CSEE 4119: Computer Networks
Sample Midterm Exam– Solutions

March 2019

Name:
UNI:

Question 1		
Question 2		
Question 3		
Question 4		
Total	100	

Please Read Carefully Before You Start:

- The time limit is 1 hour and 20 minutes.
- If additional space is needed, use the back of the page for each problem.
- Show your work and clearly write all the steps, otherwise you will not get full credit.

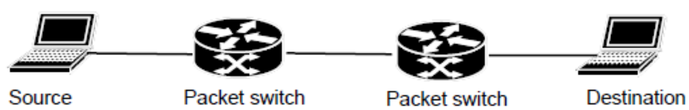
Question 1. (a) Describe one advantage and one disadvantage that circuit-switched networks have over packet-switched networks.

advantage: can guarantee a certain amount of end-to-end bandwidth for the duration of the circuit. Most packet-switched networks today cannot make any end-to-end gurantees for bandwidth. Disadvantage: especially for short-lived flows, the set up overhead of the circuit can consume relatively high overhead. Also, the reserved bandwidth if not used is wasted.

(b) Consider sending a packet from a sending host to a receiving host over a fixed route. List the delay components in the end-to-end delay computation. Which of these delays are constant and which are variable?

The delay components are processing delays, transmission delays, propagation delays, and queueing delays. All of them are fixed for a fixed-size packet except queueing delay.

(c) In modern packet-switched networks, the source host segments long, application-layer messages (for example, an image or a music file) into smaller packets and sends the packets into the network. The receiver then reassembles the packets back into the original message. We refer to this process as message segmentation. The figure below illustrates a switched network. Consider a message that is $7.5 * 10^6$ bits long to be sent from the source to the destination. (Assume header size is negligible relative to the entire message size). Suppose each link is 1.5 Mbps. Focus on transmission delays only and assume all other delay components are negligible.



(1) Consider sending the message from source to the destination without message segmentation. How long does it take to move the message from the source host to the first packet switch? Keep in mind that each switch uses store-and-forward packet switching. What is the total time to move the message from source to the destination host?

$7.5 * 10^6/(1.5 * 10^6) = 5sec$
Total delay= 5 sec $\times$ 3 hops = 15 sec.

(2) Now suppose that the message is segmented into 5000 packets, with each packet being 1500 bits long. How long does it take to move the first packet from source to the first switch? When the first packet is being sent from the first switch to the second switch, the second packet is being sent from the source host to the first switch. At what time will the second packet be fully received at the first switch?

$1500/(1.5 * 10^6) = 1$ msec
Time at which second packet is received at the first switch is the time at which first packet is received at the second switch: 2×1 msec $= 2$ msec.

(3) How long does it take to move the file from source host to destination host when message segmentation is used?

Time at which the first packet is received at the destination host is 3 msec. After this, every 1 msec one packet will be received; thus time at which the last or 5000th packet is received is 3msec + 4999 * 1msec = 5002 msec.

(4) Compare this result with our answer in part (1) and give at least one advantage and disadvantage of message segmentation.

Using segmentation, delay is significantly less than the delay in (1) – without segmentation. A drawback is that packets have to put in sequence at the destination which requires adding header to packets. The advantage is that because of store-and-forward, the overall delay in transferring the message is lower due to parallelism.

Question 2. Consider an http client that wants to retrieve a WWW document at a given URL. The IP address of the http server is initially unknown. The WWW object at the URL has one embedded GIF image that resides at the same server as the original object.

(a) What transport and application layer protocols besides http are needed in this scenario?

DNS is needed to determine the IP address of the server. TCP is used to carry the HTTP request; UDP is used to transport the DNS messages

(b) Suppose that the time needed to contact and receive a reply from any server (for any protocol) is RTT. How many RTTs are needed from when the user first enters the URL until the complete document is displayed? Assume that non-persistent http is used. Consider the delays of all protocols in your answer, not just those of http.

One RTT to do DNS, one RTT set up 1st TCP connection to WWW server, one RTT to get base HTML. one RTT to setup 2nd TCP connection, one RTT to get the embedded GIF. Total= 5 RTT

(c) In order to deal with the issue of scale, the DNS uses a large number of servers, organized in a hierarchical fashion. Write the names of different levels of this hierarchy.

root DNS servers, top-level domain (TLD) DNS servers, and authoritative DNS servers

(d) In a DNS server, what is the format of a resource record? it is a four-tuple that contains the following fields: (Name, Value, Type, TTL)

What is a type A record? Name is hostname and Value is its IP address

What is a type NS record? Name is a domain and Value is the hostname for the authoritative DNS server for that domain

(e) Write a simple message in the HTTP protocol, to retrieve the object with the following URL:

http://www.someSchool.edu/someDepartment/picture.gif

GET /someDepartment/picture.gif HTTP/1.1
Host: www.someSchool.edu

Question 3. Part One: TCP vs. UDP [8 points]

(a) List three services provided by TCP that are not provided by UDP.

Reliability, In-order delivery, Congestion control, and others

(b) [5 points] Indicate whether you think TCP or UDP would be better suited for each of the following applications and briefly explain why. State any assumptions that you are making for each application.

• Streaming video client/server

UDP. reliability is not needed, but a guaranteed transfer rate is. UDP will give a higher-quality transmission if there are limited network resources.

• Multiplayer online first-person shooting game

Probably UDP. reliability is needed here, but so is low latency. A case could be made for both, but many games use UDP in order to achieve the low latency.

• Chat client/server

Definitely TCP. reliability, in-order delivery are needed, and bandwidth/ latency guarantees are not.

• Internet telephony

UDP. Same as answer to streaming video: reliability is not needed, but a guaranteed transfer rate is. UDP will give a higher-quality transmission if there are limited network resources.

• A protocol designed to synchronize the clocks of computers over a network, what protocol should be used for packets exchanged to identify time differences?

Most likely UDP: designed particularly to resist the effects of variable latency and has no setup delay. TCP will incur additional overhead and becomes harder to synchronize the clock.

Part Two: TCP Slow Start [12 points]

There are two nodes on a network C and S. C is the client and S is the server. C wants to connect to S and send a message that is 36 kilobytes long using TCP. Assume that a single packet can hold up to 2 kilobytes of data and the headers are negligibly small. Processing time at both ends of the connection is negligible, but the propagation delay between node S and node C is 10 ms. The link transmission rate is 10 Megabits per second. Assume that control packets (SYN, ACK, etc) are very small and their transmission delay is negligible. Also assume that the connection starts off in the slow start stage and that there is no packet loss.

(a)At what time will C begin sending the message to the server S? 2 $\times$ Trip Time (SYN has to be sent to server and SYN/ACK has to be received) = 2 $\times$ 10 ms = 20 ms

(b) At what time will S send an ACK packet in response to the first data packet sent by C?

Transmission time of a packet $= \frac{2 \text{ kilobytes}}{(10/8)\times 1024 \text{ kilobytes/second}} = 1.56\ ms$
$3 \times$ Trip time + Packet Transmission time $= 3 \times 10$ ms + 1.56 ms =31.56ms

Note: here we're using kilobyte $=2^{10}$ bytes and megabyte$=2^{10}$ kilobytes. If someone uses 10^3, he/she would still get full mark

(c) How many windows will be needed for the client to send the entire message?

$2 + 4 + 8 + 16 + 32 > 36$ kilobyte. So 5 windows. 4 windows would not be enough to transmit the entire message.

Note: Here we have written the window sizes in terms of kilobytes. You can equivalently write them in terms of number of packets: 1, 2, 4, 8, 16

(d) At what time will the server finish receiving the message from the client? (Hint: it may help to draw a timing diagram)

P = transmission time of one packet = 1.56 ms
RTT= 2×5 ms=10 ms
(Set up:) RTT + (first window:) P + RTT + (second window:) P + RTT + (third window:) P + RTT + (fourth window:) P + RTT (remaining packets for fifth window:) 3P + RTT/2 = RTT * 5.5 + P * 7
Here we are assuming that RTT is much larger than P (specifically RTT $>$ 8 P).
For each window, we are measuring the time from sending the first packet in each window until the start of the next window.

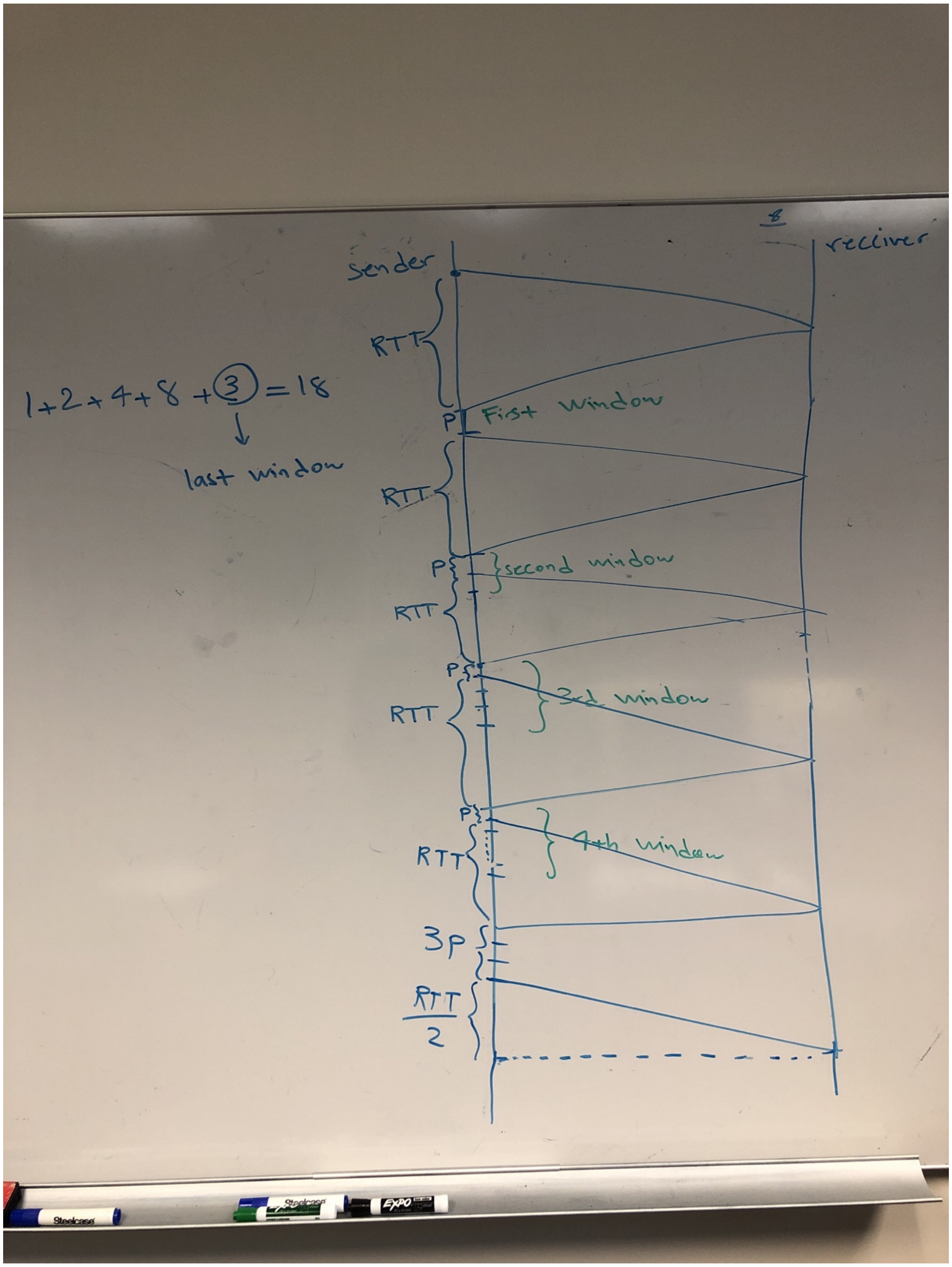
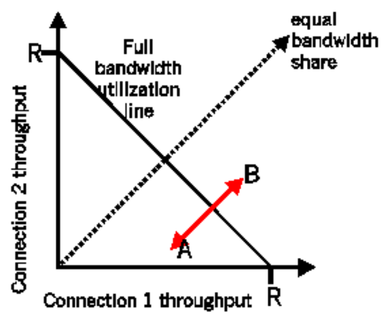


Figure 1: Timing Diagram

Part 3: AIAD instead of AIMD

Assume we modify the TCP AIMD algorithm and instead use AIAD, i.e. Additive Increase Additive Decrease. So, when no congestion is observed, a source increases its window size by 1 packet every round-trip time (RTT). Whenever congestion is observed, a source decreases its window size by 1. Given two AIAD sources sharing the same bottleneck and experiencing the same RTT, do they converge to a fair and efficient rate allocation? Support your answer graphically by showing the trajectories of the two windows assuming a synchronized model where the windows are adapted at the same time instants.

Suppose window sizes (or rates) start from point A in the plot. In this case, the throughputs never move off of the AB line segment. This could still be efficient but it is not a fair allocation. The only way that this could be fair is if the initial rate vector A is on the 45 degree line.



Question 4. Suppose one time slot corresponds to a time during which one packet can be sent. Packet and ACKs propagates so that they are received in the following time slot in the other end. The RTT is hence 2 time slots. We assume that the timeout value is set to 5.5 (i.e., after a packet is transmitted, unless an ACK is received the time out is triggered just before the 6th time slot). We assume Pipelining is used (window size 4, each packet size 1). Sequence numbers (SNs) starts at 0. ACK indicates the sequence number of the packet it acknowledges. All transmissions are successful and without corruption except:

The first packet with SN=1 sent by the sender is lost.

The first ACK sent by the receiver with SN=3 is lost.

(a) Assuming the Sender and Receiver implements GO-BACK-N, present a timing diagram for the first 11 time slots, that shows

• The packet and ACK sent and received by each host during each time slot, with their SNs.

• The set of packets currently in the window maintained by each host.

(b) Do the same diagram for SELECTIVE REPEAT.

GO-BACK-N

	Sender		Receiver	
Pkt 0 sent	0 1 2 3 4 5 6 7 8 9		0 1 2 3 4 5 6 7 8 9	
Pkt 1 sent	0 1 2 3 4 5 6 7 8 9	lost	0 1 2 3 4 5 6 7 8 9	Ack 0 sent
Pkt 2 sent	0 1 2 3 4 5 6 7 8 9		0 1 2 3 4 5 6 7 8 9	
Pkt 3 sent	0 1 2 3 4 5 6 7 8 9		0 1 2 3 4 5 6 7 8 9	Ack 0 Sent
Pkt 4 sent	0 1 2 3 4 5 6 7 8 9		0 1 2 3 4 5 6 7 8 9	Ack 0 Sent
	0 1 2 3 4 5 6 7 8 9		0 1 2 3 4 5 6 7 8 9	Ack 0 Sent
Time out Packet 1 sent	0 1 2 3 4 5 6 7 8 9		0 1 2 3 4 5 6 7 8 9	
Pkt 2 sent	0 1 2 3 4 5 6 7 8 9		0 1 2 3 4 5 6 7 8 9	Ack 1 sent
Pkt 3 sent	0 1 2 3 4 5 6 7 8 9		0 1 2 3 4 5 6 7 8 9	Ack 2 sent
Pkt 4 sent	0 1 2 3 4 5 6 7 8 9	lost	0 1 2 3 4 5 6 7 8 9	Ack 3 sent
Pkt 5 sent	0 1 2 3 4 5 6 7 8 9		0 1 2 3 4 5 6 7 8 9	Ack 4 sent
Pkt 6 sent	0 1 2 3 4 5 6 7 8 9		0 1 2 3 4 5 6 7 8 9	Ack 5 sent

2. [redacted]Do the same diagram for SELECTIVE REPEAT

SELECTIVE-REPEAT

	Sender		Receiver	
Pkt 0 sent	0 1 2 3 4 5 6 7 8 9		0 1 2 3 4 5 6 7 8 9	
Pkt 1 sent	0 1 2 3 4 5 6 7 8 9	lost	0 1 2 3 4 5 6 7 8 9	Ack 0 sent
Pkt 2 sent	0 1 2 3 4 5 6 7 8 9		0 1 2 3 4 5 6 7 8 9	
Pkt 3 sent	0 1 2 3 4 5 6 7 8 9		0 1 2 3 4 5 6 7 8 9	Ack 2 sent
Pkt 4 sent	0 1 2 3 4 5 6 7 8 9	lost	0 1 2 3 4 5 6 7 8 9	Ack 3 sent
	0 1 2 3 4 5 6 7 8 9		0 1 2 3 4 5 6 7 8 9	Ack 4 sent
Time out for pkt 1	0 1 2 3 4 5 6 7 8 9		0 1 2 3 4 5 6 7 8 9	
	0 1 2 3 4 5 6 7 8 9		0 1 2 3 4 5 6 7 8 9	Ack 1 sent
Time out for Pkt 3	0 1 2 3 4 5 6 7 8 9		0 1 2 3 4 5 6 7 8 9	
Pkt 5 sent	0 1 2 3 4 5 6 7 8 9		0 1 2 3 4 5 6 7 8 9	Ack 3 sent
Pkt 6 sent	0 1 2 3 4 5 6 7 8 9		0 1 2 3 4 5 6 7 8 9	Ack 5 sent
Pkt 7 sent	0 1 2 3 4 5 6 7 8 9		0 1 2 3 4 5 6 7 8 9	Ack 6 sent