Example sheet 5

Job models etc.

Network Performance—DJW—2010/11

Question 1. Consider a FIFO queue, with arrival rate λ, service rate μ, and finite buffer size b. Suppose that service times are exponentially distributed random variables, and that arrivals are a Poisson process. Let $\rho = \lambda/\mu$. Show that the equilibrium distribution of queue size is $\mathbb{P}(Q = q) = \pi_q$ where

$$\pi_q = \frac{(1-\rho)\rho^q}{1-\rho^{1+b}}.$$

Explain why the packet drop probability is π_b. *[Hint.* $1 + r + r^2 + \cdots + r^n = (1 - r^{n+1})/(1 - r)$ *for* $r \neq 1$.*]*

Question 2. Patient Zero has been infected and turned into a zombie. Zombies can attack humans and turn them into zombies, and they can also be killed by decapitation. Each zombie attacks humans as a Poisson process of rate λ, and each zombie unlifetime is exponentially distributed with rate μ. Draw a state space diagram, where the state is the number of zombies. What is the probability that the zombie infection dies out? What is the mean time until it dies out?

This is an example of an epidemic model. Epidemics can be used to describe the spread of gossip, viruses, etc.

Question 3. In a cable modem system, a number of households share a single uplink channel. A typical capacity for this channel is 4.71 Mb/s. At a busy time of day, maybe 10% of these households are online and using their Internet connections, and while they are online and active they initiate new TCP flows at an average rate of one flow every 2 seconds. The mean flow size is 21 kB (a figure taken from the `wischik.com` webserver logs). The contention system for sharing capacity between flows begins to break down when there are more than 10 or so simultaneous flows. Use the standard processor-sharing model to estimate how many households can be attached to a single upstream channel.

Cisco recommends 200 subscribers per upstream channel. The precise number depends on characteristics such as signal to noise ratio, and the details of the contention system. See `http://www.cisco.com/application/pdf/paws/12205/max_number_cmts.pdf`.

Question 4. The Copenhagen Telephone Company wishes to offer video calls as well as normal voice calls. A normal voice call takes up one circuit, whereas a video call takes up four. The company is concerned about a single bottleneck link, consisting of just 12 circuits.

Suppose that voice calls arrive at a rate of 2 calls per minute, and that video calls arrive at a rate of 0.5 calls per minute, and that the mean duration of both types of call is 2 minutes. Suppose also that call arrivals are Poisson processes and that call durations have an Exponential distribution.

(i) Let (T_t, V_t) be the number of voice calls and the number of video calls respectively, at time t. What is the state space, i.e. what are the possible values that the pair (T_t, V_t) can take? Draw a state space diagram.

(ii) Find the transition rates for this Markov process. Set up a rate matrix, and compute the equilibrium distribution. *[Hint. Use a computer.]*

(iii) In which states would a newly arriving voice call be blocked? What is the blocking probability for voice calls? What is the blocking probability for video calls?

One technique for improving the quality of service for video calls, at the expense of voice calls, is to reserve a certain number of circuits which may only be used by video calls. This is called trunk reservation.

Question 5. Consider a network consisting of links $\{1,2,3,4\}$. A route is a subset of links; there are three routes with traffic, $r_1 = \{1,2\}$, $r_2 = \{2,3\}$ and $r_3 = \{1,2,3,4\}$. TCP flows arrive to each of the three routes; they arrive on route i as a Poisson process with rate λ_i, and all flow sizes are exponentially distributed with mean m. TCP is a mechanism for sharing capacity between flows: let $\theta_i(n_1,n_2,n_3)$ be the throughput obtained by a flow on route i when there are n_1 flows active on route r_1 etc. Write down the state space for this system. Find the transition rates.

The function $\theta_i(\cdot)$ was discovered by Misra, Gong and Towsley in 2000. See `http://gaia.cs.umass.edu/fluid/`.

Question 6. My brother wrote a simple program for ripping radio programs from the BBC website. It downloads an audio file, then encodes it as an MP3, then downloads another audio file, and so on. The duration of a radio program is an exponential random variable with mean μ. Downloading happens in real-time, i.e. download time is exactly equal to the duration of the program. Encoding happens with an s-fold speedup, i.e. encoding time is an exponential random variable with mean μ/s. (Assume that downloading and encoding times are independent.)

This program worked but was inefficient, since the CPU is underutilized during downloads. To improve efficiency, he has programmed a multithreaded version, which runs m copies of his original program concurrently, using threads. Downloading takes the same time as before, for each thread. When there are E files being encoded, then encoding speed is E times slower for each (i.e. there is processor sharing between the threads that are encoding). He hopes that, most of the time, there will be at least one thread encoding, so that his CPU is not underutilized. He has asked me for advice on how to choose m. For the multithreaded version of his program,

(i) Let D_t be the number of threads which at time t are in the process of downloading. Draw a diagram of the state space, showing the possible values that D_t can take, and draw arrows for the possible transitions.

(ii) What are the transition rates in this system? Explain your answer in detail.

(iii) What is the equilibrium distribution of D_t?

(iv) In which state is his CPU underutilized? For what fraction of time is his CPU underutilized?

(v) How do you recommend he should choose m? *[Hint. To answer this, you should use a computer to evaluate your answer to part (iv) numerically, and from these computations derive a general rule of thumb.]*

Question 7. Consider a processor-sharing link, in which flow sizes are exponentially distributed.

(i) Assume first that flow arrivals are a Poisson process. State a formula for the average number of active flows. Explain all the terms in your formula. Explain briefly how this formula is derived.

(ii) Now assume instead that flow arrivals have the following bursty pattern: two flows arrive back to back, then there is a random exponentially distributed delay, then two new flows arrive, then there is another random exponentially distributed delay, and so on. Find a formula for the average number of active flows, in terms of the average flow arrival rate.

(iii) The PASTA property says that if arrivals are a Poisson process, then arriving flows see time-averages. In the case of bursty arrivals, do arriving flows see time-averages? *[Hint. Bursts arrive as a Poisson process, so bursts see time averages. Half of all arriving flows are the first in a burst, and half are the second in a burst.]*

Question 8. According to the standard processor sharing model of TCP, the utilization of a bottleneck link should be ρ and the mean number of active flows should be $\rho/(1-\rho)$, where $\rho = \lambda m/C$ is the traffic intensity, λ is the arrival rate, m is the mean file size, and C

is the link capacity. In practice, we observe that core links have tens of thousands of active flows, but the utilization may be as little as 20%. Obviously the processor-sharing model is inaccurate. The most likely explanation is that flows are rate-limited, i.e. when there are n flows then each flow gets throughput $\min(A, C/n)$ where A is the capacity of an access link.

(i) Draw a state space diagram, and find the transition rates.

(ii) Given some value of π_0, define π_n by

$$\pi_n = \begin{cases} \pi_0 \left(\frac{\lambda}{A/m}\right)^n \frac{1}{n!} & \text{if } n \leq \lfloor \alpha \rfloor \\ \pi_0 \left(\frac{\lambda}{A/m}\right)^{\lfloor \alpha \rfloor} \frac{1}{\lfloor \alpha \rfloor !} \left(\frac{\lambda}{C/m}\right)^{n - \lfloor \alpha \rfloor} & \text{if } n > \lfloor \alpha \rfloor \end{cases}$$

where $\alpha = C/A$ is the multiplexing ratio, and $\lfloor \alpha \rfloor$ is the floor of α, i.e. $\lfloor \alpha \rfloor \leq \alpha < \lfloor \alpha \rfloor + 1$. Show that π solves the balance equations.

(iii) For what parameter values is this system stable?

Question 9. *Let $E(\rho, C)$ be the blocking probability for an Erlang link with traffic load ρ and C circuits, and let $F(\rho, C)$ be the mean number of active circuits on such a link. Later in the course, we will prove that $F(\rho, C) = \rho(1 - E(\rho, C))$.*

Consider the capped processor-sharing model from Question 8, and assume it is stable. Show that the mean number of active flows is

$$\frac{e}{e+g} F(\rho\alpha, \lfloor \alpha \rfloor) + \frac{g}{e+g} \left(\lfloor \alpha \rfloor + \frac{1}{1-\rho} \right).$$

When there are n active flows, the utilization is $\min(nA/C, 1)$, hence the mean utilization $\sum_{a=0}^{\infty} \min(na/C, 1) \pi_a$. Show that this is equal to

$$\frac{e}{e+g} \frac{F(\rho\alpha, \lfloor \alpha \rfloor)}{\alpha} + \frac{g}{e+g}$$

where $e = 1/E(\rho\alpha, \lfloor \alpha \rfloor)$ and $g = \rho/(1-\rho)$.

For a core bottleneck link, one might observe a utilization level of 20%, and 5,000 active flows. Estimate ρ and α. *[Hint. If C is large and $\rho < 1$ then $E(\rho C, C) \approx 0$.]*

Question 10. Here is a model for a web server with active server pages, i.e. pages that cannot be served directly from the disk but instead require processing e.g. in PHP.

Suppose requests arrive at rate λ. Upon arrival they are placed in a 'task ready' queue, where they wait for the next available worker thread. The server has m worker threads. The CPU can execute c instructions per second, and when there are M threads active then each executes c/M instructions per second. When a thread becomes free, it starts work on the next task in the 'task ready' queue. When a thread is working on a task, it executes an average of i instructions, and then either it completes or it blocks, e.g. to wait for I/O. On average, each request will block b times before completing. If the task blocks, the thread is freed and the task is placed in a 'task blocked' pool. Each blocked task waits for an average of t seconds to unblock, and then it is placed in the 'task ready' queue.

What is the maximum rate at which this web server can serve requests? What is the average request completion time?