Homework 2

Please email your answers/report **in PDF format** TO "kharrazi@sharif.edu". The HW file name should be **"Your Lastname-443-HW-1"**. Also the same title should be used as the subject of your email. Please follow the formatting. **This homework is due by Aban 30th, 11:59 PM**.

Part I

1. In class we discussed how Timeouts are calculated from the RTT values. One problem with the algorithm discussed in the class (Lecture 5, Page 24, also in Section 5.2.6), which is called the Original algorithm, is that it does not take into account the variance of the sample RTTs. An alternate algorithm is the Jacobson/Karels algorithm (Section 5.2.6 of textbook). Now suppose that TCP obtains the following latency numbers from a connection, calculate the Timeout based on the Original and Jacobson/Karels algorithms:

   1 , 1.3 , 1 , .9 , .8 , 1.2 , 1.3 , 1.6 , 1.9 , 2.3 , 2 , 1.4 , 1 .9 , 0.7 , 0.8 , 0.9 , 1 , 1.2 , 0.9 , 1 , 1.2 , 1.1 , 6 , 7 , 7 , 5.5 , 3.2 , 2.4 , 2.3 , 2 , 1.6, 1.4 , 1.2 , 1.1 , 1 , 1

   Use $\delta = \frac{1}{8}$, and $\alpha = 0.8$

   Solution: I will add this part in the next few days, this type of question will not be in your midterm.

2. Per TCP header, the receiver window size is a 16 bit number, hence at max 64KB. Assume you have a link with a 50ms delay. Calculate how well the $Delay \times Bandwidth$ is being utilized in the each of the following scenarios:

   Solutions: The network bandwidth is based on the clock that paces the transmissions. For example a 10MHz clock transmits at 10Mbps. Hence, we have 128 Kbits = 128 $\times 10^3$ bits. On the other hand, when we talk about messages in the memory or on the disk, size is measures in power of 2. For example 64KB = 64 $\times 2^{10}$ = 65536 Bytes.

ADSL modem (128 Kbps)	50ms $\times$ 128Kbps = 800 Bytes	< 65536 Bytes$\rightarrow$ 100%
T1 (1.5 Mbps)	50 ms $\times$ 1.5Mbps = 9.4 KBytes	< 65536 Bytes $\rightarrow$ 100%
Ethernet (10 Mbps)	50 ms $\times$ 10Mbps = 62.5 KBytes	< 65536 Bytes $\rightarrow$ 100%
T3 (45 Mbps)	50 ms $\times$ 45Mbps = 281 KBytes	$\frac{65536}{281\times10^3} = 23.32\%$
OC 192 (2.5 Gbps)	50 ms $\times$ 2.5Gbps = 15.6 MBytes	$\frac{65536}{15.6\times10^6} = 0.42\%$
OC 768 (10 Gbps)	50 ms $\times$ 10Gbps = 62.5 MBytes	$\frac{65536}{62.5\times10^6} = 0.1\%$

   What should be the receiver window size in each case, to obtain maximum utilization?

   Solution: Maximum utilization would be obtained when the window size equal $Delay \times Bandwidth$. That is why there is an extension to TCP, with which a scaling number is stored in the options field of the TCP header. Therefore the congestion window is equal to the congestion window value in the header field (the 16 bit value) scaled by the number stored in the options field.

3. Suppose that you want to run TCP over a 1Mbps link with a latency of 100ms to transfer a 2-MB file, and the TCP receive window is 64KB. If TCP sends 1KB packets (assuming no congestion and no lost packets):

   - How many RTTs does it take until slow start opens the send window to 64KB?
   **Solution:** $1KB * 2^i \leq 64KB \Rightarrow i = 6 RTTs$ required.

   - How many RTTs does it take to send the file?
     Solution: For the first 6 RTTs we have $(1 + 2 + 4 + 8 + 16 + 32 + 64) = 127KB$, and after that we have 64KB of data sent in each RTT. Hence we have $\frac{2MB - 127KB}{64KB} = \frac{2*2^{20} - 127*2^{10}}{64*2^{10}} = 30.01 \approx 31 \Rightarrow$ we require $31 + 6 = 37 RTTs$ to transmit a 2MB file.

   - 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?
     Solution: $\frac{2MB*8}{37*100ms} = \frac{16Mb}{3.7s} = 4.32Mbps$. Therefore $\frac{4.32}{100} = 432\%$ utilization. As some of you pointed out, this number does not make real sense here. We are assuming that there is no congestion and no packet losses, in fact there will be congestion and many more RTTs required before the file is transferred.

4. Draw the congestion window as a function of time (Consider linear increase and multiplicative decrease plus slow start). Assume initial window size is 1 mss (at round 0). A triple duplicate ACK occurs at round 3 and 13, a packet loss happens at round 10 and 18. Draw the curve up to round 22.
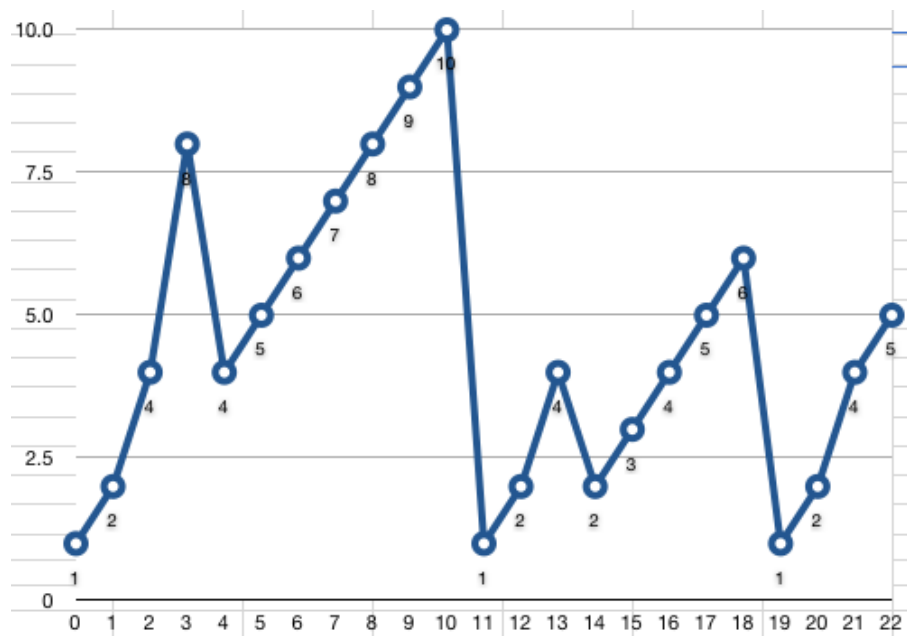
Here I am assuming that the window is increased and then checked if it passes half of the window size when the last loss happened. For example, loss happens at round 18 when we have a window size of 6. At round 19 slow start begins, and at round 21, it passes $\frac{6}{2}$, at which time we stop the slow start and increase the window size linearly.