Coursework 5

Simulation, Little's law, fixed points
Network Performance—DJW—2010/11

Recently, researchers at Google have argued[1] for an increase in TCP's initial congestion window. They argue that most Web transactions are short-lived, and that the current standard (an initial window of about 4KBytes of data) means that flow completion times are unnecessarily long.

In this coursework you will investigate how average flow completion time is affected by congestion control, including both slow start and congestion avoidance. You will simulate a simple model of congestion control, with a pessimistic version of slow start, and you will test the simulation results against approximations derived from the fixed point method.

The model of congestion control. Consider the state of a flow to be a pair (x, f) where x is the current transmission rate of the flow, and f is the amount of data left to transmit. Assume that $x = 0$ when the flow starts. The congestion control algorithm is as follows. In the absence of congestion, transmit rate grows linearly:

$$x(t+u) = x(t) + au \quad \text{for } u \geq 0,$$

where $a = 1/\mathsf{RTT}^2$ pkt/s^2, and x is measured in pkt/s and time is measured in seconds. The amount of data transmitted in this time is

$$\int_t^{t+u} x(s)\,ds = x(t)u + \frac{1}{2}au^2.$$

Obviously the flow will finish once it has transmitted all its remaining data; we can calculate how long this will take in the absence of congestion by solving $x(t)u + au^2/2 = f$; the answer is

$$u = \frac{\sqrt{x(t)^2 + 2af} - x(t)}{a}.$$

Use the following model for congestion at the link. When there are n active flows, with transmit rates $x_1(t), \ldots, x_n(t)$, then all the transmit rates grow linearly until some time s such that $x_1(s) + \cdots + x_n(s) = C$, where C is the link speed. At this time, one of the flows (selected at random) cuts its transmit rate from $x_i(s)$ to $x_i(s)(1-b)$, where $b = 1/2$.

A fixed point approximation. The following fixed point approximation is in terms of four variables: w, the average completion time for a flow; d, the average transmit rate of a flow when it completes; x, the average transmission rate that a flow gets when it is congestion avoidance (i.e. ignoring the time it takes for the flow to get started); and n, the average number of active flows. The equations, for the case that flow sizes are $\mathsf{Exp}(1/m)$ and flows arrive at rate λ, are

$$w = \exp\Big(-\frac{x^2}{2am}\Big) + \sqrt{\frac{\pi m}{2a}}\Big(2\Phi\big(x/\sqrt{am}\big) - 1\Big) \tag{1}$$

$$d = \sqrt{\frac{\pi am}{2}}\Big(2\Phi\big(x/\sqrt{am}\big) - 1\Big) \tag{2}$$

$$C - x\frac{b}{2} - \frac{d^2}{2na} = \lambda m \tag{3}$$

$$n = \lambda w. \tag{4}$$

[1] Technical paper in CCR online, July 2010, `http://ccr.sigcomm.org/online/?q=node/621`

Here $\Phi(x) = \mathbb{P}(N(0,1) \le x)$, where $N(0,1)$ is a random variable with the Normal distribution, with mean 0 and variance 1. In Python, $\Phi(x)$ is called `scipy.special.ndtr(x)` and in R it is called `pnorm(x)`.

Equations (1) and (2) come from assuming that the flow's transmit rate grows linearly until it reaches x whereupon it stays constant, then integrating to find the amount of data sent in a given time, then inverting to find how long it takes to send a given amount of data M, then letting $M \sim \mathrm{Exp}(1/m)$ and taking the expectation either of time or of current rate. Equation (3) derives from approximating how much utilization is lost due to congestion backoff and flow departures, and equating total useful work done to total amount of arriving work. Equation (4) is Little's law.

Question 1. Program a simulator of this system. Your written report should include the source code. It should also include a brief description of how your simulator works, at a similar level of detail to the student's answer for Coursework 2. *[Hint. It is possible to program this simulator by making a few small changes to the simulator for Coursework 2.]*

Test that your simulator works correctly, for some specific set of input data which you should choose (flow arrival times, flow sizes, which flow backs off when the link is full). Work out with pen and paper what should happen, and also give a printout from your simulator that demonstrates it agrees. *[Hint. Make sure your example 'touches' all parts of your code.]*

Question 2. Run simulations with $m \in \{0.2, 1, 5\}$ Mb, $C = 10$ Mb/s, and $\lambda m/C = 0.9$, and $\mathsf{RTT} = 200$ms. Run these simulations both with exponentially-distributed flow sizes, and Pareto-distributed flow sizes with $\alpha = 1.2$ using the random number generator from Coursework 2.

(i) Plot histograms[2] of the queue size distribution, and also histograms of the distribution of link utilization. *[Hint. By the PASTA property, it is sufficient measure queue size and utilization as seen by arriving flows.]* On your graphs, superimpose the corresponding results for the idealized processor sharing model.

(ii) Also report the mean utilization for each of your experiments. Using an argument similar to the proof of Little's law, explain why the mean utilization should be 0.9 in each case. *[Hint. See the last page of §4.4.]*

(iii) According to Little's law, $n = \lambda W$, where w is the average number of active flows and w is the average flow completion time. For each of your simulations, report w and n/λ, and verify that Little's law holds.

[Hint. Make sure you use consistent units, e.g. you work entirely in pkt/s or Mb/s. The rate increase parameter a is measured in pkt/s^2, and 1pkt is approximately 1500 Bytes.]

Question 3. Rewrite (3) as an equation for x, then solve the fixed point equations for the parameter values you simulated in Question 2. How accurate is this fixed point approximation?

Equations (1)–(3) are all approximations. Test the validity of each of these approximations for the simulations you conducted in Question 2 with $\mathsf{Exp}(1/m)$ flow sizes, by measuring w, d, x and n from your simulator and then evaluating each of the equations. *[Hint. Here is one way to measure x. Each time a flow's transmission rate is cut from r to $r(1-b)$, record $r(1-b/2)$, then take the average of these readings over the flow's lifetime; call this the congestion-avoidance transmission rate for that flow. Then let x be the average congestion-avoidance transmission rate across all flows that received say three or more drops.]*

The fixed point equations (1)–(3) are approximations, and they are not particularly accurate. In a full research project you would test each of the equations using a simulator, explore the reasons for the inaccuracies, and devise more accurate equations.

[2] In R, the command `histogram(~x)` plots a histogram of the values in `x`. If however you have already binned the values and counted totals, e.g. in Python, and all you want is to draw bars for your totals, use `xyplot` with a panel function like `panel.rect(x=x,y=y/2,width=1,height=y)`.

Appendix: the theory behind coursework 2

In Coursework 2 you used simulation to investigate average completion time for flows sharing a link. In lectures, we have learnt theory that predicts the simulation results. The theoretical model is a processor-sharing link (§3.1). In this idealized model, the link spends all its time either idle (no flows) or 100% utilized (one or more active flows).

Let the link speed be C. Let new flows arrive as a Poisson process of rate λ, and let average flow size be m. Since this is a symmetric queue, the average queue size depends only on m and not on the distribution of flow size, hence we may as well assume that flow sizes are exponentially distributed with mean m (§3.7). By analysing a Markov process model we can calculate the equilibrium distribution of the number of active flows N, and we find that the average number of active flows is $\mathbb{E}N = \rho/(1-\rho)$ where $\rho = \lambda m/C$, as long as $\rho < 1$ (§3.1). If $\rho > 1$ then the system is unstable and the number of flows increases the longer the system runs for (§1.9 from the online version of lecture notes, and §4.1). By Little's law, assuming the system is stable, the average flow completion time is $\mathbb{E}N/\lambda = m/(C - \lambda m)$ (§4.4). This theoretical formula is confirmed by simulation, though particular care is needed when flow sizes have a Pareto distribution (§1.6c from the online version of lecture notes).

Note that by the PASTA property, we get the same answer for average flow completion time whether we measure the average value of N observed at instants when flows arrive and then divide by λ, or whether we measure the average over all flows of their completion time (§3.6).