"We have Met the Enemy and [S/]He is Us": A View of Internet Research and Analysis

Kathleen Nichols
BCIT Internet Engineering Lab Opening

June 11, 2001

Packet Design

Everyone Loves Mythology (Especially Tech Folks)

- King Arthur
- Lord of the Rings
- Star Trek
- Star Wars
- Harry Potter
- ATM QoS

These Self-contained Worlds have Their Own Rules

- “The science of Star Trek”
- Devotees can endlessly argue about the self-consistency (or lack of it) in these mythical worlds

This is all well and good, and potentially healthy escapism, but to distinguish reality from fantasy:

- Don’t expect to fly on a broomstick
- Don’t take standards body work too seriously
- Don’t expect a PhD thesis on networking to apply to the Internet

Suppose We Challenge the Last...

- Network research doesn't have to take place in a mythical domain but can be correct, relevant, and significant

- It can "shine a little science" on the real Internet rather than developing the arcane trivia of the Mythical Internet

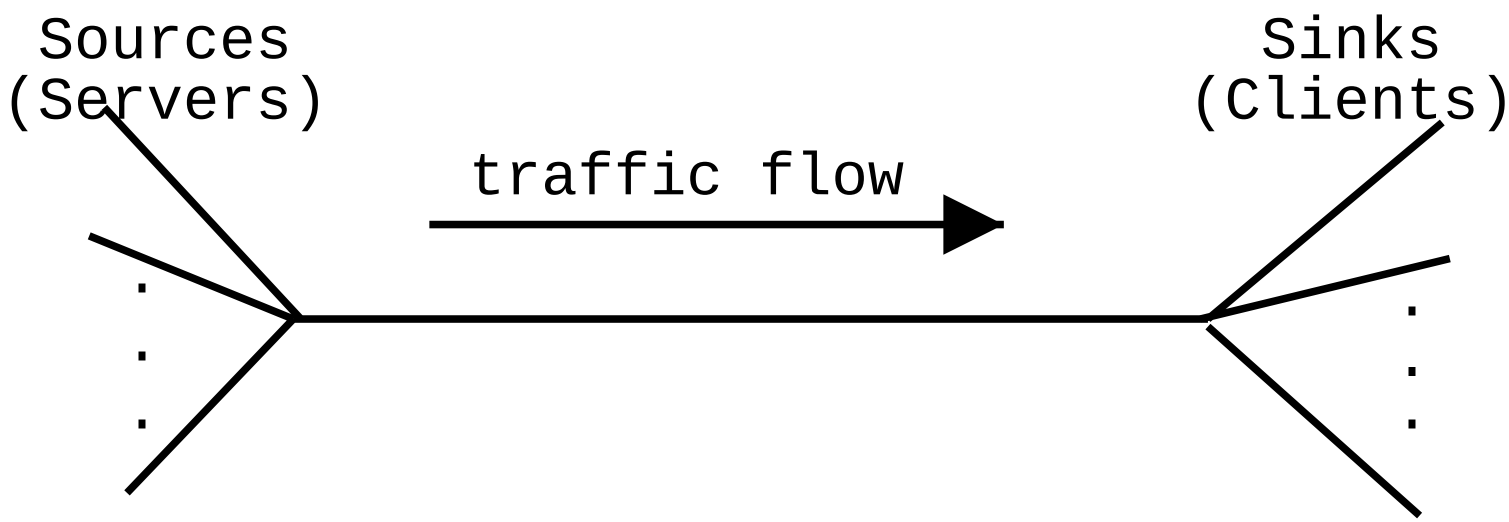
In the Mythical Internet:

A typical traffic pattern in an Internet link is (around) ten long-lived TCP connections

Toplogically, all are arranged in a dumbbell and dancehall pattern where all connections have the same RTT

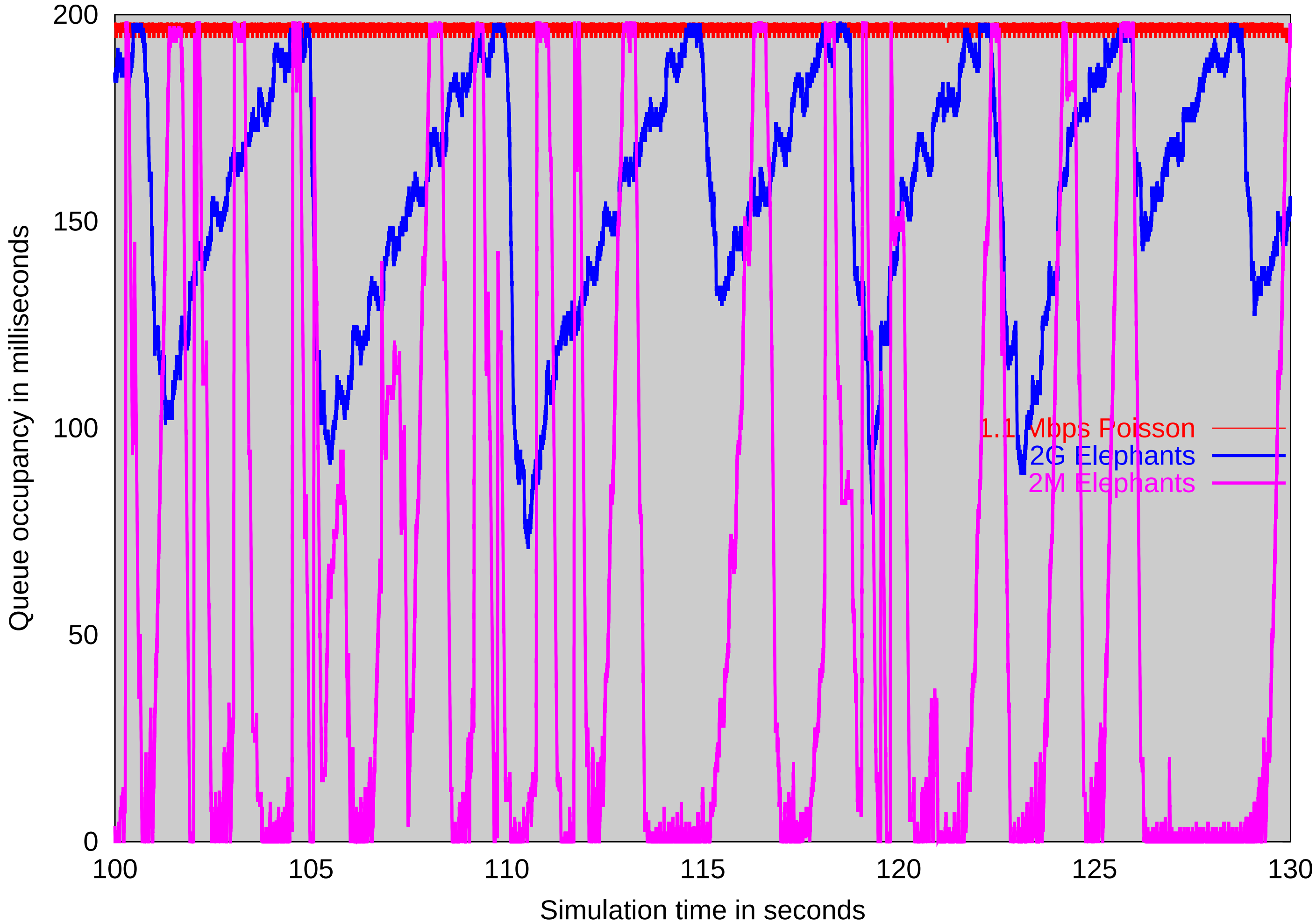
Implemented in simulations and lab studies

Quite commonly found in research papers and PhD theses



Once We Had Networks of “Swimming Fish”...

The fish have given way to elephants. All elephants are not created equal:

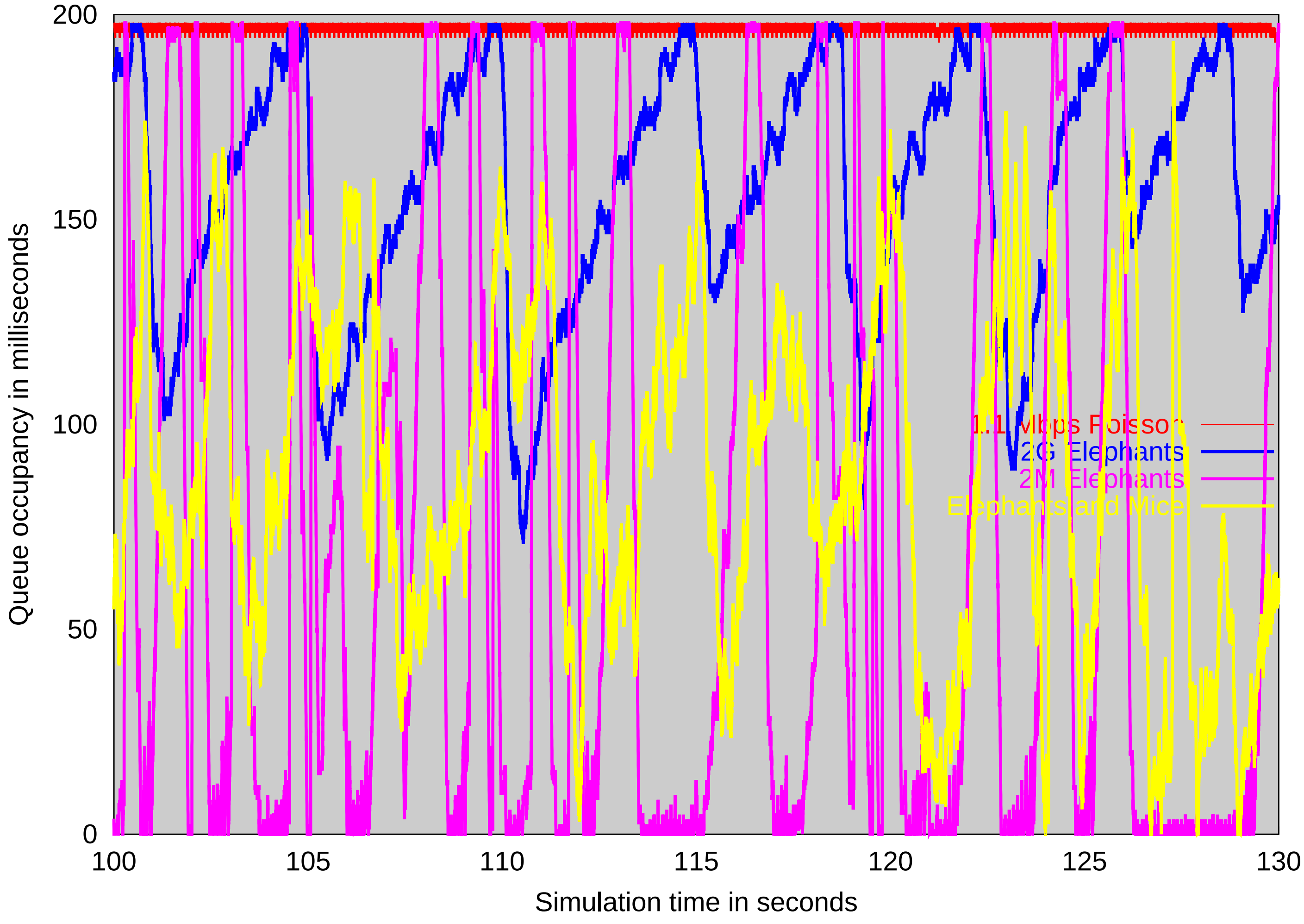


(10 connections through a 10 Mbps bottleneck with a drop tail queue)

The Real Internet Has No Fish and Mice Outnumber Elephants

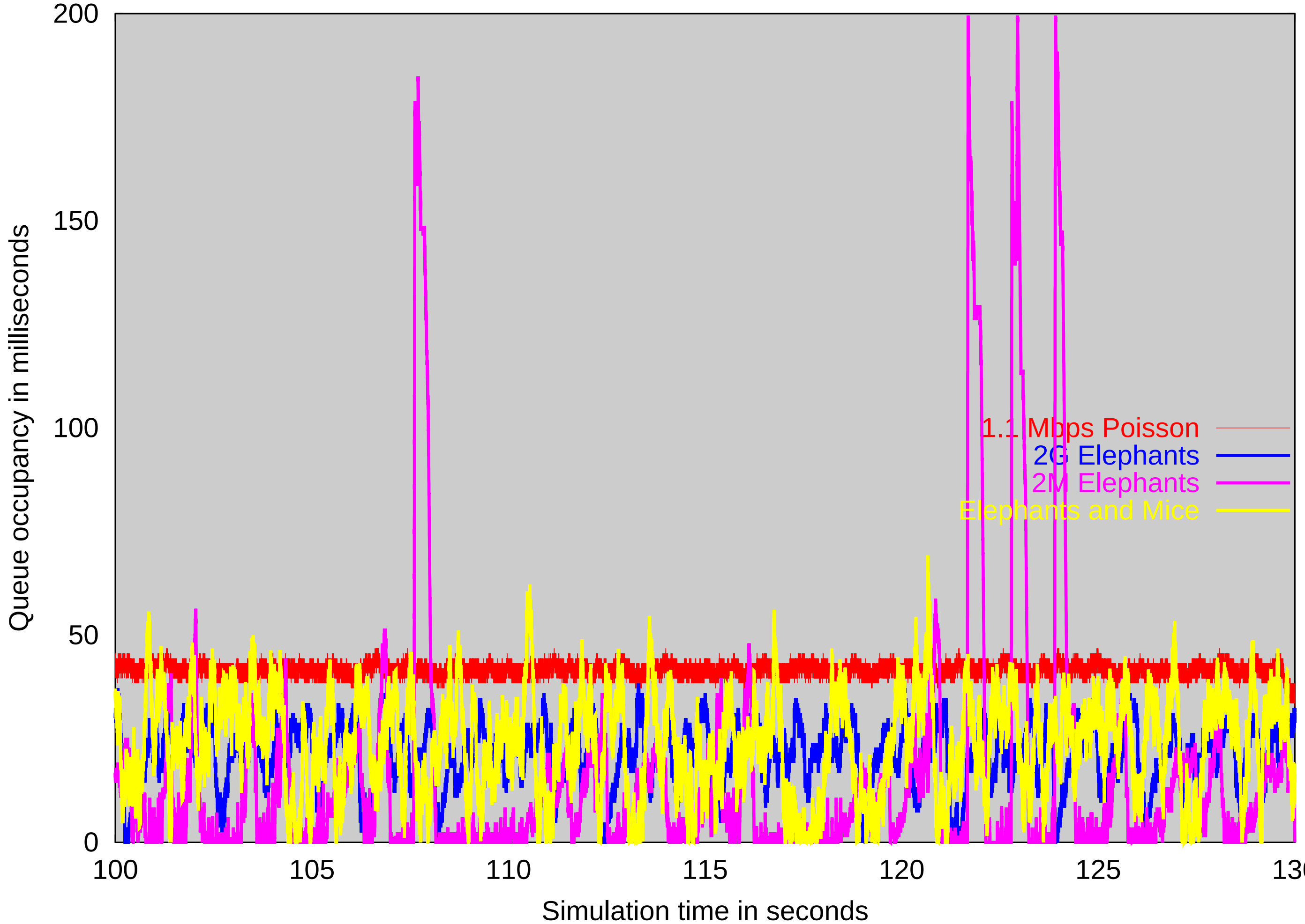
- Measurement studies consistently find HTTP acounting for 60-90% of all flows and the sizes of its fetched objects Pareto distributed
- Most objects around 1Kbyte (a single packet on most networks)
- When a user downloads a web page, typically many small transfers take place.
- Short flows have different characteristics from a long-lived TCP (like FTPing a large file)
- Measurements show that *most* flows are short, that is, they never reach TCP steady state
- CAIDA data has shown data at both OC3 and OC12 where one third of the packets per second on a backbone link have distinct source-destination pairs

Adding these Internet Mice to the Mix...



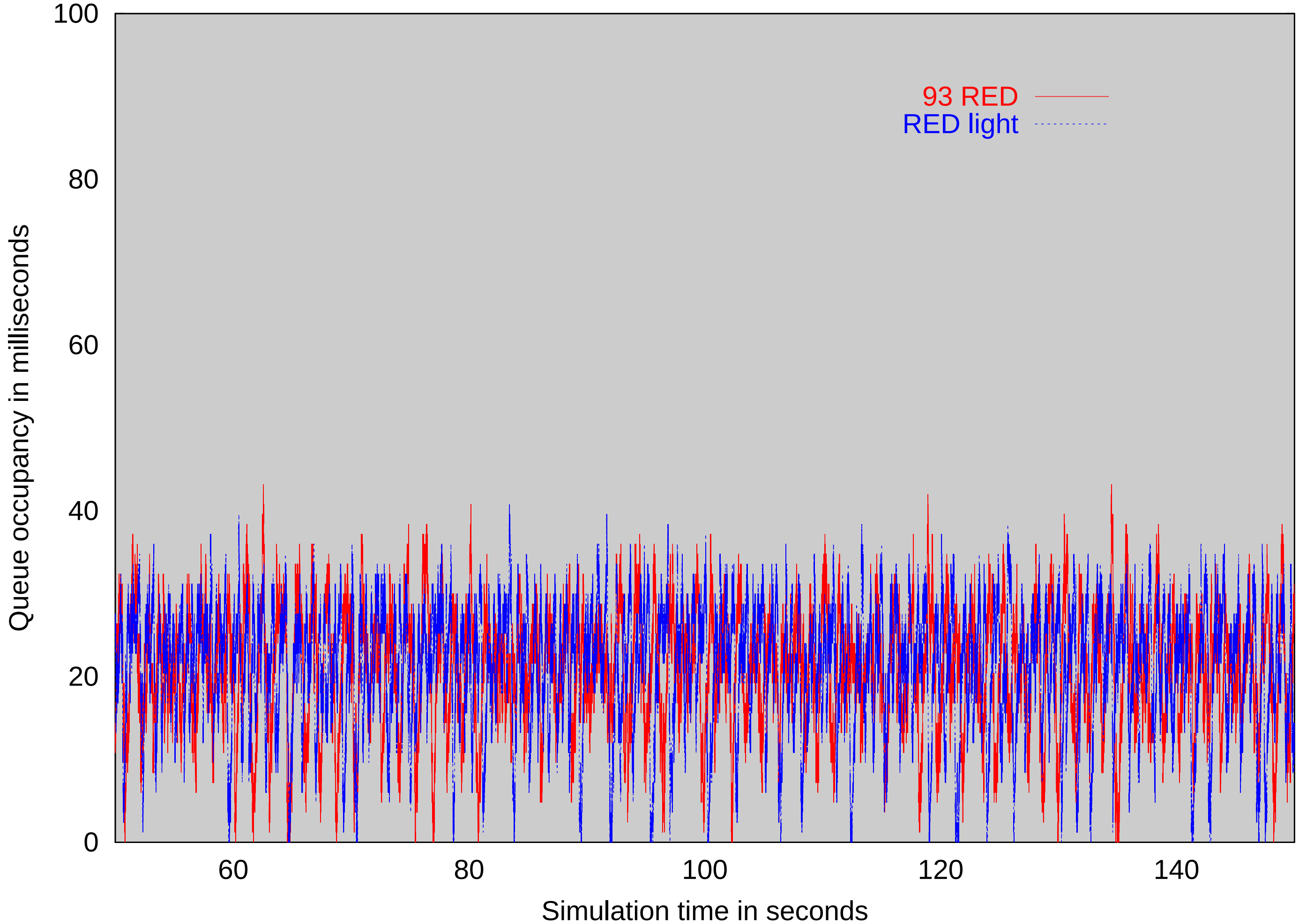
Oversimplification can Lead to Irrelevance

Look at RED (Random Early Detection) active queue management for these sources

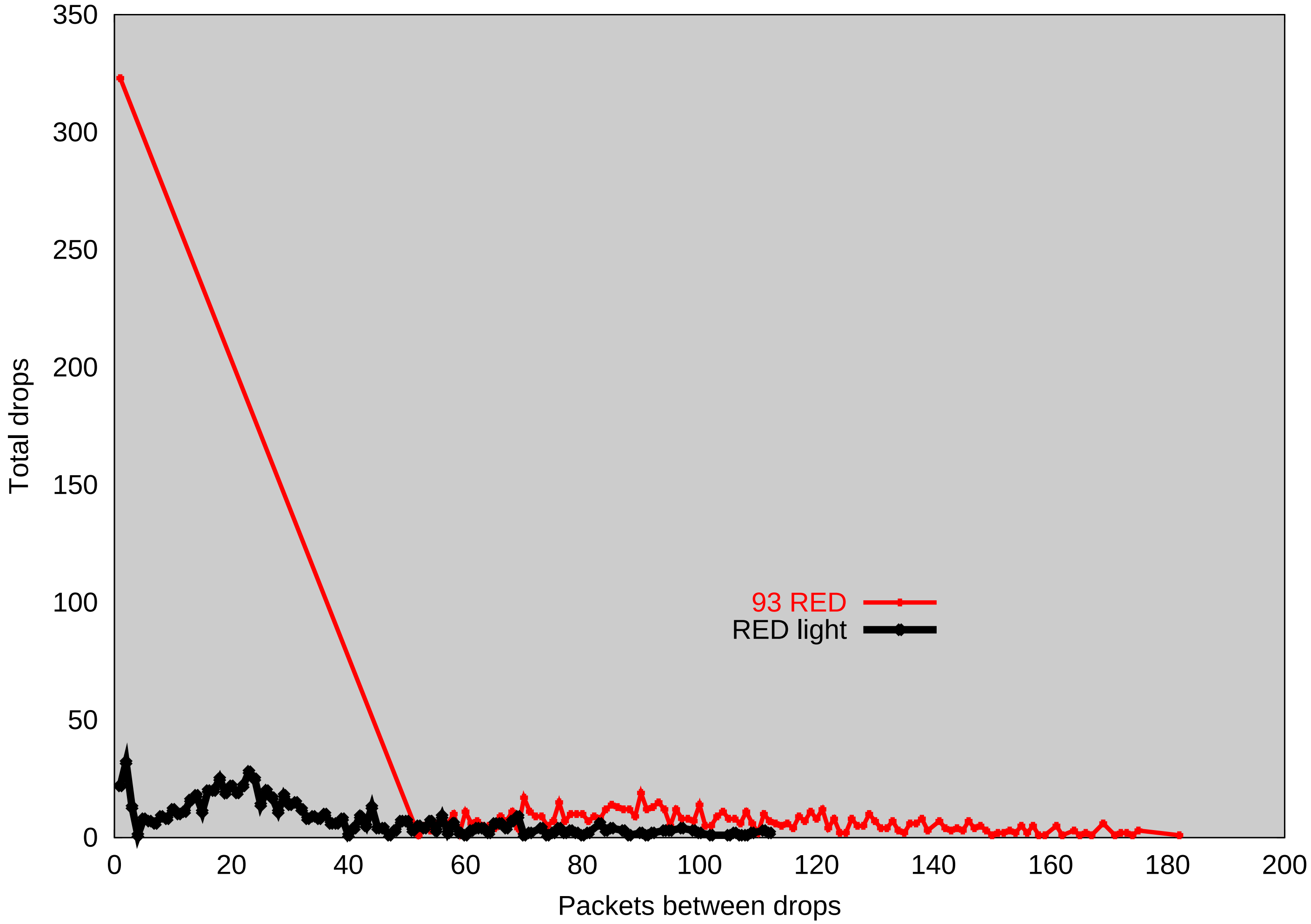


Almost Anything Works for Very Long-Lived TCPs

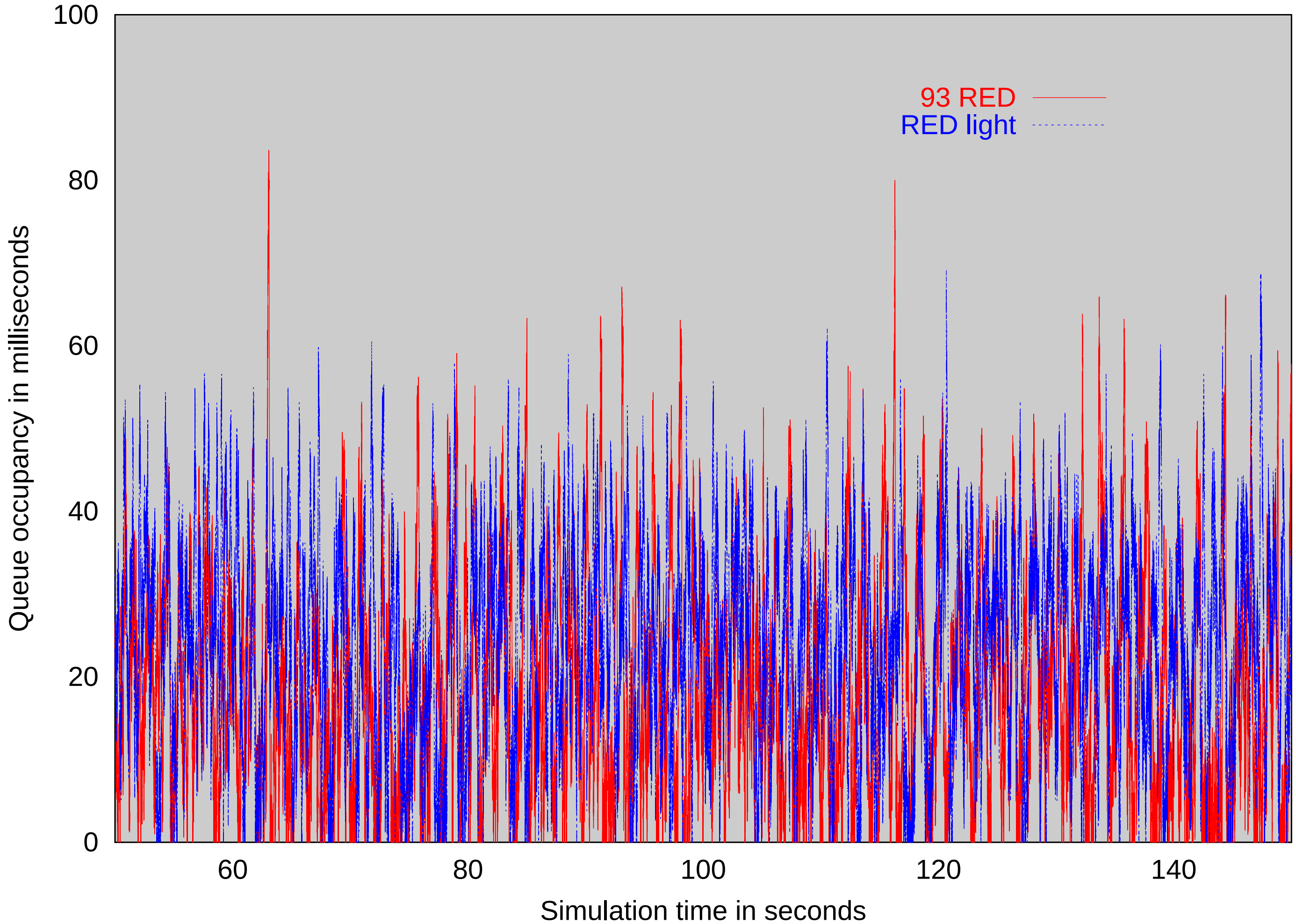
These very long-lived TCPs behave nicely (in queue occupancy) for two varieties of RED



...But How Does It Get the Job Done?

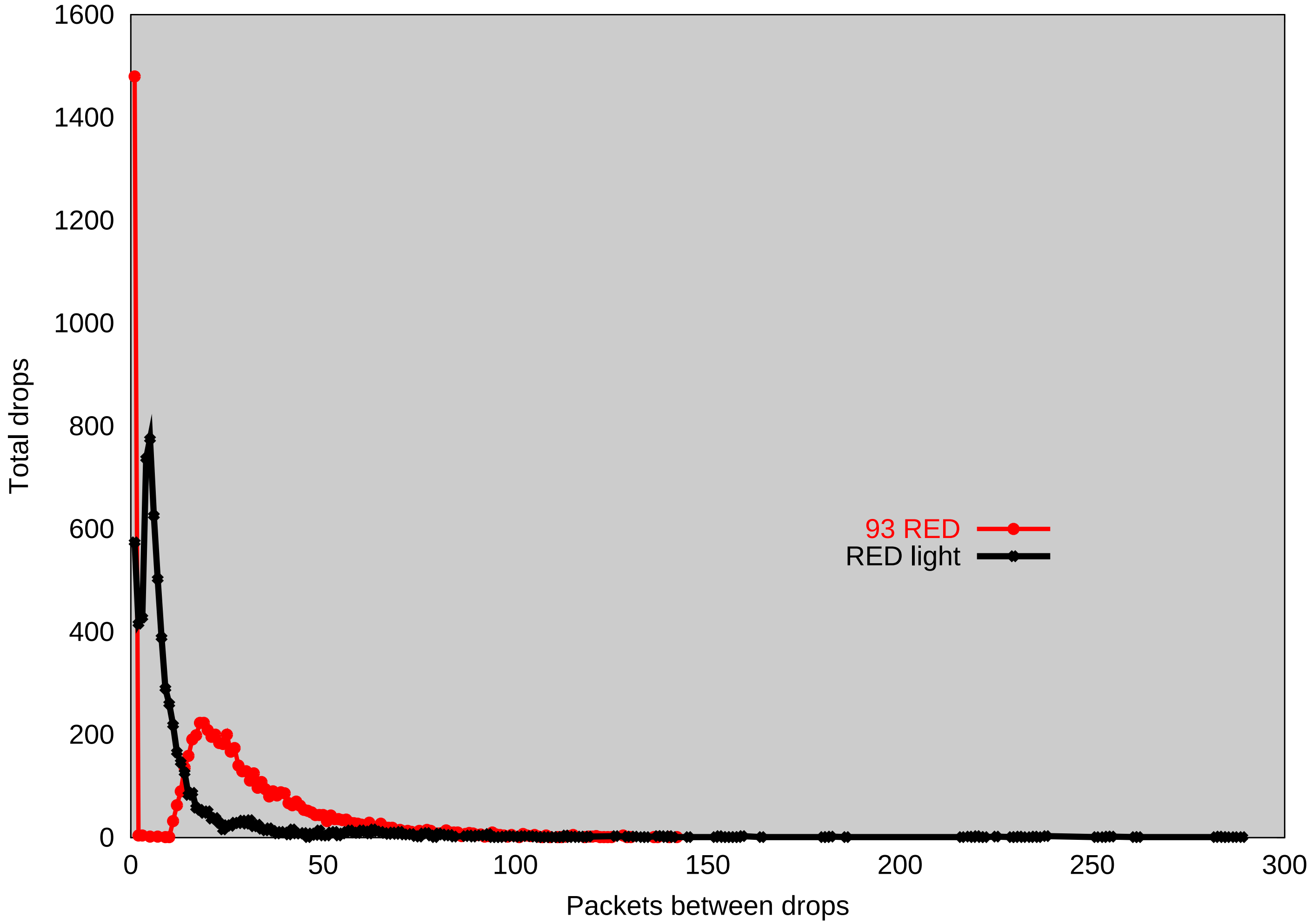


Almost Nothing Works for Web "Mice"



Packet Design

But Some Controllers “Control” Better



Challenges for a Testbed

- In the early Internet, the network **was** the testbed
- Those days are gone (but sometimes the Internet can be passively measured)
- Hooking up two computers and generating poisson traffic to drive a TCP connection is not a good use of a testbed. (Note that some measurement studies use Mythical Internet topologies and sources.)
- Getting a realistic operational scenario is going to be hard
- The “events” and anomolies you look for are going to take a lot of looking

Some Recent Myth Busting

A current fashion has it that a faster convergence time is needed than IP routing is capable of delivering. It's been suggested that some undeveloped and untried approaches would work better.

We thought we should apply a little science to this. That is, we endeavored to:

1. understand, in detail, why today's routing doesn't work as well as it should.

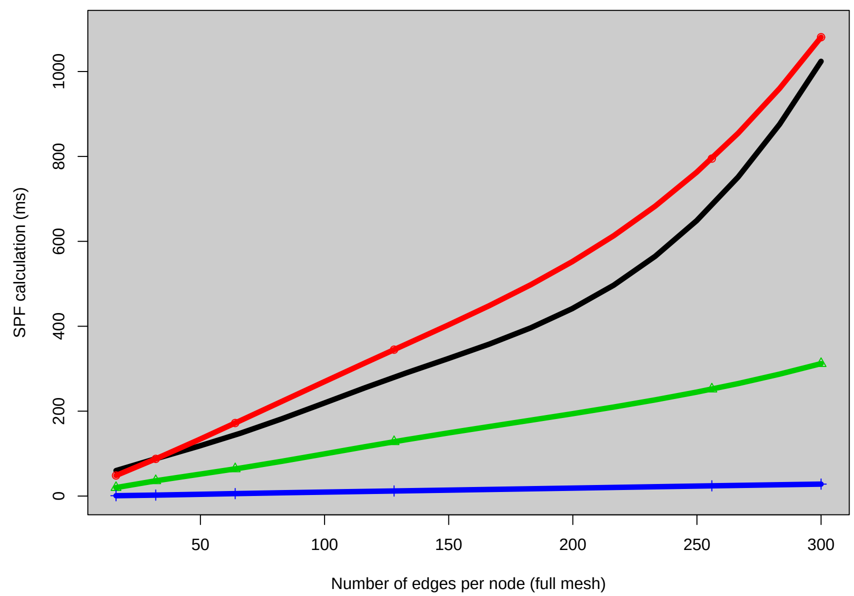
2. fix some of the implementation and specification mistakes.

"Toward Millisecond IGP Convergence" was presented at NANOG-20 by Alaettinoglu, Jacobson, and Yu) and showed problems where people weren't looking.

SPF Calculation: One Result in Brief

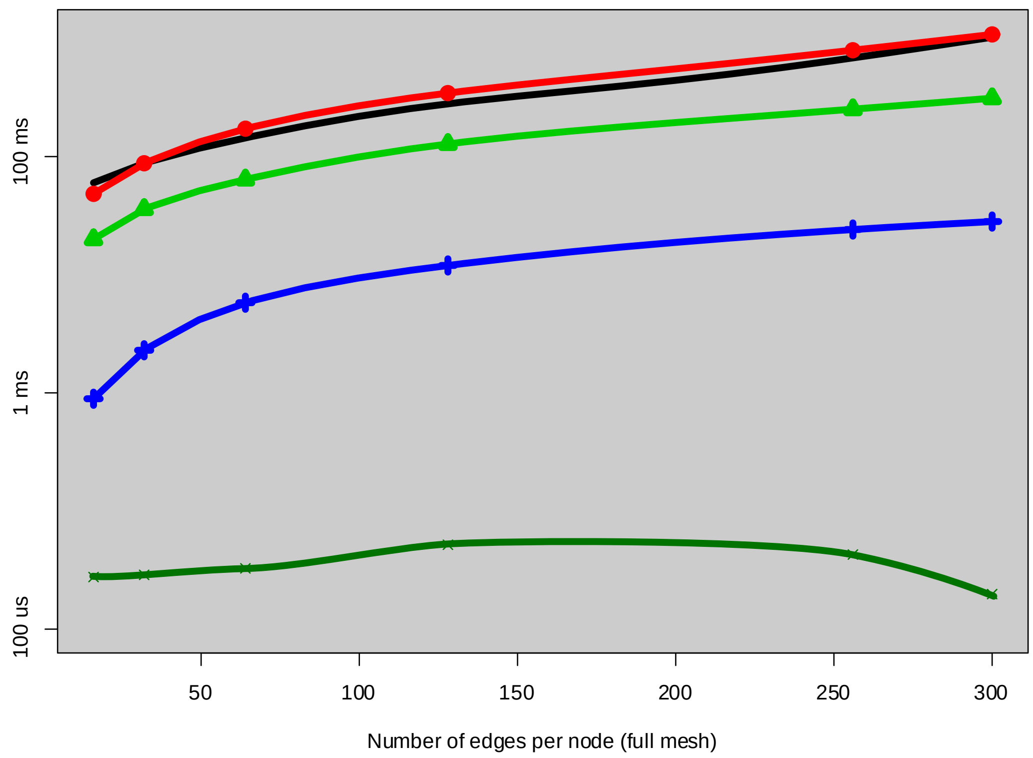
- When there is a topology change, one step that must always be performed is to compute new routes by running a Dijkstra Shortest Path First (SPF) algorithm on the changed topology.

- Calculation time of SPF on a full mesh topology: red, black, and green lines were found in commercial routers; blue is Haobo Yu's tuned SPF



Compared to a Modern Incremental Algorithm (on a log scale)

Note that even the “tuned” SPF can’t handle a fully-meshed structure the size of large ISPs (the smallest we talked to is a 400 node mesh). This one computes changes to SPF trees in $\log n$ time rather than $n \log n$. (red line is n^2!)



More Performance Myth Busting: Backbone Jitter

At Packet Design, we had the opportunity to measure a tier one ISP's transcontinental U.S. backbone

Despite some "claims" that sub-millisecond jitter cannot be acheived without changing IP (e.g., bolting on MPLS), we found:

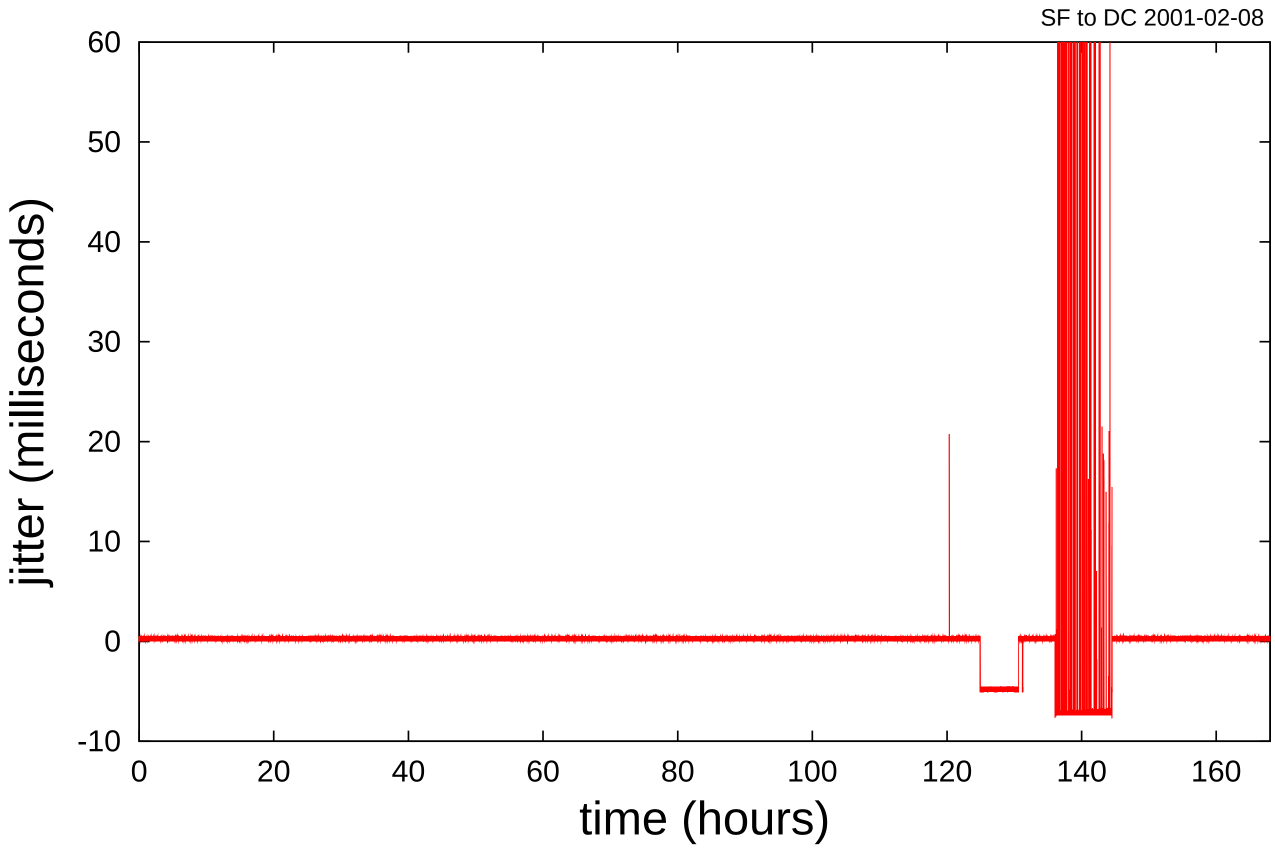
1. 99.99% of *sent* packets had a jitter below a millisecond ("four nines")

2. the cause that was keeping us from "five nines" (routing implementation bugs)

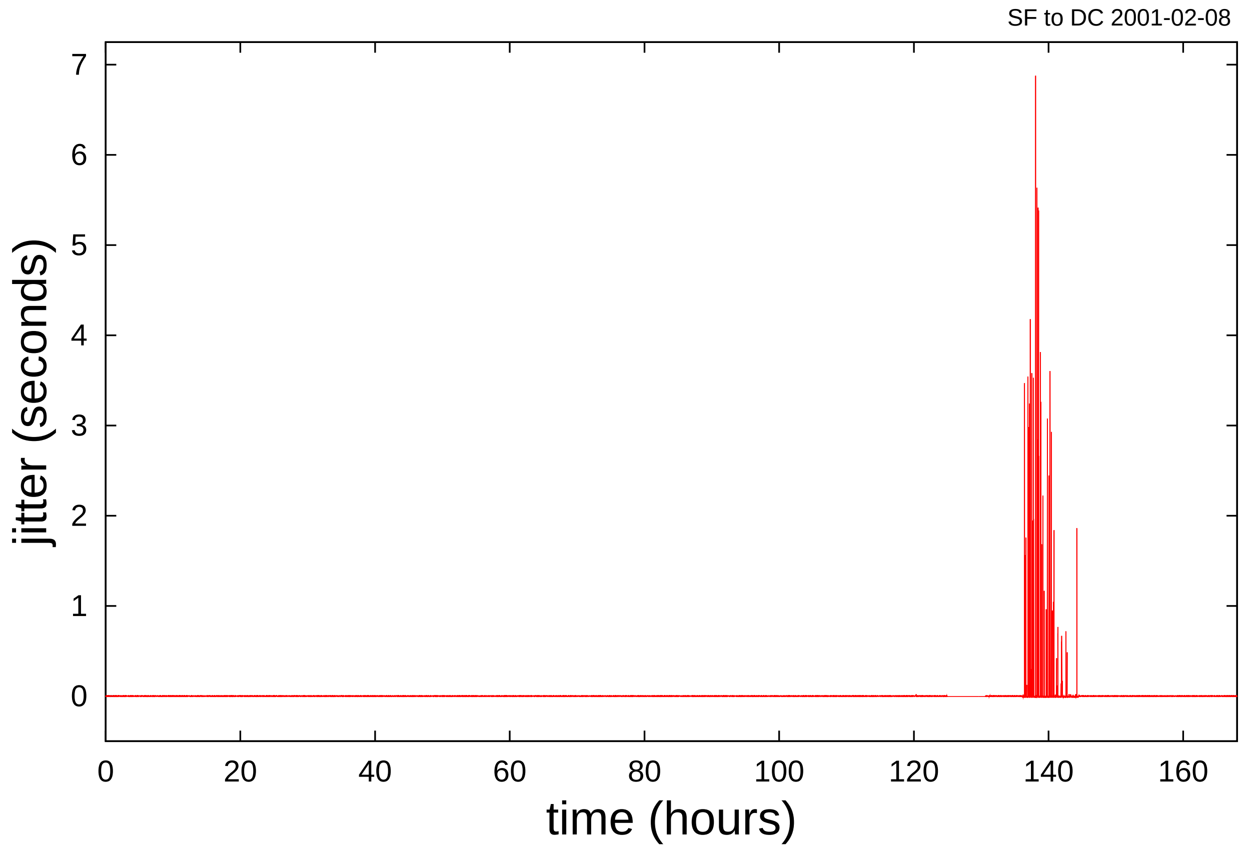
Finding these rare events takes a lot of observation time: e.g., one second is roughly 0.001% of a day

See "A Fine-Grained View of High-Performance Networking" by Casner, Alaettinoglu, and Kuan, NANOG-22 (www.nanog.org)

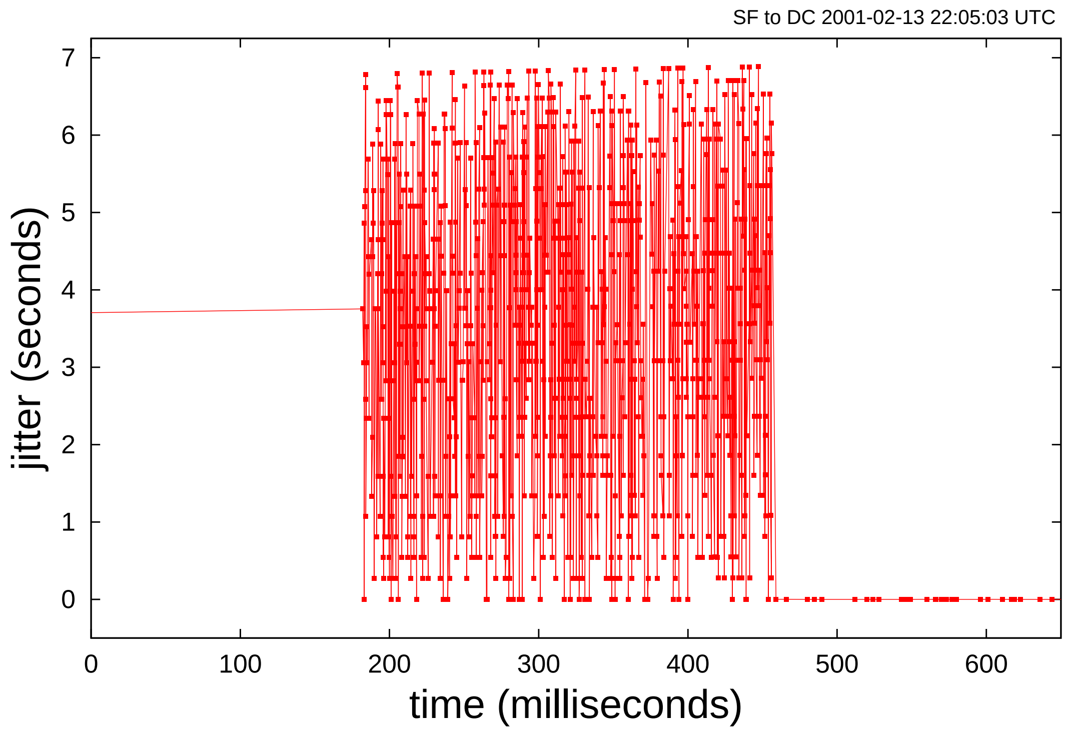
A Network that is 99.99% Clean (“four nines”)



With a Zoomed Out Vertical Axis Timescale



...But That 0.01% Was Very Interesting



Casner's "blender event", initiated by a line problem. For the story on the work, see www.nanog.org, NANOG-22.

Avoiding Myths

- No sweeping generalizations from simple models and experiments
- "Best Effort" $\neq$ "Who knows?"
- Almost anything will work for properly implemented long-lived TCPs
- Just because the Internet is 98% IP traffic, doesn't mean you can model it with LL TCPs
- Just because it's easy to analyze doesn't mean it's a good model
- It's not easy to change the networking code in all the hosts on the planet
- If the solution requires parsing flow id's in the core of the Internet, think again
- If someone tells you something is utterly broken and they have a nifty new solution, be skeptical

What Should be Worked On?

I can't tell you that: don't try to replicate this work!

Once you get the testbed right, there are two approaches I can think of:

- Identify problems and their causes
- Go after the myths and half-truths