Example sheet 3

Job models & simulation

Network Performance—DJW—2011/2012

Question 1. 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 2. 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 then 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) Suppose that at some point in time, there are D threads downloading and $m-D$ encoding. Let $T_1,\dots,T_{m-D}$ be the time until each of the currently-encoding threads finishes its current job and switches to downloading, and let $T = min(T_1,\dots,T_{m-D})$ be the soonest of these times. Show that $T \sim \text{Exp}(\mu/s)$.

(ii) Suppose the program beeps whenever a file is encoded and written to disk, i.e. whenever a new download starts. Explain why the beeps form a Poisson process. What is the rate of this Poisson process?

You have argued that new downloads start as a Poisson process. Each download takes a random time to complete, independent of the other active downloads. Therefore this system can be treated as an Erlang link; the m threads are equivalent to circuits.

(iii) When all m threads are downloading, the CPU is underutilized. For what fraction of time is this the case?

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

Question 3. A student has analysed web server logs, and concluded that the request size distribution can be modeled as $\text{Pareto}(0.12)$ Bytes, and the interarrival time distribution can be modeled as **Exp**(1.2) requests/sec. Suppose that these requests are carried over a link with service rate $C = 1$ Mb/sec. Explain why this system is unstable.

When the student simulated this system, it did not appear to be unstable. Explain why.