Network Performance, GZ02 and 4033, 2007/2008
Answer THREE of FIVE questions.
Marks for each part of each question are indicated in square brackets
Calculators are permitted

1. (a) Consider a set of TCP flows, all with the same round trip time, sharing a single bottleneck link. Let $x(t)$ be the average throughput of the flows at time t. Derive a drift model for $x(t)$, and explain your working.

[12 marks]

Jon Crowcroft, a former professor at UCL, proposed a modified form of TCP which he called MulTCP. This is much like TCP, but with two differences: TCP increases its window by $1/w$ packets per ACK whereas MulTCP increases by k/w packets; TCP decreases its window by $w/2$ per drop whereas MulTCP decreases it by $w/(2k)$. Here w is the current window size, and k is a fixed parameter which can be set by the user.

(b) Write down a drift model for MulTCP.

[9 marks]

(c) Prove that if MulTCP is stable then the average transmission rate is

$$\frac{k\sqrt{2}}{RTT\sqrt{p}}$$

where you should define the quantities RTT and p.

[6 marks]

(d) It is claimed that if everyone were to use MulTCP with a large value of k then the Internet would become unstable. Describe a simple simulator, which you might implement for example in Excel, that could test this claim.

[6 marks]

[Total 33 marks]

2. The following email was posted by Lachlan Andrews to the Transport Modeling Research Group, a group which was set up by the Internet Research Task Force to propose how replacements for TCP should be evaluated.

> In the description of delay/throughput tradeoff, it talks about "moderate congestion" as 1-2% packet loss with TCP NewReno. Unless I'm mistaken, that says "windows should be about 1/sqrt(0.01)=10 packets" (to within a small factor). I'd prefer not to quantify the load that way. Consider some scenarios:
>
> 1Mbit/s: 10 packets of 12000 bits > 100ms. That means that for 1Mb/s tests with inter-city RTTs (50ms), a moderate level of load would be less than *half* of one flow.
>
> 1Gbit/s bottleneck, 100ms path. "Moderate" congestion would be when 1000 flows each gets about 1Mb/s. To me, that is very heavy load. Indeed, however large the bottleneck bandwidth is, "moderate" congestion would be when 100ms paths give 1Mbit/s per user.
>
> I'd much prefer to specify the load in terms of the offered load as a fraction of bandwidth. I propose an alternative: The "load" is the average number of flows if the traffic was served by an M/G/1 queue with an ideal processor-sharing service discipline. My reasons are:
>
> 1. This scales properly as capacity increases, and is correctly independent of RTT
> 2. A processor-sharing M/G/1 queue is a model of roughly what we're aiming for with a single bottleneck (equal instantaneous rates).
> 3. For loads like 10%, this simply corresponds to 10% of the bandwidth.
> 4. It reflects that, even at extreme overload, we want to consider a system whose average number of flows doesn't increase with time. Otherwise, the results would be _very_ sensitive to duration, and we agreed that we should try to design tests which are _not_ sensitive to the parameters.
>
> Thoughts?

(a) State the TCP throughput equation. Use it to explain the calculation that window size is roughly 10 packets. Explain the reasoning behind Lachlan's two scenarios.

[8 marks]

(b) Explain what is meant by an M/G/1 queue with an ideal processor-sharing service discipline. State a formula for the average number of flows, and explain the variables in your formula.

[6 marks]

(c) An M/G/1 processor-sharing queue alternates between idle periods (when no flows are active) and busy periods (when one or more flows are active, and the entire link capacity is shared between them). Using Little's law, or otherwise, calculate the fraction of time that the link is busy.

[12 marks]

(d) What does Lachlan mean by "a system whose average number of flows doesn't increase with time"? What parameter values would cause the average number of flows to increase with time?

[7 marks]

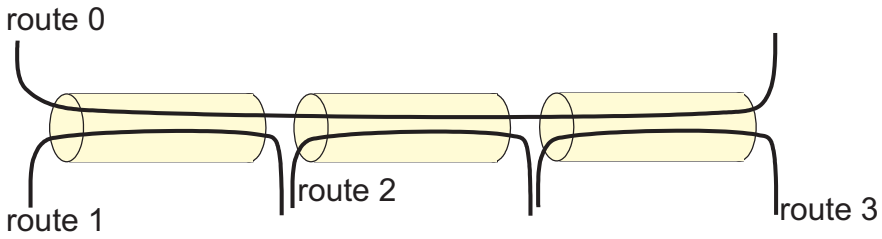
[Total 33 marks]

3. Erlang modeled a telephone exchange with C circuits, to which calls which arrive as a Poisson process of rate λ, where call holding times are exponential random variables with rate μ. Write $E(\lambda/\mu, C)$ for the blocking probability in this system.

   (a) Sketch the state space diagram for a Markov process model of the Erlang link. Include the transition rates on your diagram.

   [5 marks]

Now consider the network below which has three links and four routes. Calls on route 0 occupy all three links; calls on route i occupy only link i, for $i = 1, 2, 3$. Calls on route i arrive as a Poisson process of rate ν_i for each i, and all call holding times are exponential random variables with rate μ. The three links have capacity C.



   (b) Sketch the state space diagram for a Markov process model of this four-route system. Include the transition rates on your diagram.

   [10 marks]

Let B_i be the probability that link i has all its circuits busy. We would like to be able to calculate B_i. Since the state space diagram has very many states, it is impractical to find B_i by calculating the equilibrium distribution of the Markov process. Instead, we can use the fixed point method.

   (c) Write down a set of fixed point equations for calculating the B_i, and explain your reasoning.

   [10 marks]

   (d) Explain how to use the fixed point method to calculate the B_i.

   [8 marks]

[Total 33 marks]

4. *This question tests your understanding of the binomial and Poisson distributions.*

Google has a large database of web pages, and a steady flow of queries which must be matched against the database. The database is far too large to fit onto a single computer, and the rate of queries is far too high. Google therefore distributes the database over many computers: each computer holds several small fragments of the database, each fragment is replicated across several computers, and each query is copied to several computers. The challenge is to ensure that each query finds at least one copy of each relevant fragment. This problem is known as distributed rendezvous.

In this question, we analyse a simple random algorithm for distributed rendezvous. Let N be the number of computers, and let r and c be constants. When there is a fragment that needs to be stored, each computer makes an independent decision whether or not to store it; it chooses to store the fragment with probability r/N. When there is a query, the probability it is copied to a given computer is c/r, and the decisions for different computers are independent.

(a) The total number of copies of a given fragment is a random variable. What is its distribution? What is the mean number of copies?

[7 marks]

(b) Suppose that a certain query should match a certain fragment. What is the probability that a given computer both stores the fragment and receives the query? Show that the probability that there is at least one computer which both stores the fragment and receives the query, i.e. that the search succeeds, is approximately $1 - e^{-c}$.

[16 marks]

(c) Suppose that fragments arrive at a rate of λ_F per second, and that queries arrive at a rate of λ_Q per second. What is the average number of fragment copies, plus the average number of query copies, every second? Sketch a graph of this, as a function of r. What is the optimal choice of r?

[10 marks]

5. Here is a sample of measurements of mobile phone lifetimes, in years.

$$0.5, 1.9, 2.1, 2.1, 2.8, 3.0, 3.6, 3.9, 4.8, 5.3$$

(a) Explain how to plot the ECDF of this sample, and give a rough sketch. Explain how to calculate a 95% confidence interval for the mean lifetime.

[8 marks]

A standard model for component lifetimes is the Weibull distribution, which has density

$$f(x) = k\lambda^{-k}x^{k-1}e^{-(x/\lambda)^k}$$

and distribution function

$$\text{Prob}(X \geq x) = e^{-(x/\lambda)^k}.$$

If $k < 1$ then components are most likely to fail early on (e.g. because of bad manufacturing). If $k > 1$ then components are increasingly likely to fail as they age (e.g. because of wear and tear).

(b) We happen to know that $k = 2$. Find a formula for the maximum likelihood estimator for λ.

[13 marks]

(c) Describe a random number generator, to generate random samples from a Weibull distribution.

[12 marks]

[Total 33 marks]

END OF PAPER