CHAPTER 6

11. **(a)** First we calculate the finishing times F_i. We don't need to worry about clock speed here since we may take $A_i = 0$ for all the packets. F_i thus becomes just the cumulative per-flow size: $F_i = F_{i-1} + P_i$.

Packet	Size	Flow	F_i
1	200	1	200
2	200	1	400
3	160	2	160
4	120	2	280
5	160	2	440
6	210	3	210
7	150	3	360
8	90	3	450

We now send in increasing order of F_i: Packet 3, Packet 1, Packet 6, Packet 4, Packet 7, Packet 2, Packet 5, Packet 8.

(b) To give flow 1 a weight of 2 we divide each of its F_i by 2: $F_i = F_{i-1} + P_i/2$. To give flow 2 a weight of 4 we divide each of its F_i by 4: $F_i = F_{i-1} + P_i/4$. To give flow 3 a weight of 3 we divide each of its F_i by 3: $F_i = F_{i-1} + P_i/3$. Again, we are using the fact that there is no waiting.

Packet	Size	Flow	Weighted F_i
1	200	1	100
2	200	1	200
3	160	2	40
4	120	2	70
5	160	2	110
6	210	3	70
7	150	3	120
8	90	3	150

Transmitting in increasing order of the weighted F_i we send as follows: Packet 3, Packet 4, Packet 6, Packet 1, Packet 5, Packet 7, Packet 8, Packet 2.

15. (a) For the ith arriving packet on a given flow we calculate its estimated finishing time F_i by the formula $F_i = \max\{A_i, F_{i-1}\} + 1$, where the clock used to measure the arrival times A_i runs slow by a factor equal to the number of active queues. The A_i clock is global; the sequence of F_i values calculated as above is local to each flow.

The following table lists all events by wall clock time. We identify packets by their flow and arrival time; thus, packet A4 is the packet that arrives on flow A at wall clock time 4 (ie., the third packet). The last three columns are the queues for each flow for the subsequent time interval, *including* the packet currently being transmitted. The number of such active queues determines the amount by which A_i is incremented on the subsequent line. Multiple packets appear on the same line if their F_i values are all the same; the F_i values are in italic when $F_i = F_{i-1} + 1$ (versus $F_i = A_i + 1$).

Wall Clock	A_i	Arrivals	F_i	Sent	A's Queue	B's Queue	C's Queue
1	1.0	A1,B1,C1	2.0	A1	A1	B1	C1
2	1.333	C2	3.0	B1		B1	C1,C2
3	1.833	A3	3.0	C1	A3		C1,C2
4	2.333	B4	3.333	A3	A3	B4	C2,C4
		C4	4.0				
5	2.666	A5	4.0	C2	A5	B4	C2,C4
6	3.0	A6	5.0	B4	A5,A6	B4	C4,C6
		C6	5.0				
7	3.333	B7	4.333	A5	A5,A6	B7	C4,C6,C7
		C7	6.0				
8	3.666	A8	6.0	C4	A6,A8	B7,B8	C4,C6,C7
		B8	5.333				
9	4	A9	7.0	B7	A6,A8,A9	B7,B8,B9	C6,C7
		B9	6.333				

(*Continued*)

Wall Clock	A_i	Arrivals	F_i	Sent	A's Queue	B's Queue	C's Queue
10	4.333			A6	A6,A8,A9	B8,B9	C6,C7
11	4.666	A11	8.0	C6	A8,A9,A11	B8,B9	C7
12	5	C12	7.0	B8	A8,A9,A11	B8,B9	C7,C12
13	5.333	B13	7.333	A8	A8,A9,A11	B9,B13	C7,C12
14	5.666			C7	A9,A11	B9,B13	C7,C12
15	6.0	B15	8.333	B9	A9,A11	B9,B13,B15	C12
16	6.333			A9	A9,A11	B13,B15	C12
17	6.666			C12	A11	B13,B15	C12
18	7			B13	A11	B13,B15	
19	7.5			A11	A11	B15	
20	8			B15		B15	

(b) For weighted fair queuing we have, for flow B,

$$F_i = max\{A_i, F_{i-1}\} + 0.5$$

For flows A and C, F_i is as before. Here is the table corresponding to the one above:

Wall Clock	A_i	Arrivals	F_i	Sent	A's Queue	B's Queue	C's Queue
1	1.0	A1,C1	2.0	B1	A1	B1	C1
		B1	1.5				
2	1.333	C2	3.0	A1			C1,C2
3	1.833	A3	3.0	C1	A1		C1,C2
4	2.333	B4	2.833	B4	A3	B4	C2,C4
		C4	*4.0*				
5	2.666	A5	4.0	A3	A3,A5		C2,C4
6	3.166	A6	5.0	C2	A5,A6		C2,C4,C6
		C6	5.0				
7	3.666	B7	4.167	A5	A5,A6	B7	C4,C6,C7
		C7	6.0				
8	4.0	A8	6.0	C4	A6,A8	B7,B8	C6,C7
		B8	4.666				

(Continued)

Wall Clock	A_i	Arrivals	F_i	Sent	A's Queue	B's Queue	C's Queue
9	4.333	A9	7.0	B7	A6,A8,A9	B7,B8,B9	C6,C7
		B9	5.166				
10	4.666			B8	A6,A8,A9	B8,B9	C6,C7
11	5.0	A11	8.0	A6	A6,A8,A9,A11	B9	C6,C7
12	5.333	C12	7.0	C6	A8,A9,A11	B9	C6,C7,C12
13	5.666	B13	6.166	B9	A8,A9,A11	B9,B13	C7,C12
14	6.0			A8	A9,A11	B13	C7,C12
15	6.333	B15	6.833	C7	A9,A11	B13,B15	C12
16	6.666			B13	A9,A11	B13,B15	C12
17	7.0			B15	A11	B15	C12
18	7.333			A9	A11		C12
19	7.833			C12	A11		C12
20	8.333			A11	A11		

35. (a) We have

$$\mathsf{TempP} = \mathsf{MaxP} \times \frac{\mathsf{AvgLen} - \mathsf{MinThreshold}}{\mathsf{MaxThreshold} - \mathsf{MinThreshold}}.$$

AvgLen is halfway between MinThreshold and MaxThreshold, which implies that the fraction here is $1/2$ and so $\mathsf{TempP} = \mathsf{MaxP}/2 = p/2$. We now have

$$\mathsf{P}_{\mathsf{count}} = \mathsf{TempP}/(1 - \mathsf{count} \times \mathsf{TempP}) = 1/(x - \mathsf{count}),$$

where $x = 2/p$. Therefore,

$$1 - \mathsf{P}_{\mathsf{count}} = \frac{x - (\mathsf{count} + 1)}{x - \mathsf{count}}.$$

Evaluating the product

$$(1 - \mathsf{P}_1) \times \cdots \times (1 - \mathsf{P}_n)$$

gives

$$\frac{x-2}{x-1} \cdot \frac{x-3}{x-2} \cdots \cdot \frac{x-(n+1)}{x-n} = \frac{x-(n+1)}{x-1},$$

where $x = 2/p$.

(b) From the result of previous question,

$$\alpha = \frac{x-(n+1)}{x-1}.$$

Therefore,

$$x = \frac{(n+1)-\alpha}{1-\alpha} = 2/p.$$

Accordingly,

$$p = \frac{2(1-\alpha)}{(n+1)-\alpha}$$

48. At every second, the bucket volume must not be negative. For a given bucket depth D and token rate r, we can calculate the bucket volume $v(t)$ at time t seconds and enforce $v(t)$ being non-negative:

$$\begin{aligned}
v(0) &= D - 5 + r = D - (5-r) \geq 0 \\
v(1) &= D - 5 - 5 + 2r = D - 2(5-r) \geq 0 \\
v(2) &= D - 5 - 5 - 1 + 3r = D - (11-3r) \geq 0 \\
v(3) &= D - 5 - 5 - 1 + 4r = D - (11-4r) \geq 0 \\
v(4) &= D - 5 - 5 - 1 - 6 + 5r = D - (17-5r) \geq 0 \\
v(5) &= D - 5 - 5 - 1 - 6 - 1 + 6r = D - 6(3-r) \geq 0
\end{aligned}$$

We define the functions $f_1(r), f_2(r), \ldots, f_6(r)$ as follows:

$$\begin{aligned}
f_1(r) &= 5 - r \\
f_2(r) &= 2(5-r) = 2f_1(r) \geq f_1(r) \quad (\mathit{for}\ 1 \leq r \leq 5) \\
f_3(r) &= 11 - 3r \leq f_2(r) \quad (\mathit{for}\ r \geq 1) \\
f_4(r) &= 11 - 4r < f_3(r) \quad (\mathit{for}\ r \geq 1) \\
f_5(r) &= 17 - 5r \\
f_6(r) &= 6(3-r) \leq f_5(r) \quad (\mathit{for}\ r \geq 1)
\end{aligned}$$

First of all, for $r \geq 5$, $f_i(r) \leq 0$ for all i. This means if the token rate is faster than 5 packets per second any positive bucket depth will suffice (i.e., $D \geq 0$). For $1 \leq r \leq 5$, we only need to consider $f_2(r)$ and $f_5(r)$, since other functions are less than these functions.

One can easily find $f_2(r) - f_5(r) = 3r - 7$. Therefore, the bucket depth D is enforced by the following formula:

$$D \geq \begin{cases} f_5(r) = 17 - 5r & (r = 1, 2) \\ f_2(r) = 2(5 - r) & (r = 3, 4, 5) \\ 0 & (r \geq 5) \end{cases}$$

CHAPTER 7

2. Each string is preceded by a count of its length; the array of salaries is preceded by a count of the number of elements. That leads to the following sequence of integers and ASCII characters being sent:

```
4 M A R Y 4377 7 J A N U A R Y 7 2002 2 90000 150000 1
```

8.

INT	4	15
INT	4	29496729
INT	4	58993458

10.

```
15 be 00000000 00000000 00000000 00001111
15 le 00001111 00000000 00000000 00000000

29496729 be 00000001 11000010 00010101 10011001
29496729 le 10011001 00010101 11000010 00000001

58993458 be 00000011 10000100 00101011 00110010
58993458 le 00110010 00101011 10000100 00000011
```