EECS 489: Midterm Exam, Fall 2018

Duration: 64 Minutes

OFFLINE, open-book, open-notes exam. You may consult 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 you've 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	Total
Grade	/20	/12	/8	/12	/12	/64

1 [20 points] True or False

2 points for a correct answer; **-1 point** for a wrong answer.

(i) *(True / False) In a circuit-switched network, a connection can send data in a best-effort manner even if the reservation step fails.*

(ii) *(True / False) Queueing delays in a packed-switched network are always negligible.*

(iii) *(True / False) A server must always send the data requested in a HTTP GET request.*

(iv) *(True / False) IP is the narrow waist of the Internet.*

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

(vi) *(True / False) IPv6 packet headers have fixed size; hence, unlike IPv4, IPv6 cannot have Options-like functionalities.*

(vii) *(True / False) Transmission delay does not depend on packet size.*

(viii) *(True / False) Datacenter networks have lower latency than satellite networks.*

(ix) *(True / False) HTTP is stateless.*

(x) *(True / False) TCP provides a reliable, in-order message stream.*

2 [12 points] Web Browsing

You visit the following URL using your web browser:
`http://www.two-pics.com/index.html`
The page show two images (`img1.png` and `img2.png`).

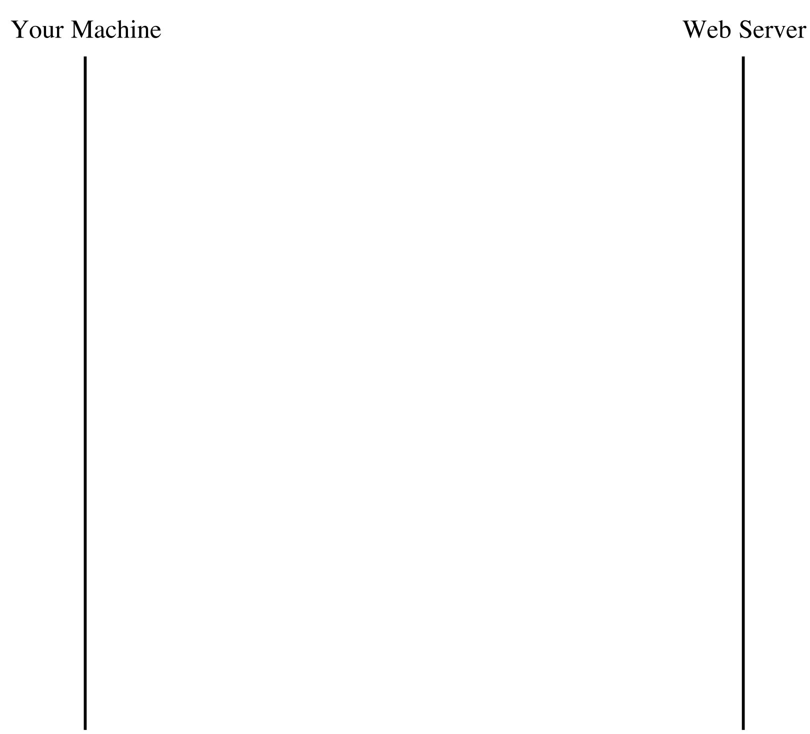
Assume that

- the HTML response returns 200 OK with a web page,
- the HTML request and response each fit in a single packet,
- there is no packet loss on the link,
- the RTT between the localhost and the server is T, and
- the DNS resolutions are already cached.

(i) **[2 points]** With *non-persistent* HTTP and the browser configured for 5 parallel connections, how much time elapses until the browser retrieves all objects?

(ii) **[2 points]** With *persistent* HTTP and the browser configured for 5 parallel connections, how much time elapses until the browser retrieves all objects?

(iii) **[8 points]** Now assume *persistent HTTP with pipelining and the browser configured for a single TCP connection*. Draw below the series of packet exchanges that will occur for your host for this process. Include all packets – control and data – from relevant protocols. Do not include 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.

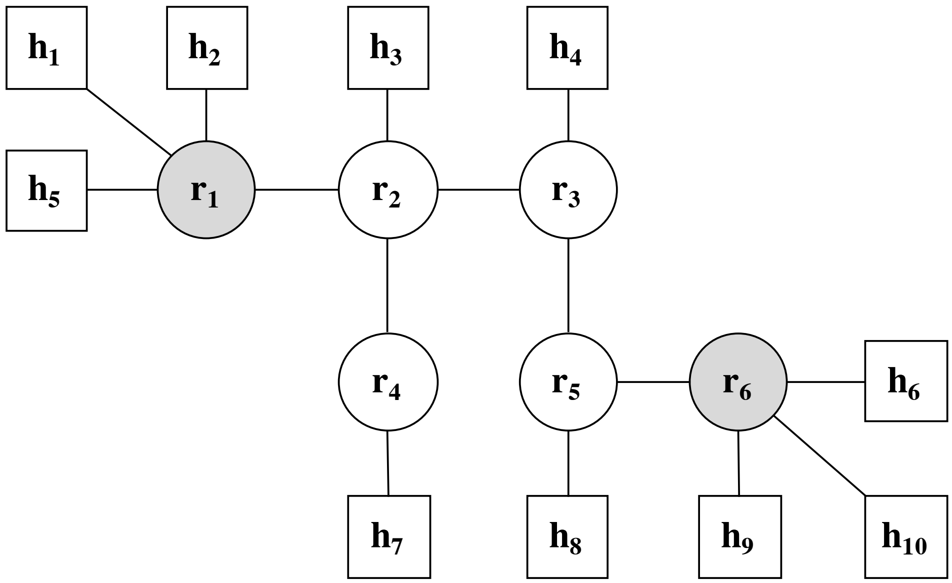


Uniqname:

3 [8 points] Delays in the Network

Consider the network below with 10 hosts and 6 routers. All links have the same propagation delay of 10 ms and same capacity of 80 Mbps. Each of the routers r_1 and r_6 has a processing delay 10 ms, but routers r_2–r_5 have no processing delay. Queueing delays are negligible.

Given the source, destination, and packet size in the following scenarios ($src \rightarrow dst; packet\_size$), what are the *RTTs* of transferring one packet for each of them on the given network? Assume that link MTUs are infinite for all links. Assume 1 Mbps = 10^6 bps.



(i) $h_1 \rightarrow h_7; 1000$ bytes

(ii) $h_2 \rightarrow h_6; 1000$ bytes

(iii) $h_3 \rightarrow h_8; 10000$ bytes

(iv) $h_4 \rightarrow h_9; 10000$ bytes

4 [12 points] TCP Mich

Your friend has designed a new TCP congestion protocol called TCP Mich. It is similar to the congestion protocol we have seen in the class but only has 2 phases: *Slow-Start* phase and *Congestion-Avoidance* phase.

TCP Mich starts in the Slow-Start phase with CWND initially set to 1 (i.e., at RTT 0, CWND = 1), and ssthresh (slow-start threshold) set to 27.

Its actions upon receiving each acknowledgement (Ack) in each of its phases are defined as follows.

- Slow-start phase: For each Ack, CWND += 2
- Congestion-Avoidance phase: For each Ack, CWND = CWND + 2 / CWND

When CWND $\geq$ ssthresh, TCP Mich exits Slow-Start and enters Congestion-Avoidance phase.

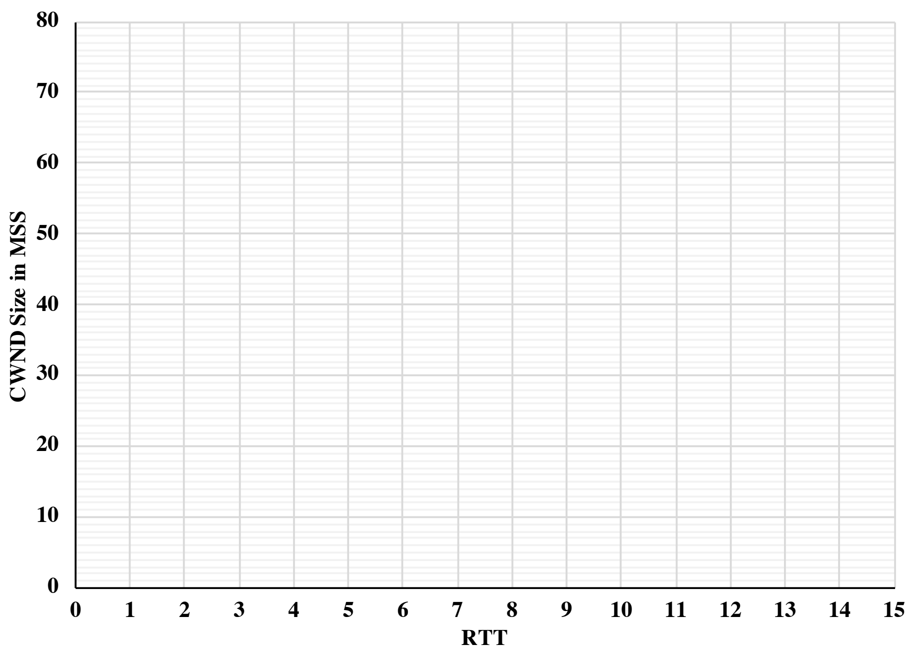
On a packet loss, TCP Mich always goes back to the Slow-Start phase and adjusts as follows.

- ssthresh = CWND / 4
- CWND = 1

Finally, when dividing, TCP Mich *rounds numbers up* to the nearest integer (e.g., 5/4 = 2, 9/2 = 5).

Based on the protocol described above, answer the following questions:

(i) **[7 points]** Using TCP Mich, draw a CWND-size vs. RTT graph for the first 14 RTTs. Assume a packet loss is detected right after the 7th RTT has passed. Clearly label the CWND size value for each RTT.



(ii) **[2 points]** Does TCP Mich ensure equal bandwidth sharing among multiple flows? Briefly explain why or why not.

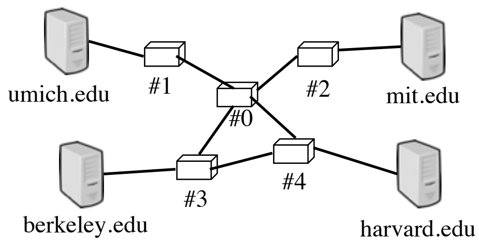
(iii) **[3 points]** Assume no loss happens and header size is negligible. Assume the MSS of your transport protocol to be 1500 bytes and RTT is fixed at 200 ms. Calculate the average throughput (in Mbps) using TCP Mich for the first 5 RTTs. Assume 1 Mbps = 10^6 bps.

5 [12 points] Forwarding Basics

Consider the following network with 4 hosts and 5 routers (and their forwarding tables). Propagation delay for all links is 10 ms. Assume all other delays (transmission, queueing, and processing) to be 0 ms. Assume that link MTUs, link bandwidths are infinite for all links.

Unless otherwise specified below, assume that the IP header has only 2 bits for TTL.

Given the source and destination in the following scenarios ($src \rightarrow dst$), what are the one-way delays (not RTT) of transferring one packet for each of them on the given network? Also write down the path (e.g., $A \rightarrow B \rightarrow C$) taken by each packet.



Forwarding Table at #0

Destination	Next Hop
UMICH	1
Berkeley	3
MIT	2
Harvard	3

Forwarding Table at #1

Destination	Next Hop
UMICH	1
Berkeley	0
MIT	0
Harvard	0

Forwarding Table at #2

Destination	Next Hop
UMICH	0
Berkeley	0
MIT	2
Harvard	0

Forwarding Table at #3

Destination	Next Hop
UMICH	0
Berkeley	3
MIT	0
Harvard	4

Forwarding Table at #4

Destination	Next Hop
UMICH	0
Berkeley	0
MIT	0
Harvard	4

(i) umich.edu $\rightarrow$ mit.edu
One-way latency:

Path:

(ii) berkeley.edu $\rightarrow$ harvard.edu
One-way latency:

Path:

(iii) mit.edu $\rightarrow$ harvard.edu
One-way latency:

Path:

(iv) harvard.edu $\rightarrow$ umich.edu
One-way latency:

Path: