Problem Set 7

Title: Congestion control and Performance Analysis

Due: start of class, Wednesday, May 3rd

Recommended Reading: Section 6.

All problems carry equal weight. To receive full credit, show all of your work. Answers MUST be typed in. Handwritten solutions will not be accepted. You can use Word, or any other software you would like to prepare your solution. Do not change the order of questions. Your solution must be converted to PDF before handing it in. Other file types (**.doc, .docx.,** *etc*) will not be graded. MacOS and Linux systems come already with PDF converters installed, there are various free software solutions to print PDFs available on the Internet.

You will hand in your solution using SVN, by adding and committing a PDF.

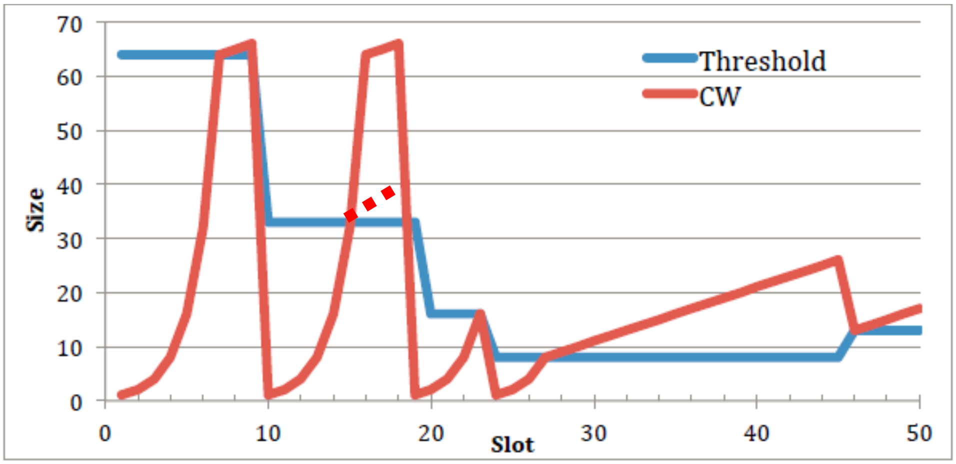
TCP Congestion

1. Consider a TCP system implementing slow start and congestion avoidance with fast retransmit and fast recovery. When a connection is setup the congestion window is initialized to one segment and the slow start threshold to 64 segments. To simplify the problem, assume that the timeout is equal to the RTT (an exact estimate) and specify time in units of RTT, such that one time slot is one RTT.

   Packet transmissions are such that at each time slot the sender sends all packets in the congestion window. If ACK's are received in the next time slot, there is no timeout. In addition, to simplify matters either the entire window is acknowledged or none of its segments are acknowledged.

   For a particular connection, ACK's are received in time slots 1-9, 11-18, 20-23, 25-45, and 47-50. Timeouts occur in slots 10, 19 and 24; in slot 46, three duplicate ACK's are received for the packets sent in time slot 45.

   For the system described, **plot** both the congestion window and the slow start threshold (on the same graph) versus time (**slots**), between slots 0 and 50. The values for each slot are the ones that the CW and the threshold assume **at the end of the slot**. Remember to consider the differences between slow start and congestion avoidance with fast retransmit and fast recovery when changing the congestion window. Ignore the value of SWS.



CW has exponential growth until it reaches 64 (slot7), then it grows linearly in slots 8 and 9, reaching 66 packets when the timeout occurs. CW goes to 1, the threshold is set to 33. The exponential growth brings the CW back to 64 (first power of two larger than the threshold), after which additive increase brings the CW to 66 when the second timeout happens. Again, the threshold is halved, and the CW brought back to 1. The CW increases exponentially to 2, 4, 8, 16 packets when one more timeout happens, at slot 24. CW is set to 1 once more, and the threshold is halved. The exponential increase brings the CW value to 8 (the threshold value) at round 27, after which the linear increase starts. At slot 45, when the three acks arrive, the C value is 26, the threshold and the CW are halved, and the CW starts growing linearly.
(Comments: The dash line is also accepted as a right answer)

Random Early Detection

2. Consider a RED gateway with MaxP = 0.08, and with an average queue length halfway between the two thresholds MinThreshold and MaxThreshold.

   a. a. Find the drop probability P_{count}, for count = 1 and count = 15

   b. Calculate the probability that none of the 15 packets are dropped. (Describe the method, and provide a numerical result approximate to the 4th decimal place)

Sol:

(a) $AvgLength = \frac{MinTh+MaxTh}{2}$

$$TempP = MaxP \cdot \frac{AvgLength - MinTh}{MaxTh - MinTh} = 0.08 \cdot 0.5 = 0.04$$

$$P_1 = \frac{TempP}{1 - 1 \cdot TempP} = \frac{0.04}{0.96} = 0.042$$

$$P_{15} = \frac{TempP}{1 - 15 \cdot TempP} = \frac{0.04}{1 - 0.6} = 0.1$$

(b) $P_{No\ Drops} = (1 - P_1) \cdot (1 - P_2) \cdot \cdots \cdot (1 - P_{15}) = 0.3750$

TCP and Network Delay

3. Consider the following two causes of a 4-second network delay (assume ACKs return instantaneously).

   - One intermediate router with a 4-second outbound per-packet bandwidth delay and no competing traffic.
   - One intermediate router with a 100-ms outbound per-packet bandwidth delay and with a steadily replenished (from another source) 40 packets in the queue.

   a. How might a transport protocol in general distinguish between these two cases?

   b. Suppose TCP Vegas sends over the above connections, with an initial CongestionWindow of 6 packets. What will happen to CongestionWindow in each case? Assume BaseRTT = 4 second and β is one packet per second.

   Sol:

   (a) If we send a single packet, we see a 4 seconds delay (in either case). Now consider the case where we send a burst of 10 packets. In the first case we will get ACKs back at 4 seconds intervals and the last packet will have an RTT of 40 seconds. However, the second case will have a 4 second RTT for the first packet and a 5 second RTT for the last packet.

   The technique of packet-pairs (sending multiple instances of two consecutive packets right after one another and analyzing the minimum time difference between their ACKs) achieves the same effect. In fact, packet-pair is sometimes thought of as a technique to determine the minimum path bandwidth. In the first case, the two ACKs of a pair will always be 4 second apart. In the second case, the two ACKs will sometimes be only 100 ms apart.

   (b) In the first case, TCP-Vegas will measure RTT=24 as soon as there is a full window of outstanding packets. This means the ActualRate is down to 0.25 packet/second. However, BaseRTT is 4 second, so ExpectedRate = CongestionWindow/BaseRTT =1.5 packets/second. Hence, Diff is 1.25 packets/second which is larger than beta so CongestionWindow will decrease.

   In the second case, when a burst of these packets is sent, the measured RTTs are 4.0, 4.1, 4.2, 4.3, 4.4, and 4.5seconds. Expected rate is: Expected Rate = (6 pkts)/(4s). Note that the actual rate is 6/4.5 (because it takes 4.5 seconds to get those five packets so diff is .166, which is smaller than beta, so you do not decrease so will increase or stay the same depending on whether alpha is greater than or less than 0.166

Fair Queueing

1. Suppose that the following packets arrivals:
Flow 1 A:5, B:4, C:6, D:7
Flow 2 E:6, F:7, G:2, H:4, I:6
Flow 3 J:1, K:5, L:1, M:1, N:6, O:5, P:4
Flow 4 Q:6, R:6, S:6, T:5

Assume that at the current time, all the packets have arrived and are now sitting in the per-flow queues. The number after the colon is the size of the packet, so packet A is 5 units in size.

(a) List the order of departure of these packets under the packetized weighted fair queuing scheme, with equal weights for each flow. Break ties by going with the lowest-numbered flow.
Sol:
Bit-by-bit schedule is:
Flow 1: AAAAABBBBCCCCCCDDDDDDD
Flow 2: EEEEEEFFFFFFGGHHHHIIIIII
Flow 3: JKKKKLMNNNNNNOOOOOPPPP
Flow 4: QQQQQRRRRRRSSSSSSTTTTT
Order of departure is therefore: J, A, E, K, Q, L, M, B, R, F, N, C, G, S, H, O, D, P, T, I.

Comments: If follow the scheme mentioned in the slides, the solution should be:
AEJQKBFLRMNCSGODHTIP
(both the above solutions are right)
(b) Now list the order of departure assuming that the four flows have weights 3, 1, 2, 1 respectively.
Sol:
To model different sending weights, we multiply packet sizes for the flows by 2, 6, 3, 6 respectively.
Then we have
Flow 1: AAAAAAAAAABBBBBBBBCCCCCCCCCCCCDDDDDDDDDDDDDD
Flow2:
EEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFF
FFFFFFFGGGGGGGGGGGGHHHHHHHHHHHHHHHHHHHHHHHHIIIIIIIIIIIIIIIIIIIIIIIIIIIIII
Flow3:
JJJKKKKKKKKKKKKKLLLMMMNNNNNNNNNNNNNNNNNNOOOOOOOOOOOOOOOPPPPPPP
PPPPP
Flow4:
QQQQQQQQQQQQQQQQQQQQQQQQQQQQQQRRRRRRRRRRRRRRRRRRRRRRRR
RRRRRRRSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSTTTTTTTTTTTTTTTTTTTTTTTTTTTTTT
Order of departure is therefore: J, A, B, K, L, M, C, E, Q, N, D, O, P, R, F, G, S, H, T, I.

Comments: If follow the scheme mentioned in the slides, the solution should be:
AEJQKBCLMNDFROPSGHTI
(both the above solutions are right)