Coursework 2

Network Performance—DJW—2006

[Answers are due at noon on 8 January 2007. You may discuss the model and the design of the simulator with your fellow students, but you should answer the questions, program the simulator, and choose what to simulate on your own. This coursework is graded, and is worth 7.5% of your final grade for the course. The majority of the marks are for well-reasoned answers; only a small number of marks are for programming.]

Professor Theodore believes that TCP is designed to give flows a fair share of capacity. He considered a processor-sharing model for flows on a single link with capacity C Mb/second, in which new flows arrive as a Poisson process of rate λ flows/second and flow sizes have an exponential distribution with mean size f Mb. He calculated that the mean number of active flows should be $\rho/(1-\rho)$, where $\rho = \lambda f/C$.

Engineers Atticus, Brittel, and Ciscelia, all working for large telecommunications companies, have measured the mean number of active flows in core links in the Internet, and found it to be much higher than what Professor Theodore predicts. Obviously, his model is wrong. The three engineers disagree about what the major flaw in the model is:

- Atticus believes the major flaw is that it does not take account of the fact that users give up when the network is too congested.
- Brittel believes the major flaw is that flow sizes in the Internet are not exponentially distributed. (He went to a conference, and heard that flow sizes actually have a Pareto distribution.)
- Ciscelia believes the major flaw is that most users have slow access links. Suppose that each user has an access link of capacity A: then, when there are n flows, she says that each flow gets throughput $\min(C/n, A)$.

1. Explain briefly why engineer Atticus is mistaken.

2. Engineer Ciscelia's model of the network can be described by a Markov process. Draw the state space diagram, and explain how you found the transition rates.

3. Program a simulator of your model from Question 2, and estimate the mean number of active flows for a suitable range of parameters λ, f, C and A.

[It is up to you what programming language to use. It is also up to you to choose a sensible range of parameters; explain the reasoning behind your choice. Remember to calculate confidence intervals for your measurements.]

4. Let π be the equilibrium distribution of the number of active flows, from your model in Question 2. The mean number of active flows is given by

$$m = \sum_{i=0}^{\infty} i\pi_i.$$

EITHER calculate m algebraically OR estimate m numerically.

[You can estimate m numerically as follows. Construct a new Markov process like your model from Question 2 but with a maximum of N active flows. This is now a Markov process with a finite state space, so you can find its equilibrium distribution in the normal way. Let m_N be the mean number of active flows in this reduced Markov process. Calculate m_N for a range of values of N, until you find that increasing N any further does significantly alter m_N. This final value is a good approximation to m.]

5. A typical core Internet link may have on average around 5000 active flows. What does this tell you about A?

[You may use either your simulator from Question 3 or your technique from Question 4 to answer this question.]

6. Modify your simulator from Question 3, to make it so that flow sizes have a Pareto distribution with mean f and shape parameter α. Does the mean number of active flows depend on α?

[You need not run exhaustive simulations over a range of values of C, f, λ and A. It is sufficient to choose a single set of values. Run simulations for several values of α in the range 1.1 to 3.]

Note. You can generate an exponential random variable as follows. Let U be a random variable, uniformly distributed in $[0, 1]$. Most programming languages provide a primitive for generating such a random variable, e.g. `java.lang.Math.random()` or `random.random()` in Python. Now let $X = -\log(U)/\lambda$. Then X is an exponential random variable with rate λ, which means that its mean is $1/\mu$.

You can generate a Pareto random variable as follows. Let U be a uniformly distributed random variable, as above. Now let $Y = \nu/U^{1/\alpha}$, for some $\nu > 0$ and $\alpha > 1$. Then Y is a Pareto random variable with shape parameter α, and mean $\nu\alpha/(\alpha - 1)$.