EECS 489: Midterm Exam, Winter 2017

Duration: 72 Minutes

OFFLINE, open-book, open-notes exam. You may consult the course textbooks, Kurose and Ross, Computer Networking: A Top-Down Approach, any edition, your own notes, notes from discussion sections, programming assignments, your solutions to them, any solutions provided by the instructors, and other notes provided by the instructors, including the course lecture slides. The above listed material may be accessed in hard copy or electronic form, offline. A calculator or a calculator program is also permitted for the exam. These are the only uses of a computer permitted. Any other use of the computer and accessing the Internet/local network is strictly forbidden. You are also not allowed to compile and run any programming code during the exam. You must not consult any resources other than those listed above.

Write legibly. If the grader cannot read what youve written, they will assume that you meant the illegible portion as a note to yourself and will ignore it. If you lose points because part of your answer is illegible, you will not be given the opportunity to explain what it says. Illegible scribble will earn zero points.

Do not ask questions during the exam. They disturb other students. Figuring out what the question is asking is part of the exam. If you think you have to make some assumption to answer a problem, note your assumption on the test. The answers to most questions should be short; you need not use all the space provided to answer the questions. If you find yourself writing an excessively long response, you may want to think more carefully about the question.

Write your uniqname on the lower-left corner of every page.

Honor code pledge. You are to abide by the University of Michigan/Engineering honor code. Sign below to indicate that you have kept the honor code pledge.

"I have neither given nor received unauthorized aid on this examination, nor have I concealed any violations of the Honor Code."

Name:

Signature:

Uniqname:

	Q1	Q2	Q3	Q4	Q5	Q6	Total
Grade	/21	/11	/10	/10	/10	/10	/72

1 [21 points] True or False; with Justification

3 points for a correct answer with justification; **1 point** for a correct answer with incorrect/poor justification; **-1 point** for a wrong answer.

(i) *(True / False)* The Open Systems Interconnection model started with seven layers, but we use only five layers in practice today.
As a result, we cannot reap the benefits of session and presentation layers.

False. We can implement many of their functionalities in the application layer.

(ii) *(True / False) Caching is useful for web pages that contain only static content.*

False. They can still serve the static parts of a web page that also includes dynamic content. Moreover, some CDNs can perform some dynamic processing on behalf of the content provider.

(iii) *(True / False)* A CDN usually uses DNS to direct clients to replicas in a round-robin fashion. Despite the use of round-robin, certain replicas can become more heavily loaded than others.
The root cause behind this imbalance is not taking client latencies from the CDN servers into account.

False. The root cause is different clients imposing different amount of workload.

(iv) *(True / False)* An application has to send 100 KB of data using a stop-and-wait reliability protocol, which splits the data into segments that have a 1 KB application data payload. Each segment fits in a single IP packet. The RTT is 50 milliseconds that is equally divided in each direction. There is no transmission, queueing, or processing delay, and the network does not drop, duplicate or corrupt any packets.
Given that the transmission only incurs propagation delay, it will take only 25 milliseconds to send the entire 100 KB.

False. It will take 5 seconds (50 * 100 ms).

(v) *(True / False) IPv4 can work over underlying link layer technologies that have different MTUs.*

True. It supports fragmentation.

(vi) *(True / False)* Consider two TCP/IP flows: one has to transfer 1 KB and the other 1 GB of data. Now consider a datacenter network with 1 millisecond RTT and a satellite network with 1 second RTT; everything else in both networks are the same (e.g., same bandwidth, no loss, and no corruption etc.). *Although the short flow will complete faster in the datacenter network in comparison to the satellite network, the long flow will take the same amount of time in both networks.*

False. Long flow will also take longer because TCP throughput depends on RTT.

(vii) *(True / False)* IPv6 packet headers have fixed size and thus are more efficient to process. *However, because an IPv6 header uses 128-bit source and destination addresses instead of 32-bit ones, it is larger than any IPv4 header.*

False. IPv4 header can be as long as 60 bytes, whereas IPv6 packets are always 40 bytes.

2 [11 points] Browsing the Web

You visit the following URL using your web browser:

`http://cse.umich.edu/eecs/faculty/csefaculty.html`

The page lists all the faculty in the CSE division. To know more about what they have accomplished, you type the following URL into your web browser:

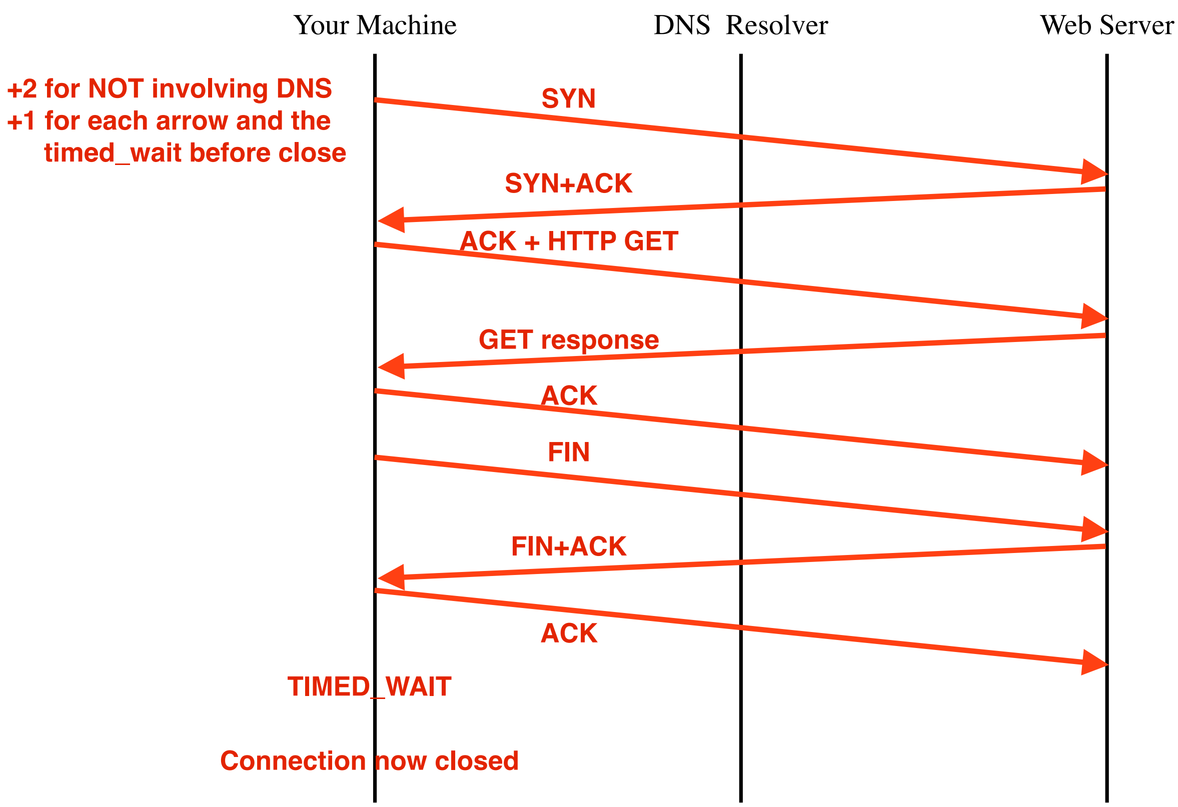
`http://cse.umich.edu/eecs/awards/csefacultyawards.html`

You close the tab after a minute of browsing the page.

Assuming that

- your DNS resolver is located it 192.12.80.214,
- your browser and DNS has caching enabled,
- DNS always uses UDP, and
- the HTML response returns 200 OK with a web page,
- the HTML request and response each fit in a single segment, and
- the web page requires loading no additional resources,

draw below the series of packet exchanges that will occur for your host to receive **only** the second web page. Include **all packets** – control and data – from relevant protocols. You do not need to write down message formats. Each packet is a labeled arrow, where the label has the *protocol name and message type* and the arrow starts from the source and ends at the destination for that packet. In the case of the HTTP request, clearly state the path of the file requested in the GET.



3 [10 points] Watching Videos

You want to watch a video that is 640 MB in size from your home. The video can be watched in only one quality setting in the video player of your browser: at 16 MB/s rate. Your browser will be receiving the video via a proxy, and your machine is connected to the proxy by an infinite capacity link. The video itself is stored as 1 MB fragments in 3 CDN servers that the proxy knows about. The video is fully replicated across all CDN servers. The bandwidth between the proxy and the CDN servers are 2 MB/s, 3 MB/s, and 5 MB/s.

(i) **[5 points]** How long the video must be buffered so that you can watch the video without any interruption once the video starts playing?

Video duration = 640/16 = 40 s; Download time = 640/x s; buffer for (640/x-40) s;
x = 10 Mb/s in the best case (proxy can use all servers in parallel to increase throughput!).
So, buffer for 24 sec.

+3 Correct approach
+2 Using all three servers

(ii) **[1 points]** As soon as you finish watching, your roommate watch the same video from her own laptop. However, she does not experience any buffering at all! What is the likely cause?

Proxy caching

(iii) **[2 points]** You then go to the department and show your best friend the same video from the same browser you watched the video on. Unlike your roommate, your friend experience the same buffering as you did earlier. Why?

+1 Different proxy
+1 the video was removed from your browser cache

(iv) **[2 points]** The next day, you try to watch the same video again connecting to the same proxy. You end up waiting as long as the video takes to play – even longer than the day before! What exactly went wrong?

Link bandwidth changed OR a server failed OR both.

4 [10 points] Sending Small Messages

Two endpoints (A and B) in a voice-over-IP session are connected by a path of 4 routers. All links are running at 10 Mb/s, and the hosts are separated by 3000 km. All packets are of size 1500 Bytes. Assume the bit propagation speed is 2×10^8 m/s. Assume 1 Mb/s = 10^6 bits/s.

Show calculations for each of the following questions.

(i) What is the minimum round trip time (RTT), assuming there is no queueing and processing delay?

RTT = 2*(propagation + transmission * 5 hops);
propagation = 3000km/2x10^8= 15 ms; transmission = 1500 Bytes / 10 Mbps = 1.2 ms;
So, RTT = 2*(15+1.2*5) = 2 * 21 = 42 ms

(ii) For this part only let us assume that one router on the path has a steady queue occupancy of 5 packets. What is the end-to-end delay (one way, not round trip) in this case?

Avg. queueing delay = 5 * 1.2 = 6 ms (i.e., 5 * time to transmit each packet)
so, end-to-end delay = 6 + 21 = 27 ms

(iii) Now let us assume the maximum queue occupancy for *every* router queues is 5 packets. What is the maximum end-to-end delay (one way, not round trip)?

End-to-end delay = 6 * 4 + 21 = 45 ms

(iv) Finally, let us assume that there are no queueing delays, but one of the routers inspects every packet toward B, and it adds 10 milliseconds processing delay. What is the minimum RTT?

RTT = 42 + 10 = 52 ms (processing delay is added once because it only exists in one direction)

5 [10 points] Downloding Large Files

You are downloading a 100 MB file from a server. The bottleneck capacity between your machine and the server is 1 MB/s, and the RTT between them is 1 second.

(i) **[1 points]** What is the minimum time to download this file?

100 / 1 = 100 s

(ii) **[4 points]** Your machine uses TCP Michigan with an MSS of 1 KB. TCP Michigan is different from all other TCP variants in that it does not implement a `ssthresh` for slow start. Furthermore, it treats packet loss and timeout in the same fashion (similar to TCP Tahoe). Approximately, at least how long will it end up taking to download the file using TCP Michigan? Assume 1 MB = 1000 KB.

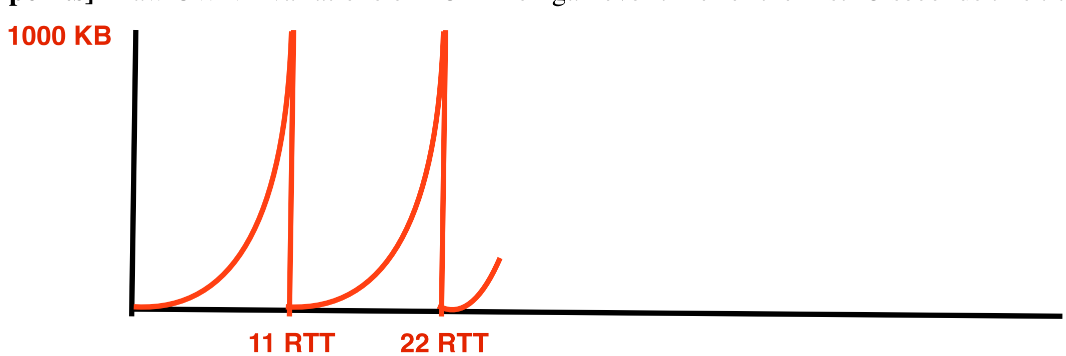
TCP Michigan is a sequence of Slow Starts, where packet loss happens when we reach the link bandwidth.
Roughly speaking, in 11 RTT we get to CWND = 2^10 > 1MB/s bandwidth (2^0=1 on 1st RTT, 2^1=2 on 2nd RTT, and so on)

Approximately, 2^11-1 MSS in 11 seconds => Throughput for each round before loss = 2047/11 KB/s

So, completion time >= 100 MB / (2047/11 KB/s) = 537 s

2023/11 KB/s => ~544 s is also accepted.

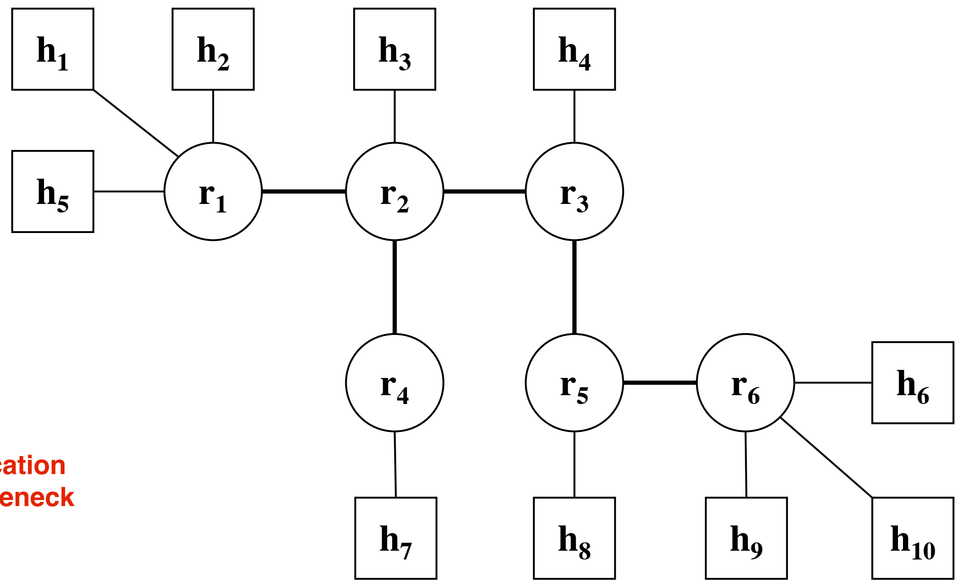
(iii) **[5 points]** Draw CWND variations of TCP Michigan over time for the first 25 seconds this transfer.



6 [10 points] Sharing the Network

Consider the network below with 10 hosts and 6 routers. Each host-to-router link has capacity 100 Mb/s and each router-to-router link has capacity 200 Mb/s.

Given the source, destination, and demand of the following TCP flows ($src \rightarrow dst; demand$), what are the *steady-state* max-min fair share allocations of each of them on the given network? Also write down the bottleneck link (e.g., $r_5 \rightarrow r_6$) that caused the eventual allocation.



For each,

+1.5 Allocation
+1 Bottleneck

(i) $h_1 \rightarrow h_9$; 100 Mb/s
Allocation: 70 Mbps

Bottleneck link: r1 -> r2

(ii) $h_2 \rightarrow h_7$; 60 Mb/s:
Allocation: 60 Mbps

Bottleneck link: None

(iii) $h_3 \rightarrow h_8$; 50 Mb/s:
Allocation: 50 Mbps

Bottleneck link: None

(iv) $h_5 \rightarrow h_6$; 200 Mb/s:
Allocation: 70 Mbps

Bottleneck link: r1 -> r2