Exam answers for 2008/9 & resit from 2007/8

Q1 from 2008/9

All bookwork. Note! students have been instructed that, for this course, "write an essay on" simply means "assemble and copy out from memory the relevant lecture notes".

(i) Example from drift models:

Appropriate example. 4

A drift model is an equation for the expected change in a quantity per unit time. For example, in TCP, let $w(t)$ be the congestion window at time t. By considering the congestion control algorithm,

$$\mathbb{E}\, w(t+\delta) = w(t) + \frac{1}{w(t)}\frac{w(t)\,\delta}{RTT} + p\,\frac{w(t)\,\delta}{RTT}\,\frac{w(t)}{2}$$

So drift $= \dfrac{\mathbb{E}\,w(t+\delta) - w(t)}{\delta} = \dfrac{1}{RTT} - p\,\dfrac{w(t)^2}{2RTT}.$

What is drift? 3

A fixed point is where there is zero drift. In this case,

$$\text{drift} = 0 \Rightarrow \frac{1}{RTT} = \frac{p w^2}{2RTT} \Rightarrow w = \sqrt{\frac{2}{p}}.$$

What is zero drift? 3

To find the fixed points numerically, we could choose some small δ and repeatedly update these equations, e.g. in Excel

t	$w(t)$	drift
0	initial guess	use drift formula, applied to ← cell
$=\uparrow + \delta$	$= \uparrow + \delta \times \nearrow$	"
"	"	"

Algorithm 5

After a while this should settle down.
If it doesn't, then either
(a) there are no stable fixed points nearby
or (b) δ is too large — try making it smaller, or use a proper diff. eq. solver.

Bonus
Insight into algorithm failures

(ii) Example of solving simultaneous equations:

Appropriate example 4

In a loss network, we might have concluded

$$B_1 = E(\text{some function of } B_1 \text{ and } B_2,\ c_1) := f(B_1, B_2)$$
$$B_2 = E(\text{some other func. of } B_1 \text{ and } B_2,\ c_2) := g(B_1, B_2).$$

where B_1 and B_2 are blocking probabilities, c_1 and c_2 are link capacities measured in no. of circuits, and E is the Erlang function

We could solve this numerically by making an initial guess e.g.
$$B_1^{(0)} = \tfrac{1}{2}, \qquad B_2^{(0)} = \tfrac{1}{2}$$
and repeatedly updating
$$B_1^{(n+1)} = f(B_1^{(n)}, B_2^{(n)})$$
$$B_2^{(n+1)} = g(B_1^{(n+1)}, B_2^{(n)}).$$

Algorithm: 5

Keep iterating until the values settle down to some limit $B_1^{(\infty)}$, $B_2^{(\infty)}$.
These limiting values must solve the original simultaneous equations.

limiting value idea: 3

Bonus: insight into alg. failures

(iii) A fixed point x_0 is <u>stable</u> if, when you start either of these update procedures from close to x_0, you head towards x_0.

It is <u>unstable</u> if you head away from x_0.

stable+unstable: 2+2

link to both kinds: 2

Q1 from 2007/8 resit

(a) Sort the dataset.

lookwork

let $n = \#$ observations.

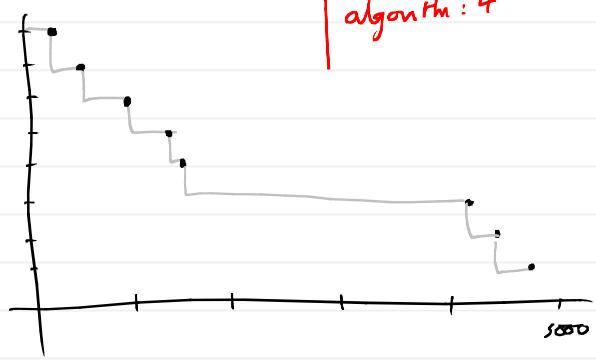
Plot

x	y
smallest obs	n/n
.	$\frac{n-1}{n}$
.	.
.	.
largest obs	$1/n$

algorithm: 4

ie

x	
186	1
431	7/8
1108	6/8
1409	5/8
1427	4/8
4215	3/8
4315	2/8
4713	1/8



execution: 2

(b) $$\text{lik}(\lambda,\mu) = \prod_{i=1}^{n} f(x_i)$$ where x_i are the observations

correct lik(·): 5

$$\log \text{lik}(\lambda,\mu) = \sum_{i=1}^{n} \begin{cases} \log\lambda - \lambda x_i & \text{if } x_i \le 1024 \\ \log(\lambda+\mu) - (\lambda+\mu)x_i + 1024\mu & \text{if } x_i > 1024 \end{cases}$$

$$= n_{\le 1024}\log\lambda - \lambda S_{\le 1024}$$
$$+ n_{>1024}\log(\lambda+\mu) - (\lambda+\mu)S_{>1024} + 1024\mu\, n_{>1024}$$

where $n_{\le 1024} = \#\, x_i \le 1024$

$S_{\le 1024} = \sum_{i: x_i \le 1024} x_i$ etc.

$$\frac{\partial}{\partial\lambda} = 0 \Rightarrow \frac{n_{\le 1024}}{\lambda} + \frac{n_{>1024}}{\lambda+\mu} = S_{\le 1024} + S_{>1024}$$

$$\frac{\partial}{\partial\mu} = 0 \Rightarrow \frac{n_{>1024}}{\lambda+\mu} = S_{>1024} - 1024\, n_{>1024}$$

idea of maximizing by using $\frac{\partial}{\partial\cdot} = 0$: 3

idea of simultaneous soln: 3

execution: 4

Hence $$\frac{n_{\le 1024}}{\lambda} = S_{\le 1024} + S_{>1024} - S_{>1024} + 1024\, n_{>1024} \Rightarrow \hat{\lambda} = \frac{n_{\le 1024}}{S_{\le 1024} + 1024\, n_{>1024}}.$$

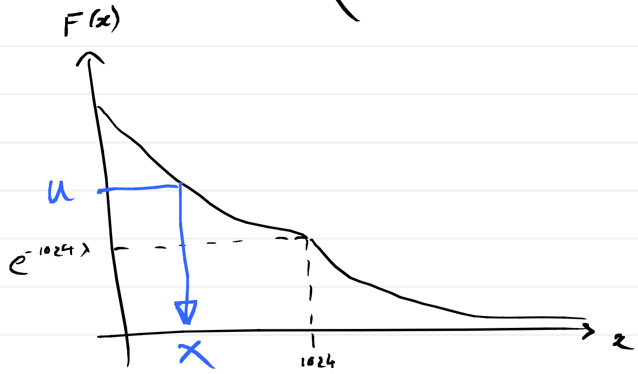
So $$\hat{\mu} = \frac{n_{>1024}}{S_{>1024} - 1024\, n_{>1024}} - \hat{\lambda}.$$

(c) Let $U \sim \text{Uniform}[0,1]$.

Choose Uniform : 2

Solve $F(x) = \begin{cases} e^{-\lambda x} & \text{if } x \leq 1024 \\ e^{-(\lambda+\mu)x + 1024\mu} & \text{if } x > 1024 \end{cases} = U$

correct link to $F(\cdot)$: 5



This X is the output of the random number generator.

In particular, if $U \geq e^{-1024\lambda}$ then solve $e^{-\lambda X} = U \Rightarrow X = -\frac{1}{\lambda} \log U$,

else solve $e^{-(\lambda+\mu)X + 1024\mu} = U \Rightarrow -(\lambda+\mu)X + 1024\mu = \log U$

$$\Rightarrow X = \frac{-1}{\lambda+\mu}\left(\log U - 1024\mu\right).$$

calculation : 5

In summary: let U = random uniform $[0,1]$

let $X = -\frac{1}{\lambda}\log U$ if $U \geq e^{-1024\lambda}$ else $-\frac{1}{\lambda+\mu}(\log U - 1024\mu)$

Return X.

I'd rather see the answer presented in pseudo-code like this; but, absent pseudo-code, I'll be satisfied if the intention is clear from the maths.

Q2 from 2008/2009

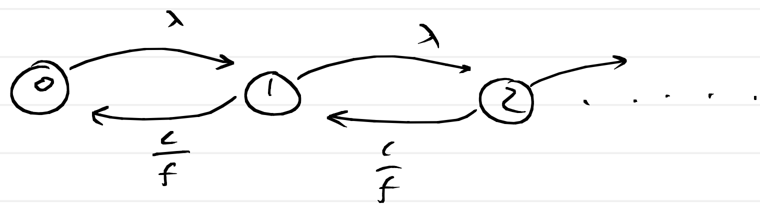
All bookwork.

(a) A processor sharing link is a model for a system in which jobs arrive, stay in the system until all the work they brought has been served, and then depart; while there are n jobs in the system, each of them receives service at rate $\frac{C}{n}$ where C is the link capacity.

| 4 marks. Service rate / Job size / Arrive/depart.

(b) Assume arrivals are a Poisson process of rate λ, and that job sizes are Exponential with mean f. The number of jobs active at time t, N_t, is a Markov process:

| proper setup : 3



| correct diagram : 2

- Arrivals are a Poisson process, so interarrival times are $\sim Exp(\lambda)$. By the memoryless property, if I turn up at some time t, the time until the next arrival is $\sim Exp(\lambda)$. Hence the up-rates. | 2 marks.
- If there are n jobs active, the residual job sizes $X_1, \cdots, X_n$ are all $\sim Exp(\frac{1}{f})$ by the memoryless property. So the residual service times are $X_1/\frac{C}{n}, \cdots, X_n/\frac{C}{n} \sim Exp(\frac{C}{nf})$. So the time until the next departure is $\min(X_1/C/n, \cdots, X_n/C/n) \sim Exp(n \times \frac{C}{nf})$, hence the down-rates. | 3 marks

(c) Detailed balance equations: $\pi_n \lambda = \pi_{n+1} \frac{C}{f} \Rightarrow \pi_{n+1} = \frac{\lambda f}{C} \pi_n = \rho \pi_n$.

So $\pi_1 = \rho \pi_0$, $\pi_2 = \rho^2 \pi_0$, etc.

To normalize, $\pi_0 + \pi_1 + \cdots = 1 \Rightarrow \pi_0(1+\rho+\rho^2+\cdots) = 1 \Rightarrow \pi_0 = 1-\rho$.

Thus $\pi_n = (1-\rho)\rho^n$.

So, eqm # jobs $\sim$ Geom$(1-\rho) - 1$.

$\mathbb{E}$ # jobs $= \frac{1}{1-\rho} - 1 = \frac{\rho}{1-\rho}$.

| Balance : 3
| Normalize. 2
| Execution of eqm dist : 2
| Mean # jobs : 3

(d) Little's Law: $N = \lambda W$ where N = mean occupancy, λ = arrival rate, W = mean occupancy time, for jobs in a system.

| 2: statement
| 2: explain terms

(e) $W = \frac{N}{\lambda} = \frac{\rho}{1-\rho} \cdot \frac{1}{\lambda} = \frac{\lambda f/C}{1-\lambda f/C} \cdot \frac{1}{\lambda} = \frac{f}{C-\lambda f}$.

| 3: bring in correct terms
| 2: answer

Q2 from 2007/8 resit

All bookwork apart from (e)

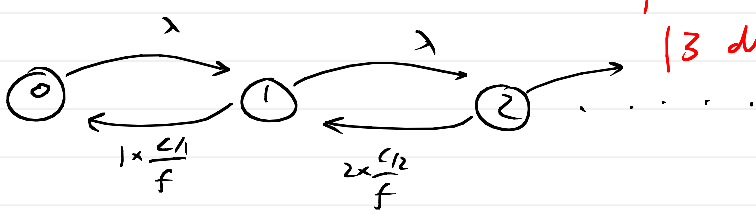
(a) A processor sharing link is a model for a system in which jobs arrive, stay in the system until all the work they brought has been served, and then depart; while there are n jobs in the system, each of them receives service at rate $\frac{C}{n}$ where C is the link capacity.

| 3 marks. throughput / bring work / arrive + depart.

(b) Assume arrivals are a Poisson process of rate λ, and that job sizes are Exponential with mean f. The number of jobs active at time t, N_t, is a Markov process:

| 2 setup

| 3 diagram.



(c) Detailed balance equations: $\pi_n \lambda = \pi_{n+1}\frac{C}{f} \Rightarrow \pi_{n+1} = \frac{\lambda f}{C}\pi_n = \rho\,\pi_n$.

So $\pi_1 = \rho\pi_0$, $\pi_2 = \rho^2\pi_0$, etc.

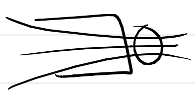
To normalize, $\pi_0 + \pi_1 + \cdots = 1 \Rightarrow \pi_0(1+\rho+\rho^2+\cdots) = 1 \Rightarrow \pi_0 = 1-\rho$.

Thus $\pi_n = (1-\rho)\rho^n$.

So, eqm # jobs $\sim$ Geom$(1-\rho) - 1$.

$\mathbb{E}$ # jobs $= \frac{1}{1-\rho} - 1 = \frac{\rho}{1-\rho}$.

| Balance: 3
| Normalize. 2
| Execution of eqm dist: 2
| Mean # jobs: 3

(d) 

Let there be N flows, each sending at rate x, sharing a link of capacity C with buffer B.

| setup: 3

$$x = \frac{\sqrt{2}}{RTT\sqrt{p}} \qquad p = \frac{\rho^B(1-\rho)}{1-\rho^{B+1}}$$ (drop formula for M/M/1/B queue) where $\rho = \frac{Nx}{C}$

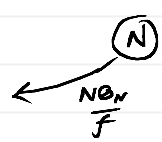
| appropriate eqns: 3

Alternatively, $$\rho = \frac{\sqrt{2}}{RTT\cdot\frac{C}{N}\sqrt{p}} \quad \text{and} \quad p = \frac{\rho^B(1-\rho)}{1-\rho^{B+1}}.$$

| solution concept: 3

Plot these two on a graph of p against ρ and find the intersection; this lets you find $x = \frac{\rho C}{N}$ and actual throughput $x(1-p) = \frac{\rho C}{N}(1-p)$.

| actual throughput: 1

(e) Down rates are now  where θ_N is throughput with N flows.

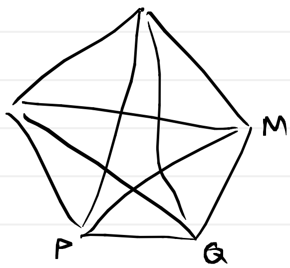
| correct term: 3. Correct place: 2.

Q3

13 marks

(a)

bookwork



A call from e.g. P to Q would first of all try the direct link. If that link has a circuit free, route the call P–Q.

Otherwise, pick a node M at random from the others. If P–M and M–Q both have a circuit free, route it P–M–Q. Otherwise, the call is blocked.

(b) λ = arrival rate of calls between any pair of nodes.

terms: 4 marks

bookwork

μ^{-1} = mean call duration

c = # links on each circuit

$E(\frac{\lambda}{\mu}, c)$ = Erlang formula for blocking probability on an isolated link with c circuits, arrival rate λ, mean call duration μ^{-1}.

Consider the link P–M. The offered traffic is

idea of offered load : 2

putting it together: 4

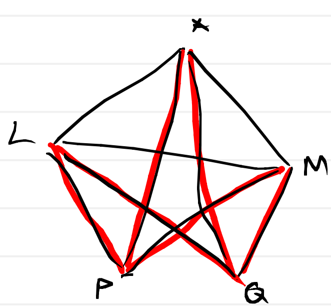
$$\underset{\text{direct traffic}}{\lambda} + \underbrace{2(n-2)}_{\substack{\text{\# of other nodes Q}\\ \text{that could use P–M,}\\ \text{either P–M–Q or}\\ \text{M–P–Q}}} \lambda B \underbrace{\tfrac{1}{n-2}}_{\substack{\text{chance that}\\ \text{P–Q or M–Q call}\\ \text{chooses to route over}\\ \text{M–P}}} (1-B)$$

B: the P–Q or M–Q call must be blocked on the direct route

$1-B$: the other leg of the two-hop route must not block.

Hence $B = E\left(\frac{\lambda + 2(n-2)\lambda B \frac{1}{n-2}(1-B)}{\mu}, c\right)$ as per the equation.

(c)



Offered traffic on PM:

λ (direct traffic)

$+ \lambda \frac{1}{n-2}(1-B_{red})$ (traffic P–Q that chooses waypoint M)

$+ (n-3)\lambda B_{red} \frac{1}{n-3}(1-B_{black})$ (traffic P–L that chooses 2-hop route via M)

$+ (n-3)\lambda B_{black} \frac{1}{n-2}(1-B_{red})$ (traffic M–L that chooses 2-hop route via P)

So $B_{red} = E\left[\frac{\lambda}{\mu}\left(1 + \frac{1}{n-2}(1-B_{red}) + B_{red}(1-B_{black}) + \frac{n-3}{n-2} B_{black}(1-B_{red}\right), c\right]$

4: types of traffic. 5: induced rates. 1: answer.

Offered traffic on LM:

λ (direct traffic)

$+\lambda 4 B_{red} \frac{1}{n-3}(1-B_{red})$ (traffic P-L or P-M or Q-L or Q-M that chooses 2-hop over L-M)

$+ 2(n-4)\lambda B_{black} \frac{1}{n-2}(1-B_{black})$ (traffic L-* or Q-* that chooses 2-hop over L-M)

So $B_{black} = E\left[\frac{\lambda}{\mu}\left(1+\frac{4}{n-3}B_{red}(1-B_{red}) + \frac{2(n-4)}{n-2}B_{black}(1-B_{black})\right), \quad C\right]$

| 4: types of traffic. 5: induced rates. 1: answer.

Q4

(a) On receipt of ACK, increase $w(t)$ by $\frac{1}{w(t)}$.
On detection of a drop, cut $w(t)$ by $\frac{w(t)}{2}$.

| 2 marks

| 1 mark

Packets are being sent at rate $x(t) = \frac{w(t)}{RTT}$, so ACK rate $= x(t)$ & drop rate $= p\,x(t)$.

$$\mathbb{E}\, w(t+\delta) = w(t) + \underbrace{\delta x(t)}_{\text{chance of getting an ACK in interval of length } \delta} \cdot \frac{1}{w(t)} - \underbrace{\delta p x(t)}_{\text{chance of detecting a drop in an interval of length } \delta} \cdot \frac{w(t)}{2}$$

| 3 marks

$$\Rightarrow \text{drift} = \mathbb{E}\,\text{rate of change of } w(t) = \frac{\mathbb{E}\, w(t+\delta) - w(t)}{\delta} = x(t)\frac{1}{w(t)} - p\,x(t)\frac{w(t)}{2}$$

$$= \frac{1}{RTT} - \frac{p\,w(t)^2}{2RTT}.$$

The fixed point is when drift $=0$

$$\Rightarrow \frac{1}{RTT} = \frac{p\,w^2}{2RTT} \Rightarrow w = \sqrt{\frac{2}{p}} \Rightarrow x = \frac{\sqrt{2}}{RTT\sqrt{p}}.$$

| 2 marks

(b)

$$\frac{dw_A(t)}{dt} = x_A(t)\frac{1}{w_A(t)+w_B(t)} - p_A x_A(t)\frac{w_A(t)}{2} = \frac{1}{RTT}\frac{w_A}{w_A+w_B} - \frac{p_A w_A^2}{2RTT}$$

$$\frac{dw_B(t)}{dt} = x_B(t)\frac{1}{w_A(t)+w_B(t)} - p_B x_B(t)\frac{w_B(t)}{2} = \frac{1}{RTT}\frac{w_B}{w_A+w_B} - p_B\frac{w_B^2}{2RTT}$$

| 3 marks : right place to tweak
| 3 marks : right tweak

Fixed point : $\frac{w_A}{w_{A+B}} = \frac{p_A w_A^2}{2} \Rightarrow w_A = \frac{2/p_A}{w_A+w_B}$

& $w_B = \frac{2/p_B}{w_A+w_B}$.

From the first eqn, $w_A+w_B = \frac{2/p_A}{w_A} \Rightarrow w_B = \frac{2/p_A}{w_A} - w_A$.

Substituting into 2nd eqn, $\frac{2/p_A}{w_A} - w_A = \frac{2/p_B}{\frac{2/p_A}{w_A}} = \frac{w_A p_A}{p_B}$

$$\Rightarrow \frac{2/p_A}{w_A} = w_A\left(1+\frac{p_A}{p_B}\right) \Rightarrow w_A = \sqrt{\frac{2/p_A}{1+p_A/p_B}} = \sqrt{\frac{2}{p_A + p_A^2/p_B}} = \frac{\sqrt{2}}{\sqrt{\frac{p_A}{p_B}(p_B+p_A)}}$$

likewise, $w_B = \dfrac{\sqrt{2}}{\sqrt{\frac{p_B}{p_A}(p_B+p_A)}}$

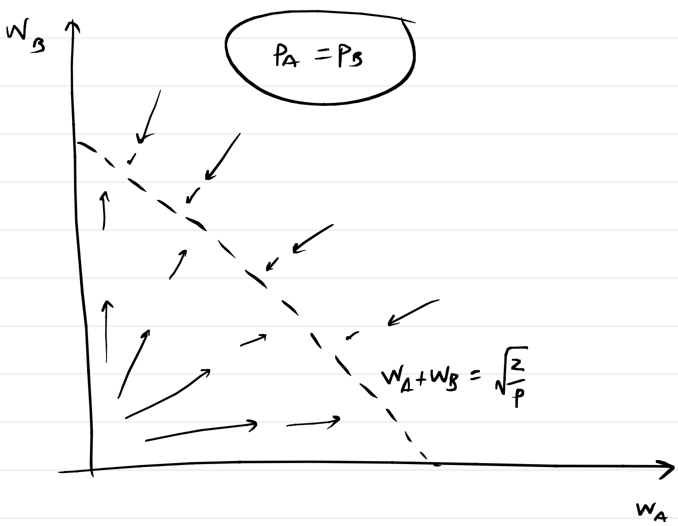
Total throughput is $\dfrac{w_A+w_B}{RTT} = \dfrac{\sqrt{2}}{RTT}\dfrac{1}{\sqrt{(p_A+p_B)\left(\frac{p_A}{p_B}+\frac{p_B}{p_A}\right)}}$

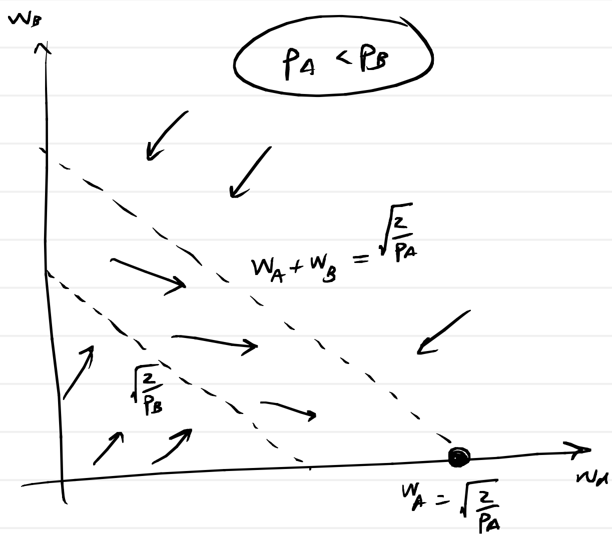
3 marks: right place to tweak
3 marks: right tweak

(c) $$\frac{dw_A(t)}{dt} = x_A(t)\frac{1}{w_A(t)+w_B(t)} - p_A x_A(t)\frac{w_A(t)+w_B(t)}{2} = \frac{x_A(t)}{w(t)}\left(1-p_A\frac{w(t)^2}{2}\right)$$

$$\frac{dw_B(t)}{dt} = x_B(t)\frac{1}{w_A(t)+w_B(t)} - p_B x_B(t)\frac{w_A(t)+w_B(t)}{2} = \frac{x_B(t)}{w(t)}\left(1-p_B\frac{w(t)^2}{2}\right).$$

(d)





Drifts to the dotted line and stays there, anywhere on the line.

Drifts to $w_A = \sqrt{\frac{2}{p_A}}$, $w_B = 0$

3 marks: right axes, idea

4: identify special lines/zones

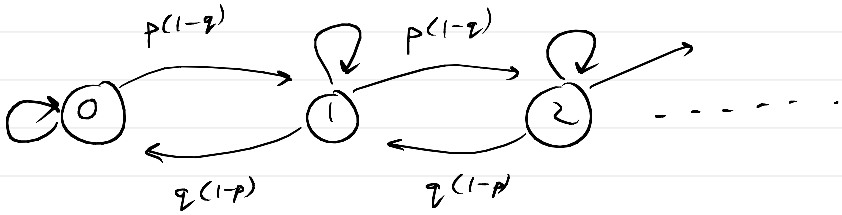
3: arrows in right direction

3: execution

Bonus: interesting comparison to the first proposal

Q5

(a)



4: right ideas about how transitions occur

2: execution

(The unlabelled transitions are such as to make jump-out probs. sum to 1 at each node.)

(b) The detailed balance equations are

4: balance + normalization

$$\pi_n \, p(1-q) = \pi_{n+1} \, q(1-p)$$

$$\Rightarrow \quad \pi_n = \rho^n \pi_0 \quad \text{where } \rho = \frac{p(1-q)}{q(1-p)}$$

To make $\pi_0 + \pi_1 + \cdots = \pi_0 (1 + \rho + \cdots) = 1$, we need $\pi_0 = 1 - \rho$.

Thus

$$\pi_n = (1-\rho)\rho^n, \quad \text{ie queue size} \sim \text{Geom}(1-\rho) - 1.$$

4: correct eqm + mean

$P(\text{queue is empty}) = \pi_0 = 1 - \rho.$

Mean queueing size $= \frac{1}{1-\rho} - 1 = \frac{\rho}{1-\rho}$.

Mean queueing delay $= \frac{N}{\lambda}$ by Little, $= \frac{\rho}{1-\rho} \Big/ p$ timeslots

2: appeal to Little

2: correct delay

(c)

$$\mathbb{P}(\text{node 1 successfully transmits}) = q_1 = r_1 \big(\overbrace{1 - \rho_2 + \rho_2(1-r_2)}^{\mathbb{P}(\text{node 2 doesn't attempt to xmit})} \big)$$

$$\mathbb{P}(\text{node 2 successfully transmits}) = q_2 = r_2 \big(1 - \rho_1 + \rho_1(1-r_1)\big)\big(1 - \rho_3 + \rho_3(1-r_3)\big)$$

$$\mathbb{P}(\text{node 3 successfully transmits}) = q_3 = r_3 \big(1 - \rho_2 + \rho_2(1-r_2)\big)$$

where

$$\rho_i = \frac{p_i(1-q_i)}{q_i(1-p_i)}$$

3 use ρ in the prob. term
3: notice special form for queue 2
3: correct Q2 eqn
3: correct Q1 and Q3

3: identify ρ from first part of question.

These 6 equations may be solved for $q_1, q_2, q_3, \rho_1, \rho_2, \rho_3$, using the fixed pt method