Computer Communications HW1C

1(a)

Start Time	Counters A	B	C	Current	Queue
0	0	0	0	A1	N/A
8	8	0	0	—	N/A
10	0	0	0	A2	B1
18	8	0	0	B1	N/A
38	8	5	5	B2	N/A
56	8	10	5	C1	A3, B3, C2, C3
68	8	10	15	A3	B3, C2, C3, A4
76	16	10	15	B3	C2, C3, A4
96	16	15	15	C2	C3, A4, C4
106	16	16	25	A4	C3, C4, A5
114	24	24	25	A5	C3, C4
122	32	25	25	C3	C4
132	35	35	35	C4	N/A
142	0	0	0	—	N/A

(b) Comparing the above solution to the fair queueing ex. from class, only our packet B3 is served at a different time. In class it went at 94, for me it went at 76. The weight for B3 allowed it to go earlier as it changes the speed the counter goes up for the highest packet size, meaning it gives a lower average queueing delay.

2 Packets

RTT	1	2	3	4	5	6	7	8
Sent	1	2-3	4-6	7-10	9-10	11-13	14-17	18-22
WinSize	1	2	3	4	2	3	4	5
Packet lost	–	–	–	9	–	–	–	–
Ack'd	1	3	6	8	10	15	17	22

RTT	9	10	11	12	13	14	15
Sent	23-28	25-27	29-32	30-31	33-35	36-39	38-39
Size	6	3	4	2	3	4	2
Lost	25	–	30	–	–	38	–
Ack	24	28	29	32	35	37	39

RTT	16	17	18	19
Sent	40-42	43-46	47-51	50-51
Size	3	4	5	2
Lost	–	–	50	–
Ack	42	46	49	51

3 Slow start is on until about $t=0.5$, and then again from $t=1.9$ until around $t=2.5$ or so, where TCP changes to linear increase. At $t=5.3$ another packet is sent & lost, which is recognized by TCP at $t=5.5$ ish b/c of retransmits. Figure 6.11 does not have this feature. There is no fast recovery as the window drops to size 1. Then slow start opens to half the previous size until linear increase takes back over around $t=5.7$.

Computer Communications HW10 (cont.)

4 Only when the average queue length exceeds the MaxThreshold are packets automatically dropped. If average queue size is less than MaxThreshold, incoming packets may be queued even if the real queue size is larger than the value of MaxThreshold

5 DEC bit should not work better with ECN than RED, as DEC bit can quickly reduce a window size and disconnect if a certain flow has many marked packets in a burst.

6 Neither flow nor congestion control will matter here as the conditions for them will be trivial here. Simply the rate of returning ACKs will control the speed of data transfer. This is self-clocking.