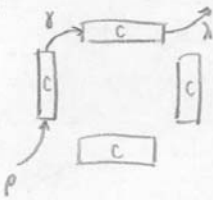


1) a)



Assign high priority on the second link and low priority on the first link to avoid wasting flow that has already taken capacity on the first link.

$$\text{for } p > \frac{c}{2}: p \geq \delta \quad \delta = \lambda$$

$$\text{If } D \text{ is fraction lost, } (1-D)p = \delta$$

$$(1-D)p + \delta = c$$

$$\delta + \delta = c$$

$$\delta = \frac{c}{2} = \lambda$$

(10)

b) Both protocols prevent congestion collapse, since both round robin and weighted fair queuing keep flows' exit rates constant regardless of whether on the first or the second link. So the exit rate is $\frac{c}{2}$ for $p > \frac{c}{2}$

(10)

(10)

2) a) $p = \min(\lambda_i)$ since the flow is only as fast as its slowest router.

(10)

$b = \min(b_i)$ since the minimum queue size is what the flow can handle when it gets a load of packets (assuming some general processing distribution)

b) $p' = p, b' > b$ will cause overflow when the leaky bucket becomes full, since $b = \min(b_i)$

[the minimum queue size] cannot handle $b' > b$. So packets will be lost at the first router.

(10)

c) $p' > p, b' = b$ loses packets anytime the rate reaches p' , since the slowest router will lose the packets after its queue fills up.

(10)

3) a) If you increase flow 4's rate, you don't have to decrease any lesser or equal flow rate (flows 1,2 have greater rates)

\therefore it's not min-max fair.

(20)

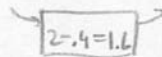
b) assign $p_i = 0$ for all i

increase all p_i by the same amount until some link reaches capacity



(20)

remove any p_i that passes through a full link, and start again.



$$p_3 = 1.6$$

$$p_1, p_2, p_4 = 0.4$$