Homework 6 (120 points) Due Date*: 10:00am 04/06/2020 Cutoff Deadline: 10:00am 04/08/2020**

***Late penalty will apply for past-due late submission; **Submission will NOT be accepted after the cutoff deadline**

Submission: Upload and submit it on **Blackboard**. If your homework is in hand-written, please try to SCAN it into a .pdf file or take a clear photo, and then upload/submit it. (NO Paper/Hardcopy or Email submission please!)
Please 1) **include your First Name and Last Name** on the page(s) that you upload & submit on Blackboard
2) Name your HW file as "**HW<#><LastName><FirstName>**", e.g., **HW6PolisJared.pdf** for HW6.
Grading: I will also grade it in Adobe Acrobat and post the graded work on Blackboard for everyone who submits it.

PLEASE ORGANIZE YOUR WORK IN THE SEQUENCE GIVEN IN THE ASSIGNMENT!!!

Problem A. Consider transferring an enormous file of L bytes from Host A to Host B. Assume an MSS (Max Segment Size) of 536 bytes. (hint: MSS: Maximum Segment Size, i.e., the maximum size of the payload of a TCP segment, not including the TCP segment header. It's essentially the largest amount of application-layer data, in Bytes, that a TCP segment can carry.)

(a) What is the maximum value of L such that TCP sequence numbers are not exhausted? Recall that the TCP sequence number field has 4 bytes.

(b) For the L you obtain in (a), find how long it takes to transmit the file. Assume that a total of 66 bytes of transport, network, and data-link header are added to each segment before the resulting packet is sent out over a 155 Mbps link. Ignore flow control and congestion control so A can pump out the segments back to back and continuously. (Hint: it is assumed that the window size N is large enough such that the segments can be sent out back to back without the need to wait for the next sequence number to be available and there is no loss of packet or ack.)

Problem B. Host A and B are communicating over a TCP connection, and Host B has already received from A all bytes up through byte 126. Suppose Host A then sends two segments to Host B back- to- back. The first and second segments contain 80 and 40 bytes of data, respectively. In the first segment, the sequence number is 127, the source port number is 302, and the destination port number is 80. Host B sends an acknowledgment whenever it receives a segment from Host A.

(a) In the second segment sent from Host A to B, what are the sequence number, source port number, and destination port number?

(b) If the first segment arrives before the second segment, in the acknowledgment of the first arriving segment, what is the acknowledgment number, the source port number, and the destination port number?

(c) If the second segment arrives before the first segment, in the acknowledgment of the first arriving segment, what is the acknowledgment number? (Hint: The reliability control used in TCP is basically a GBN approach with some variation.)

(d) Suppose the two segments sent by A arrive in order at B. The first acknowledgment is lost and the second acknowledgment arrives after the first time-out interval. Draw a timing diagram, showing **these segments and all other segments and acknowledgments** sent. (Assume there is no additional packet loss.) For each segment in your figure, provide the sequence number and the number of bytes of data; for each acknowledgment that you add, provide the acknowledgment number. **(hint: draw all the segments and ACKs that are exchanged between Hosts A and B involved in delivering these two segments)**

Problem C. Referring to the Slide 38 titled "TCP Flow Control", if RcvBuffer = 4096 bytes, 1280 bytes data is buffered,

(a) what is the **`rwnd`** value in the TCP header of the next receiver-to-sender segment?

(b) When the sender receives the above TCP segment, it has sent out 2560 bytes not yet ACKed. At most how many more bytes can the sender send out before receiving any ACK?

Problem D. (a) List the following three TCP segments in the order of being transmitted in the TCP 3-way handshake for establishing a TCP connection; (b) describe the sender and receiver of EACH TCP segment as *client-to-server* or *server-to-client*; (c) what is the initial sequence number chosen by the client; (d) what is the initial sequence number chosen by the server?

TCP Segment [SYNBit = 1, Seq = 58, ACKbit = 1, ACKnum = 126]
TCP Segment [ACKbit = 1, ACKnum = 59]
TCP Segment [SYNBit = 1, Seq = 125]

Problem E. Given that the client initiates the procedure of closing a TCP connection, the **last byte** sent from client to server **before** the FIN segment is **byte #1,742**, the **last byte** sent from server to client **before** the FIN segment is **byte #6,029**, (a) list four TCP segments exchanged between client and server for closing a TCP connection, where EACH TCP Segment must be described in the **format** used in Problem B; (b) describe the sender and receiver of EACH TCP segment as *client-to-server* or *server-to-client*.

(more on next page!)

Problem F. The initial ssthresh is 16, the sender experiences a ***3-duplicate-ACKs*** event **right after the 9th transmission round** in both Tahoe and Reno cases, and then the sender experiences ***a timeout*** event **right after the 16th round in a TCP Tahoe case** and **after right after the 19th round in a TCP Reno case**, FILL the following table to illustrate the congestion window size in segments (**cwnd**) and **ssthresh** as functions of transmission round for the time from the **1st** to the **22th** round if (a) TCP Tahoe is used for congestion control; and (b) TCP Reno is used for congestion control. (**Hint**: be aware that the figure in the slide titled "TCP: switching from slow start to CA" illustrates both cases in the same graph.)

	TCP Tahoo				TCP Reno		
	Trans. Round	cwnd	ssthresh		Trans. Round	(cwnd + 1)	Ssthresh
Slow Start	1st	1	16		1st	1	16
	2nd	2			2nd	2	
	3rd	4			3rd	4	
	4th	8			4th	8	
	5th	9			5th	9	
	6th	10			6th	16	
	7th	11			7th	11	
Congestion Avoidance	8th	12	Send ACK8		8th ≥3 Dup Ack	12	
Send missing packet → Time out RxMT	9th ≥3 Duplicate ACK send FAST RxMT 8	(1)	12/2 = 6		9th RXTX 8	~~9~~	12/2+3 = 9
ssth = cwnd/2 Slow Start	10th	2	6		10th	10	9
cwnd = 1	11th	4			11th	11	
	12th	6			12th	12	
	13th	7			13th	13	
	14th	8			14th	14	
	15th	9			15th	15	
RCV ACK 16	16th	10	Send ACK16		16th	16	16/2 = 8
	17th	(1)	10/2 = 5		17th	(8)	8
	18th	2			18th	9	
	19th	4			19th	10	
	20th	8			20th	11	
	21st	9			21st	12	
	22nd	10			22nd	13	

12/2 = 6
cwnd = ssth
Send missing pkt.

Problem A

(a) The TCP sequence number Field size is calculated by

$4 \times 8 = 32$ bits

2^{32} possible sequence numbers

The sequence number of bit are 2^{32} so, the maximum File size that can be sent from Host A to B equals:

$$= \frac{4294967296}{1024^3} \quad \begin{array}{l} \leftarrow \text{possible sequence numbers } 2^{32} \\ \leftarrow \text{GB's} \end{array}$$

$\approx$ 4 GB max File size

The sequence number increments by length or bytes of data being sent. The max File size won't depend on MSS here.

(b) Num of segments $\left(\frac{2^{32}}{536 \times 8}\right) = 1001624.836$ segments $\times$ 66 bytes

$= 66107239.16$ bytes

$= 66107239.16 + 2^{32}$

$= 4361074435$ Total bytes xmt

Total time (T.O.T.) $\longrightarrow \frac{4361074435 \times 8 \text{ bits}}{155 \text{ Mbps}}$

$= 249$ seconds

Problem B

(a) The first segment sequence number is 127. The source port is 302. The destination port number is 80

(b) (Sequence number of the 1st segment + length of datagram in bytes)

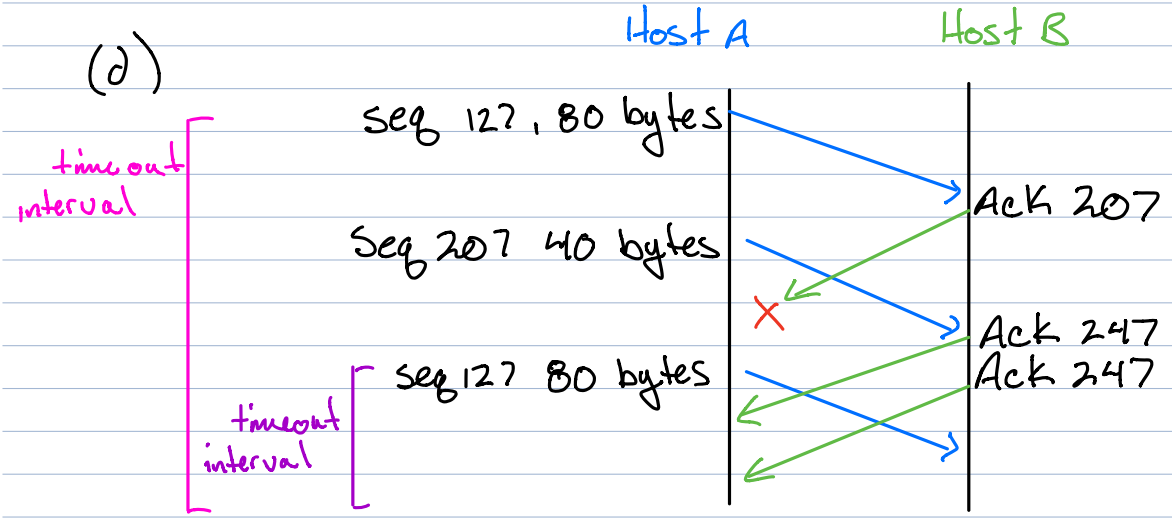
2nd segment number = 127 + 80 ⇒ (207)

The source port and destination are the same (302, 80)

Also the acknowledgement number is (207) if the first segment arives before the second segment.

(c) (127), indicating that B is waiting for bytes greater than the current sequence.

(d)



Problem C

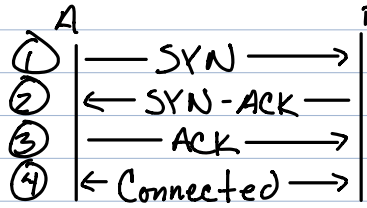
(a) rwnd val: 4896 − 1280 = (3616) data bytes

(b) Sender can send a minimum of 3616 − 2560 = (1056) bytes

Problem D

(a)
- Host A sends TCP SYN packet to Host B
- B recieves A's SYN
- B sends SYN-ACK
- A recieves B SYN-ACK
- A sends ACK
- B recieves ACK
- Connected ✓

(b)

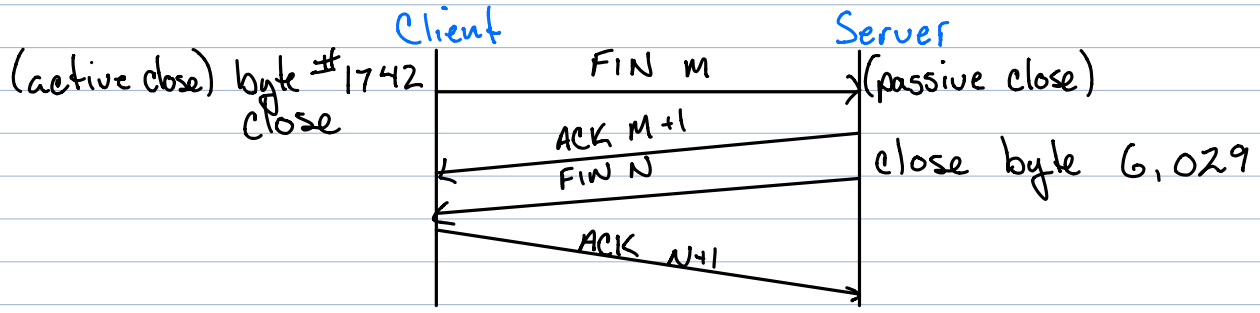


"Three way handshake"

(c) 58

(d) 125

Problem E



(a) FIN occupies one byte of the sequence number. The ACK of each FIN is the sequence number of the FIN plus one. There is a possibility for the data to flow "half-closed" From the end doing a passive close to the end doing the active close.

(b) TCP segments the sender and reciever of each TCP segment of client-to-server server-to-client.

First: The server recieves clients FIN M packet. (client-to-server).
second: client recieves servers ACK M+1 packet. (server-to-client).
third: Client recieves servers FIN N packet. (server-to-client).
Fourth: Server recieves clients ACK N+1 packet (client-to-server).