CS118 Midterm

Richard Sun

TOTAL POINTS

90 / 100

QUESTION 1

20 pts

1.1 (4 / 4)

- **- 0 Correct**
- - 0.5 Base object also requested by proxy
- - 1 One object is missing

1.2 (4 / 4)

- **- 0 Correct**
- - 0.5 Base object is also requested by proxy
- - 4 Incorrect
- - 1 One object missing

1.3 (12 / 12)

- - 3 1st wrong
- - 3 2nd wrong
- - 3 3rd wrong
- - 0 (not graded) foo.com/logo.png and bar.com/logo.png not related, even if IP match
- **- 0 Correct**

QUESTION 2

20 pts

2.1 (5.5 / 6)

- **+ 6 Correct**
- + 1 know connection setup RTT + http RTT
- + 2 calculate connection setup RTT correctly
- + 2 calculate http RTT correctly
- + 1 know 4 connections one-by-one
- **- 0.5 calculation error**
- + 0 wrong

2.2 (5.5 / 6)

- **+ 6 Correct**
- + 1 know two rtt
- + 2 calculate parallel transmission correctly (queueing)
- + 2 calculate queuing correctly
- + 1 calculate last one correctly
- **- 0.5 calculation error**
- + 0 wrong

2.3 (4 / 4)

- **+ 4 Correct**
- + 1 setup tcp
- + 1 4 requests
- + 1 one request calculation
- + 1 total
- + 0 wrong

2.4 (4 / 4)

- **+ 4 Correct**
- + 1 connection
- + 1 4 back-to-back response
- + 2 queueing delay
- + 0 wrong
- - 1 end-to-end delay error

QUESTION 3

20 pts

3.1 (4 / 4)

- **+ 4 Correct**
- + 1 query to root
- + 1 query to com
- + 1 query to google.com
- + 1 query to caching resolve
- + 0 wrong
- + 1 partial

3.2 (4 / 4)

- **+ 4 Correct**
- + 1 query to com
- + 1 query to amazon.com
- + 1 query to resolver
- + 1 caching
- + 1 partial

3.3 (3 / 3)

+ 3 Correct
+ 1 root
+ 1 .com
+ 1 google.com
+ 1 no cache
+ 1 partial
- 1 additional queries

3.4 (3 / 3)

+ 3 Correct
+ 1 amazon.com
+ 2 caching
- 1 additional query
+ 0 wrong

3.5 (5 / 6)

+ 1 a) is right
+ 1 b) is right
+ 1 c) is right
+ 1 d) is wrong
+ 1 e) is right
+ 1 f) is wrong

QUESTION 4

20 pts

4.1 (3 / 6)

- 0 Correct
- 0.5 SYN in 2nd
- 0.5 no SYN in 3rd packet
- 0.5 no SYN in 3-9th packets
- 0.5 no SYN in 3-9th packets
- 0.5 no RST flags anywhere
- 0.5 No FIN in last packet
- 0.5 no FIN in 1st-6th packets
- 0.5 no FIN in 1st-6th packets
- 1 FIN 8th, no FIN 9th
- 0.5 FIN in 7th and 8th
- 0.5 no ACK in 7th packet
- 0.5 ACK in 3rd packet
- 0.5 ACK in 4 or in 4&5th packet
- 0.5 ACK in 6th packet
- 0.5 ACK in 8th packet
- 0.5 ACK in 9th packet
- 0.5 One or two unspecified flags
- 2 4 unspecified/incorrect flags
- 6 Most flags not filled
- 2 4 unspecified/incorrect flags

4.2 (5.5 / 8)

- 0 Correct
- 1 No. 3 incorrect
- 1 No. 4 incorrect
- 1 No. 5 incorrect
- 1 No. 6 incorrect
- 0.5 No. 7 incorrect (no ACK)
- 1 No. 8 incorrect
- 1 No. 9 incorrect
- 8 Incorrect
- 0.1 Error in calculations
- 0.5 Some mixing of ack and seq positions
- 0.5 SeqNo offset error

4.3 (1.5 / 3)

- 0 Correct
- 3 Incorrect
- 1.5 Incomplete/unclear reasoning
+ 1 2 correct reasons

4.4 (3 / 3)

- 0 Correct
- 1.5 Issue with reasoning
- 0.5 Minor issue
- 3 Incorrect/Missing answer

QUESTION 5

20 pts

5.1 (4 / 4)

- 0 Correct
- 2 Didn't mention no need for a connection (extra RTT)
- 4 Incorrect

5.2 (4 / 4)

- 0 Correct
- 4 Incorrect
- 2 Incorrect explanation
- 1 Incomplete explanation (no delay mentioned)

+ 1 (extra credit: firewall issue)

5.3 (3 / 4)

- 0 Correct
- 1 (b) is not secured at all (HTTP), so can be monitored or you can be talking not to amazon
- 1 **(c) is encrypted (no monitoring), but you will send data not to amazon, but a potential attacker**
- 2 Missing/wrong/incomplete discussion of unsafe cases
- 1 Some issues with discussion
- 2 (a) is safe = DV validated cert
- 1 (a) is pretty safe

> When some items loaded over HTTP, the address bar will still show green. Or may be yellow. Red and strikeout means something really bad

5.4 (4 / 4)

- 0 **Correct**
- 2 Incorrect/incomplete reasoning
- 1 Issue with reasoning
- 4 Incorrect/missing answer

5.5 (4 / 4)

+ 2 **Head-of-line blocking**
+ 2 Reduce size of headers / overhead
+ 2 **Request prioritization/request-response interleaving**
+ 2 Proactive push of objects
+ 2 Non-optional pipelining
+ 1 Almost correct item
+ 0 **Correct**
+ 0 Incorrect/Missing

CS118
Spring 2016 Midterm Exam

1 hour 50 minutes

Close book and closed notes; NO use of any device except calculators.

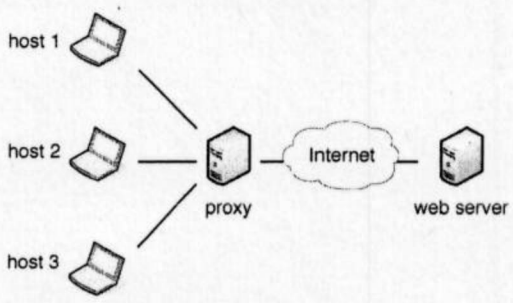
- This exam has 6 pages, including this cover page. Do all your work on these exam sheets.
- Cross out all the scratch work that you do not want to be counted as part of your answer before you submit the exam.
- Show ***all*** your work, including unfinished problems that you wish to be considered for partial credit.
- Be *specific, clear,* ***concise*** in your answers, and *explain* your answers.
- When the answer to a problem is not immediately clear, do not simply dump everything, relevant or irrelevant, on the paper. Irrelevant answers may lead to point-deduction as they show the lack of understanding of the problem.

Your name: Richard Sun

Student ID: 904444918

Problem 1 (20 points) Three hosts share the same web caching proxy whose cache is empty at the beginning. The browser on host 1 sends the proxy a request for `http://foo.com/info?uid=tom`. This initial object contains three referenced objects, which are then retrieved by the browser on host 1:

```
http://foo.com/logo.png
http://foo.com/profile?uid=tom
http://foo.com/footnote
```



10 seconds later, the browser on host 2 sends a request for `http://foo.com/info?uid=jerry`. This initial object also contains three referenced objects:

```
http://foo.com/logo.png
http://foo.com/profile?uid=jerry
http://foo.com/footnote
```

1.1 (4 points) Please circle one or more HTTP requests that were sent from the caching proxy in the first 10 seconds.

(a) `http://foo.com/info?uid=tom`
(b) `http://foo.com/logo.png`
(c) `http://foo.com/profile?uid=tom`
(d) `http://foo.com/footnote`
(e) `http://foo.com/info?uid=jerry`
(f) `http://foo.com/profile?uid=jerry`

1.2 (4 points) Please circle one or more the HTTP requests were sent from the caching proxy after the first 10 seconds.

(a) `http://foo.com/info?uid=tom`
(b) `http://foo.com/logo.png`
(c) `http://foo.com/profile?uid=tom`
(d) `http://foo.com/footnote`
(e) `http://foo.com/info?uid=jerry`
(f) `http://foo.com/profile?uid=jerry`

1.3 (12 points) Please circle True or False.

False (True or False) If another host, host 3, sends a request for `http://foo.com:8080/logo.png`, the proxy will not send any HTTP request.

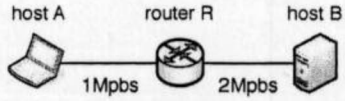
True (True or False) If host 3 sends a request for `http://foo.com:80/logo.png`, the proxy will not send any HTTP request.

False (True or False) Host3 sends another request for `http://bar.com/logo.png`. When the caching proxy sends the DNS query for name `bar.com`, DNS returns the same IP address as the IP address for `foo.com`. This must be an error.

False (True or False) If `http://foo.com/logo.png` object is already cached in the proxy, the caching proxy will not send a separate HTTP request for `http://bar.com/logo.png`.

C S

Problem 2 (20 points) A web browser is running on the host A. A web server on the host B. Host A and B are connected to a router R. The bandwidth of Link A-R is 1 Mbps (10^6 bits/sec), while the bandwidth of Link R-B is 2 Mbps (2×10^6 bits/sec). The propagation delay of both links is 10 msec.



A sends 4 HTTP requests to B, each HTTP responses message is sent back in a 1250-byte packet.
Assume that the size of HTTP requests and TCP SYN/SYN-ACK messages are small enough so that their transmission delay can be ignored. Also assume that TCP flow and congestion control window sizes are big enough so that they do not slow down data transmission. There is also no packet loss.

2.1 (6 points) Assuming the browser uses HTTP/1.0 to retrieve the data. The browser only uses a single TCP connection at any given time. Starting from sending the first TCP connection setup (SYN) packet, how long will it take for the browser to receive all the 4 pieces of data?

TCP setup for each request: SYN: 10+10=20ms, SYN-ACK: 10+10=20ms, total 40ms
Sending one request: 10+10=20ms
Sending one response: $10+10+d_{trans_{B\to R}}+d_{trans_{R\to A}} = 10+10+1250(8\text{b})\left(\frac{1s}{10^6}\right)+1250\left(\frac{8}{1}\right)\left(\frac{1s}{2\times10^6}\right) = 20.015ms$
Time for one request/response: 40+20+20.015 = 80.015ms
Time for 4 pieces of data: 4×80.015 = 320.06 ms

2.2 (6 points) To speed up the retrieval, the browser opens 3 TCP connections in parallel. Again starting from sending the first TCP connection setup (SYN) packet, how long will it take for the browser to receive all 4 pieces of data?

SYN+SYNACK for first 3: 40 ms
First 3 requests: 20ms
Responses have queueing delay at R because bandwidth shrinks
First packet arrives at R: $10+1250\left(\frac{8}{2\times10^6}\right) = 10.005$ ms
Delay before 3rd packet leaves R: $3\times1250\left(\frac{8}{100}\right) = 0.03$ ms
Propagation delay for 3rd packet R→A: 10 ms
Last packet is retrived separately: 80.015 ms
Total: 40+20+10.005+0.03+10+80.015 = 160.05 ms

2.3 (4 points) Assuming the browser uses HTTP/1.1 ***without pipelining*** to retrieve the data over a single TCP connection. How long will it take for the browser to receive all 4 pieces of data in this case?

SYN+SYNACK: 40ms (only do setup one time)
each request/response: 20+20.015 = 40.015 ms

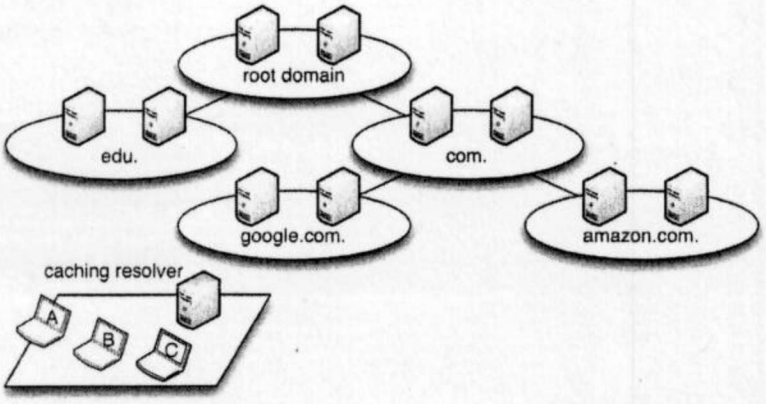
Time for 4 HTTP requests: 40+4(40.015) = 200.06 ms

2.4 (4 points) Assuming the browser uses HTTP/1.1 ***with pipelining*** to retrieve the data over a single TCP connection. How long will it take for the browser to receive all 4 pieces of data in this case? Is the delay the same as the one of parallel connections? If so, why we still prefer HTTP/1.1 with pipelining?

SYN+SYNACK: 40 ms
all requests arrive at server: 20ms
Queuing delay at R because the bandwith shrinks
1st packet arrives at R: $10+1250\left(\frac{8}{2\times10^6}\right) = 10.005$ ms
Delay until 4th leaves R: $4\times1250\left(\frac{8}{100}\right) = 0.04$ ms
4th packet propagation R→A: 10 ms
Total: 40+20+10.005+0.04+10 = 80.045 ms

This delay is approximately the same as if there were 4 parallel TCP connections (1 for each request).
HTTP/1.1 with pipelining is preferred because only one socket is needed, which saves resources.

Problem 3 (20 points) Consider the following DNS resolution process:
at time T=0: the caching resolver in the figure has an empty cache. Host-A sends a query to resolve the DNS name *www.google.com* and get the IP address.
T=30 minutes: Host-B sends a query for the IP address of *www.amazon.com* and gets the answer.
T=70 minutes: Host-C sends a query for DNS name *hangout.google.com* and another query for DNS name *video.amazon.com*.



Assuming that it takes 10 msec for packet resolver (10 msec is the round trip delay), and it takes 100 msec for the caching resolver to get a reply from any of the authoritative DNS servers. All authoritative servers support iterative queries only. All the DNS data has a TTL value of 1 hour. There is no packet loss.

3.1 (4 points) How long does it take for Host-A to get the answer back for the IP address of *www.google.com*?

RTT between Host A and caching resolver: 10ms
Caching resolver queries .com: 100ms
queries google.com: 100ms
queries www.google.com: 100ms
Total: 10 + 100 + 100 + 100 = 310 ms

3.2 (4 points) How long does it take for Host-B to get the answer back for the IP address of *www.amazon.com*?

RTT between host and caching resolver: 10ms
.com is cached: 0 ms
query amazon.com: 100 ms
query www.amazon.com: 100ms
Total: 10 + 100 + 100 = 210 ms

3.3 (3 points) How long does it take for Host-C to get the answer back for the IP address of *hangout.google.com*?

RTT between host and caching resolver: 10ms
.com exceeded its TTL, so query .com again: 100 ms
google.com exceeded its TTL, so query again: 100 ms
query hangout.google.com: 100 ms
Total: 10 + 100 + 100 + 100 = 310 ms

3.4 (3 points) How long does it take for Host-C to get the answer back for the IP address of *video.amazon.com*?

RTT host to caching resolver: 10ms
.com cached: 0 ms
amazon.com cached: 0 ms
query video.amazon.com: 100ms
Total: 10 + 100 = 110 ms

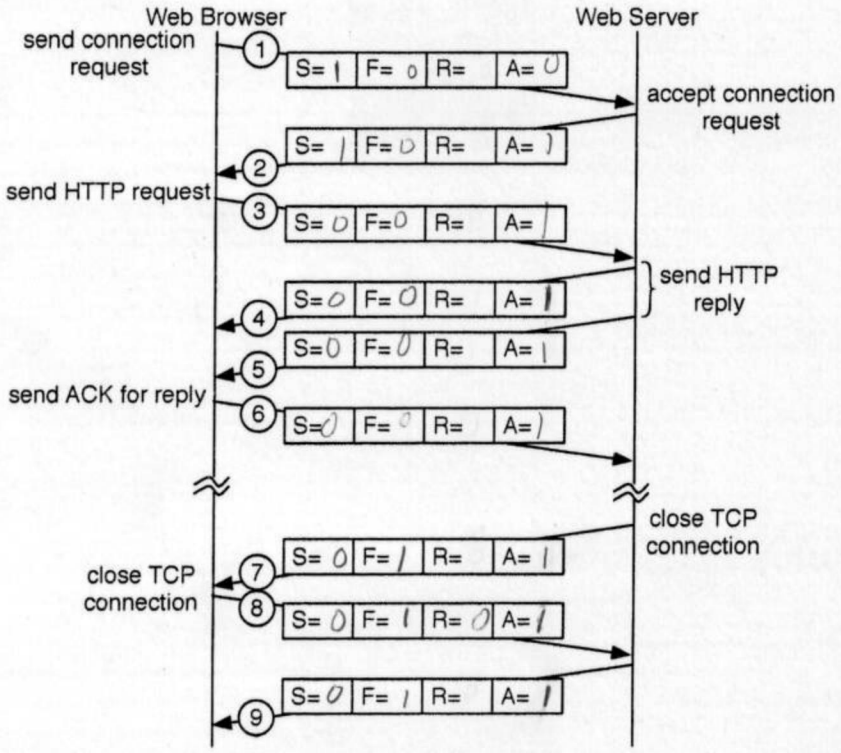
3.5 (6 points) At T=100 minutes, all the authoritative servers of .com go offline. Which domain names below can be resolved by Host-A? Circle those domain names:

(a) www.google.com (b) hangout.google.com (c) doc.google.com
(d) www.amazon.com (e) video.amazon.com (f) aws.amazon.com

google.com cached, amazon.com is not

Problem 4 (20 points) The following diagram shows a sequence of TCP packets for a session between a web browser and a web server. The HTTP in use is version 1.0 (non-persistent HTTP).

4.1 (6 points) Fill in all the missing flag values for the SYN, FIN, RST, and ACK flags in the TCP headers (when the flag is set, the value is 1, otherwise is 0).



4.2 (8 points) If the web browser starts its TCP connection with the initial sequence number 308, and web server picks 1110 as its initial sequence number, the HTTP request size is 150 bytes, and the HTTP reply is made of 2 packets with 1500 byte data each. What is the sequence number and acknowledge number on the ***numbered*** packets?

No	Sequence No.	Ack No.
1	308	--
2	1110	309
3	309	1111
4	1111	459

No	Sequence No.	Ack No.
5	2611	459
6	459	~~2612~~
7	2612	~~460~~
8	~~460~~	2613
9	2613	461

4.3 (3 points) Why the sequence number at each end of a TCP connection starts from a random number, instead of zero?

This makes sure that both hosts have actually established a connection, starting from a fixed number each time makes the server vulnerable to a SYN flood attack, where a client can pretend to establish a lot of connections and overload the server.

4.4 (3 points) How does the web server know that the browser has received the last packet (packet 9)?

After sending packet 9, the server keeps the connection open for some amount of time. If nothing else is received in that time, the server can assume that the browser has closed its end of the connection.

Problem 5 (20 points)

5.1 (4 points) You have learned four application layer protocols: HTTP, FTP, SMTP, and DNS. Only one of them can run over UDP. Which protocol is it? Why it is preferred to run over UDP than TCP? **(in one sentence, otherwise you will not get any credit)**

DNS. UDP is better because it is faster, and DNS does not require reliable communication.

5.2 (4 points) If you are asked to develop a real-time online conferencing application, will you choose TCP as the transport layer protocol? Justify your answer.

No. Less latency is more important than reliable transport. Some packet loss or corruption is okay for voice-streaming applications because humans are able to infer what is missing from a noisy channel, so the application is still usable if it has unreliable transport.

5.3 (4 points) You went to amazon.com website and Chrome shows you the above state in the address bar. In which case you can safely send your Amazon login and password information and why? If in some cases it is not safe, list those and explain why is it not safe and/or what could have gone wrong.

(a)

(b)

(c)

Safe, the communication is encrypted with HTTPS. Only the server has the private key and can decode requests.

Unsafe. Communication uses unencrypted HTTP, so passwords are sent in plaintext. This happens if the website does not use a SSL/TLS certificate.

Maybe safe. This state means that the site uses HTTPS, but some items were loaded over HTTP. The page could reference an object that uses HTTP. Your data could be send with HTTPS, or not.

5.4 (4 points) Some major email service providers recently announced that they have adopted HTTPS-like approach to secure the email communication (each connection between client and SMTP server and between SMTP servers is secured using HTTPS-like connection). Do you think their solution can secure email communication and eliminate all spam? Justify your answer.

This secures email communication assuming that all SMTP servers cooperate. If the emails are encrypted and decrypted between every SMTP server jump, a rouge server could look at the plaintext email before sending it on. This is not an issue if emails are signed, eg. with PGP. This does not eliminate spam because anyone can send you an email if they have your email address, and securing connections does not check for spam.

5.5 (4 points) HTTP 1.1 already allows a client to send multiple requests in a single connection. Why we still need multiple streams in HTTP 2.0?

Multiple streams allow for further optimization. The streams can be interleaved based on which packets are available at a given time. For example, suppose the server is sending several very large files at once. In HTTP/1.1, the files must be sent roughly in order so that the sent packets are within the receiver window. In HTTP/2.0 with multiple streams, parts of any file can be sent whenever the packets are ready, increasing utilization of the server and network.