Computer Networks
Exercise #2 Suggested Solutions

Problem 15 (10%)

It takes 9.6 microseconds (or 0.0096 milliseconds) to send a packet, as $1200*8/10^9$=9.6 microseconds. In order for the sender to be busy 95 percent of the time, we must have

$$util = 0.97 = (0.0096n)/30.0096$$

or n approximately 3032 packets.

Problem 26 (20%) (各小題 10%)

There are $2^{32} = 4{,}294{,}967{,}296$ possible sequence numbers.

a) The sequence number does not increment by one with each segment. Rather, it increments by the number of bytes of data sent. So the size of the MSS is irrelevant -- the maximum size file that can be sent from A to B is simply the number of bytes representable by $2^{32} \approx 4.19\,\text{Gbytes}$.

b) The number of segments is $\left\lceil \frac{2^{32}}{512} \right\rceil = 8{,}388{,}608$. 64 bytes of header get added to each segment giving a total of 536,870,912 bytes of header. The total number of bytes transmitted is $2^{32} + 536{,}870{,}912 = 4.832 \times 10^9$ bytes.

Thus it would take 249 seconds to transmit the file over a 155 Mbps link.

Problem 40 (30%) (答對一小題給 3%)

a) TCP slowstart is operating in the intervals [1,6] and [23,26]
b) TCP congestion avoidance is operating in the intervals [6,16] and [17,22]
c) After the 16th transmission round, packet loss is recognized by a triple duplicate ACK. If there was a timeout, the congestion window size would have dropped to 1.
d) After the 22nd transmission round, segment loss is detected due to timeout, and hence the congestion window size is set to 1.
e) The threshold is initially 32, since it is at this window size that slow start stops and congestion avoidance begins.
f) The threshold is set to half the value of the congestion window when packet loss is detected. When loss is detected during transmission round 16, the congestion windows size is 42. Hence the threshold is 21 during the 18th transmission round.

g) The threshold is set to half the value of the congestion window when packet loss is detected. When loss is detected during transmission round 22, the congestion windows size is 29. Hence the threshold is 14.5 (or 14) during the 24th transmission round.

h) During the 1st transmission round, packet 1 is sent; packet 2-3 are sent in the 2nd transmission round; packets 4-7 are sent in the 3rd transmission round; packets 8-15 are sent in the 4th transmission round; packets 16-31 are sent in the 5th transmission round; packets 32-63 are sent in the 6th transmission round; packets 64 – 96 are sent in the 7th transmission round. Thus packet 70 is sent in the 7th transmission round.

i) The threshold will be set to half the current value of the congestion window (8) when the loss occurred and congestion window will be set to the new threshold value + 3 MSS . Thus the new values of the threshold and window will be 4 and 7 respectively.

j) Threshold is 21, and congestion window size is 1.

k) round 17, 1 packet; round 18, 2 packets; round 19, 4 packets; round 20, 8 packets; round 21, 16 packets; round 22, 21 packets. So, the total number is 52.

Problem 46 (15%) (各小題 5%)

a) Let W denote the max window size measured in segments. Then, W*MSS/RTT = 15Mbps, as packets will be dropped if the maximum sending rate exceeds link capacity. Thus, we have W*1200*8/0.16=15*10^6, then W is about 250.

b) As congestion window size varies from W/2 to W, then the average window size is 0.75W=187.5 segments. Average throughput is 187.5*1200*8/0.16=11.25Mbps.

c) (250/2) *0.16= 20 seconds, as the number of RTTs (that this TCP connections needs in order to increase its window size from W/2 to W) is given by W/2. Recall the window size increases by one in each RTT.

Problem 50 (25%)

a) (15%) The key difference between C1 and C2 is that C1's RTT is only half of that of C2. Thus C1 adjusts its window size after 50 msec, but C2 adjusts its window size after 100 msec. Assume that whenever a loss event happens, C1 receives it after 50msec and C2 receives it after 100msec. We further have the following simplified model of TCP. After each RTT, a connection determines if it should increase window size or not. For C1, we compute the average total sending rate in the link in the previous 50 msec. If that rate exceeds the link capacity, then we assume that C1 detects loss and reduces its window size. But for C2, we compute the average total sending rate in the link in the previous 100msec. If that rate exceeds the link capacity, then we assume that C2 detects loss and reduces its window size. Note that it is possible that the average sending rate in last 50msec is higher than the link capacity, but the average sending rate in last 100msec is smaller than or equal to the link capacity, then in this case, we assume that C1 will experience loss event but C2 will not.

The following table describes the evolution of window sizes and sending rates based on the above assumptions.

	C1		C2	
Time (msec)	Window Size (num. of segments sent in next 50msec)	Average data sending rate (segments per second, =Window/0.05)	Window Size(num. of segments sent in next 100msec)	Average data sending rate (segments per second, =Window/0.1)
0	10	200 (in [0-50]msec]	10	100 (in [0-50]msec)
50	5 (decreases window size as the avg. total sending rate to the link in **last 50msec** is 300= 200+100)	100 (in [50-100]msec]		100 (in [50-100]msec)
100	2 (decreases window size as the avg. total sending rate to the link in **last 50msec** is 200= 100+100)	40	5 (decreases window size as the avg. total sending rate to the link in **last 100msec** is 250= (200+100)/2 + (100+100)/2)	50
150	1 (decreases window size	20		50

	as the avg. total sending rate to the link in last 50msec is 90= (40+50)			
200	1 (no further decrease, as window size is already 1)	20	2 (decreases window size as the avg. total sending rate to the link in **last 100msec** is 80= (40+20)/2 + (50+50)/2)	20
250	1 (no further decrease, as window size is already 1)	20		20
300	1 (no further decrease, as window size is already 1)	20	1 (decreases window size as the avg. total sending rate to the link in **last 100msec** is 40= (20+20)/2 + (20+20)/2)	10
350	2	40		10
400	1	20	1	10
450	2	40		10
500	1 (decreases	20	1	10

	window size as the avg. total sending rate to the link in last 50msec is 50= (40+10)			
550	2	40		10
600	1	20	1	10
650	2	40		10
700	1	20	1	10
750	2	40		10
800	1	20	1	10
850	2	40		10
900	1	20	1	10
950	2	40		10
1000	1	20	1	10

Based on the above table, we find that after 1000 msec, C1's and C2's window sizes are 1 segment each.

b) (10%) No. In the long run, C1's bandwidth share is roughly twice as that of C2's, because C1 has shorter RTT, only half of that of C2, so C1 can adjust its window size twice as fast as C2. If we look at the above table, we can see a cycle every 200msec, e.g. from 850msec to 1000msec, inclusive. Within a cycle, the sending rate of C1 is (40+20+40+20) = 120, which is thrice as large as the sending of C2 given by (10+10+10+10) = 40.