1

(a) Let there be N flows, with common round trip time RTT. Let

$W(t)$ = sum of window sizes at time t

$w(t)$ = $W(t)/N$ = av. window size

$X(t)$ = sum of xmit rates at time t

$x(t)$ = $X(t)/N$ = av. throughput at time t

$p(t)$ = packet drop probability at time t

Since each flow sends a window of packets every RTT,

$x(t) = w(t)/RTT$. ✓

2 marks: x & W

TCP rules for updating its window size cwnd are:

* increase cwnd by 1/cwnd every ACK
* decrease cwnd by cwnd/2 every drop

An alternative expression for the increase rule is:

there are cwnd ACKs every RTT, so cwnd increases by
$\text{cwnd} \times \frac{1}{\text{cwnd}} = 1$ packets every RTT,
i.e. cwnd increases at rate 1/RTT.

2 marks: what the LHS is

2 marks for each of the 3 parts

Now we can write down a drift model:

$$W(t+\delta) = W(t) + \underbrace{\delta \cdot \frac{1}{RTT} \cdot N}_{\text{each flow increases its window at rate } 1/RTT\text{, so in time } \delta \text{ each window increases by } \delta/RTT} - \underbrace{\delta X(t)\, p(t)}_{\text{total xmit rate is } X(t)\text{, so in time } \delta \text{ we expect } \delta X(t) p(t) \text{ drops. If } \delta \text{ small, this is} \approx \#\text{flows which get drops}} \cdot \underbrace{\frac{w(t)}{2}}_{\text{each drop results in a decrease of roughly } w(t)/2 \text{ on average.}}$$

$$\Rightarrow \quad w(t+\delta) = w(t) + \delta\left[\frac{1}{RTT} - x(t)\,p(t)\,\frac{w(t)}{2}\right]$$

$$\Rightarrow \quad x(t+\delta) = x(t) + \delta\left[\frac{1}{RTT^2} - p(t)\,\frac{x(t)^2}{2}\right]$$ ✓

2 marks: answer

Or, as a differential equation,

$$\frac{d}{dt}x(t) = \frac{1}{RTT^2} - p(t)\,\frac{x(t)^2}{2}$$ ✓

bonus: diff.eq

bonus: expression for p.

A plausible eqn for $p(t)$ is: $p(t) = \left[\frac{X(t) - C}{X(t)}\right]^+$ ✓ = fraction of work that exceeds link capacity.

The three terms of the drift model were poorly explained, especially the term for #flows which get drops.

Several students didn't give their answer in terms of x, as the question asks.

12 marks

(b) The only difference is in the increase and decrease rules. MulTCP will

* increase cwnd by k/cwnd packets every ACK,
ie by $\text{cwnd} \times \frac{k}{\text{cwnd}} = k$ packets every RTT,
ie at rate $\frac{k}{RTT}$.

* decrease cwnd by $\text{cwnd}/2k$.

9 marks

The revised drift model is

$$W(t+\delta) = W(t) + \delta \cdot \frac{k}{RTT} \cdot N - \delta X(t)\, p(t) \cdot \frac{W(t)}{2k}$$

$$\Rightarrow x(t+\delta) = x(t) + \delta\left[\frac{k}{RTT^2} - p(t)\frac{x(t)^2}{2k}\right]$$

$$\Rightarrow \frac{dx}{dt} = \frac{k}{RTT^2} - p(t)\frac{x(t)^2}{2k}.$$

3 marks each: know which bits of the equation to change,

3 marks: right answer

(c) If the system is stable, then there is 0 drift. Let x, p be the steady-state throughput & pkt drop probability. Then

6 marks

$$\text{drift} = \frac{k}{RTT^2} - \frac{p x^2}{2k} = 0$$

$$\Rightarrow x^2 = \frac{2k^2}{RTT^2 p} \Rightarrow x = \frac{k\sqrt{2}}{RTT\sqrt{p}}.$$

3 marks: formulate the question as solving for 0 drift

3 marks: solve.

Observe that this is simply the total throughput obtained by k simultaneous TCP flows.

(d) If everyone were to replace each TCP flow by k, then there would be more traffic demand, which may cause instability problems.
To test this, a simple spreadsheet like

6 marks

t	δ	x	p	$\frac{dx}{dt}$
t_1		x	$p = \left[\frac{x-c}{x}\right]^+$	$= \frac{k}{RTT^2} - \frac{px^2}{2k}$
t_2	$\delta = t_2 - t_1$	$= x + \delta\frac{dx}{dt}$		

3 marks: appropriate use of the drift equation

3 marks: p depends on x

Plot x as a function of t. See if it stabilizes, or if it looks unstable.

bonus: drift diagram.

Hardly anyone gave a good answer.

* The question says: describe a simulator. This means, in effect, "write pseudocode". Very many students just said "simulate it" without saying how.

* Many students got confused with the flow-level simulator from Coursework 2. The question refers to Excel — this is a hint that it's about the Excel simulation of the drift model that we saw in lectures

* The notion of instability here is oscillations, NOT some quantity $\to \infty$.

2

(a) The TCP throughput equation is

$$x = \frac{\sqrt{2}}{RTT\sqrt{p}}$$ where x = throughput of a flow, RTT = round trip time of the flow, p = packet drop probability.

8 marks

2 marks: throughput eqn.

Let w = window size.
Since w packets are sent every RTT, $x = \frac{w}{RTT}$.

$$\Rightarrow \quad w = \frac{\sqrt{2}}{\sqrt{p}}.$$

If packet drop prob. is $p = 1\% = 0{\cdot}01$, then $w = \frac{\sqrt{2}}{\sqrt{0{\cdot}01}} = \frac{\sqrt{2}}{0{\cdot}1} \approx \frac{1}{0{\cdot}1} = 10.$

2 marks: calc 10 pkts

First scenario:

$$x = \frac{w}{RTT} = \frac{10\text{ pkts}}{50\text{ ms}} = \frac{120\text{ kbit}}{\frac{1}{20}\text{ s}} = 2{\cdot}4\text{ Mbit/s.}$$

2 marks for calculations for each of the scenarios

So, moderate load means each flow gets 2·4 Mbit/s.
If the link has capacity 1 Mbit/s, it will be moderately loaded with just half a flow.

Second scenario:

$$x = \frac{w}{RTT} = \frac{10\text{ pkts}}{100\text{ ms}} = \frac{120\text{ kbit}}{\frac{1}{10}\text{ s}} = 1{\cdot}2\text{ Mbit/s.}$$

If a 1 Gbit/s link is "moderately loaded" there must be $\frac{1\text{ Gbit/s}}{1{\cdot}2\text{ Mbit/s}} \approx 1000$ flows active.

For some reason, hardly anyone explained the scenarios. All you need to do is show the working — use the throughput eqn you wrote down, and solve it! Do NOT just parrot Lachlan's qualitative assertions.

(b) Consider a single link with ideal processor sharing, with capacity C Mb/s. By proc. sharing we mean that, if there are n flows active, each flow gets $\frac{C}{n}$ Mb/s throughput.

6 marks

2 marks: what is processor sharing

Suppose that flows arise as a Poisson process of rate λ flows/sec, and that each flow has to transfer an average of μ Mbit. Once all this data is transmitted, the flow terminates.

[M means: Poisson arrivals.
G means: arbitrary distribution of flow sizes.]

2 marks: Kendall notation

Let $\rho = \frac{\lambda\mu}{C}$. Then, the mean number of active flows is $\rho/1-\rho$, assuming $\rho < 1$.

2 marks: formula, and definition of ρ.

Too many students got confused with Erlang link and with FIFO queue! I was very disappointed. After all, you did Coursework 2 on exactly this model!

1c) Little's Law says:

12 marks

$$N = \lambda W$$

av. time spent in the system.

av. occupancy of a system

arrival rate to the system

2 marks: Little's Law

Suppose that the link was FIFO, ie earliest job served first. This would not affect whether the link is busy, it would simply alter the order of service.
Now, consider the slot at the head of the FIFO queue:

2 marks: consider slot at head of queue

$$\text{av. occupancy} = \lambda \times W = \rho$$

av. duration of service in the FIFO queue $= \frac{\mu}{c} = \frac{\text{job size}}{\text{service rate}}$

2 marks: answer ρ

arrival rate of flows to the link = arrival rate of flows to the spot at the head of the queue, since all jobs must go past the head of the queue

3 marks: identify the parameters λ and $\frac{\mu}{c}$

Now, whenever the link is busy, the slot at the head of the queue has occupancy 1.
Therefore, av. occupancy = fraction of time that the link is busy. $= \rho$.

2 marks: relate av. occupancy to $\mathbb{P}(\text{busy})$.

Alternatively, fraction of time busy $= \frac{m}{1+m}$ where $m = \frac{\rho}{1-\rho}$ = av. #flows active.

We did this in lectures, as one of two examples of Little's Law. But hardly anyone remembered.

Another way to calculate this is from the distribution of #jobs active.
We know that $\mathbb{P}(\#\text{active flows} = r) = (1-\rho)\rho^r$, from the Markov model of an M/M/1 processor sharing link.

Thus $\mathbb{P}(\#\text{active flows} = 0) = \mathbb{P}(\text{link idle}) = 1-\rho$

$\Rightarrow \quad \mathbb{P}(\text{link is busy}) = \rho$.

By the theorem about invariant distributions of Markov processes,
$\rho = \mathbb{P}(\text{link is busy}) = \text{fraction of time that link is busy}$.

1d) If $\rho \geq 1$, then the equation for av. # of active flows doesn't work any more. This reflects the fact that the system is unstable, ie work arrives faster than it can be served, which means that # active jobs just keeps on increasing.

7 marks

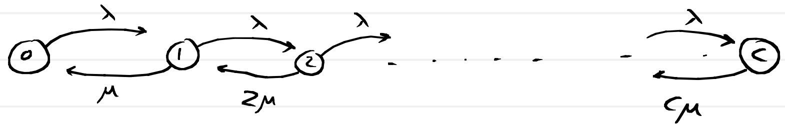
If you run a simulation, you'll see

Any sensible discussion will do. It should describe
* the basic problem
* how instability manifests itself.

Coursework 2 was precisely about this sort of instability — I was disappointed that so few students made the connection.

3

(a) State = # active calls.



5 marks

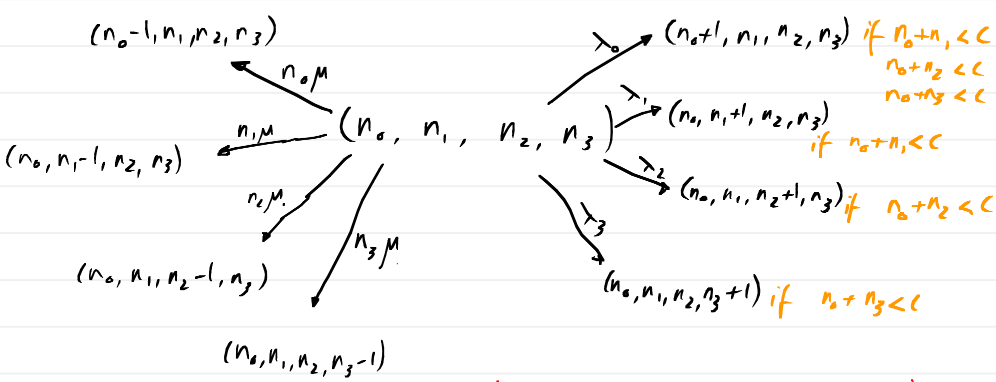
✓ state space: 2 marks

✓ rates: 3 marks

(b) State = (# calls on route 0, # on route 1, # on route 2, # on route 3).

✓ state: 3 marks

10 marks



✓ rates: 3 marks

✓ transitions: 2 marks

✓ capacity constraints on up transitions: 2 marks

Only one student understood that there are capacity constraints on the up transitions (as for the Erlang link, or for the example with voice & video that we went through in lectures)

Many students got muddled up with processor sharing. The key words here are "Erlang" and "circuit" which tell you it's not processor-sharing.

(c) In the fixed point method, we apply Erlang's formula for the blocking probability on a link, to each of the three links, separately.

10 marks

$$B_1 = E\left(\frac{\lambda_1 + \lambda_0(1-B_2)(1-B_3)}{\mu}, C\right)$$

✓ equations: 4 marks

$$B_2 = E\left(\frac{\lambda_2 + \lambda_0(1-B_1)(1-B_3)}{\mu}, C\right)$$

$$B_3 = E\left(\frac{\lambda_3 + \lambda_0(1-B_1)(1-B_2)}{\mu}, C\right)$$

The offered load on link 3 comes from dedicated arrivals for link 3, at rate λ_3, plus calls on route 0 which are not blocked on either link 1 or link 2. The arrival rate of calls on route 0 is λ_0; the arrival rate of calls not blocked on link 1 or 2 is $\lambda_0(1-B_1)(1-B_2)$.

✓ 3 marks: mixing the right traffic flows

✓ 3 marks: correct "thinking"

Generally well answered, except that there were many mistakes with slightly incorrect thinking.

(d) Start with some initial estimates, say

$B_1 = \frac{1}{2}$
$B_2 = \frac{1}{2}$
$B_3 = \frac{1}{2}$

$\boxed{8 \text{ marks}}$

✓ initial guess : 2 marks

Then, repeatedly apply the equations above:

let $B_1 \leftarrow E\left(\frac{\lambda_1 + \lambda_0(1-B_2)(1-B_3)}{\mu}, c\right)$ ✓ update rule : 3 marks

then using this new value of B_1 do

$$B_2 \leftarrow E\left(\frac{\lambda_2 + \lambda_0(1-B_1)(1-B_3)}{\mu}, c\right)$$

and run through the equations over and over again until the values of the B_i settle down. This might be tested visually, or by testing to see if a round of updates has changed each of the B_i by less than some small threshold.

✓ how to terminate : 3 marks.

Some students just wrote "update your estimates" without explaining how. Most students didn't put together a clean concise answer — there was much too much waffle

Some students were confused about fixed point & drift. They are two completely separate ideas.

4

(a) Let F = total # computers which store a given fragment.
Each computer makes an independent decision of whether or not to store the fragment; it stores with probability r/N.

✓ params: 2 marks ✓ dist: 2 marks

Thus $F \sim \text{Bin}(N, \frac{r}{N})$ ← the Binomial distribution

7 marks

$$\mathbb{P}(F = f) = \binom{N}{f} \left(\frac{r}{N}\right)^f \left(1-\frac{r}{N}\right)^{N-f}.$$

The mean # copies is

$$\mathbb{E}F = N \times \frac{r}{N} = r.$$

✓ work out what to calculate & get answer: 3 marks

(b) $\mathbb{P}$(a given computer stores the fragment, & receives the query)

16 marks

$= \mathbb{P}$(stores the fragment) $\times$ $\mathbb{P}$(receives the query)

$= \frac{r}{N} \times \frac{c}{r} = \frac{c}{N}$. ✓ prob: 3 marks

So, # computers which store fragment & receive query i.e. # matches is

$M \sim \text{Bin}(N, \frac{c}{N})$ ✓ distribution: 4 marks

$\approx$ Poisson(c), for N large, by a standard approximation. ✓ limit: 3 marks

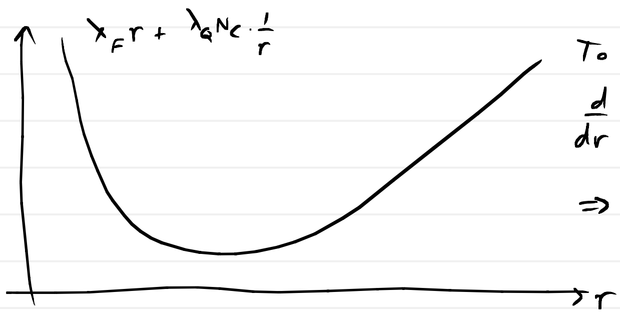
So $\mathbb{P}$(match succeeds) $= 1 - \mathbb{P}(M=0) = 1 - e^{-c}$. ✓ answer: 3 marks.

(c) Fragments arrive at rate λ_F, each stored at r computers on average.
Queries arrive at rate λ_Q, each stored at $\frac{N_c}{r}$ computers on average.

10 marks

✓ 2 marks: what to calculate ✓ ✓ 2 marks each: right terms

So total # objects stored per second $= \lambda_F r + \lambda_Q \frac{N_c}{r}$.



To minimize this, we solve ✓ 2 marks: what to solve

$$\frac{d}{dr} = \lambda_F - \frac{\lambda_Q N_c}{r^2} = 0$$

$$\Rightarrow r = \sqrt{\frac{\lambda_Q N_c}{\lambda_F}}.$$ ✓ 2 marks: answer

Few people chose to answer this question. Those who did seemed to find it very straightforward.

5

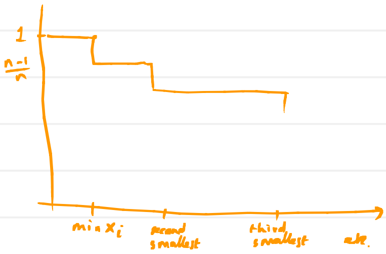
(a) The ECDF plots the empirical cumulative distribution function

$$F(x) = \mathbb{P}(X \geq x) = \text{fraction of observations that are} \geq x$$

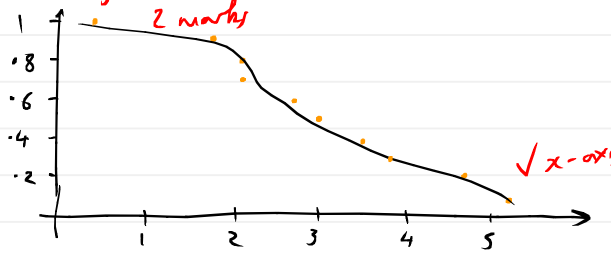
as a function of x.

8 marks

bonus: explain what the ECDF is trying to estimate



✓ y-axis: 2 marks



✓ x-axis: 2 marks

A 95% confidence interval for the sample mean is $\left[\hat{\mu} - 1.96\frac{\hat{\sigma}}{\sqrt{n-1}}, \hat{\mu} + 1.96\frac{\hat{\sigma}}{\sqrt{n-1}}\right]$. ✓ 2 marks: expression

where

μ = sample mean = $\frac{1}{n}\sum x_i$

n = # observations

σ^2 = sample std. deviation = $\frac{1}{n}\sum (x_i - \mu)^2$.

✓ 2 marks: terms

(b) The likelihood of parameters λ, k given the observations $x_1, \dots, x_n$ is

13 marks

✓ product of densities: 3 marks

$$\text{Lik}(\lambda, k \mid x_1, \dots, x_n) = \prod_{i=1}^{n} f(x_i \mid \lambda, k) = k^n \lambda^{-nk} \left(\prod x_i\right)^{k-1} e^{-\sum x_i^k / \lambda^k}.$$

The maximum likelihood estimator for λ, $\hat{\lambda}$, is the value of λ that maximizes this function. To calculate this, it's easier to maximize

$$\log \text{Lik} = n \log k - nk \log \lambda + (k-1)\sum \log x_i - \lambda^{-k} \sum x_i^k.$$

✓ work out loglik in a usable form

$$\frac{d}{d\lambda} \log \text{Lik} = 0 \Rightarrow \frac{-nk}{\lambda} - \left(\sum x_i^k\right)(-k)\lambda^{-k-1} = 0$$

✓ what to solve: 3 marks

$$\Rightarrow \frac{k}{\lambda^{k+1}} \sum x_i^k = \frac{nk}{\lambda} \Rightarrow \frac{\lambda}{\lambda^{k+1}} = \frac{n}{\sum x_i^k}$$

$$\Rightarrow \hat{\lambda} = \left(\frac{1}{n}\sum x_i^k\right)^{1/k}$$

✓ answer: 2 marks

(c) The standard way to generate a random variable X from distribution function $F(x) = P(X \geq x)$, is:

12 marks

- Generate $U \sim \text{Uniform}(0,1)$ ✓ 2 marks: start with uniform dist.
- let $X = F^{-1}(U)$. ✓ 3 marks: apply inverse of F

Here, $F(x) = e^{-(x/\lambda)^k}$. ✓ 2 marks: choose what to take inverse of

We want to solve

$$U = F(X) \Rightarrow U = e^{-(X/\lambda)^k} \Rightarrow -\log U = \left(\frac{X}{\lambda}\right)^k$$

$$\Rightarrow X = \lambda(-\log U)^{1/k}.$$ ✓ 3 marks: correct manipulation

So, generate U using a standard function like rand();
and let $X = \lambda(-\log U)^{1/k}$, for given parameters λ and k. ✓ 2 marks: answers.

The few students who attempted this question all gave good answers.