6.896 Lecture 3: Methodology I: Design Principles and Analysis *

Hari Balakrishnan

September 16, 1998

1 Outline

- **Readings: SRC84, Bol93**
- ALF and ILP (from last class)
- The end-to-end argument
- Inferring delay and loss characteristics of the Internet

2 The end-to-end argument

- Folklore, formalized by Saltzer, Reed and Clark in what has become a classic systems paper.
- System designer's main job: figure out what functions to implement (modularize), and where to place these functions in the system. This principle tries to guide the latter process.
- "Place it where it can be done completely, or if it only does an incomplete job someplace, then it'd better be a significant performance enhancer (or cost savings, although they didn't articulate it specifically)."
- Numerous examples from experience with many systems.
  - Careful file transfer: app-level checks for integrity.
  - MIT router swapping bytes.
  - Encrypted transmission.
  - RISC analogy.
- What it does not tell us: where to place early checks for performance improvement, how much to place in a "lower" layer, economic issues.
- Caveat: It's a fairly general and powerful engineering principle, but it's only that. It's not the law!

* These notes are based on the end-to-end arguments paper by Saltzer, Reed and Clark and Jean Bolot's SIGCOMM 1993 paper on end-to-end delay and loss characteristics. We will spend the first 20 minutes on the ALF/ILP paper from last class.

3 Annpouncements

- Reminder: signup or send me email. Going to create mailing list.
- Next class: CK74 and Pax96. Find out what the original Internet designers were thinking. Find out all about the routing pathologies that occur in the Internet. The fact that it works at all seems like a miracle! Also read the tcpdump man page (will do a quick overview of BPF and tcpdump).
- Get familiar with *ns*. Will spend some time talking about simulation as a research tool in networking.
- Will put these notes online.

4 End-to-end delay and loss characterization

Jean Bolot's paper from SIGCOMM 1993. Uses *NetDyn,* a tool from Maryland (Sanghi), written to diagnose aspects of the Internet.

4.1 Methodology

- Simple experiments: The Internet is complex; complex experiments only serve to complicate.
- Send a small UDP packet, get an echo reply. Draw topology. Draw topology w/ bottleneck link. Explain notation: s_n, r_n, rtt_n, δ. (Q: why not simply ping?).
- Cute analysis: Clever application of straightforward queueing theory and algebra—the beauty of this work is its simplicity!
- Analysis tool: phase plots. Idea: from phase spaces in dynamical systems—eliminate time and deal with other coordinates. (In dynamical systems, work with momentum and position. We use rtt_{n+1} and rtt_n).

4.2 Delay analysis

- Goal: Distribution of end-to-end delay. By-products: bottleneck bandwidth, "average" packet size of Internet packets.
- Draw close-up picture of queue. D, K (buffer size), probes, cross traffic, μ (service rate). Waiting time w_n. Hidden assumption here—what is it?
- Light load: w_{n+1} approx w_n. $w_{n+1} = w_n + \epsilon_n$. ϵ_n has mean 0 and small variance. Now, in general $rtt_{n+1} = D + w_{n+1} + P/\mu$ and likewise for rtt_n, which gives us $rtt_{n+1} - rtt_n = w_{n+1} - w_n$. For light load, this is ϵ_n. This explains "$y = x$" line on phase plot.
- Heavy load: And small δ. Probe compression occurs in this case. Suppose two probes are separated by B bits. Before this buffer clears, $k - 1$ more probes arrive (so now there are k probes waiting to be serviced). $w_{n+1} = w_n + B/\mu - \delta$, approx equal to $w_n + B/\mu$. For the remaining packets, $w_{n+i} - w_{n+i-1} = P/\mu - \delta$. This is *probe compression*. RHS is smaller than 0. Why? Probe compression explains large number of points on line $y = x + (P/\mu - \delta)$.
- Bottleneck bandwidth now trivial to estimate! How?

- *Principle: Queues at bottleneck (no queues ==¿ not a bottleneck!) cause back-to-back probe packets to get spaced according to the bottleneck bandwidth.* This is the principle (but not the same machinery) used in packet pair, TCP's ack clock based congestion control etc.). Pause for absorption...

- Packet sizes. Use Lindley's recurrence. Related waiting times of successive customers into a queue. Draw picture.

- Apply it twice: once to get waiting time of cross-traffic burst and once to get waiting time of probe $n+1$.

- End up with $w_{n+1} - w_n = (b_n + P)/\mu - \delta$. Explain this intuitively based on picture (no need for Lindley's recurrence, as it turns out).

- Picture showing distribution of $w_{n+1} - w_n + \delta$. Different peaks at values. "Frequency analysis."

- Pragmatics: Need to pick "right" value of δ (too small means overwhelm router; to large means too much intervening traffic). Need to deal with real-world issues like coarse-grained clocks. Still, things work out and some wonderful conclusions can be made, esp. for small-enough δ.

- Turns out left-most freq is the highest, corresponding to later probes queued behind a large packet. Then comes empty queue.

- Third frequency peak gives packet size estimate. Treating packet size as an r.v., what statistic does this correspond to? Why?

- $w_{n+1} - w_n > P/\mu - \delta$ implies one or more Internet packets have arrived in the interim. Using curve fit, has evidence of exponential distribution. While consistent with other observations of the time, doesn't gel with subsequent observations of L.R.D and self-similarity.

4.3 Loss analysis

- p_u is uncond. loss probability. p_c is conditional loss probability. g is "gap" or size of a burst loss.

- $p_u <= p_c$. Why?

- Finds that g is close to 1. Augurs well for FEC. This finding is very inconclusive, though.

- $g = 1/(1 - p_c)$ under certain assumptions (rtt_n is stationary and ergodic). Under stronger assumptions, this is easy to show.

4.4 Summary

- Delays, as expected, arise because of cross-traffic. We can now, from end-to-end measurements, determine packet sizes of Internet cross-traffic. We can also get bottleneck bandwidths.

- Exponential inter-arrival times for Internet packets at a router. Makes sense, from independent users. But more recent work shows LRD.

- Losses are not as bursty as one might expect. But then again, this part is not as thorough or conclusive.

- Simple experiments and clever analyses can accomplish a great deal!