CS118
Spring 2016 Midterm Exam

1 hour 50 minutes

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

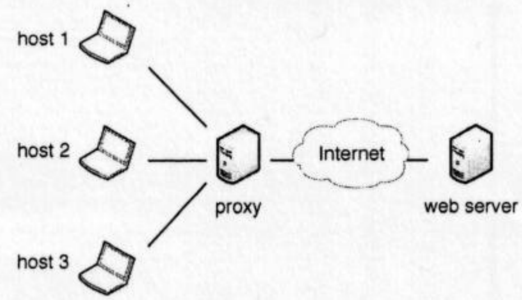
- This exam has 6 pages, including this cover page. Do all your work on these exam sheets.
- Cross out all the scratch work that you do not want to be counted as part of your answer before you submit the exam.
- Show ***all*** your work, including unfinished problems that you wish to be considered for 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: Srikanth Ashtalakshmi V Rajendran

Student ID: 804478782

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.

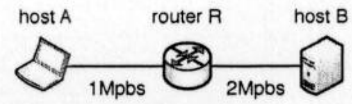
False (True or False) If another host, host 3, sends a request for `http://foo.com:8080/logo.png`, the proxy will not send any HTTP request.

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.

True (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`.

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.

RTT → round trip time t_p → propagation time
t_r → transmission time

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?

~~Total time = 4RTT + 4RTT = 8RTT (4RTT for TCP connection setup, 4RTT for request and response)~~

Total time for TCP connection = (10 ms + 10ms) × 2 × 4 = 160 ms

t_r for packets $= 4\times\left(\frac{1250\times8}{10^6}+\frac{1250\times8}{2\times10^6}\right) = 0.06s = 60\,ms$

t_p for requests & responses $= 4\times2\times(10ms+10ms) = 160\,ms$

Total time = 380 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?

Total time for TCP connection $= 2\times2\times(10ms+10ms) = 80\,ms$

t_r for packets $= \left(\frac{1250\times8}{10^6}+\frac{1250\times8}{2\times10^6}\right)\times(3+1) = 60\,ms$ → * Bandwidth is shared in parallel TCP connection

t_p for requests & responses $= 2\times2\times(10ms+10ms) = 80\,ms$

Total time = 220 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?

Time for connection $= 2\times(10ms+10ms) = 40\,ms$

$t_r + t_p$ for requests and responses = 60ms + 160 ms = 220 ms

Total time = 260 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?

Time for requests to arrive = 10 ms + 10 ms = 20 ms

Time for connection = 40 ms

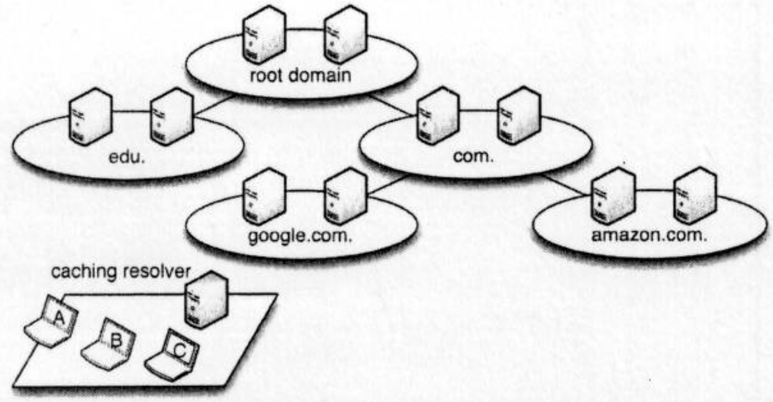
Time after which last packet leaves $= 3\times\left(\frac{1250\times8}{10^6}\right) = 0.03\,s = 30\,ms$

Time for last packet to reach client $= \frac{1250\times8}{10^6}+\frac{1250\times8}{2\times10^6}+0.02 = 35\,ms$

Total time = 125 ms

No, the delay is not the same as that of parallel connections

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*?

Total time = 10 ms + 100 ms + 100 ms + 100 ms = 310 ms

10 ms for RTT between host A and caching resolver
100 ms for reply from root domain
100 ms for reply from TLD server (com.)
100 ms for reply from google.com

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*?

Total time = 10 ms + 100 ms + 100 ms = 210 ms

No need to query root domain, as caching resolver already knows the IP address of com. TLD server.

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*?

Total time = 10 ms + 100 ms + 100 ms + 100 ms + 100 ms = 410 ms

the caching resolver has removed data about www.google.com after 60 mins.

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*?

Total time = 10 ms + 100 ms + 100 ms = 210 ms

to query amazon.com

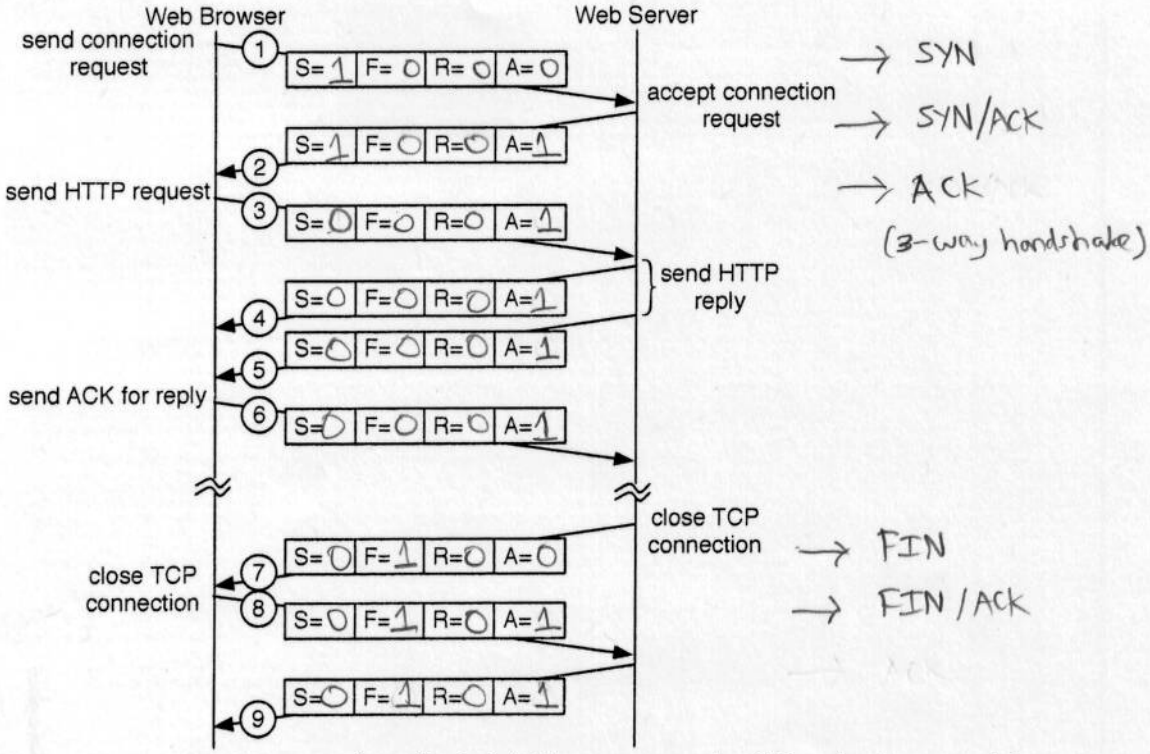
to query video.amazon.com

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`

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?

309
+150
+ 1
460

No	Sequence No.	Ack No.
1	308	--
2	1110	309
3	309	1111
4	1111	460

1111
+1500
+1500
+ 1
4112

No	Sequence No.	Ack No.
5	2611	460
6	460	4112
7	4112	—
8	461	4113
9	4113	462

SYN, ACK and FIN consume one sequence no.

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

This is because there may be packets from previous TCP sessions that could finally make its way to the client. By starting a new session with a different random number, the client-side can discard packets from earlier sessions.

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

The server knows that the browser has received the first FIN message (pkt 7). The last packet (packet 9) just acknowledges that the server has obtained the FIN/ACK from the browser. Hence, the server does not know if ~~the~~ packet 9 reached, but it still doesn't affect the closing process.

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**)

only DNS can run over UDP, and this is preffered because UDP allows for a quicker response as it doesn't have as much overhead as TCP has.

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.

I would not choose TCP as TCP has a large overhead and it ensures that all data packets reach the other end in the same order. However, this feature of TCP could reduce the speed at which data is transferred, thereby negatively affecting the quality of the conference calls.

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)

(a) It is safe to send in this case, as this address has been certified by a certificate authority.

(b) It is probably safe to send the password, but we know there isn't an extended domain name certification

(c) It's unsafe to send such details in this case as there is no certification of the domain name

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.

No, as spam messages could be sent by certified senders, but the receiver may not desire this message. The user's preferences would have to be understood to prevent 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?

HTTP 2.0 needs multiple streams so that data chunks with higher priority can be delivered to the client even if one of the streams is congested. This mechanism ensures a large chunk does not cause other chunks with higher priority to wait for its transmission.