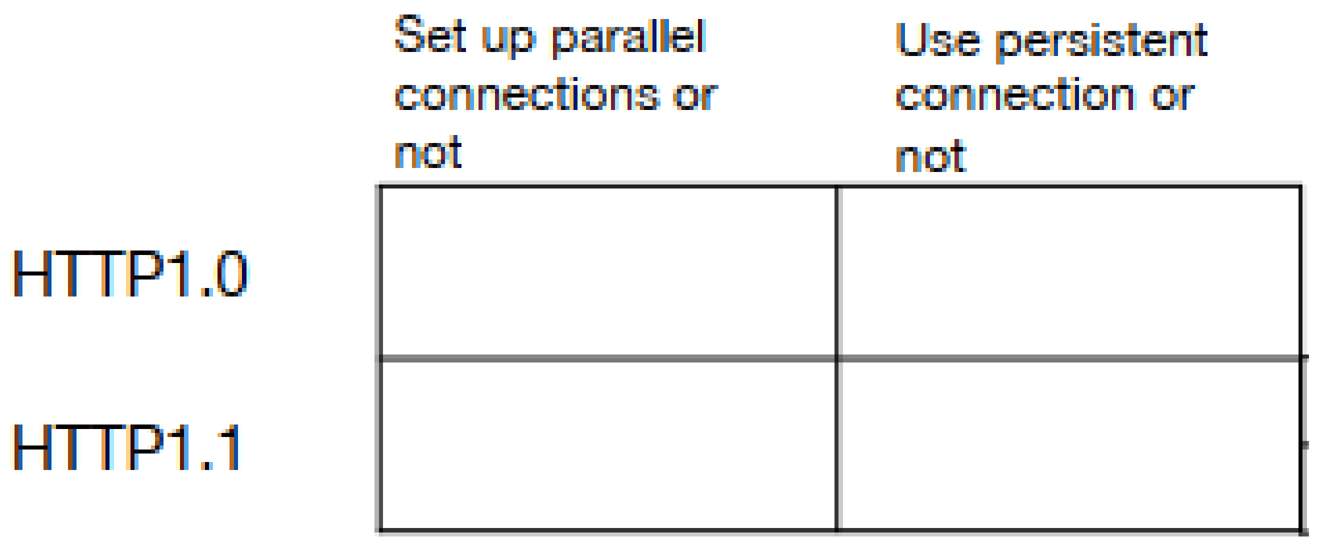
Application Layer: Protocols

- HTTP: a stateless protocol on top of TCP
  - HTTP is based on pull model
  - Persistent HTTP V.S. Non-persistent HTTP
  - Method Types: GET, HEAD, POST, PUT, DELETE, Conditional GET
  - What if we want stateful service (e.g. shopping cart)?
  - Web Caches (proxy server)

Non-persistent v.s. Persistent v.s. Pipelining



Three factors in HTTP fetching



	Set up parallel connections or not	Use persistent connection or not
HTTP1.0		
HTTP1.1		

Question

- How many TCP connections do we need to get one HTML file with 5 embedded images? How many RTTs shall we need?
- Demo

HTTP Header: request

- Request message elements:
  - Method
  - URL
  - HTTP Version
  - Header lines
  - CRLF (carriage return and line feed)

HTTP Header: response

- Response message elements:
  - HTTP Version
  - Status line
  - Header lines
  - CRLF
  - Data requested

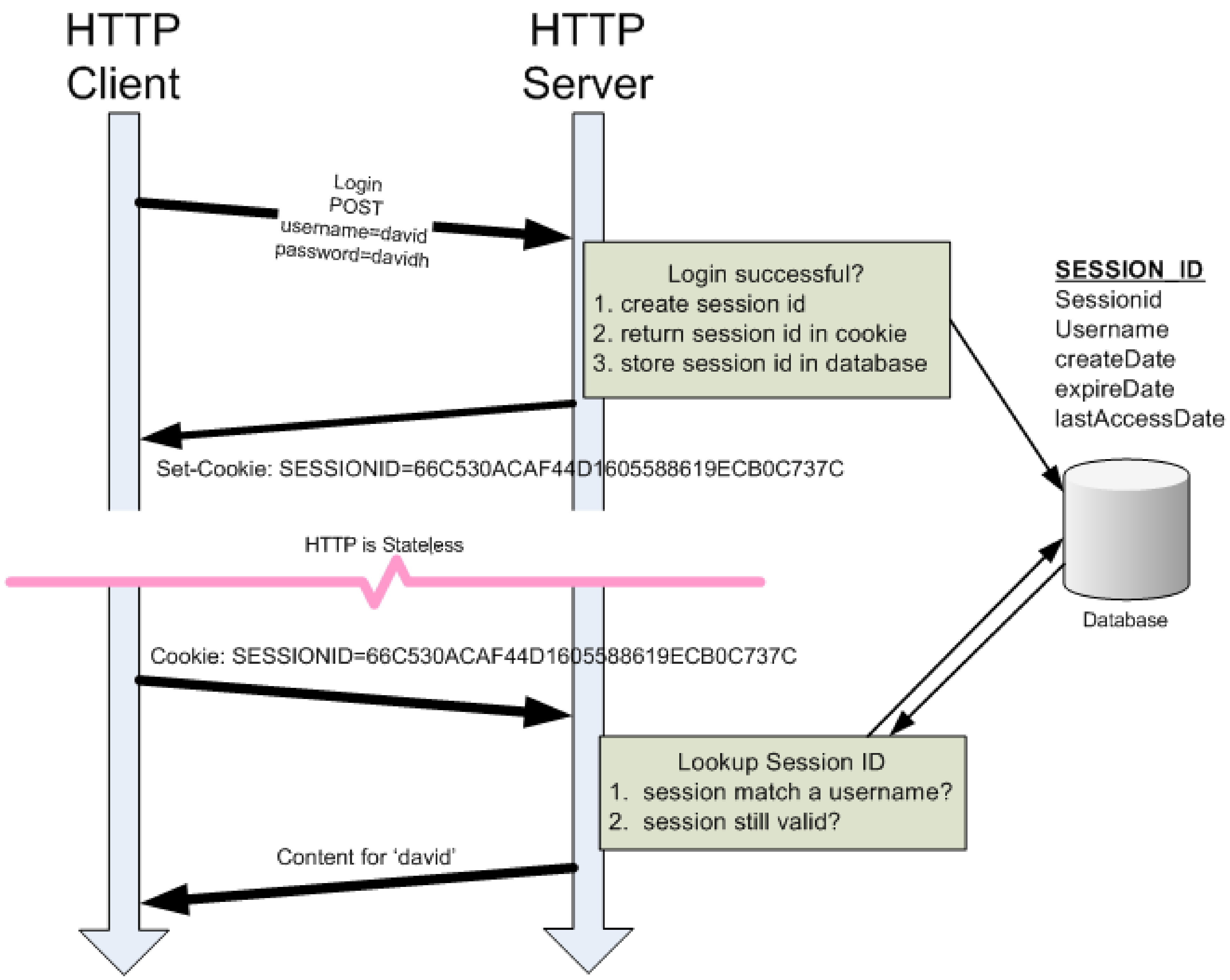
Try HTTP GET yourself

- telnet google.com 80
  - Get / HTTP/1.1
  - Host: google.com
  - <Enter>
  - <Enter>

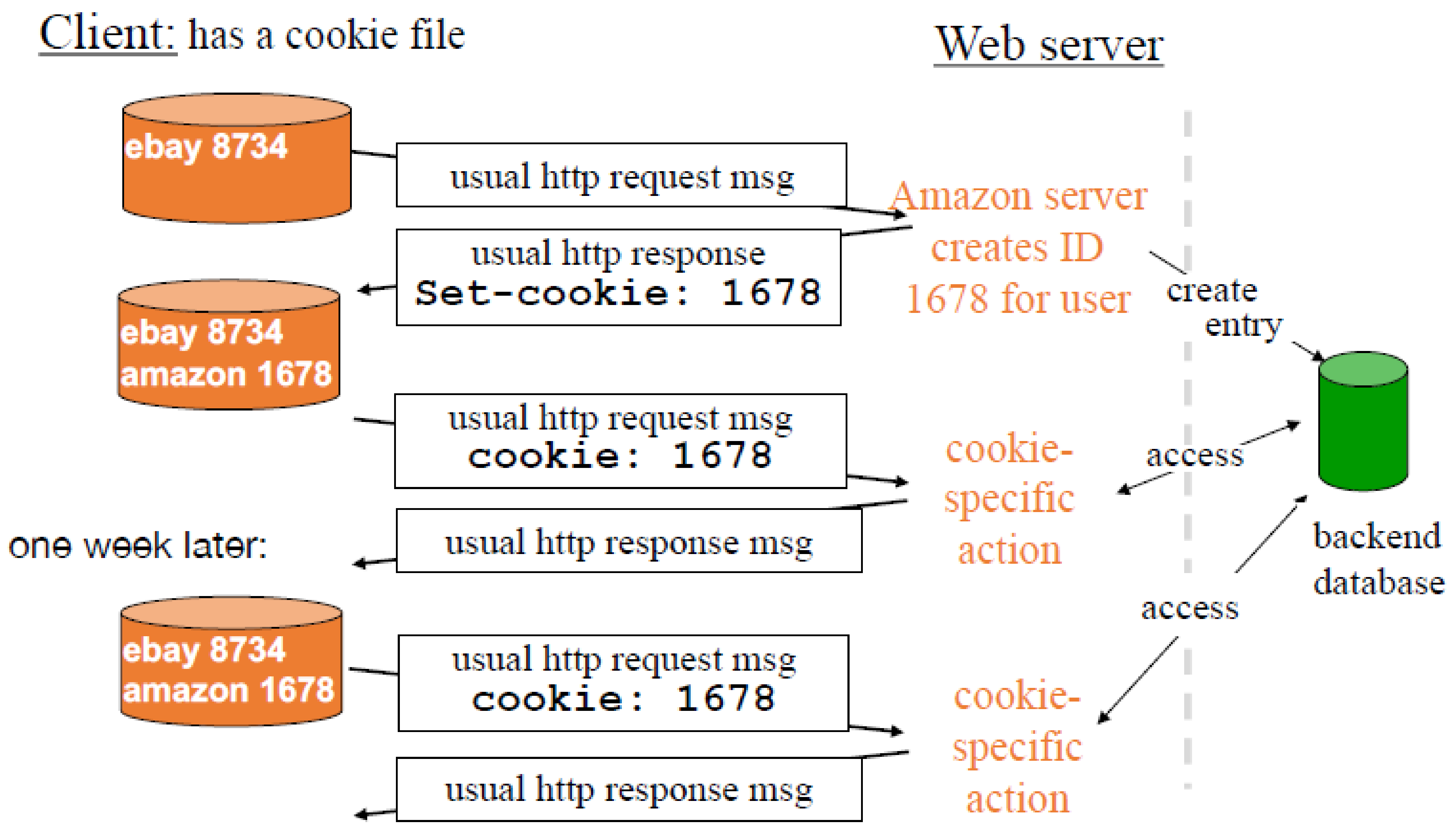
Cookie

- Bring statefulness into HTTP
- Components
  - Cookie header line of HTTP response message
  - Cookie header line of HTTP request message
  - Cookie file on the browser
  - Back-end database at web-site

Cookie: make HTTP stateful



Cookie: operations



Web caching: Proxy v.s. CDN

- Proxy acts both as client and server
    - What if cache is stale?
        - HTTP conditional GET
- CDN: Content Distribution Network
    - Globally distributed network of web servers
    - Stores and replicates images, videos and other files

HTTP conditional GET

Request:

```
GET /sample.html HTTP/1.1
Host: example.com
```

Response:

```
HTTP/1.x 200 OK
Via: The-proxy-name
Content-Length: 32859
Expires: Tue, 27 Dec 2005 11:25:11 GMT
Date: Tue, 27 Dec 2005 05:25:11 GMT
Content-Type: text/html; charset=iso-8859-1
Server: Apache/1.3.33 (Unix) PHP/4.3.10
Cache-Control: max-age=21600
Last-Modified: Wed, 01 Sep 2004 13:24:52 GMT
Etag: “4135cda4”
```

Cache-Control: It tells the client the maximum time in seconds to cache the document.

Last-Modified: The document’s last modified date

Etag: A unique hash for the document.

HTTP conditional GET

Request:

```
GET /sample.html HTTP/1.1
Host: example.com
```

Response:

```
HTTP/1.x 200 OK
Via: The-proxy-name
Content-Length: 32859
Expires: Tue, 27 Dec 2005 11:25:11 GMT
Date: Tue, 27 Dec 2005 05:25:11 GMT
Content-Type: text/html; charset=iso-8859-1
Server: Apache/1.3.33 (Unix) PHP/4.3.10
Cache-Control: max-age=21600
Last-Modified: Wed, 01 Sep 2004 13:24:52 GMT
Etag: “4135cda4”
```

Request:

```
GET /sample.html HTTP/1.1
Host: example.com
If-Modified-Since: Wed, 01 Sep 2004 13:24:52 GMT
If-None-Match: “4135cda4”
```

Response:

```
HTTP/1.x 304 Not Modified
Via: The-proxy-server
Expires: Tue, 27 Dec 2005 11:25:19 GMT
Date: Tue, 27 Dec 2005 05:25:19 GMT
Server: Apache/1.3.33 (Unix) PHP/4.3.10
Keep-Alive: timeout=2, max=99
Etag: “4135cda4”
Cache-Control: max-age=21600
```

Question

Consider the following string of ASCII characters that were captured by Wireshark when the browser sent an HTTP GET message.

a. What is the URL of the document requested by the browser?

b. What version of HTTP is the browser running?

c. Does the browser request a non-persistent or a persistent connection?

```
GET /118/index.html HTTP/1.1<cr><lf>Host: gai
a.cs.umass.edu<cr><lf>User-Agent: Mozilla/5.0 (
Windows;U; Windows NT 5.1; en-US; rv:1.7.2) Gec
ko/20040804 Netscape/7.2 (ax) <cr><lf>Accept:ex
t/xml, application/xml, application/xhtml+xml, text
/html;q=0.9, text/plain;q=0.8,image/png,*/*;q=0.5
<cr><lf>Accept-Language: en-us,en;q=0.5<cr><lf>
AcceptEncoding: zip,deflate<cr><lf>Accept-Charset: ISO
-8859-1,utf-8;q=0.7,*;q=0.7<cr><lf>Keep-Alive: 300<cr>
<lf>Connection:keep-alive<cr><lf><cr><lf>
```

Question

Consider the following string of ASCII characters that were captured by Wireshark when the browser sent an HTTP GET message.

d. What is the IP address of the host on which the browser is running?

e. What type of browser initiates this message? Why is the browser type needed in an HTTP request message?

```
GET /118/index.html HTTP/1.1<cr><lf>Host: gai
a.cs.umass.edu<cr><lf>User-Agent: Mozilla/5.0 (
Windows;U; Windows NT 5.1; en-US; rv:1.7.2) Gec
ko/20040804 Netscape/7.2 (ax) <cr><lf>Accept:ex
t/xml, application/xml, application/xhtml+xml, text
/html;q=0.9, text/plain;q=0.8,image/png,*/*;q=0.5
<cr><lf>Accept-Language: en-us,en;q=0.5<cr><lf>
AcceptEncoding: zip,deflate<cr><lf>Accept-Charset: ISO
-8859-1,utf-8;q=0.7,*;q=0.7<cr><lf>Keep-Alive: 300<cr>
<lf>Connection:keep-alive<cr><lf><cr><lf>
```

Question

The text below shows the reply sent from the server in response to the HTTP GET message in the question above.

a. As the server able to successfully find the document or not? What time was the document reply provided?

b. When was the document last modified?

c. How many bytes are there in the document being returned?

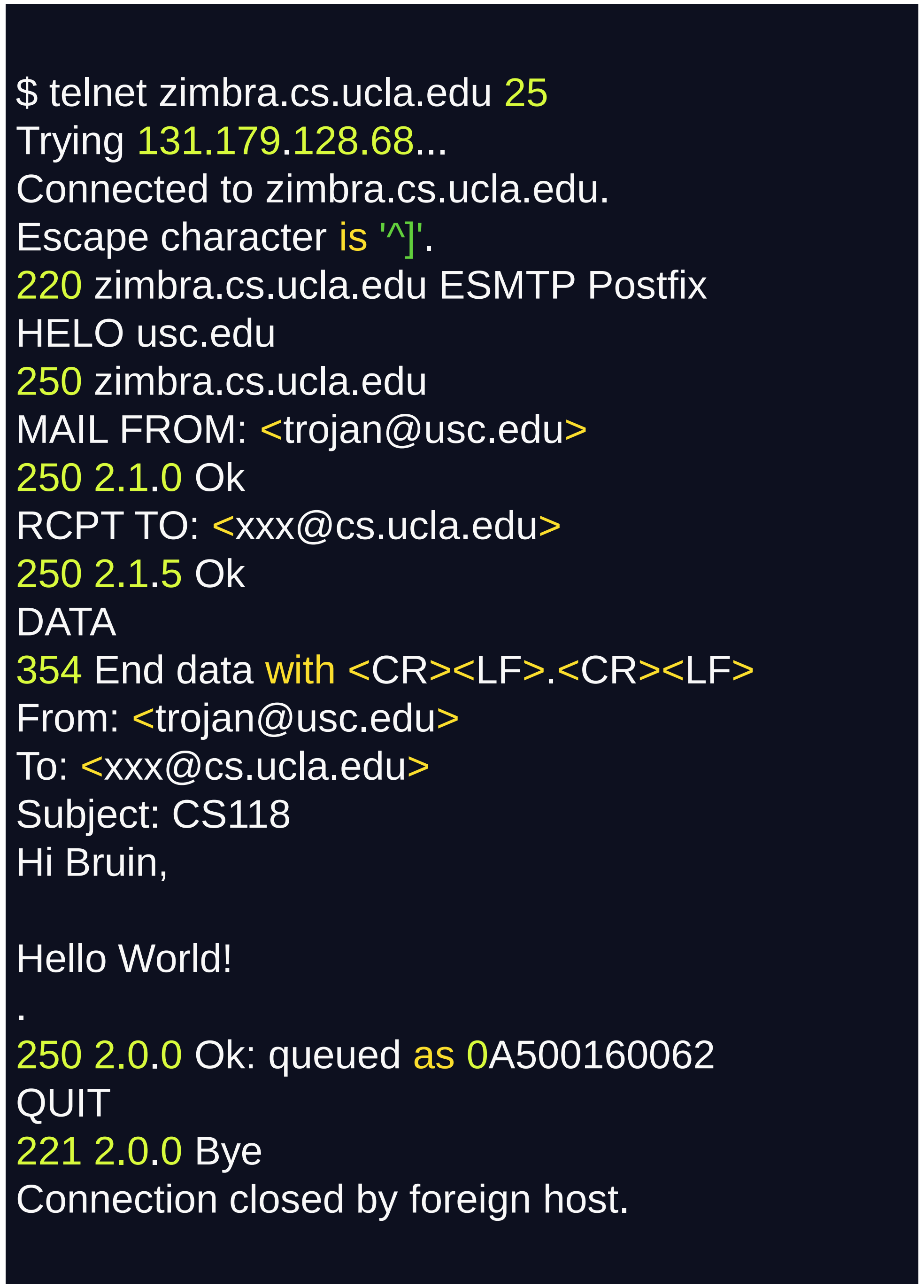
d. What are the first 5 bytes of the document being returned? Did the server agree to a persistent connection?

```
HTTP/1.1 200 OK<cr><lf>Date: Tue, 07 Mar 2008
12:39:45GMT<cr><lf>Server: Apache/2.0.52 (Fedora)
<cr><lf>Last-Modified: Sat, 10 Dec2005 18:27:46
GMT<cr><lf>ETag: "526c3-f22-
a88a4c80"<cr><lf>Accept- Ranges:
bytes<cr><lf>Content-Length: 3874<cr><lf> Keep-Alive:
timeout=max=100<cr><lf>Connection: Keep-
Alive<cr><lf>Content-Type: text/html; charset =ISO-
8859-1<cr><lf><cr><lf><!doctype html public "-
//w3c//dtd html 4.0 transitional//en"><lf><html><lf>
<head><lf> <meta http-equiv="Content-Type"
content="text/html; charset=iso-8859-1"><lf><meta
name="GENERATOR" content="Mozilla/4.79 [en]
(Windows NT 5.0; U) Netscape]"><lf> <title>CMPSCI
453 / 591 / NTU-ST550A Spring 2005
homepage</title><lf></head><lf> <much more
document text following here (not shown)>
```

Application Layer: Protocols

- FTP: separate control from data transmission (“out-of-band”)
- SMTP: protocol for email exchange between email servers
  - SMTP is based on push model
  - Mail access protocol: POP, IMAP, HTTP-based
- P2P: no always-on server, peers are intermittently connected
  - BitTorrent: tracker and torrent. Files are divided into multiple chunks.

A fun experiment: SMTP



```
$ telnet zimbra.cs.ucla.edu 25
Trying 131.179.128.68...
Connected to zimbra.cs.ucla.edu.
Escape character is '^]'.
220 zimbra.cs.ucla.edu ESMTP Postfix
HELO usc.edu
250 zimbra.cs.ucla.edu
MAIL FROM: <trojan@usc.edu>
250 2.1.0 Ok
RCPT TO: <xxx@cs.ucla.edu>
250 2.1.5 Ok
DATA
354 End data with <CR><LF>.<CR><LF>
From: <trojan@usc.edu>
To: <xxx@cs.ucla.edu>
Subject: CS118
Hi Bruin,

Hello World!
.
250 2.0.0 Ok: queued as 0A500160062
QUIT
221 2.0.0 Bye
Connection closed by foreign host.
```

Be responsible and be aware of legal consequences!

Application Layer: Protocols

- DNS: convert hostname to IP address (and more)
- A distributed and hierarchical database
  - Root DNS servers
  - Top-level domain (TLD) servers
  - Authoritative DNS servers
  - local DNS server (aka, DNS resolver)

Application Layer: protocols

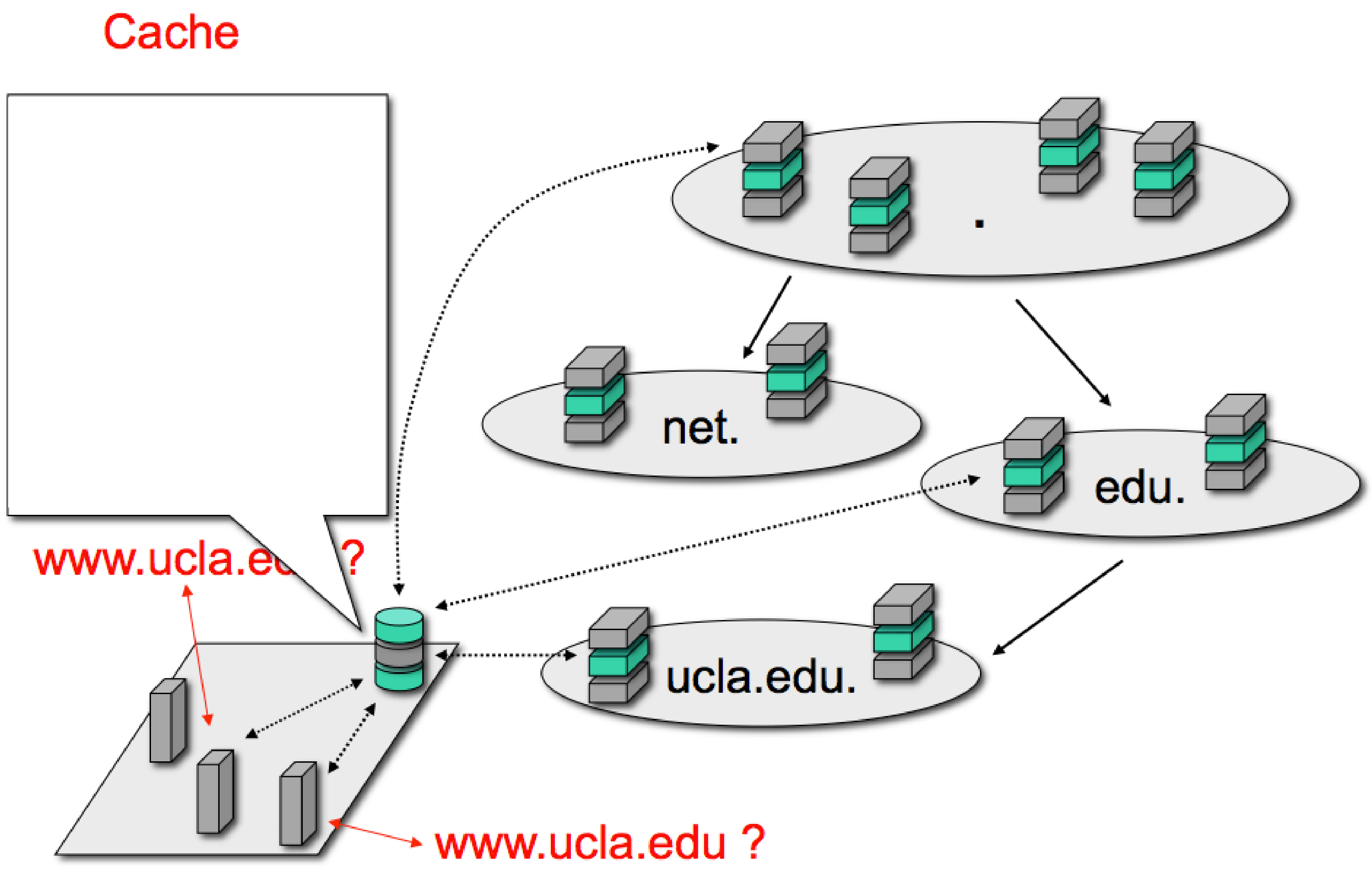
- DNS:
  - What is the transport layer protocol?
  - How the scalability is achieved?
  - Who will use iterative/recursive query?
  - Why is DNS resolver needed?

DNS protocol: exercise

- Assume the cache is empty initially
- Host A queries www.ucla.edu, how many queries should the resolver issue?
- After A’s DNS query, host B queries www.mit.edu, how many queries should the resolver issue?

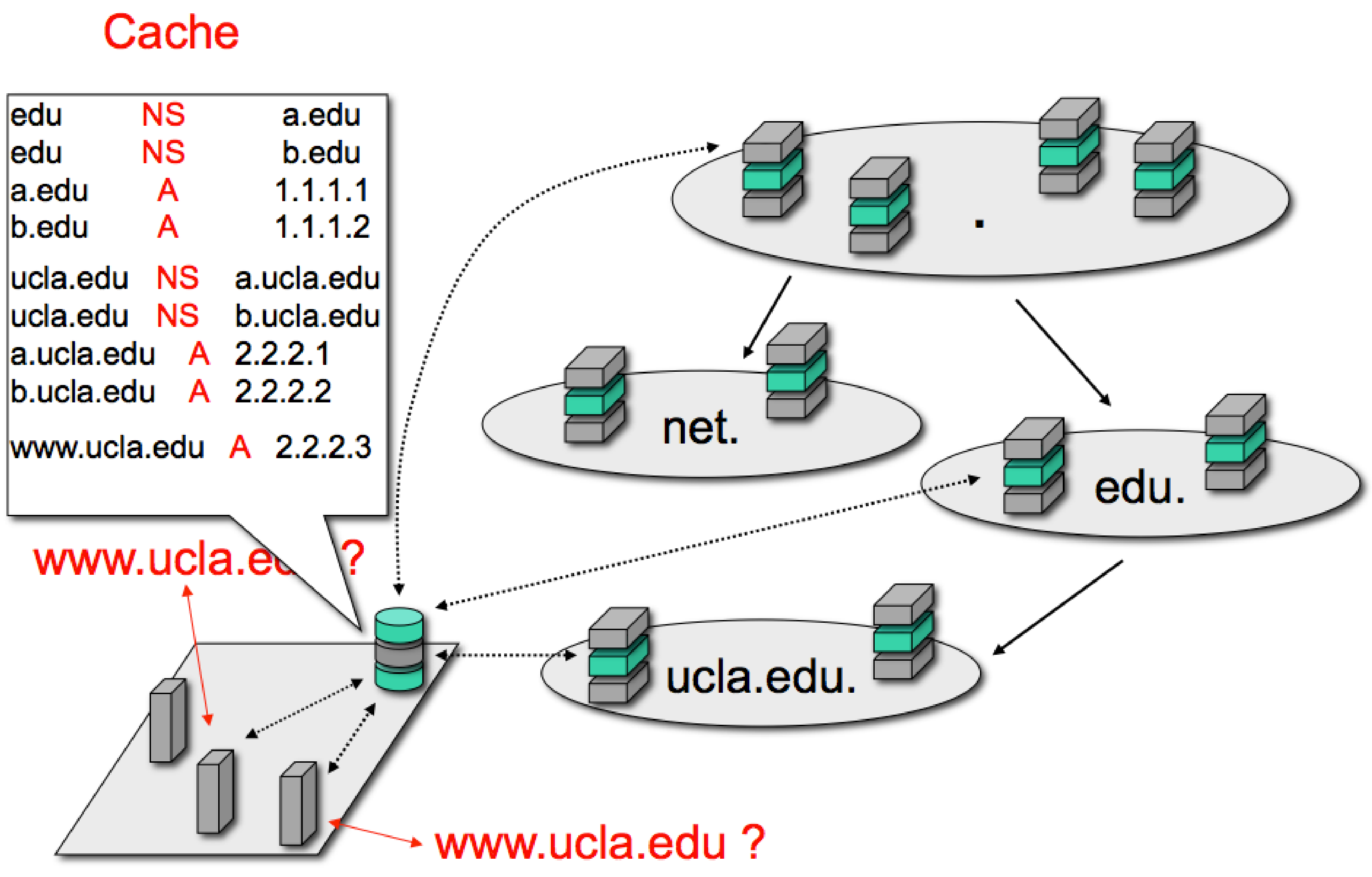
DNS protocol: exercise

- Assume the cache is empty initially
- Host A queries www.ucla.edu, how many queries should the resolver issue?



DNS protocol: exercise

- Assume the cache is empty initially
- Host A queries www.ucla.edu, how many queries should the resolver issue?



DNS protocol: exercise

- Assume the cache is empty initially
- Host A queries www.ucla.edu, how many queries should the resolver issue?
- After A’s DNS query, host B queries www.mit.edu, how many queries should the resolver issue?

A fun experiment: DNS query

```
$ dig google.com
; <<>> DiG 9.8.3-P1 <<>> google.com
;; global options: +cmd
;; Got answer:
;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 44777
;; flags: qr rd ra; QUERY: 1, ANSWER: 1, AUTHORITY: 4, ADDITIONAL: 4

;; QUESTION SECTION:
;google.com.            IN  A

;; ANSWER SECTION:
google.com.     76  IN  A   172.217.2.14

;; AUTHORITY SECTION:
google.com.     85950   IN  NS  ns3.google.com.
google.com.     85950   IN  NS  ns4.google.com.
google.com.     85950   IN  NS  ns1.google.com.
google.com.     85950   IN  NS  ns2.google.com.

;; ADDITIONAL SECTION:
ns1.google.com.     59591   IN  A   216.239.32.10
ns2.google.com.     50756   IN  A   216.239.34.10
ns3.google.com.     40354   IN  A   216.239.36.10
ns4.google.com.     36005   IN  A   216.239.38.10

;; Query time: 84 msec
;; SERVER: 158.69.209.100#53(158.69.209.100)
;; WHEN: Thu Jan 19 20:37:48 2017
;; MSG SIZE  rcvd: 180
$ dig any mit.edu
```

DNS parameters note

Bit-torrent

- Alice sends chunks to those four peers currently sending her chunks at highest rate，other peers are choked by Alice (do not receive chunks from her)

- re-evaluate top 4 every10 secs

- every 30 secs: randomly select another peer, starts sending chunks

    - “optimistically unchoke” this peer，newly chosen peer may join top 4