Prof. Athina Markopoulou
Fall 2016

Introduction to Computer Networks
EECS148/CompSci 132

Midterm Solution Sketch

Student's Last Name:

Student's First Name:

UCINETID:

Seat:

Problem #	Points	Out of Total
1		30
2		40
3		30
Total		100

Academic Honesty Policy:
I agree to abide by the UCI Academic Senate Policy on Academic Honesty (Appendix VIII.B), which specifies that students have responsibility for:

1. *Refraining from cheating and plagiarism.*
2. *Refusing to aid or abet any form of academic dishonesty.*
3. *Notifying professors and/or appropriate administrative officials about observed incidents of academic misconduct. The anonymity of a student reporting an incident of academic dishonesty will be protected.*

Student's Signature:

GOOD LUCK!

1. *(30 Points)* **Packet Switching, Delay and Throughput.**

We are sending a 30 Mbit MP3 file from a source host S to a destination host D. All links in the path between source and destination have a transmission rate of 10 Mbps, unless otherwise noted (*e.g.,* in question (g)). Packet switching is used unless otherwise stated (*e.g.,* as in question (f)). Assume that the propagation speed is $2 \cdot 10^8$ meters/sec, and the distance between source and destination is $10,000$ km.

(a) Initially suppose there is only *one link* between source and destination. Also suppose that the entire MP3 file is sent as one packet. The transmission delay is:

☐ 50 milliseconds
☐ 3.05 seconds
☑ 3 seconds
☐ none of the above

Solution: The transmission delay is $\frac{L}{D} = \frac{30*10^6 bps}{10*10^6 s} = 3sec$.

(b) Referring to the above question, the end-to-end delay (defined from the time S starts transmitting the first bit, until D receives the last bit of the MP3 file) is

☐ 6 seconds
☐ 3 seconds
☑ 3.05 seconds
☐ none of the above

Solution: The E2E delay is transmission delay + propagation delay = 3.05 sec

(c) Referring to the above question, how many bits will the source have transmitted when the first bit arrives at the destination.

☐ 1 bit
☐ 30,000,000 bits
☑ 500,000 bits
☐ none of the above

Solution: The first bit arrives at the destination after one propagation delay (0.05sec), during which $10 \cdot 10^6$ (b/sec) $\cdot 0.05$ (sec) $= 500,000$ bits are sent.

(d) Now suppose there are *two links* between source and destination, with one router connecting the two links. Each link is 5,000 km long. Again suppose the MP3 file is sent as *one packet*. Suppose there is no congestion, so that the packet is transmitted onto the second link as soon as the router receives the entire packet. The end-to-end delay is

☐ 6.1 seconds
☑ 6.05 seconds
☐ 3.05 seconds
☐ none of the above

Solution: The router uses store and forward, and d_{prop} of a link is now $\frac{5000*10^3}{2*10^8} = 0.025sec$. Therefore, $E2E = 2(d_{tr} + d_{prop}) = 2(3 + 0.025) = 6.05$ sec.

(e) Assume again two links with the previous parameters, but now suppose that the MP3 file is broken into *3 packets*, each of 10 Mbits. Ignore headers that may be added to these packets. Also ignore router processing delays. Assuming store and forward packet switching at the router, the total delay is

☐ 3.05 seconds
☐ 6.05 seconds
☑ 4.05 seconds

☐ none of the above

Solution: $d_{tr} = \frac{1}{3} \cdot 1 = 1$ (sec). $d_{prop} = 0.25$sec. The last bit of the first packet arrives after $2 \cdot (d_{prop} + d_{tr}) = 2 \cdot 1.025 = 2.05$sec. There are two more packets to be received, each with $d_{tr} = 1$sec. Therefore, $E2E = 2.05 + 2 \cdot 1 = 4.05$sec.

(f) Now suppose there is only *one link* between source and destination, and there are 10 TDM channels in the link. The MP3 file is sent over one of the channels, using circuit switching. Ignore setup time, *e.g.,* to allocate channels to connections. The end-to-end delay is

☑ 30.05 seconds
☐ 30 seconds
☐ 300 microseconds
☐ none of the above

Solution: In TDM the bandwidth is divided equally between the 10 users and each circuit uses $BW = \frac{10Mbps}{10} = 1Mbps$. So $d_{tr} = \frac{30MB}{1Mpbs} = 30sec$, and $E2E = d_{prop} + d_{tr} = 0.05 + 30 = 30.05$ sec.

(g) Suppose there are *two links* between S and D. The first link has transmission rate 100 Mbps and the second link has transmission rate 10 Mbps. Assuming that the only traffic in the network comes from the source, what is the throughput for a large file transfer?

☐ 110 Mbps
☐ 100 Mbps
☑ 10 Mbps
☐ 1 Gbps

Solution: The bottleneck link here is the second link with 10 Mbps, so the throughput is limited to 10Mbps.

Consider again the parameters mention in the beginning of this problem. Provide answer and brief justification to the remaining, non-multiple choice, questions.

(h) Consider one link between S and D, as in questions (a)(b). Would it help to break down the file into multiple packets....

- .... in terms of end-to-end delay? Assume the packet header is negligible. If no, explain why. If yes, explain what the optimal number of packets is.
  Solution: No, there is no router involved, so we cannot have pipelining to improve the delay.
- .... in terms of reliability, assuming there is packet loss? If no, explain why. If yes, which reliable data transfer (RDT) protocol would you use and why?
  Solution: Yes, because if we have packet loss, the sender just needs to retransmit lost packets only, instead of the whole message.

(i) Consider two links between S and D, as in questions (d)(e). What is the optimal number of packets to break down the MP3 file so as to minimize the end-to-end delay? Assume negligible packet headers.

Solution: Since there is no overhead due to headers, we should break the MP3 file to as small packets as possible, in order tomaximize pipelining and minimize delay. At the extreme, this would mean that we should break the file into as many packets as bits. (This is clearly an idealized scenario).

2. *(40 Points)* **Browsing: HTTP, TCP, DNS**

I sit with my laptop (called `MacBook-Air-9`) at home, I open my browser and I type "`http://www.mit.edu/`", I press `Enter`, and I wait until the page is displayed.

(a) *(12 Points)* **DNS.** First, the browser needs to make some DNS requests. Assume the browser gets similar results with what I get with the command line (`nslookup`) I ran on my laptop and provided below.

```
MacBook-Air-9:~ athina$ nslookup www.mit.edu
Server: 68.105.28.11
Address: 68.105.28.11#53

Non-authoritative answer:
www.mit.edu canonical name = www.mit.edu.edgekey.net.
www.mit.edu.edgekey.net canonical name = e9566.dscb.akamaiedge.net.
Name: e9566.dscb.akamaiedge.net
Address: 173.223.1.54

MacBook-Air-9:~ athina$ nslookup 173.223.1.54
Server: 68.105.28.11
Address: 68.105.28.11#53

Non-authoritative answer:
54.1.223.173.in-addr.arpa name = a173-223-1-54.deploy.static.akamaitechnologies.com.

Authoritative answers can be found from:

MacBook-Air-9:~ athina$ nslookup 68.105.28.11
Server: 68.105.28.11
Address: 68.105.28.11#53

Non-authoritative answer:
11.28.105.68.in-addr.arpa name = cdns1.cox.net.

Authoritative answers can be found from:
```

---

i. What is the IP address of `www.mit.edu`?

ii. What is the name of the DNS server that answers the DNS request of my browser? Is it a local server for my laptop? Is it an authoritative server for `mit.edu`?

iii. List the DNS requests and DNS responses, in the sequence they occur, until my browser gets the IP address of `www.mit.edu`.

<ins>Answers:</ins>

i. The IP address of `www.mit.edu` is 173.223.1.54.

ii. The server that answers the DNS request of the browser is the one with name `cdns1.cox.net` and IP address of 68.105.28.11. It is a local server, and is not an authoritative server for the `mit.edu` domain.

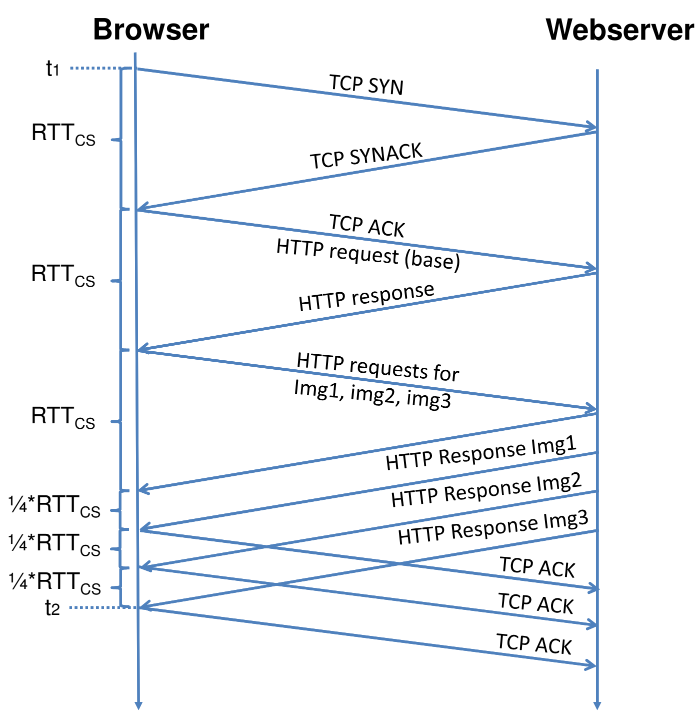
iii. The laptop sends a DNS query (with the name `www.mit.edu`) to the local DNS server `cdns1.cox.net` and this sends a DNS response (with the IP address 173.223.1.54) back to the laptop. The local DNS server must learn this information, since it is not authoritative for the domain `mit.edu`. If this is the first time that a request for domain `mit.edu` is seen by the local DNS server, it must send a DNS query to another DNS server (*e.g.,* TLD, which in turn may ask the Authoritative) to resolve it; if the record is already cached at the local DNS server, it does not need to send such a request. Interestingly, it seems that `www.mit.edu` is hosted by the Akamai CDN.

(b) *(16 Points)* **Timing of HTTP, TCP messages.**

Assume that the MIT webpage consists of a base HTML file (`http://www.mit.edu/index.html`) and 3 images. Let RTT_{cs} be the RTT between my laptop and the MIT webserver. The base HTML file is small, with negligible transmission delay. However, the images are large: (an HTTP response for) each image fits in exactly one TCP segment (each with the maximum segment size, MSS Bytes) and transmission delay $\frac{1}{4} \cdot RTT_{cs}$. The transmission delays of all other messages are negligible. (Additional simplifying assumptions: no packet is lost; ignore the TCP window effect; ignore any processing delays; all TCP segments have either negligible or maximum (MSS) size, as specified above.)

Consider that your browser uses **persistent HTTP with pipelining.** Complete the diagram below. Indicate the *TCP connection establishment segments* and the *HTTP messages*, exchanged right after DNS is resolved (t_1) and before the webpage is ready to be displayed by the browser (time t_2). How long is this delay $t_2 - t_1$?

<u>Answer:</u>



Total time until the browser has all the information to display the page is: $t_2 - t_1 = \frac{15}{4} \cdot RTT_{cs}$. Please note that at that point the TCP connection is not complete (still ACKs are sent and the connection is not closed.

Since the protocol is persistent, the same TCP connection is used to get all the objects. Three different HTTP requests are sent, one for each of img1. img2, img3, serially, one after the other. However, because their transmission delay is assumed negligible, they are drawn one on top of the other on the figure. In contrast, the HTTP responses (with img1, img2, img3 respectively, and also send serially over the the same TCP connection) have non negligible transmission delay $RTT/4$, thus they are easier to distinguish on the figure.

Finally notice that the third part of the TCP handshake (TCP ACK set to 1) also carries data (the HTTP Request for index.html) in the same packet.

(c) *(12 Points)* **Content of HTTP Request/Response.** I imitated my browser and sent an HTTP request using the following command lines from my laptop:

```
MacBook-Air-9:~ athina$ telnet www.mit.edu 80
Trying 2600:1406:1f:18c::255e...
Connected to e9566.dscb.akamaiedge.net.
Escape character is '^]'.
GET /index.html HTTP/1.1
Host: www.mit.edu
```

Below is the HTTP Response I got from the webserver:

```
HTTP/1.1 200 OK
Server: Apache/1.3.41 (Unix) mod_ssl/2.8.31 OpenSSL/0.9.8j
Last-Modified: Mon, 24 Oct 2016 04:01:21 GMT
ETag: "10e87af6-4b88-580d8791"
Accept-Ranges: bytes
Content-Length: 19336
X-Cnection: close
Content-Type: text/html
Date: Mon, 24 Oct 2016 05:15:54 GMT
Connection: keep-alive

<!DOCTYPE html>
<!--[if lt IE 7]>      <html class="no-js lt-ie9 lt-ie8 lt-ie7">  ....
......
......much more document text following here (not shown)....
......
MIT.clicktrack.init();
</script>

</body>
</html>
```

Answer the following questions and briefly justify it by referencing the HTTP Response.

i. Was the server able to successfully find the document or not?

ii. At what date and time was this response sent?

iii. When was the document last modified?

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

v. What are the last 5 bytes of the document being returned?

vi. Is the connection persistent or nonpersistent?

Solution:

i. Yes, the server responses by code 200 and phrase OK.

ii. Response was sent at Mon, 24 Oct 2016 05:15:54 GMT

iii. The document last modified was at Mon, 24 Oct 2016 04:01:21 GMT

iv. There are 19336 bytes in the document being returned

v. The last 5 bytes of the document being returned are “$html >$”

vi. The connection is persistent as indicated by `Connection: keep-alive`.

3. *(30 Points)* **Reliability: Go-back-N vs. Selective Repeat.**

Suppose that host A sends $N = 4$ data packets to host B, using one of the idealized pipelined protocols we learnt in class: (a) Go-Back-N (GBN) and (b) Selective Repeat (SR). All protocols have (sender and/or receiver if applicable) window sizes equal to 3 packets and Timeout value equal to 8 ms (assume that the timeout starts *after* the packet is sent). The one-way propagation delay between host A and B, as well as from B to A, is 2.5 ms, the transmission delay of each packet is 1ms, and the transmission delay of each ACK packet is 0 ms (negligible).

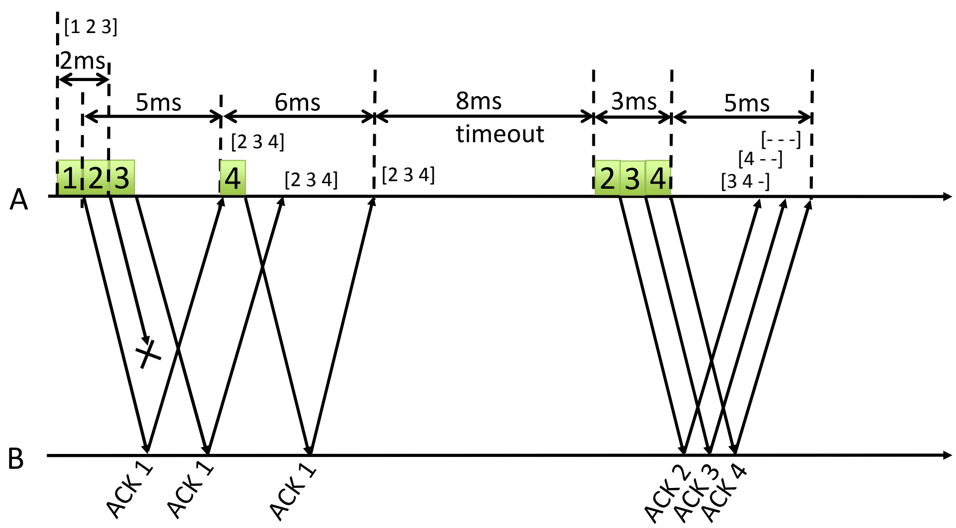
Suppose that Packet No. 2 is lost on the way from host A to host B and no other packet gets lost. The beginning of the space-time diagrams is provided in the diagrams below; the sequence numbers refer to packets.

Fill out the rest of the diagrams until all 4 packets are sent and acknowledged. Show all packets (transmissions, retransmissions and acknowledgements), their sequence numbers and the times they were sent/received. For each scenario, write down the total time, i.e., when the all 4 packets are successfully acknowledged at the server.

(a) *(15 Points)* **GBN with Packet 2 lost.**

Answer: Please see the state machines in Fig. 3.20. 3.21, for a justification

Total time = 28 ms



(b) *(15 Points)* **Selective Repeat with Packet 2 lost.**

Answer: Please see the state machines in Fig. 3.24. 3.25, for a justification of the figure.

Total time = 16 ms