CSE 473 – Introduction to Computer Networks

Review Questions 12

***Your Name**:*

Please print out this form (two-sided, if you can) and write your answers *legibly* in the spaces provided. If you can't write legibly, type.

1. Consider a TCP session in which the sender attempts to write a buffer with 10 MB of data, but the receiving application receives data by repeatedly invoking the method *socket.recv*(1), with a 1 second delay between each successive call to *recv*. How many bytes are carried in the TCP packets from the sender? Assume that the MSS is 1,000 bytes and that the receive buffer has space for 1 MB.

The application will receive 1 byte every second. Because the RecvBuffer is 1 MB and the MSS is 1 KB, the sender will initially be able to send 1,000 packets of 1 KB each, which will fill the RecvBuffer. After that, it will keep probing the receiver with 1 byte packets, which will every so often, i.e., every 1 sec when the RecvBuffer opens by 1 byte, make it through (be acked).

2. Assume that a TCP sender *A* is connected to router *X* by a 100 Mb/s link, that the corresponding receiver *B* is connected to router *Y* by a 100 Mb/s link and that the link connecting *X* and *Y* is 10 Mb/s. Also, assume that the roundtrip propagation time between *A* and *B* is 50 ms, that the MSS is 1250 bytes and that *ssthresh*=64 KB. Suppose that *A* starts in the slow-start state. At what rate is *A* sending after 400 ms?

The transmission time of a 1250 bytes message is 0.1 ms on a 100 MB/s link and 1 ms on a 10 Mb/s link. We will ignore queueing delays and the transmission time of ACKs, so that the corresponding ACK arrives back at the sender 51.2 ms after its packet was scheduled for transmission by the sender. If we now consider two "back-to-back" packets. The last bits of the two messages are spaced by 0.1 ms when they arrive at router X. The first message takes 1 ms to be transmitted and so does the second. Note that as long as the last bit of the second packet arrives before the first packet has finished transmission, their difference in arrival time is irrelevant when it comes to establishing the difference in their departure time from router X. It will be equal to the transmission time of the second packet, i.e., 1ms. So they arrive at router Y 1 ms apart. The first packet took 0.1 ms to be transmitted after it arrived and so will the second that arrived 1 ms later, so that the two packets will still be 1 ms apart when they arrive at the receiver. This means that the ACKs of back-to-back packets arrive back at the sender 1 ms apart. In general, the spacing between consecutive packets is determined by the bottleneck link.

Considering now the slow-start behavior at the sender A, it first sends a single MSS (its cwnd is equal to 1 MSS). It gets and ACK after 51.2 ms, at which point it transmits two back-to-back MSS, since its cwnd increased to two. The ACK for the first one returns again after 51.2 ms, i.e., at t=102.4 ms, and as discussed above the second arrives 1 ms later. Those two ACKs together increase cwnd to four. The first ACK triggers the transmission of two back-to-back MSS and the second does the same 1 ms later. The first ACK for those four packets arrives 51.2 ms later after the corresponding packet was sent, i.e., it *arrives at t=153.6 ms, with the second arriving 1ms later, and the third and the fourth following suit again 1ms apart. Those four ACKs will in turn increase cwnd to 8 and trigger the transmission of 8 packets. The process repeats until cwnd reaches ssthresh, at which point the sender enters congestion avoidance. Note that 64 kbytes = 2^{16} bytes = 65,536 bytes, or just over 52 MSS given the MSS size of 1,250 bytes. The evolution of the value of cwnd is shown below in number of MSS and as a function of time in ms. The column "t" indicates time, and the column "cwnd" gives two numbers. The first is the value of cwnd at that time in units of MSS, and the second is the RTT "round" to help track the progression of the ACKs. Note that the growth of cwnd is exponential as a function of the RTT rounds, i.e., it grows as 1,2,4,8,16,32…*

t	*cwnd*	*t*	*cwnd*	*t*	*cwnd*	*t*	*cwnd*	*t*	*cwnd*
0	*1/0*	*208.8*	*13/4*	*264.0*	***25/5***	*311.2*	*37/6*	*323.2*	*49/6*
51.2	***2/1***	*209.8*	*14/4*	*265.0*	***26/5***	*312.2*	*38/6*	*324.2*	*50/6*
102.4	*3/2*	*210.8*	*15/4*	*266.0*	***27/5***	*313.2*	*39/6*	*325.2*	*51/6*
103.4	*4/2*	*211.8*	*16/4*	*267.0*	***28/5***	*314.2*	*40/6*	*326.2*	*52/6*
153.6	***5/3***	*256.0*	***17/5***	*268.0*	***29/5***	*315.2*	*41/6*		
154.6	***6/3***	*257.0*	***18/5***	*269.0*	***30/5***	*316.2*	*42/6*		
155.6	***7/3***	*258.0*	***19/5***	*270.0*	***31/5***	*317.2*	*43/6*		
156.6	***8/3***	*259.0*	***20/5***	*271.0*	***32/5***	*318.2*	*44/6*		
204.8	*9/4*	*260.0*	***21/5***	*307.2*	*33/6*	*319.2*	*45/6*		
205.8	*10/4*	*261.0*	***22/5***	*308.2*	*34/6*	*320.2*	*46/6*		
206.8	*11/4*	*262.0*	***23/5***	*309.2*	*35/6*	*321.2*	*47/6*		
207.8	*12/4*	*263.0*	***24/5***	*310.2*	*36/6*	*322.2*	*48/6*		

Note that because the maximum possible value for cwnd is 64 kbytes (about 52 MSS), there will be no further increase beyond it when the sender enters congestion avoidance mode. From that point onward, the sender will be sending one packet every 1 ms, i.e., a transmission rate of exactly 10 Mb/s.

3. Consider the same setup as the last problem with one change, namely, that *ssthresh*=**100 kBytes,** or 80 MSS. Note that the value of 100kBytes exceeds the standard maximum value of 64kBytes and so would require the use of some TCP options to extend the value. We will assume that this had been done but that *cwnd* then cannot exceed 100 kBytes.

*The reasoning during the slow-start period is as in the previous question; with the only difference that cwnd reaches ssthresh during the 7th RTT. Note that 80*1250*8/50ms = 16 Mbits/sec exceeds the capacity of the 10 Mbits/sec link between X and. However, in subsequent RTTs the connection* *can only send whenever it receives and ACK (we assume that cwnd cannot grow beyond 100 kBytes), and ACKs are spaced 1 ms apart since this is the time it takes to transmit one packet on the 10 Mbits/sec link. Hence, from the 8th RTT onward, the connection is limited to transmitting at 10 Mbits/sec.*

Note that because of the excess ~28 packets that were transmitted in the 7ht RTT, the buffer preceding the link between routers X and Y, will from now on contain (at least) those 28 packets.

3. Consider the same setup as the last problem. Assume that *A* has reached a transmission rate of 10 Mb/s and that another TCP connection starts, which then results in a packet loss that is detected by *A* at time *t*. This causes *A* to cut its sending rate from 10 Mb/s to 5 Mb/s. If the buffer at router *X* can store 50 KB worth of data, how long does it take for the buffer at *X* to fully drain, assuming that *A* keeps its rate at 5 Mb/s and that the transmission rate of the other connection can be ignored?

If A keeps sending at 5 Mb/s, the buffer draining rate is 5 Mb/s, so that it will take $400,000/5x10^6=80$ *ms for the buffer to fully drain. Note that the assumption that A does not increase its rate is reasonable, as 80ms is less than 2 RTTs. So, even if A used fast recovery, its cwnd would have increased by less than 2 MSS during that period, which corresponds to a rate increase of less than 16,000/0.051=314 kb/s.*

The other connection would most likely also have lost a packet and have restarted slow-start (it was in slow-start when the loss occurred). Its ability to grow its rate back up would, however, depend on its own RTT value.