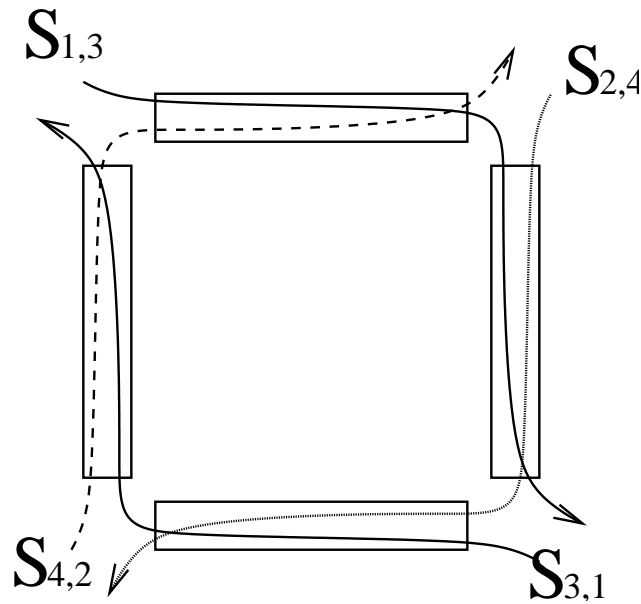


HW #8 Solutions



1. Consider the network pictured above where each link has capacity C , each session transmits into the network at rate ρ , and each session's transmissions traverse two links. In class, we showed how increasing the flow rate beyond $C/2$ led to congestion collapse. How can priority queueing be used to prevent collapse where the priority mechanism does not know the capacity of the link (i.e., you cannot simply restrict the entry rate of the flow into the link to $C/2$)? (Hint: How should each link prioritize the session data that it carries so that flow is not "wasted" by passing through the first link and then being dropped at the second?)

Answer:

In this problem first of all we have to notice the symmetry of the problem. For each flow we have to pass through two different links so for each flow we have to assign two priorities P^1 and P^2 corresponding to the first and the second link that each flow traverse. Assigning the same priority is like assigning no priority and collapse will occur. Also since only two flows share each link we can only have two priority values 1 and 2. For a flow (lets say $S_{1,3}$) we have the following four possibilities:

- a) Assign a high priority to both links so $P_{S_{1,3}}^1 = 1$ and $P_{S_{1,3}}^2 = 1$. With this configuration the other flows that share the links with $S_{1,3}$ i.e $S_{4,2}$ and $S_{2,4}$ must have $P_{S_{2,4}}^1 = 2$ and $P_{S_{4,2}}^2 = 2$. But packets for $S_{4,2}$ that have passed link 4 may drop in link 1 because it has to compete with $S_{1,3}$ that has higher priority.
- b) Assign a low priority to both links so $P_{S_{1,3}}^1 = 2$ and $P_{S_{1,3}}^2 = 2$. With this configuration the other flows that share the links with $S_{1,3}$ i.e $S_{4,2}$ and $S_{2,4}$ must have $P_{S_{2,4}}^1 = 1$ and $P_{S_{4,2}}^2 = 1$. But packets for $S_{2,4}$ that have passed link 2 may drop in link 3 because it has to compete with $S_{3,1}$ that may have higher priority.
- c) Assign a high priority to first link and low priority to the second so $P_{S_{1,3}}^1 = 1$ and $P_{S_{1,3}}^2 = 2$. With this configuration flow $S_{1,3}$ will have to compete on link 2 with $S_{2,4}$ that will have a higher priority and maybe it will drop packets. Therefore if all flows send with rate $> C$ then all packets will cross the link and will be dropped on the second link since they will have lower priority.
- d) Assign a low priority to first link and high priority to the second so $P_{S_{1,3}}^1 = 2$ and $P_{S_{1,3}}^2 = 1$. With this configuration flow $S_{1,3}$ will have to compete on link 2 with $S_{2,4}$ that will have a lower priority so every packet

that started from link 1 will surely pass through link 2.

Therefore the solution for all links is to assign low priority on their first link and high priority on their second link in order to prevent packets that are already transmitted to drop on the second link.

2. A flow will traverse a set of routers R_1, \dots, R_n where router R_i will process its packets at (deterministic) rate λ_i and can queue up to b_i packets at a time without dropping any.
 - (a) Assuming the flow has reserved access to the router resources (such that no other flow competes for the resources described above), what is the leaky bucket configuration (ρ, b) that the flow should use (where ρ is the rate at which tokens enter the bucket and b is the maximum number of tokens the bucket can hold) to maximize the transmission rate such that a packet is never lost?

Answer:

In this exercise we assume that we do not know the position of each router. In this case we have that:

$$\rho = \min\{\lambda_i, 1 \leq i \leq n\}$$

and

$$b = \min\{b_i, 1 \leq i \leq n\}$$

If $\rho > \min\{\lambda_i, 1 \leq i \leq n\}$ then we can have a flow that has a constant rate of at least ρ that after a finite time t will overflow the buffer of the link k with $\lambda_k = \min\{\lambda_i, 1 \leq i \leq n\}$ since the arrival rate is bigger than service rate (traffic intensity > 1). On the other hand if we have a bucket size $b > \min\{b_i, 1 \leq i \leq n\}$ and we can assume that the first router has $b_1 = \min\{b_i, 1 \leq i \leq n\}$ since we don't know the position of the router with the lower buffer. Then we can have a flow f that waits until the bucket is full of tokens b and then it creates a peak of b packets this will overflow router 1 since $b > b_1$.

- (b) Give an example of transmission sequence that leads to a packet loss when the leaky bucket is configured using (ρ', b') , where $\rho' = \rho$ and $b' > b$.

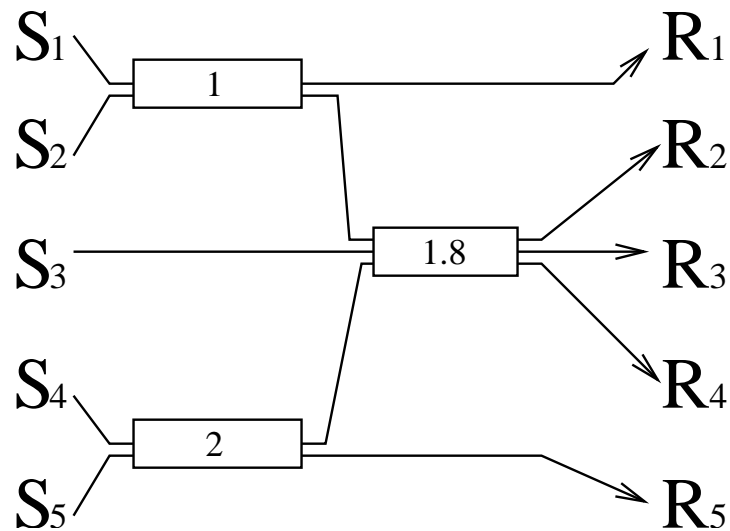
Answer:

The simplest exam is the one with only the first router. If $\rho' = \rho$ and $b' > b$ with $b = \min\{b_i, 1 \leq i \leq n\} = b_1$ then $b' > b_1$. Now if we have a flow f that waits until the bucket is full of tokens b' and then it creates a peak of b' packets this will overflow router 1 since $b' > b = b_1$.

- (c) Give an example of transmission sequence that leads to a packet loss when the leaky bucket is configured using (ρ', b') , where $\rho' > \rho$ and $b' = b$.

Answer:

Again the simplest example is the one with just the first router. If $\rho' > \rho$ and $b' = b$ then $\rho' > \min\{\lambda_i, 1 \leq i \leq n\} = \rho_1$. Now we have a flow that has a constant rate of at least ρ' that after a finite time t will overflow the buffer b_1 since the arrival rate is bigger than service rate (traffic intensity > 1).



3. Consider the network shown above.

- (a) Explain why an allocation of rates 0.4, 0.6, 0.6, 0.6, 1.4 respectively to flows 1,2,3,4,5 is not max-min fair.

Answer:

Increasing $S_1 = 0.4$ to $S_1 = 0.4 + \epsilon$ requires a decrease in the $S_2 = 0.6$ flow. Since we can increase S_1 without being forced to decrease any flow whose rate is 0.4 or less, the original allocation cannot be max-min fair.

- (b) Explain why an allocation of rates 0.5, 0.5, 0.3, 1, 1 respectively to flows 1,2,3,4,5 is not max-min fair.

Answer:

Increasing $S_3 = 0.3$ to $S_3 = 0.3 + \epsilon$ requires a decrease in one or both $S_2 = 0.5$ or $S_4 = 1$ flows. Since we can increase S_3 without being forced to decrease any flow whose rate is 0.3 or less, the original allocation cannot be max-min fair.

- (c) What is the max-min fair allocation for the depicted network?

Answer:

The max-min fair allocation for the depicted network is $S_1 = 0.5$, $S_2 = 0.5$, $S_3 = 0.65$, $S_4 = 0.65$, $S_5 = 1.35$ for flows 1,2,3,4,5 respectively.