CSC467 | # Assignment 2 | Jakob Roberts v00484900

1

If the source sends data at half the link rate, that would by 0.75Mbit/s. If it is transferring for 5 seconds, that makes a transfer of 3.5Mbits. If each packet is on average of 2000 bits, it would be sending 1750 total.

2

a) A weighted round robin system schedule could work as follows: take the inverse of the given weight A is 1, B is 1/2, C is 1/3, D is 1/4. This would mean that A would get 4 times as much traffic allowance as D, this would ensure that all A, B, C, and D get bandwidth time but still giving a priority to A. As it is round robin, each flow would go in order but cease when it runs out of packets, as A has the most available packets, it would begin first and run out last.

b) If we assume that the link bandwidth is 1.5Mbit/s, we can add up all the weights and get a percentage of usage. A get 48%, B gets 24%, C gets 16%, and D gets 12%. This comes out to the values of A having 0.72Mbit/s, B having 0.36Mbit/s, C having 0.24Mbit/s, and D having 0.18Mbit/s.

c) As "cylces" are not really defined in this question, I am going to assume that they are 1 second per cycle. Over 4 cycles, this would give values of A having transfered 2.88Mbit, B having transfered 1.44Mbit, C having transfered 0.96Mbit, and D having transfered 0.72Mbit.

d) Due to the weight distribution we can consider even packets that match the weights based on packets of multiples of 25.

   If we send out 24 packets for example, 12 will be from A, 6 will be from B, 4 will be from C, and 3 will be from D. As there is nothing but the priority of the Flows to base the packet sequence on, there is not any possible way of getting an accurate representation with a finer resolution than an even packet distribution based on WFQ.

e) The bandwidth that each flow gets is similar as before: A having 0.72Mbit/s, B having 0.36Mbit/s, C having 0.24Mbit/s, and D having 0.18Mbit/s.

f) WRR is going to have better sustained for all the flows, but WFQ could have better burst if all the packets for a single flow go in an order.

g) Not particularly possible to make them any more similar than they already are, they are both weighted and run off a schedule that is bandwidth based. The schedules are relatively similar as they have to have the same per-cycle output to meet the weight requirements.

h) As the weighted flow is related not on packet size but on bandwidth being transfered, the transfer rates would remain the same but the number of packets sent from each flow would change. As stated: over 4 cycles, this would give values of A having transfered 2.88Mbit, B having transfered 1.44Mbit, C having transfered 0.96Mbit, and D having transfered 0.72Mbit.

i) As stated: over 4 cycles, this would give values of A having transfered 2.88Mbit, B having transfered 1.44Mbit, C having transfered 0.96Mbit, and D having transfered 0.72Mbit.

3

Out of 100 packets, I would give 10 to flow A, 18 to flow B, 24 to flow C and 48 to flow D. This meets the weight division requirements for each of the flows.

4

Assuming that 10 and 15 arrive at time 0, 5 arriges at time 9, and 10 arrives at time 17. With SCFQ, we would do packet 15, 10, 10, 5. This may seem unfair but that is how it works as SCFQ takes the maximum of the packets in queue.

5

Assume connection 1 has a load of 0.2 with a mean wait time of 0.3s. Assume connection 2 has a load of 0.4 with a mean wait time of 0.4s. If the delay of connetion 1 drops from 0.3s to 0.1s then this will change the waiting time of connection 2 to 0.5s.

6

Assuming that each flow has an equal weight, we can say that each flow should be given 16.667Mb/s flow rate. Unfortunately not each flow requires that much so we can divide it a bit further. Lets take flow f1, f2, and f3: they are all below 16.667Mb/s so they can all receive their max rates. this leaves us with 100 - 1 - 2 - 10 = 87Mb/s split between f4, f5, and f6. This means that each can receive 29Mb/s bandwidth. Unfortunately f4 is capped at 20Mb/s bandwidth and therefore will get it's full speed. This leaves 67Mbit/s split between f5 and f6. This means that both f5 and f6 get 33.5Mbit/s.

Summary: f1 = 1Mb/s, f2 = 2Mb/s, f3 = 10Mb/s, f4 = 20Mb/s, f5 = 33.5Mb/s, f6 = 33.5Mb/s

7

Initially we can say that, based on the weights, the flows should be given: f1 = 5Mb/s, f2 = 5Mb/s, f3 = 10Mb/s, f4 = 10Mb/s, f5 = 20Mb/s, f6 = 50Mb/s.

We have to limit f1 to 1Mb/s and f2 to 2Mb/s, where f3 can have the maximum that it requested. Now that we have removed those 3 from the equation as they have reached their maximum, we have f4, f5, and f6 to consider again. With the remaining 87Mb/s with weights w4=20, w5=40, w6=100 we can say that f4 can have 10.875Mb/s, f5 can have 21.75Mb/s, and f6 can have 54.375Mb/s.

Summary: f1 = 1Mb/s, f2 = 2Mb/s, f3 = 10Mb/s, f4 = 10.875Mb/s, f5 = 21.75Mb/s, f6 = 54.375Mb/s

8

a) 1/250 of a second per packet assuming that Class 2 is not sending any data

b) 1/125 of a second per packet assuming that Class 1 is not sending any data

c) these arrival rates give us a queue utilization of 0.5 (50%) with 75 packets of varying size. Quantity of 27.5 customers.

   Class 1 has a wait time of only 0.1s to complete all its packets because it has high priority, whereas Class 2 has to wait 0.5s (full queue utilization) as it has to wait for the higher priority Class 1 to finish.

d) these arrival rates give us a queue utilization of 0.6 (60%) with 100 packets of varying size. Quantity of 40 customers.

   Class 1 has a wait time of only 0.2s to complete all its packets because it has high priority, whereas Class 2 has to wait 0.6s (full queue utilization) as it has to wait for the higher priority Class 1 to finish.

e) these arrival rates give us a queue utilization of 0.8 (80%) with 150 packets of varying size. Quantity of 80 customers.

   Class 1 has a wait time of only 0.4s to complete all its packets because it has high priority, whereas Class 2 has to wait 0.8s (full queue utilization) as it has to wait for the higher priority Class 1 to finish.

f) Given an increase in delay for Class 1, Class 2 must incur the same increase in delay as Class one has a higher priority than Class 2.