Notes on FCT, Slowdown, Heavy Tail Distributions

Balaji Prabhakar
Stanford University

Overview

- Infinitely long-lived flows vs dynamically arriving flows
  - Unit step response: pick parameters for control-theoretic stability
  - Flow completion time (FCT): what the users care about
- Heavy-tailed flow size distributions
  - Pareto distribution
  - Mice and elephants
  - Scheduling algorithms for exploiting them

Unit step response vs FCT

- Historically, congestion control research has considered the performance of a scheme under infinitely long-lived flows
  - This gives the unit step response of the scheme
  - Very useful for control-theoretic analysis and hence for picking the parameters for the stability of the control loop
  - But, it does not capture dynamic situation of flows arriving and departing (which is the actual situation)
  - It does not have a notion of "load" which can be increased; it is always at 100% load
  - It does not capture flow completion time (FCT), a quantity users care about

- The recent literature takes a 2-step approach
  - First study scheme under infinitely long-lived flows
  - After picking parameters and ensuring stability of control loop, consider FCT
  - This is consistent with CPU performance under "workloads" consisting of files and brings the role of algorithms into focus
  - Key metric: In addition to FCT, it is "Slowdown"
  - Slowdown for job or flow of size x = FCT of flow / x = 1 / Bdwdth given to the flow

Heavy tailed Distributions

- Let $X \geq 1$ be a random variable which denotes job/file sizes
  - let $E(X) = \int_1^\infty P(X > t)dt < \infty$ and $E(X^2) = \infty$
  - e.g. for $t \geq 1$, $P(X > t) = t^{-\alpha}$ or density $f_X(t) = \alpha t^{-\alpha-1}$

    if $\alpha \in (1, 2)$, then $EX < \infty$ but $E(X^2) = \infty$

  $\rightarrow$ for α as above, this is the Pareto distibution and α is called the shape parameter
- Some properties of the distribution
  1. decreasing failure rate; failure rate $FR(t) = \frac{f_X(t)}{P(X>t)} = \frac{\alpha t^{-\alpha-1}}{t^{-\alpha}} = \frac{\alpha}{t}$

     note: $P(X > t + s|X > t) = \left(\frac{t}{t+s}\right)^\alpha$ increases with t
  2. heavy tails: e.g. if $\alpha = 1.1$, then the largest 1% of the jobs constitute 50% of the load
- Heavy tails are prevalent
  - CPU process life-time distributions
  - Web file sizes
  - FTP file transfers, etc

Scheduling algorithms

- With FCT, the role of scheduling algorithms come into play
- For simplicity, let us assume a single server queue first
  - This corresponds to flows passing through a single link
  - We can do the network case, where there are multiple links, later
- Scheduling algorithms can be divided into categories
  - Job-size based or not
  - Pre-emptive or not

	Not job-size based	Job-size based
Not pre-emptive	FIFO	SJF
Pre-emptive	Processor Sharing (PS)	SRPT

Scheduling algorithms: General conclusions

- In terms of FCT and Slowdown, FIFO is v.bad for heavy tail distributions
  - FIFO not relevant in networking because all congestion control schemes transmit packets simultaneously from different files, FIFO is not provided by network
  - Included as a useful benchmark
- SRPT is optimal but basically unimplementable
  - Don't know how many packets remain to be transmitted
- Processor Sharing (PS)
  - Has constant slowdown; i.e. it gives equal bandwidth to all flows, regardless of their size
  - But, this can be very bad when compared to a job-size based scheme which gives more bandwidth to short jobs
- This is because
  - The small (mice) flows do not really contribute to congestion and they are not easy to detect; so just let them through quickly
  - Large (elephant) flows cause congestion and need to be controlled
  - Under HT distribution, there are many mice and a few elephants, so helping mice dramatically reduces overall FCT

In the Data Center

- Not yet clear what the flow size distribution going to be
  - However, there will like be inter-process communication (IPC) traffic: short, delay-sensitive. Treat these as mice and get them transferred quickly
  - There will also likely be large disk transfers. These will need to be congestion controlled. Think of these as the elephants
  - Bottomline: Control the elephants, get the mice out asap
- So favoring the mice by giving them more bandwidth (or reducing their slowdown)
  - Benefits performance by reducing FCT
  - Makes for easier implementation: We only need per-elephant rate limiters, as opposed to per-flow rate limiters
- In terms of actual simulation studies re the above
  - We have already seen Davide's presentation on FCT
  - We have also seem Mitch and Cyriel's sims
  - On our side, we have simulated QCN and will be presenting that shortly
  - Some work was done at Stanford in 2004 on an algorithm called SIFT which detected and favored the packets of short flows at Internet routers; it showed how there can be a huge improvement in FCT for *all* flows, not just the mice; please email me if you're interested in that paper