Solution: CS120 Homework Assignment #3

Name: ______________ ID: ___________________

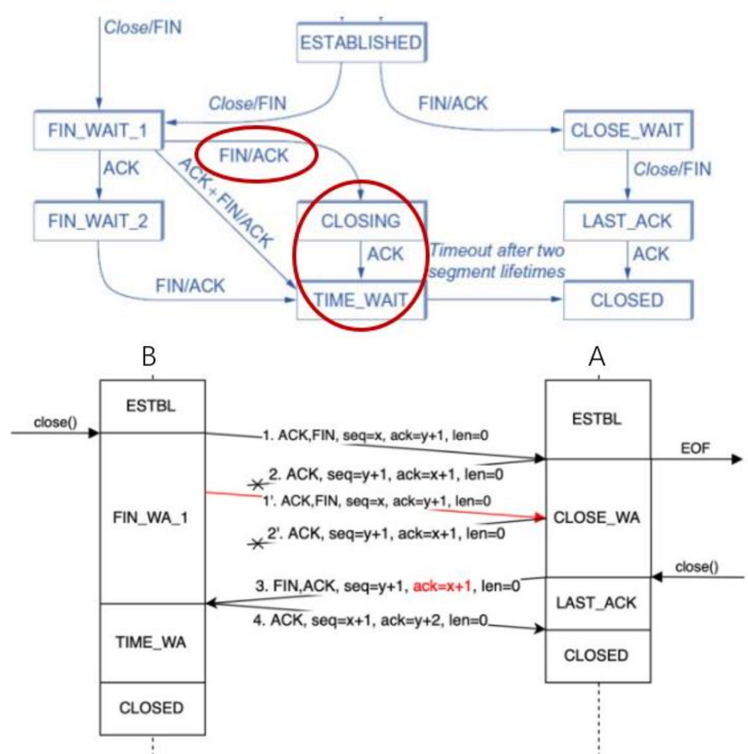
Please read the following instructions carefully before answering the questions:

- This assignment is to be completed by each student **individually**.
- There are a total of **9** questions.
- When you write your answers, please try to be precise and concise.
- Fill your name and student ID at the first page.
- Please typeset the file name and format of your submission to the following one: YourID_CS120_HW3.pdf (Replace "YourID" with your student ID). Submissions with wrong file name or format will **NOT** be graded.
- Submit your homework through Blackboard.

1. (5 points) When closing a TCP connection, why is the two-segment-lifetime timeout not necessary on the transition from LAST_ACK to CLOSED?

   When the endpoint(A) is in LAST_ACK state, the only thing it needs to do is waiting to receive the ACK for the FIN it sent.

   If it finally receives the ACK, it knows its FIN is received by the other side(B). Note that the FIN also contains the information of ACK. Then whatever B had received the ACK sent by A or not, once B receives the FIN, it also receives the ACK. According to the state transition diagram of TCP, B transits to TIME_WAIT, and won't resend the FIN. Thus, the two-segment-lifetime timeout not necessary on the transition from LAST_ACK to CLOSED



2. (10 points) You are hired to design a reliable byte-stream protocol that uses a sliding window (like TCP). This protocol will run over a 1-Gbps network. The RTT of the network is 100 ms, and the maximum segment lifetime is 30 seconds. How many bits would you include in the **AdvertisedWindow** and **SequenceNum** fields of your protocol header?

   The advertised window should be large enough to keep the pipe full; delay (RTT) × bandwidth here is 100ms × 1Gbps = 100Mb = 12.5 MB. This requires 24 bits if we assume the window is measured in bytes (224 ≈= 16million) for the **AdvertisedWindow** field. The **SequenceNum** field must not wrap around in the maximum segment lifetime. In 30 seconds, 30 Gb = 3.75 GB can be transmitted. 32 bits allows a sequence space of about 4GB, and so will not wrap in 30 seconds. (If the maximum segment lifetime were not an issue, the sequence number field would still need to be large enough to support twice the maximum window size; see "Finite Sequence Numbers and Sliding Window" in Section 2.5.)

3. (10 points) The Nagle algorithm, built into most TCP implementations, requires the sender to hold a partial segment's worth of data (even if PUSHed) until either a full segment accumulates or the most recent outstanding ACK arrives.

   (a) Suppose the letters **abcdefghi** are sent, one per second, over a TCP connection with an RTT of 4.1 seconds. Draw a timeline indicating when each packet is sent and what it contains. (5 points)

   (b) Suppose that mouse position changes are being sent over the connection. Assuming that multiple position changes are sent each RTT, how would a user perceive the mouse motion with and without the Nagle algorithm? (5 points)

   (a) T=0.0 'a' sent
   T=1.0 'b' collected in buffer
   T=2.0 'c' collected in buffer
   T=3.0 'd' collected in buffer
   T=4.0 'e' collected in buffer
   T=4.1 ACK of 'a' arrives, "bcde" sent
   T=5.0 'f' collected in buffer
   T=6.0 'g' collected in buffer
   T=7.0 'h' collected in buffer
   T=8.0 'i' collected in buffer
   T=8.2 ACK of 'bcde' arrives; "fghi" sent
   T=12.3 ACK of 'fghi' arrives

   (b) With the Nagle algorithm, the mouse would appear to skip from one spot to another. Without the Nagle algorithm the mouse cursor would move smoothly, but it would display some inertia: it would keep moving for one RTT after the physical mouse were stopped.

4. (10 points) Suppose, in TCP's adaptive retransmission mechanism, that EstimatedRTT is 4.0 seconds at some point and subsequent measured RTT's all are 1.0 second. How long does it take before the TimeOut value, as calculated by the Jacobson/Karels algorithm, falls below 4.0 seconds? Use $\delta = 1/8$. Use initial deviation value of 1.0 (Hint: You can use Excel to calculate).

   Using initial Deviation =1.0 it took 20 iterations for TimeOut to fall below 4.0.

Iteration	SampleRTT	EstRTT	Dev	diff	TimeOut
0.00	1.00	4.00	1.00		
1.00	1.00	3.63	1.25	-3.00	8.63
2.00	1.00	3.30	1.42	-2.63	8.98
3.00	1.00	3.01	1.53	-2.30	9.13
4.00	1.00	2.76	1.59	-2.01	9.12
5.00	1.00	2.54	1.61	-1.76	8.99
6.00	1.00	2.35	1.60	-1.54	8.76
7.00	1.00	2.18	1.57	-1.35	8.46
8.00	1.00	2.03	1.52	-1.18	8.12
9.00	1.00	1.90	1.46	-1.03	7.74
10.00	1.00	1.79	1.39	-0.90	7.35
11.00	1.00	1.69	1.32	-0.79	6.95
12.00	1.00	1.60	1.24	-0.69	6.55
13.00	1.00	1.53	1.16	-0.60	6.16
14.00	1.00	1.46	1.08	-0.53	5.78
15.00	1.00	1.40	1.00	-0.46	5.41
16.00	1.00	1.35	0.93	-0.40	5.06
17.00	1.00	1.31	0.86	-0.35	4.73
18.00	1.00	1.27	0.79	-0.31	4.42
19.00	1.00	1.24	0.72	-0.27	4.13
20.00	1.00	1.21	0.66	-0.24	3.86

5. (5 points) Suppose we want returning RTCP reports from receivers to amount to no more than 5% of the outgoing primary RTP stream. If each report is 84 bytes, the RTP traffic is 320 kbps, and there are 1000 recipients, how often do individual receivers get to report? What if there are 10,000 recipients?

   Each receiver gets 1/1000 of 5% of 320 kbps, or 16bps, which means one 84-byte RTCP packet every 42 sec. At 10K recipients, it's one packet per 420 sec, or 7 minutes.

6. (10 points) Suppose a congestion-control scheme results in a collection of competing flows that achieve the following throughput rates: 200 KBps, 160 KBps, 110 KBps, 95 KBps, and 150 KBps.

   (a) Calculate the fairness index for this scheme. (5 points)

   (b) Now add a flow with a throughput rate of 1000 KBps to the above, and recalculate the fairness index. (5 points)

   (a) 0.9360

   (b) 0.4419

7. (20 points) Suppose a router has three input flows and one output. It receives the packets listed in Table 1 all at about the same time, in the order listed, during a period in which the output port is busy but all queues are otherwise empty. Give the order in which the packets are transmitted, assuming:

   (a) Bit-Level fair queuing. (10 points)

   (b) Weighted bit-level fair queuing, with flow 2 having weight 4, and the other two flows having weight 1. (Hint: when calculating finishing time, transmission time = actual transmission time/weight) (10 points)

Table 1 Packets for Exercise 7

Packet	Size	Flow
1	100	1
2	100	1
3	100	1
4	100	1
5	190	2
6	200	2
7	110	3
8	50	3

(a) First we calculate the finishing times Fi. We don't need to worry about clock speed here since we may take Ai = 0 for all the packets. Fi thus becomes just the cumulative per-flow size, i.e. Fi = Fi−1 + Pi.

Packet	Size	Flow	Fi
1	100	1	100
2	100	1	200
3	100	1	300
4	100	1	400
5	190	2	190
6	200	2	390
7	110	3	110
8	50	3	160

So the packet order is: Packet 1, Packet 7, Packet 8, Packet 5, Packet 2, Packet 3, Packet 6, Packet 4.

(b) To give flow 2 a weight of 4 we divide each of its Fi by 4, i.e. Fi = Fi−1 + Pi/4

Packet	Size	Flow	Fi
1	100	1	100
2	100	1	200
3	100	1	300
4	100	1	400
5	190	2	47.9
6	200	2	97.5
7	110	3	110
8	50	3	160

So the packet order is: Packet 5, Packet 6, Packet 1, Packet 7, Packet 8, Packet 2, Packet 3, Packet 4.

8. (20 points) Assume that TCP implements an extension that allows window sizes much larger than 64 KB. Suppose that you are using this extended TCP over a 1-Gbps link with a latency of 50 ms to transfer a 10-MB file, and the TCP receive window is 1 MB. If TCP sends 1-KB packets (assuming no congestion and no lost packets):

   (a) How many RTTs does it take until slow start opens the send window to 1 MB? (5 points)

   (b) How many RTTs does it take to send the file? (5 points)

   (c) If the time to send the file is given by the number of required RTTs multiplied by the link latency, what is the effective throughput for the transfer? What percentage of the link bandwidth is utilized? (10 points)

   (a) In slow start, the size of the window doubles every RTT. At the end of the ith RTT, the window size is 2^i KB. It will take 10 RTTs before the send window has reached 2^10 KB = 1MB.

   (b) After 10 RTTs, 1023KB = 1MB − 1KB has been transferred, and the window size is now 1 MB. The congestion window will continue to increase but the sending window is limited by the receive window 1 MB. So it takes another 10 RTTs to send the file. The send size of each RTT is 1 KB, 2 KB, 4 KB, 8KB, 16 KB, 32 KB, 64 KB, 128 KB, 256KB, 512 KB, 1 MB, 1 MB, 1 MB, 1 MB, 1 MB, 1 MB, 1 MB, 1 MB, 1 MB, 1 MB, 1 KB.

   (c) It takes 1 second (20 RTTs) to send the file. The effective throughput is (10MB / 1s) = 10MBps = 80Mbps. This is 8% of the available link bandwidth.

9. (10 points) Consider a RED gateway with $MaxP = 0.01$, and with an average queue length halfway between the two thresholds.

(a) Find the drop probability P_{count} for $count = 1$ and $count = 100$. (5 points)

(b) Calculate the probability that none of the first 50 packets is dropped. Note that this is $(1 - P_1) \times \cdots \times (1 - P_{50})$. (5 points)

(a) TempP = MaxP/2 = 0.005. For count=1 the drop probability is 1/199; for count=100 the drop probability is 1/100.

(b) 149/199, or 0.7487