Alcatel·Lucent

Bufferbloat

Dark Buffers in the Internet

or: how I became paranoid, and you should be paranoid too

Jim Gettys

Bell Labs

June 14, 2011

james.gettys@alcatel-lucent.com, jg@freedesktop.org

“It's the Latency, Stupid!” – Stuart Cheshire

This is a personal history – but I've provided only a few pieces of the puzzle, and assembled the puzzles. It's not a pretty picture.

Many ***more pieces***, ***and much more important pieces***, come to me from many other people, including Rich Woundy, Dave Clark, Dave Reed, Van Jacobson, Nick Weaver, Vern Paxson, and many others some of whom I do not know... My apologies if I have overlooked your contributions to solving the puzzle.

TCP Congestion Control and Avoidance

TCP will fill any buffer just before the bottleneck of a path

TCP's design assumption is that a congested network will generate ***timely*** notification of congestion, by packet loss or ECN (Explicit Congestion Notification). To do otherwise ***destroys TCP and other congestion avoiding protocol's control loops***

We judge other Internet transport protocols by whether they are "*at least as good*" as TCP at avoiding congestion

What happens if TCP's timeliness assumption is **badly *violated***? What happens if packet loss is avoided by buffering? Dropping a packet is evil, isn't it? But....

*A small amount of **timely** packet loss is not only normal, it is **essential** for correct TCP operation under congestion! And the edge of the network is always congested, among other places.*

The Myth of the Bandwidth Delay Product

Buffering is necessary. But how much is “right”?

A single TCP flow requires the bandwidth/delay product of buffering (less for more flows)

But...

- We've presumed 100ms (CONUS diameter) delay
  - But the world is big, and CDN's often local
- And bandwidth now varies
  - Ethernet 10Mbs to 10Gps
  - Wireless varies by orders of magnitude, as you move

How could **any** static buffer sizes possibly be correct?

Statically sized buffers can't possibly be correct in today's systems! One size fits all means wrong size.

“The Internet is slow today, Daddy!”

You've heard this for years. I tried to debug my network many times including a number of service calls.

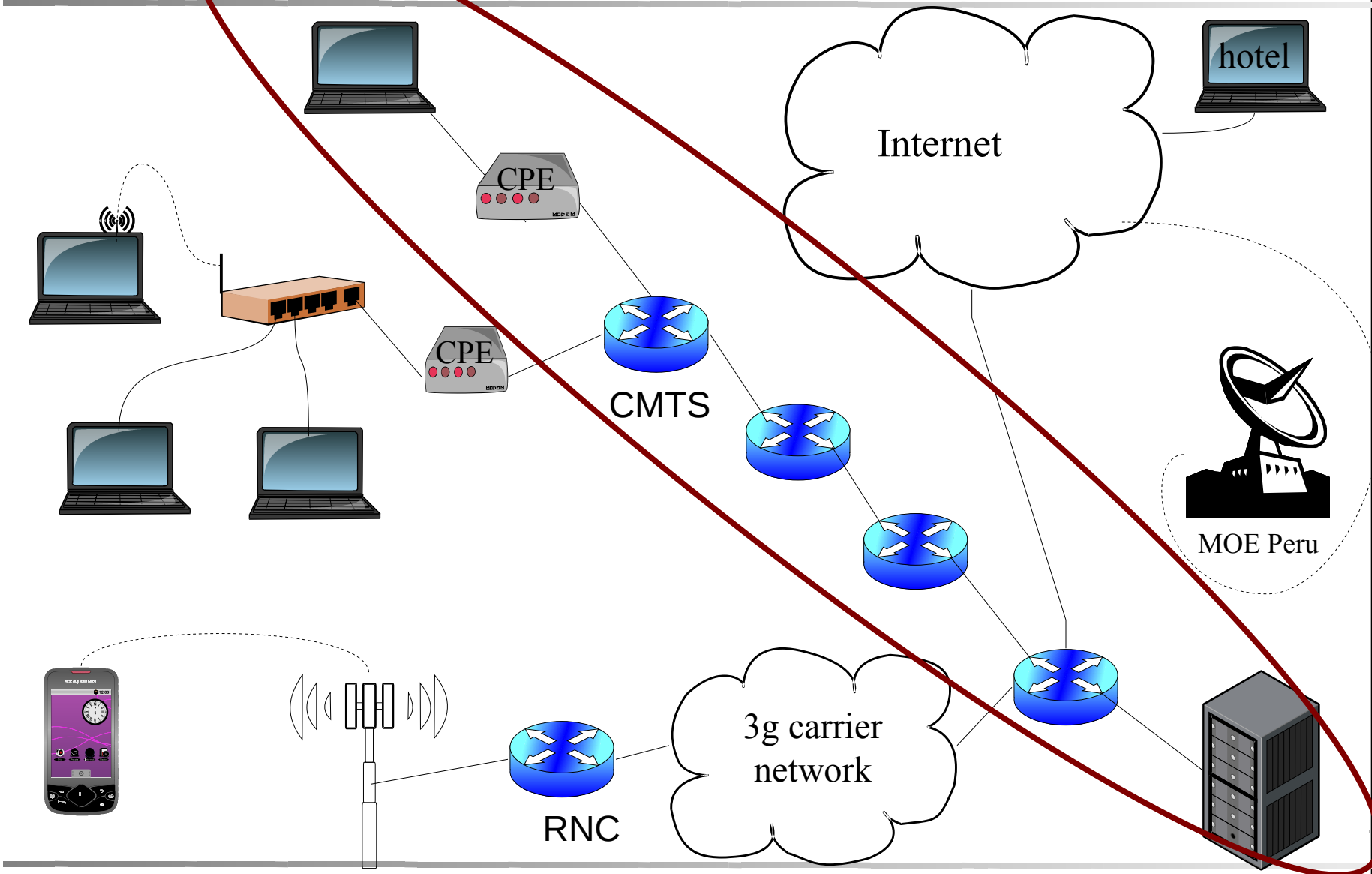
Every time I would go seriously debug, the network would stop mis-behaving.

> April, 2010 – simple bandwidth/latency test clearly demonstrated a problem: poor latency during continuous data transfers:
> ***1-2 seconds latency, with very rapidly varying 1-2 seconds jitter***

- In following up in July, I thought Comcast Powerboost might be the problem but again had trouble reproducing the problem.

I arranged to have lunch with one of PowerBoost's creators; the morning of lunch I finally figured out exactly how to tickle the problem

Network Map – Home to MIT Co-Lo center

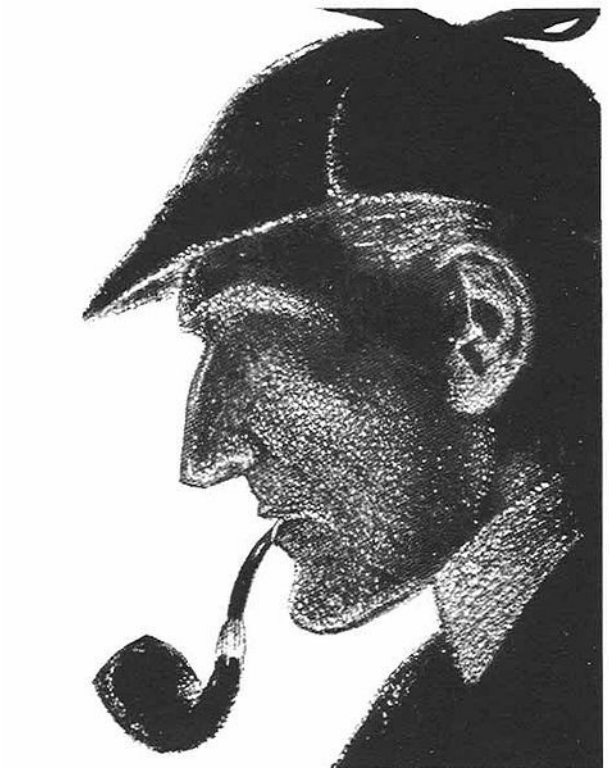


Lunch with Comcast... Many puzzle pieces

Lunch with Rich Woundy of Comcast provided many, many, puzzle pieces that fell into place later

- Suggests the “big buffers” problem, which he been chasing on suggestion of Dave Clark for over two years
  - I could drop back to DOCSIS 2, and the differences between DOCSIS 2 and DOCSIS 3 could be used to rule out TurboBoost
- RED (the most commonly available form of AQM, Active Queue Management) is often not enabled in major networks. AQM is often distrusted due to the need to tune RED, which has made many network operators averse to it. Some network operators run with AQM, some without
- ECN (Explicit Congestion Notification) is blocked in some networks
- Pointer to ICSI netalyzr

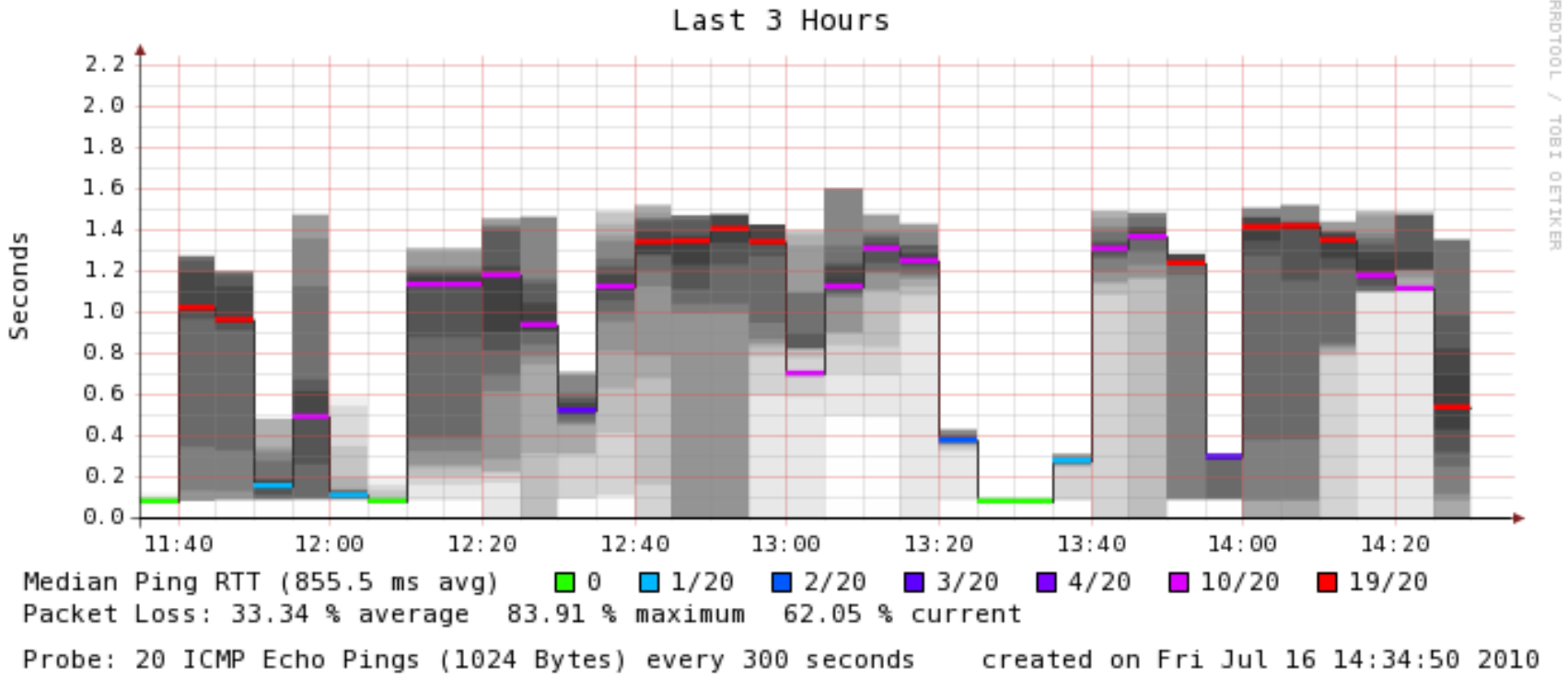
The Smoking Gun – bufferbloat is killing latency in the Internet



"We rushed into the captain's cabin... there he lay with his brains smeared over the chart of the Atlantic... while the chaplain stood with a smoking pistol in his hand at his elbow."

Sherlock Holmes story, *The Gloria Scott*, by Arthur Conan Doyle

dslreports smokeping: my home link – 7/16

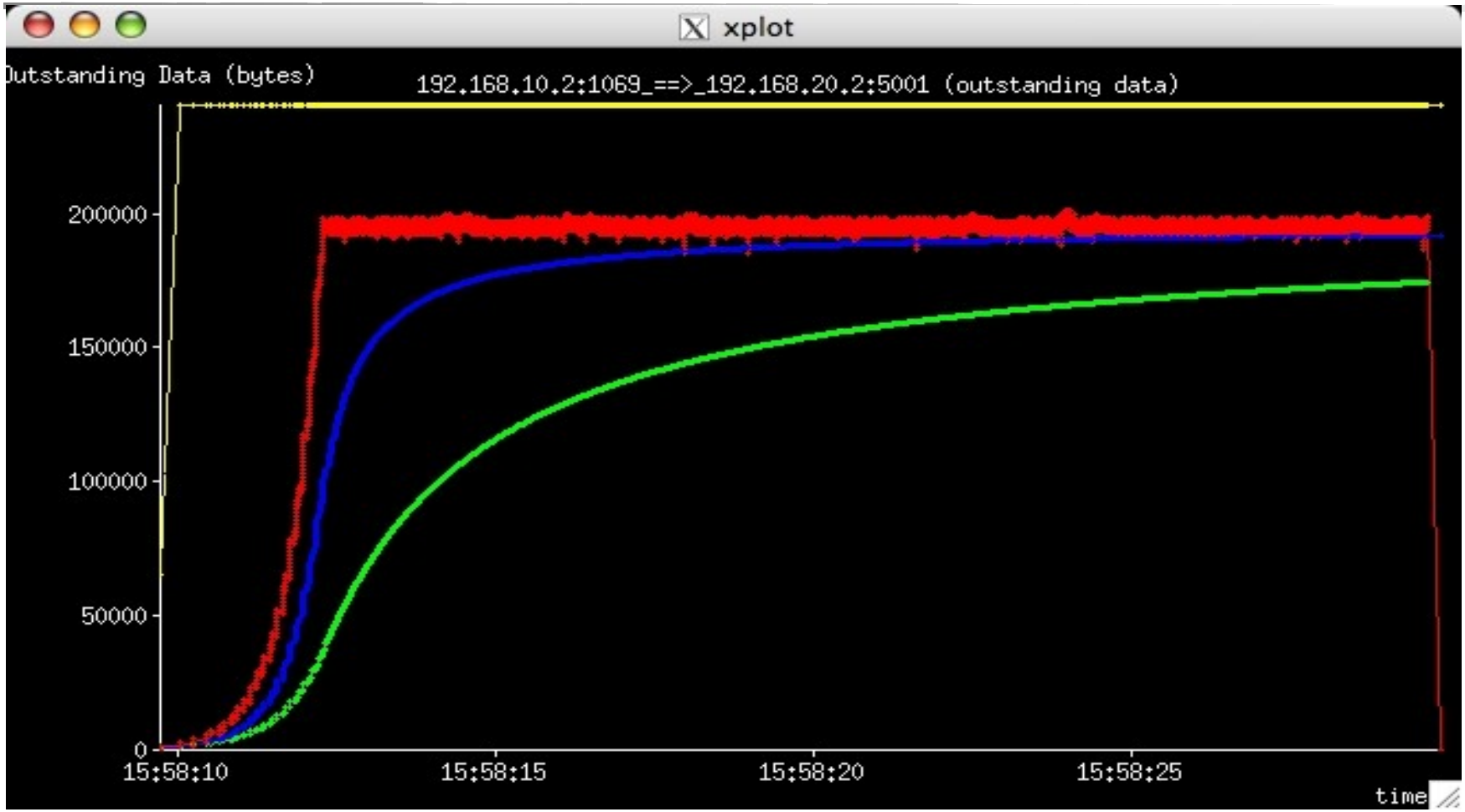


RTT of this path is less than 10ms! Essentially unusable service.

Scp of X Consortium archives from my house to expo.x.org.

The periods of “good behavior” are when I suspended the copy to get work done.

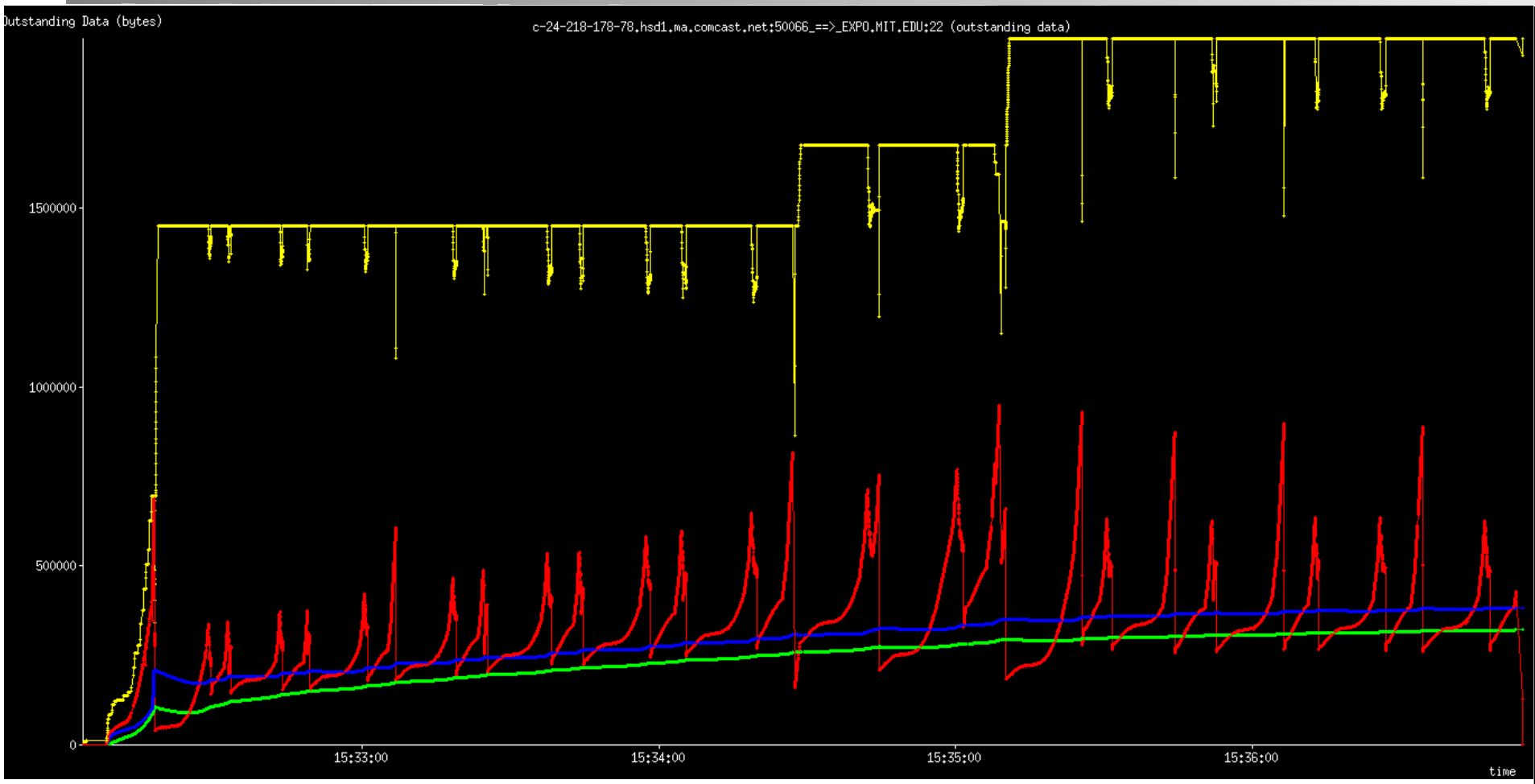
“Typical” tcptrace plot of a TCP session



Yellow: Receive Window
Red: Instant outstanding data

Blue: average outstanding data
Green: weighted ave. outstanding data

Half a megabyte in flight over a 10ms path??? Spikes???



Yellow: Receive Window
Red: Instant outstanding data
Blue: average outstanding data
Green: weighted ave. outstanding data

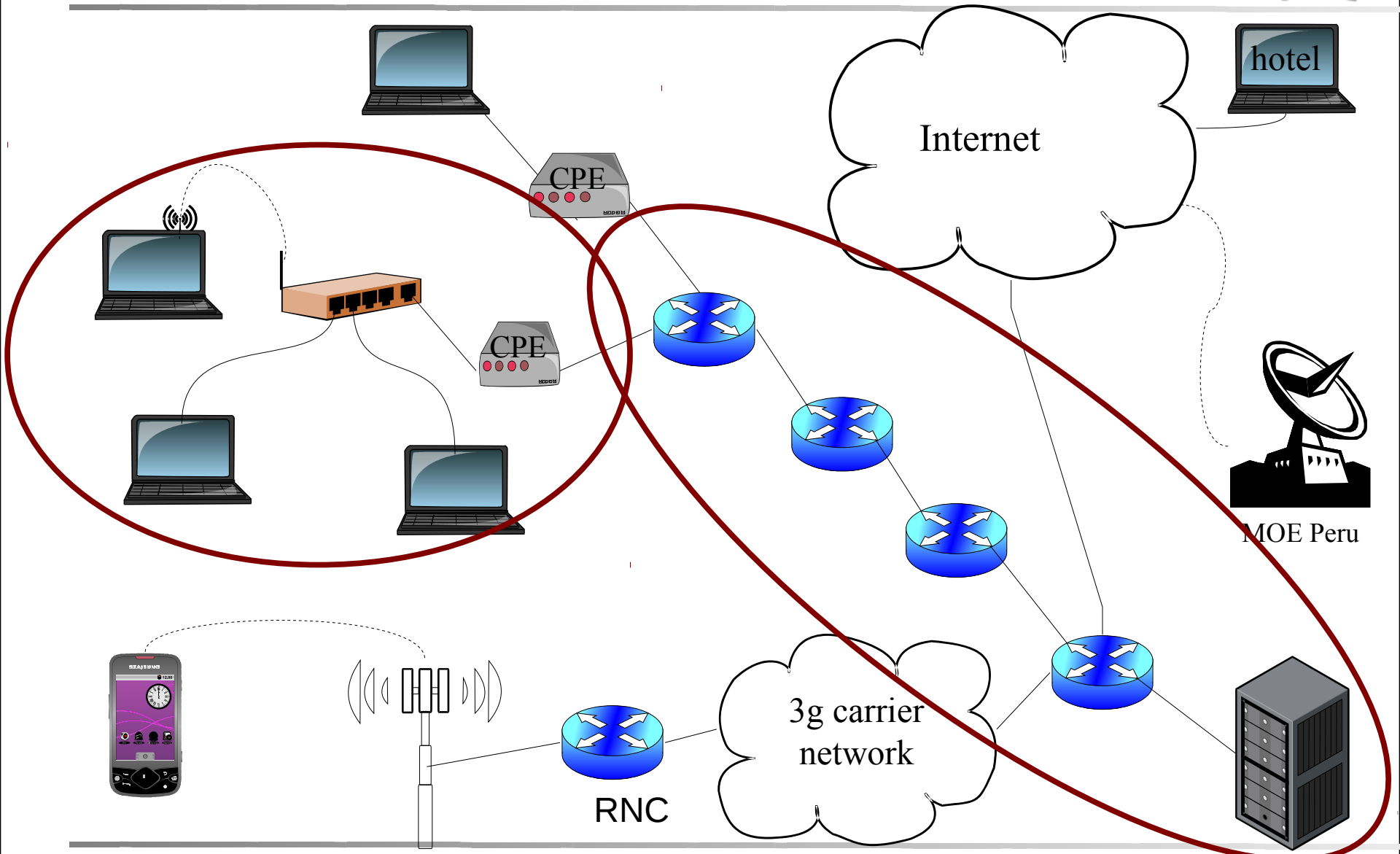
Looks like no TCP trace I've ever seen....

Bursts of duplicate acks; bursts of: retransmits; lots of SACK's; excessive packet drops – on long timescales

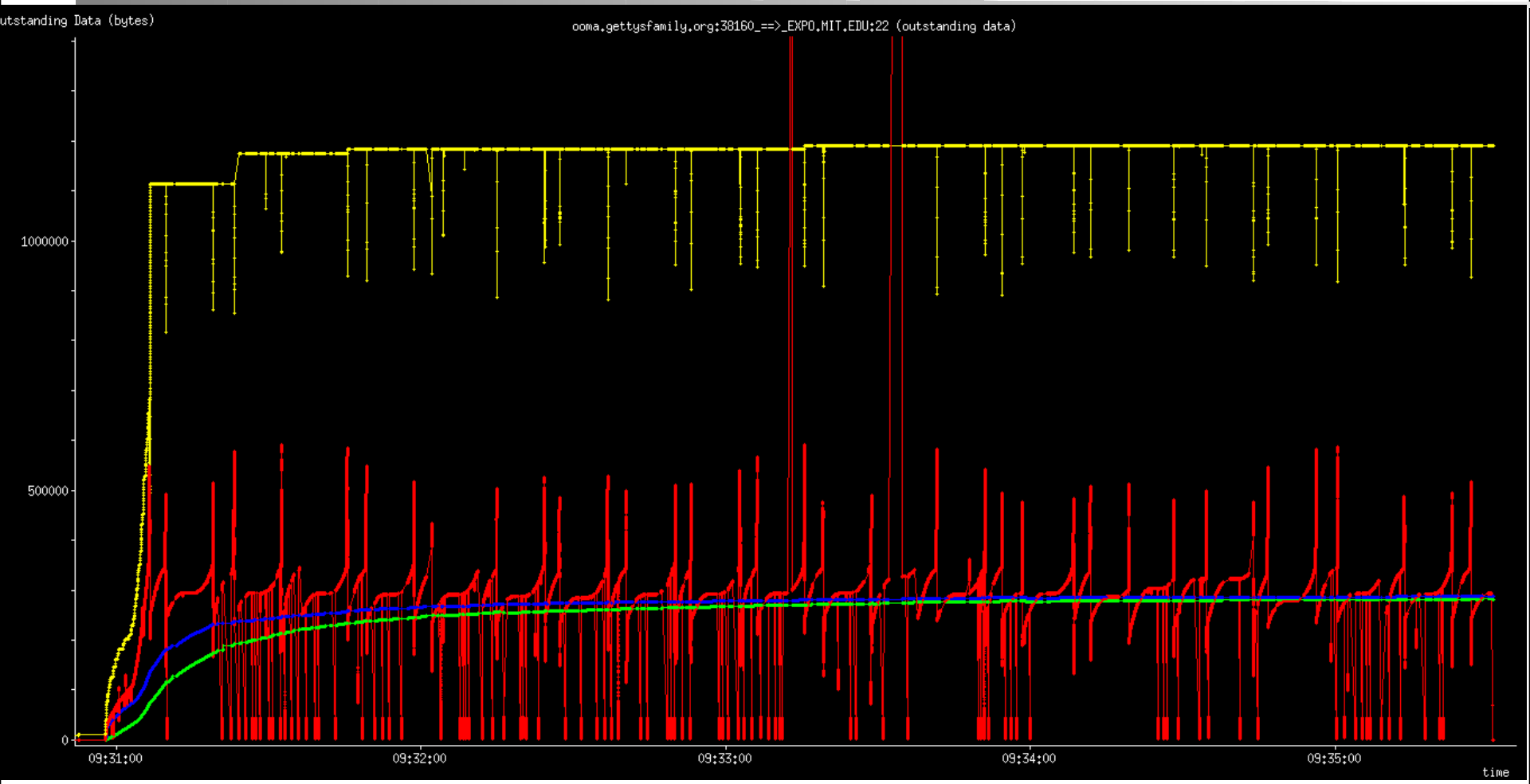
Cable move at my house due to lightning means that technician tests JG's cable at home end, and Comcast has technicians check my cable at CMTS end.

Nominal cable (once my interior TV wiring was removed, anyway); as good as Cable service ever gets outside of a lab

The Plot Thickens. My Inlaw's FIOS service.

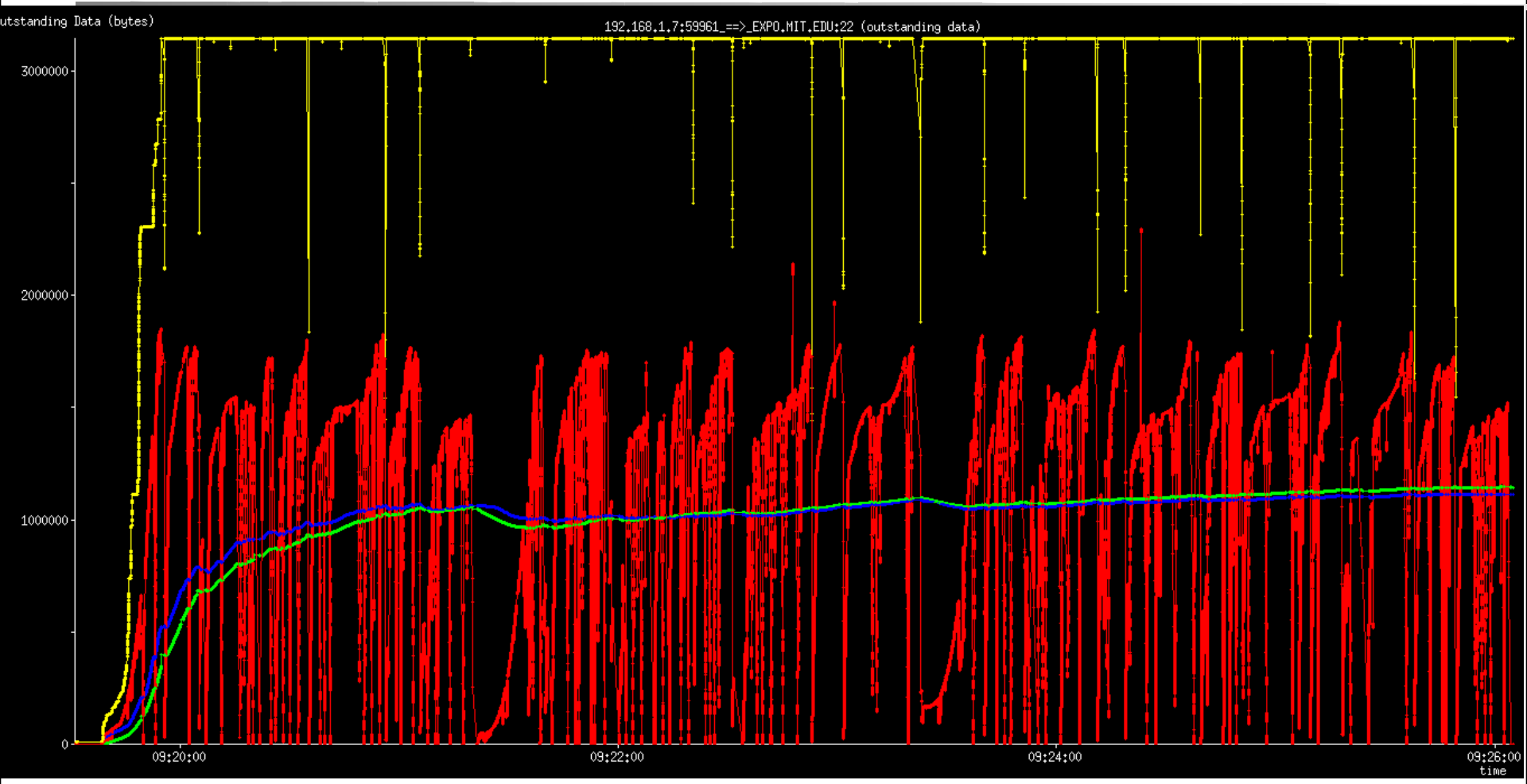


The Plot Thickens. My Inlaw's FIOS service.



My inlaws wired FIOS service in Summit, New Jersey, 25/25 service, 7/30/2010
1/4Mbyte outstanding data on a 20ms path? 200Ms latency?

And thickens. Wireless side of the FIOS home router



Wireless side of inlaw's FIOS service, Verizon provided router, > 400ms, 9% loss

I called the fire chiefs/consulting detectives for help

Some of you may be real TCP experts; I'm not...

- Dave Clark, Dave Reed, Vern Paxon and Dick Sites have all looked the traces over, and agree with the conclusions
- Van Jacobson says there are timestamps in my data which proves the case for bufferbloat, since both ends were recent enough Linux systems and implement TCP timestamps by default
- What is more, the bloated buffers are *defeating congestion avoidance* since the buffers do not allow ***timely*** notification. Slow start is also overshooting badly.

Traffic classification (QOS) can not help you. These are stupid devices. There is only a single queue, much less any fair queuing. Even if they did classify, that would only move where and when the pain occurs.

Triggers – saturation of the path

Uploads and downloads – uploads are often be worse than downloads due to details of hardware and asymmetric provisioning, but some examples include...

- YouTube uploads, Crash dump uploads
- Email with large attachments
- BitTorrent
- File copies/backup; CIFS (Samba)
- TCP based IPSEC tunnels can totally lose
- Downloads of movies to disk
- Video teleconferencing
- Web browsing image heavy sites as YouTube, Google Images, etc. causes transient bufferbloat

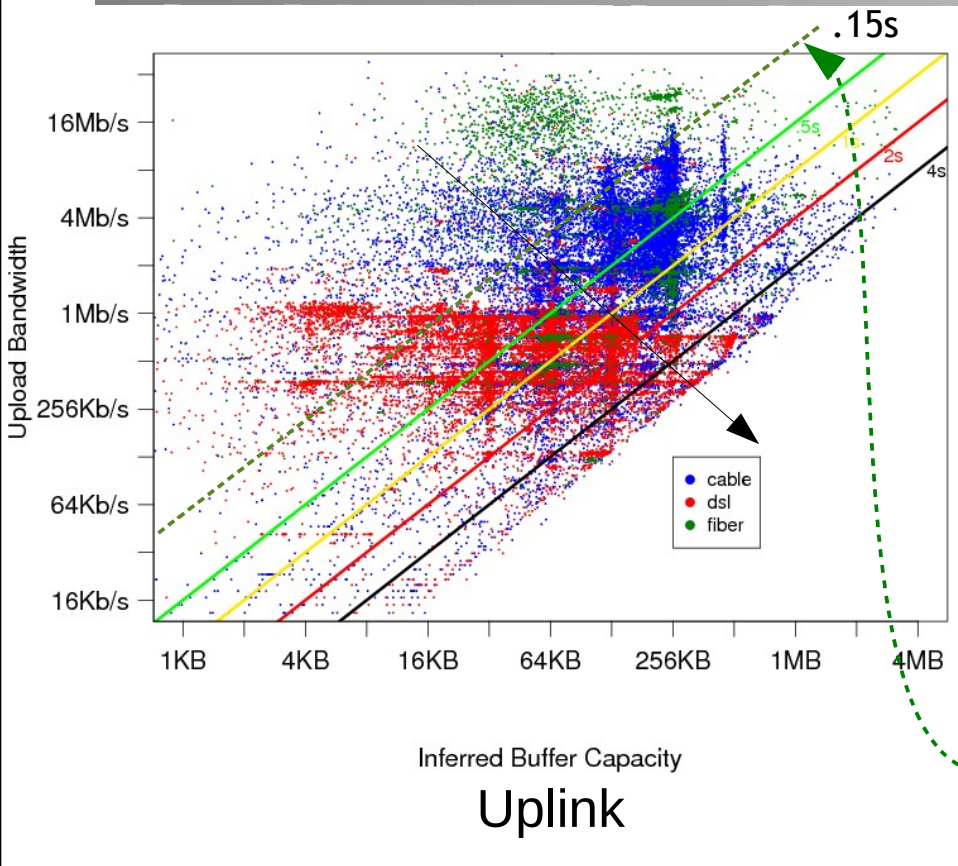
Protocols fail due to both packet loss & high latency timeouts

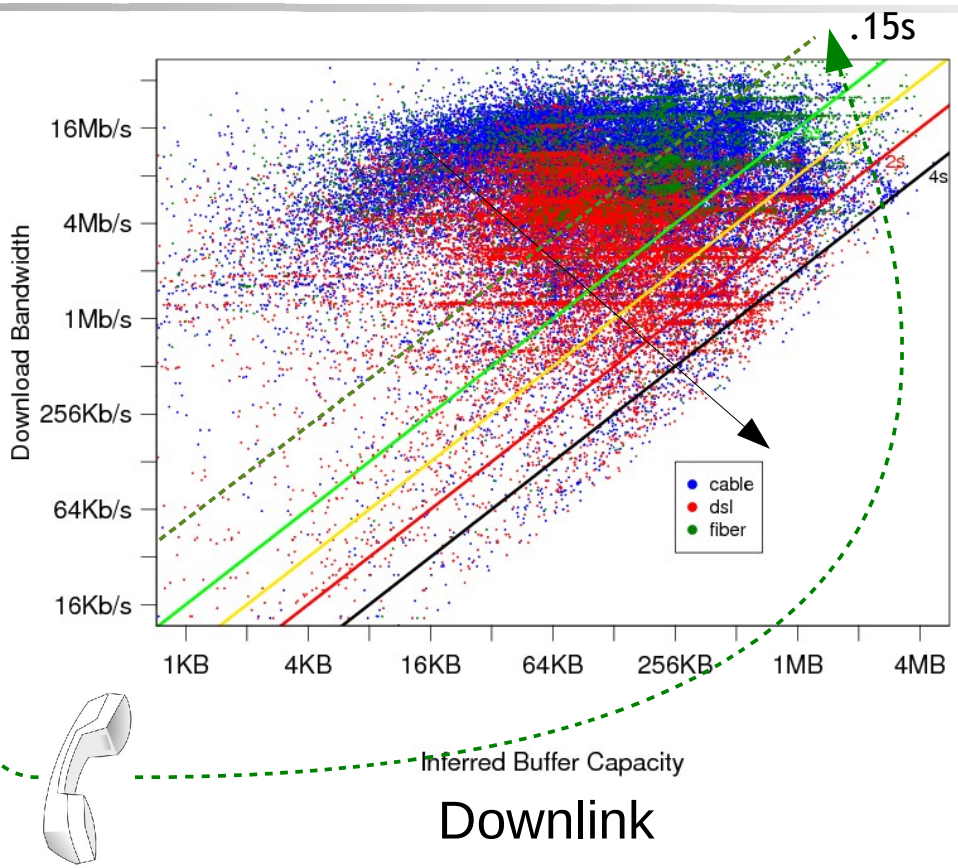
Once a network/link exhibits high latency and bad packet loss, statistically insignificant but mission critical packets can't do their jobs

- DNS – adding 100's ms or seconds of latency to lookups **kills** web browser performance and losses cause lookup failures
- ARP - relies on timely resolution to find other devices
- DHCP - if these packets are lost or excessively delayed, machines can't get on the network
- RA and ND - essential for Ipv6 functioning
- VOIP - needs about a single packet per 10ms flow in order to be good, and less than 30ms jitter.
- Gamers - will get fragged more often with latencies above their twitch factor
- Responsiveness of all applications, web or otherwise, suffers

What protocols do you run that will die with timeouts & when encountering excessive packet loss?

"Netalyzr: Illuminating Edge Network Neutrality, Security, and Performance"
C. Kreibich, N. Weaver, B. Nechaev, and V. Paxson





Arrow direction is increasing latency

This data is a *lower* bound on the severity of the broadband bufferbloat problem; there was a bug in netalyzr causing it to sometimes fail to fill buffers and therefore detect buffer sizes

This data mixes wireless and wired traffic and is contaminated with host

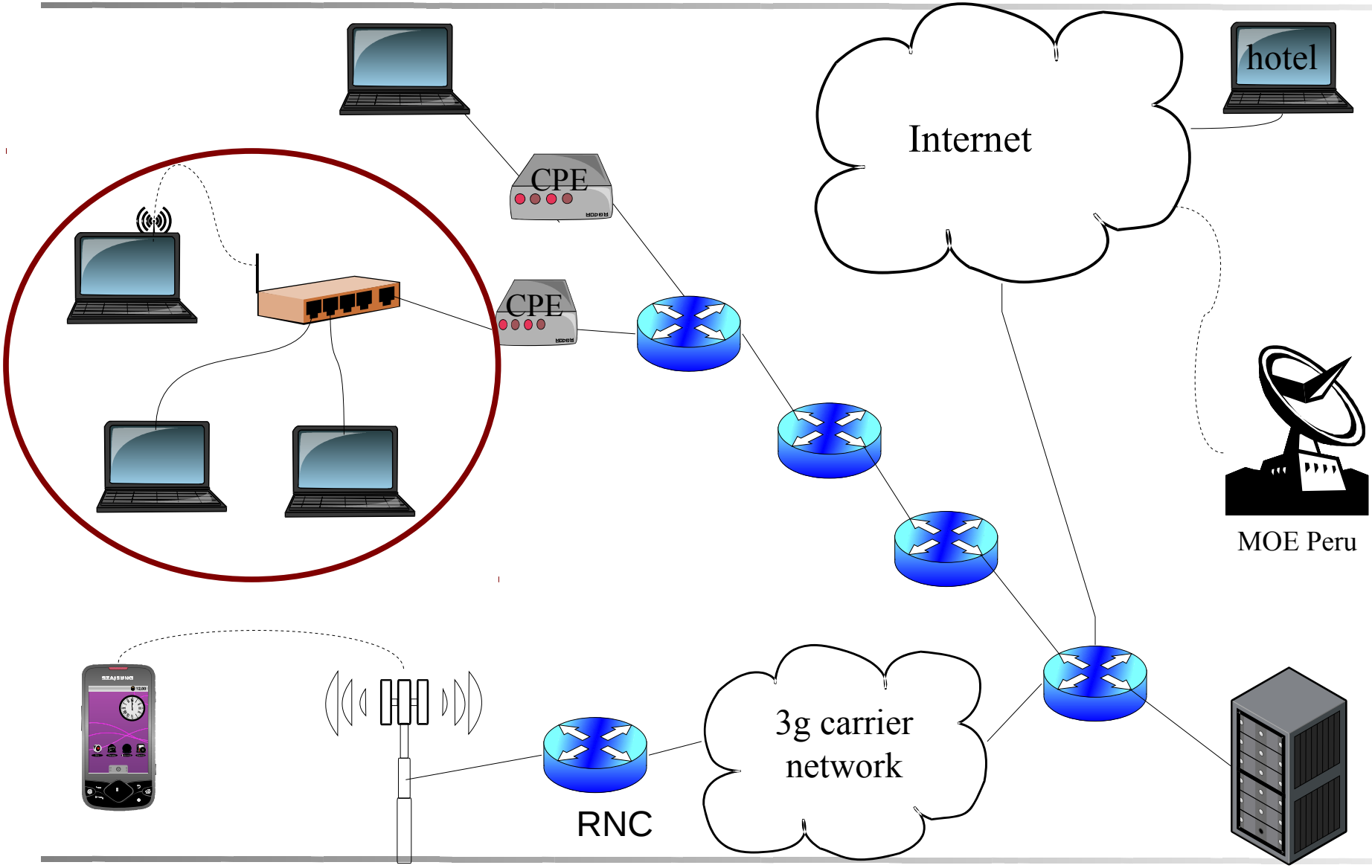
Complete Paranoia Sets In

ICSI has proven the broadband edge is broken

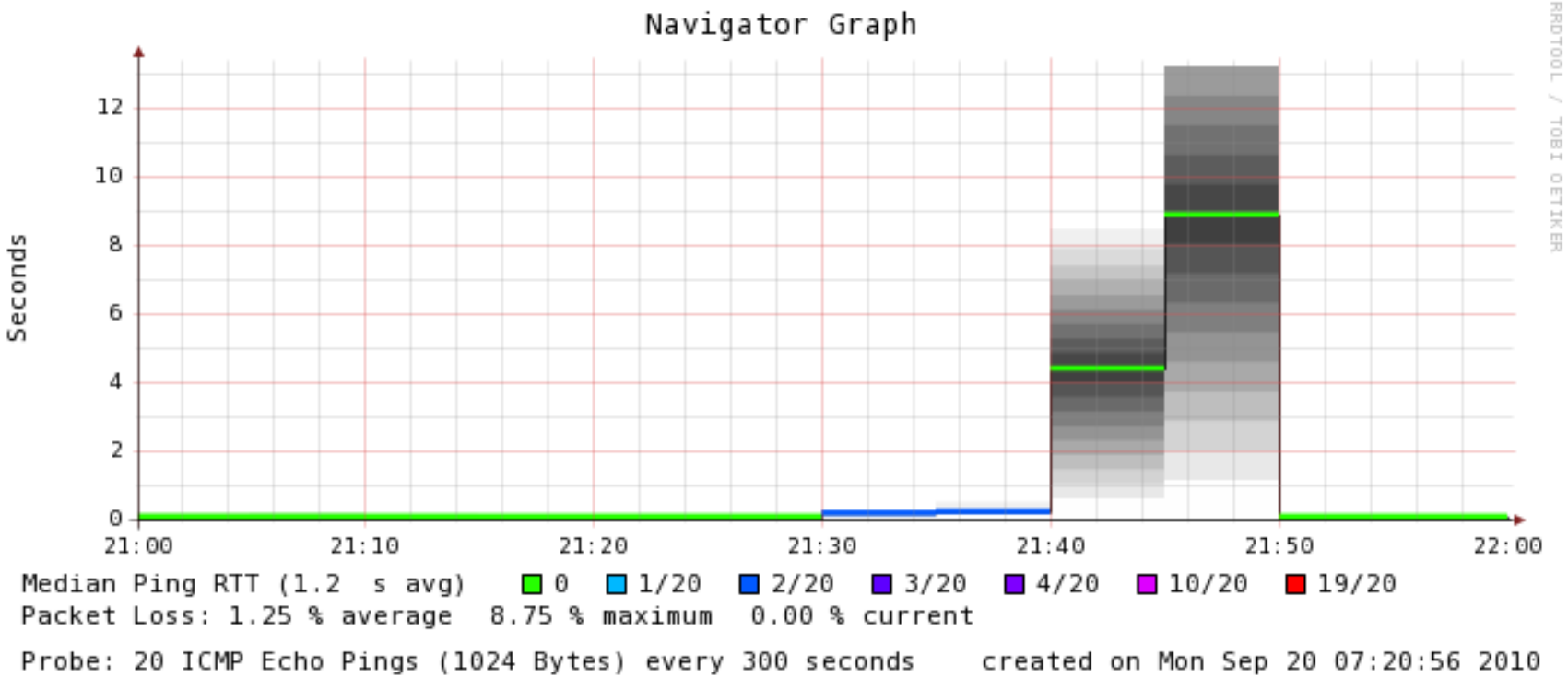
But...

I think bufferbloat hides nearly everywhere!

The Plot Thickens. Home router experiments.



During my tests, I thought I caught my home router doing terrible things!



8 second latency over a path of less than 10ms! Much worse than my broadband only test

scp of files from my house to expo.x.org, while running speedtest.net

Recent commercial home router, 50/10 Comcast service, 9/10/2010

Home router bufferbloat led me to host bufferbloat

Eight second latencies? Insanity... What is going on?

- I replicated this problem on several modern home routers

I installed OpenWRT on a router to understand

- I had realized that Linux's transmit queue might be a problem
- I set the txqueuelen knob to zero on the router, but nothing happened...
- More hair lost, and then I realized: the bloat is on my ***laptop*** in the upload direction! And on the ***home router*** in the download direction!

__Anytime__ broadband's bandwidth exceeds wireless transfer rates at home, the bottleneck becomes the device/home router hop.

We must fix the operating systems and home routers too!

Host bufferbloat, and your home router

Since home routers are usually using general purpose operating systems under the covers (e.g. Linux), bufferbloat is on both sides of your wireless link

Buffers hide in multiple places in modern OS's + hardware (*Linux*, *MacIntosh*, *Windows alike*) – more about this later

Let's do a simple calculation, presuming 10Mbps:

- 256 packets is of order 3,000,000 bits. So here's 1/3 of a second (one way)

What happens at a busy conference, where your “fair share” might be 100Kbps?

30 seconds: applications (and users) timeout entirely....

And your excessive packet loss rate induced by bufferbloat is increasing failures further

Why don't I see bufferbloat on ethernet?

You **DO** see bufferbloat on Mac OSX and Linux on 100Mbps ethernet:

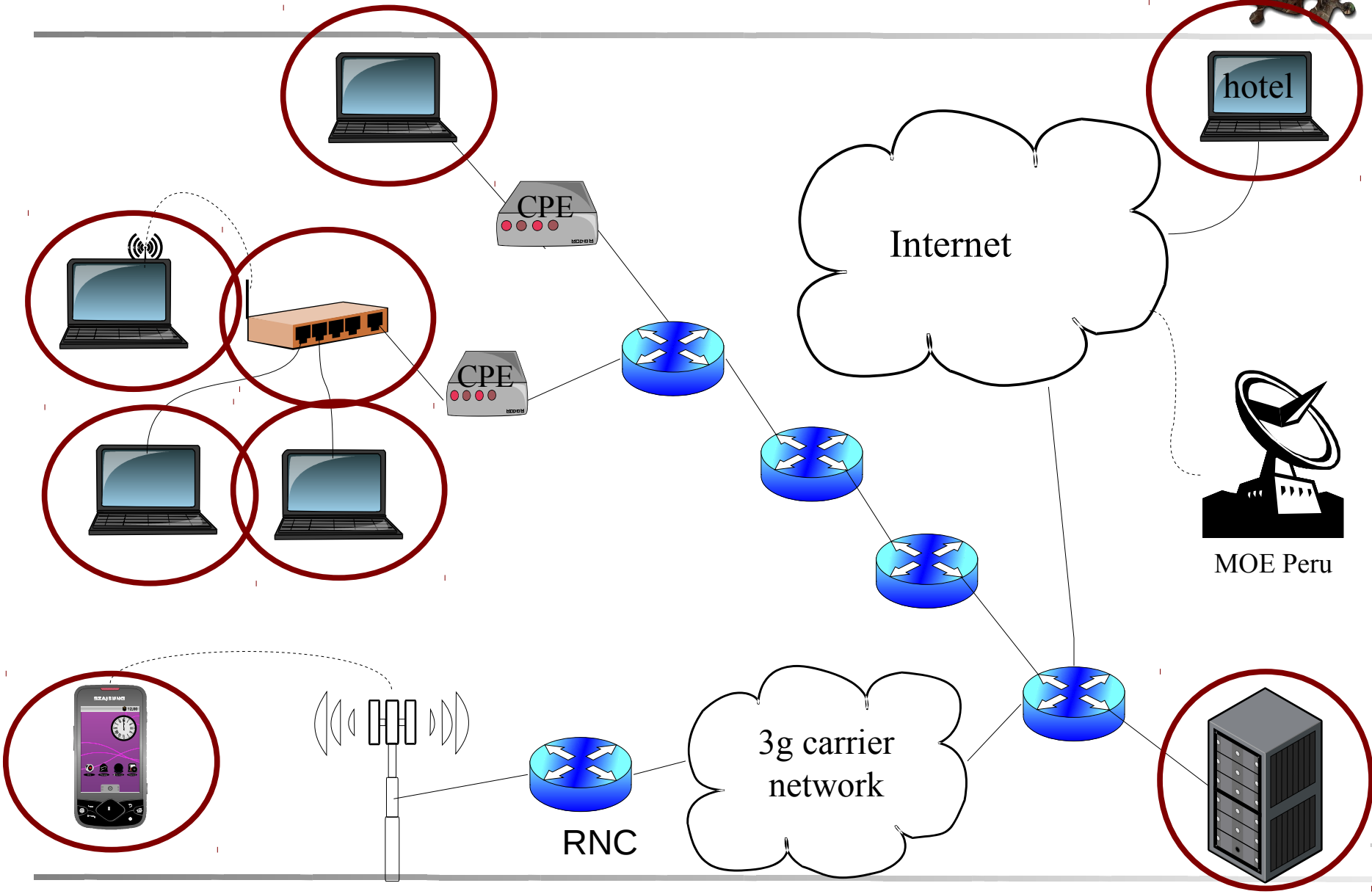
- Use a test program capable of saturating the connection....
- The network is slower than 1Gbps NIC, buffers build on your host
- The driver ring buffers are about 256 packets in size on modern hardware: you get ~10ms of latency with simple file copies

Windows is “interesting”... you get good latency, relative to Mac OSX and Linux

- Any version of Windows defaults to about 85Mbps; so the wire is faster than the OS, and therefore the OS buffers never fill
- A Microsoft tech note explains that to get better multimedia experience, Windows bandwidth shapes outgoing traffic

Ethernet switch chips have >=1Mbit of SRAM....?....?

Where does host bufferbloat hurt?



Aggregate network behavior – Back to the Future

What happens if a network has buffers “all over”? Such as any wired network without AQM enabled in its routers... Classic congestion of the 1990's... Or wireless:

- Base station protocol adaptation, for error correction
- Static buffers used to cover radio bandwidth variation
- Wireless protocols themselves (802.11, 3g)

Buffers start working in an aggregate fashion

Latency will go up when loaded; but you won't observe much packet drop - Expect a diurnal (daily) pattern: people timeout before packets do

- This is what Dave Reed reported in 2009 on 3G networks
- I've observed up to 6 second latency: others up to 60 seconds
- Bufferbloat is now known to exist in some (all?) RNC's, and observed in back haul networks
- Back-haul networks are also failing to run RED or other AQM

Conversations with Van Jacobson

Over ten years ago, Kathie Nichols walked into Van's office one afternoon, and showed him that RED has two bugs

Van & Kathie failed to get the paper with RED's problems and fixes published (twice).

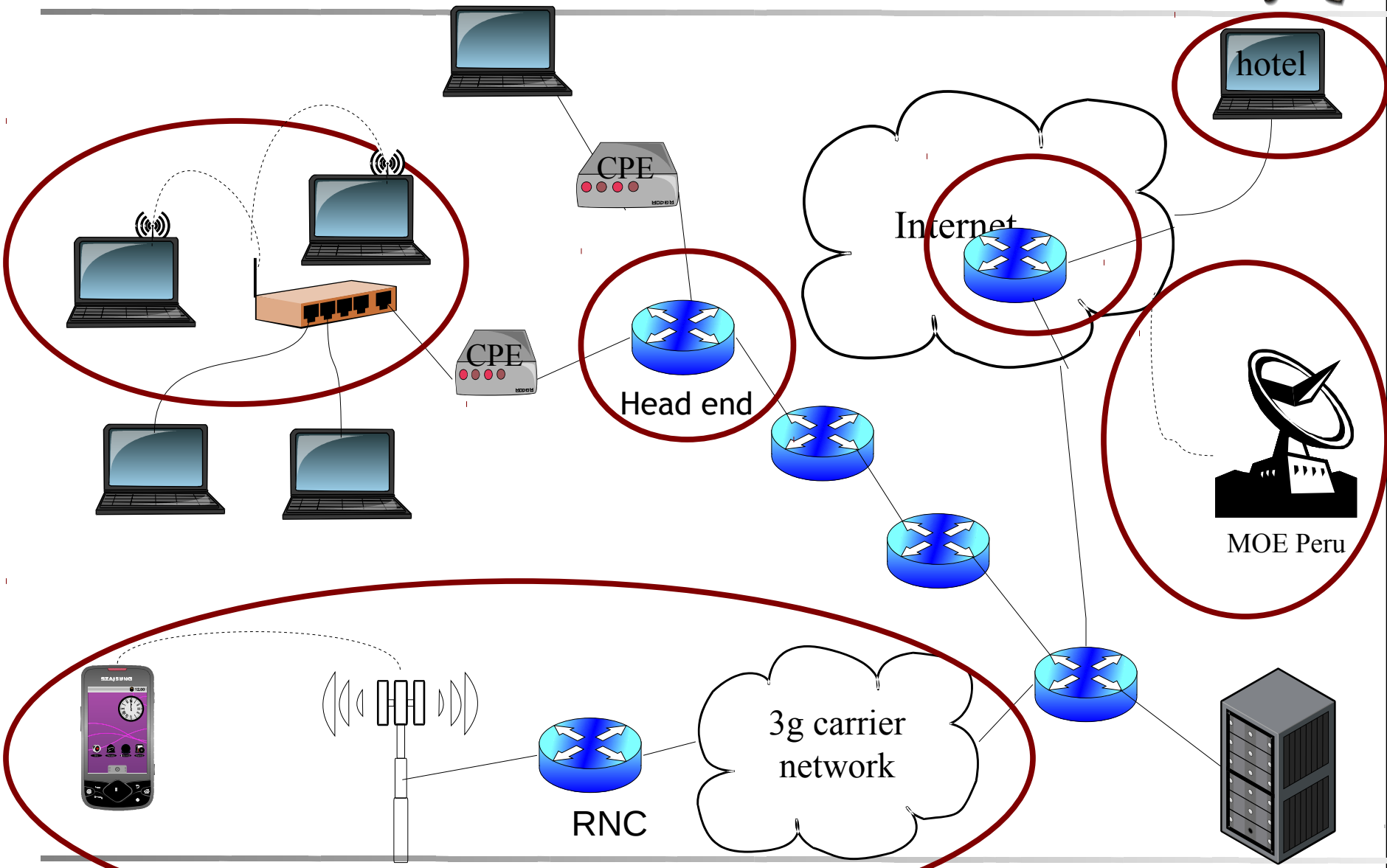
Van says RED has several problems; these problems mean that RED requires tuning, and the 100 or more papers about RED tuning in the last decade confirm this.

Network operator's reluctance to enable RED is understandable, even if their fears are usually excessive: RED must be used carefully!

A 1999 draft of RED in a Different Light draft did escape

Van warns that time based behavior in Internet gear could also cause congestion and instability. Recent broadband technologies have this property. Bunching of initially well paced packets has been observed already. Synchronization and global resonance does occur!

Aggregate bufferbloat locations



Fat Subnets: e.g. 802.11 or many other wireless technologies

Concrete example, 25 802.11 nodes, each with a single packet buffer, trying to transmit to an access point.

- Some nodes are far away, and the AP adapts down to, say 2Mbps?
- You have 25 * 1500 bytes of buffering; this is > .15 seconds excluding any overhead, if **everything** goes well.

What happens if:

- You buffer 20 packets instead of 1 on each node? 200? or 1000?
- You keep trying to retransmit packets in the name of “reliability”? (some MAC's try to transmit up to 255 times; 8 times in common)
- And, in the name of “reliability”, an inherently unreliable multicast/broadcast medium drops the radio bandwidth to minimum?
- You then try to run WDS or 802.11s, which both forward packets and/or respond to any multicast (e.g. ARP) with routing messages?

Result? An ugly, complicated sight

Conferences using 802.11
Schools
Hotels
Some network operator's networks

Example failure: OLPC's network melted under load

- In Mongolia the teachers have the children all open their laptops at the same minute in the morning?
- Net result: no packets would get through
- Our applications failed due to timeouts before meltdown

We had several problems including mesh routing problems; but we missed bufferbloat **entirely** in our failure analysis

Buffers are **only** detectable when they are **next** to a saturated bottleneck – at other times they are "dark"

Your hosts, your apps, and in socket buffers and network layers

- Your MAC itself may have packet buffers internally and Network device drivers themselves
- Your network interface's ring buffer potentially buffers large numbers of packets, often put there to hide x86 SMM (system management mode) behavior and for marketing
- And a VM system may add yet more layers

Your wireless access, at both ends

- 3g has buffers for fragmentation reassembly
- 802.11 has similar issues: long packet delays destroy timely notification

Your switch fabric: how many hops, how congested?

Your home router – potentially megabytes

Your CPE/cable modem/FIOS box – potentially megabytes

The head-ends of those connections (e.g. DSLAM, CMTS, etc.)

Every router in your path, and the line cards in those routers

Speed of light in glass and vacuum

Other Locations of Bufferbloat

Corporate networks

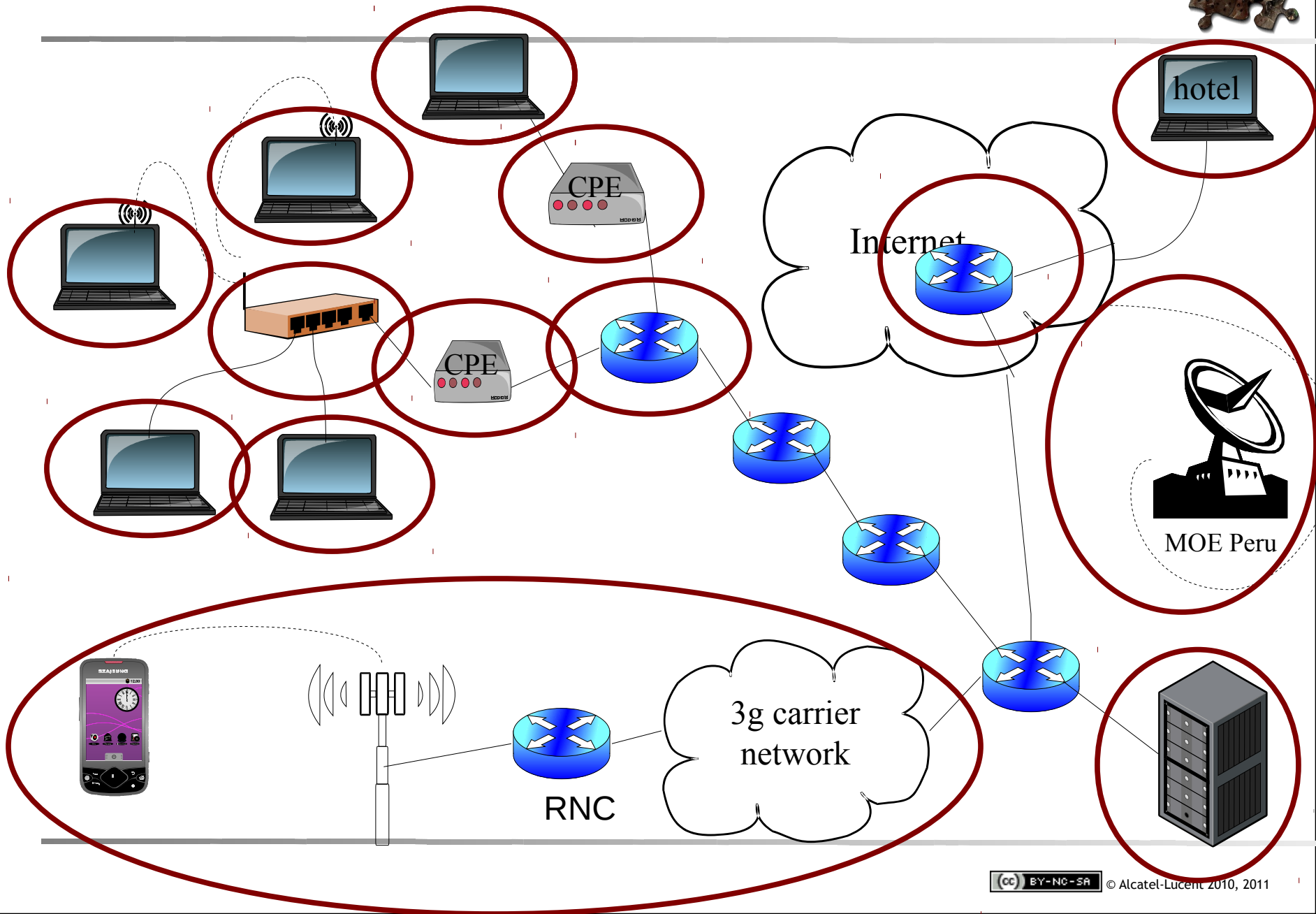
- Smokeping made me suspect my corporate network wasn't right: latency spikes were present
- ALU corporate network example: sophisticated classification is present, but no AQM enabled; as we convert to Windows 7 and other systems, this becomes an immediate problem, as all hosts will be able to saturate **all** network links

Example Middlebox

- Tunnel devices
  - Detected in my company's IPSEC infrastructure, at the firewall complex where the tunnels land
- Firewall relays?
  - If not in the OS, then in the relay applications

Elsewhere? Example: encryption buffers

Where do bloated buffers hide?



Mitigation/solution depends buffer location

High-end home routers sometimes have QOS knobs that can immediately partially mitigate the damage

- Can't deal with dynamic bandwidth (e.g. TurboBoost, TOD)
- Can't deal with upstream buffers in devices like DSLAM or CMTS's

In networks not running with AQM, “just” turn it on

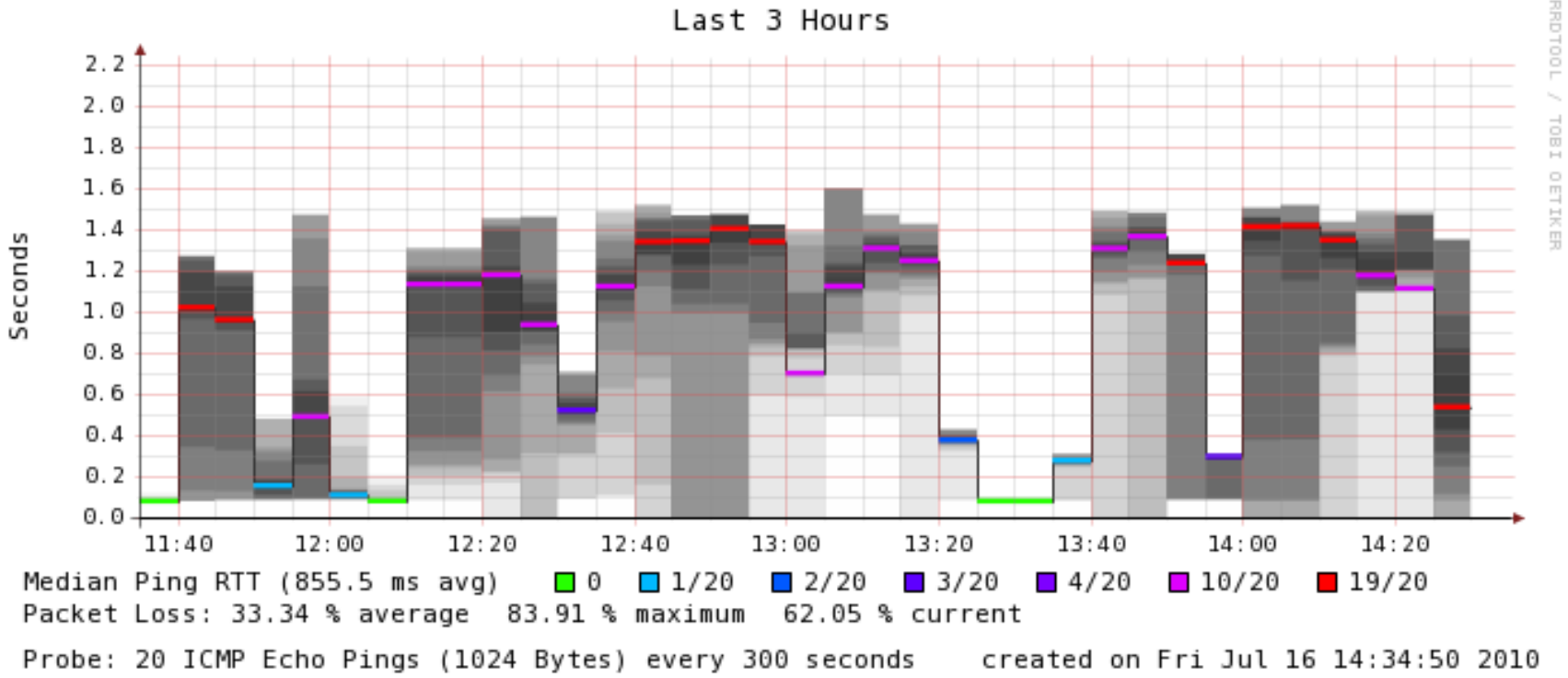
- Education may be hard in some organizations – RED has its problems, but should be used where it can be

Improved AQM algorithms need research and deployment

Hosts need both immediate mitigation and better queue management solutions

Classic RED 1993 won't work to solve 802.11 or 3g bufferbloat due to wireless's highly variable bandwidth: SFB, RED Light or other algorithms need testing and deployment

dslreports smokeping service observing my home link – **Before**

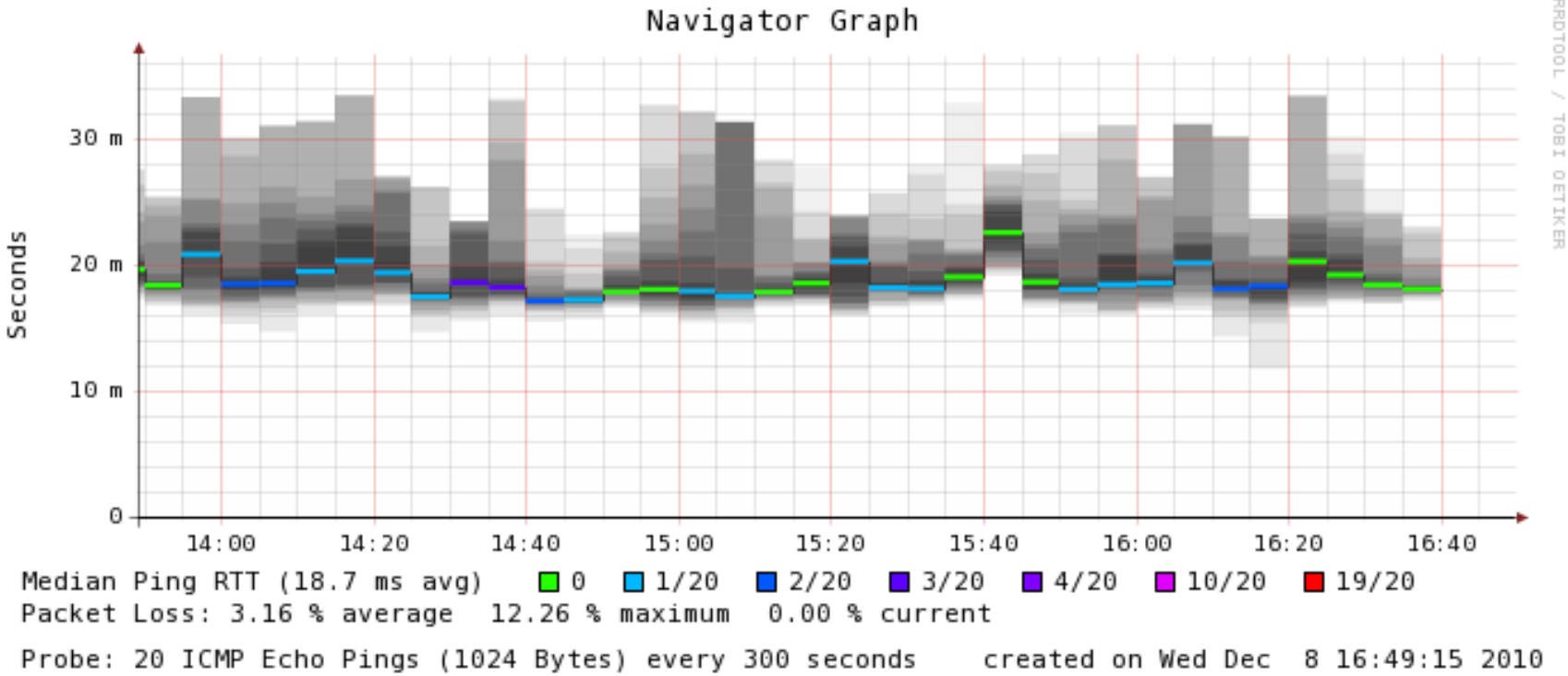


RTT of this path is less than 10ms!

Scp of X Consortium archives from my house to expo.x.org.

The periods of “good behavior” are when I suspended the copy to get work done.

dslreports smokeping service observing my home link – After



QOS knobs in router set empirically – stable latency at all loads

From > 1 second, to less than 10 milliseconds jitter and latency – 100 times better!

Network Neutrality & Bufferbloat

I believe (part of) Bittorrent's problems were incorrectly diagnosed.

- Buffers were already grossly excessive when bittorrent deployed, and uplinks were much slower (typically 384Kbps)
- Windows XP does not enable TCP Window scaling, so it was not obvious what was happening

Carrier's telephony currently has a major quality advantage over SIP VOIP or Skype

- I do not believe it intentional: *ISP's* get the service calls due to bufferbloat and everyone who builds equipment has been making the same mistake

Innovation is at risk

- *Reliable* low latency applications (games, hosted desktops, immersive teleconferencing) have become impossible to deploy

Fixing bufferbloat is essential to innovation in the Internet

The situation is getting worse

Windows XP is retiring at long last: all other platforms routinely saturates **any** link with even a single TCP connection as they enable the TCP Window Scaling option

The application mix is shifting toward saturating applications: e.g. video upload/download, backup, BitTorrent, email with large attachments, etc

The web server/client changes are deploying

The hardware has been getting more bloated every generation

The situation will be worse before it gets better

What should network operators do?

If you don't know latency under load (buffersize at a bandwidth) for each device you use, you should...

- By ensuring the bloated buffers are in **not** the bottleneck link you may be able to make bufferbloat harmless (dark buffers)

Push back on your suppliers: bandwidth specs without latency specs are useless. Push back on your customers: CLA's specifying zero packet loss spec are broken.

Tools & Tests

- The tests are easy: test latency under load

Mitigations:

- Buffering can never be static: it must be in relationship to bandwidth: e.g. the DOCSIS changes underway to control the buffers in cable modems

Think beyond “layers”: moving packets interacts with transport protocols. Timely packet drop or ECN marking is ***necessary*** for TCP to react properly!

What should network operators do?

Take RFC 2309 to heart: the RED Manifesto

"*Characterizing Residential Broadband Networks*" by Dischinger et. al., IMC 2007: section 4.3 reports many operators do not run AQM who should do so

Full solution of bufferbloat will involve:

- AQM everywhere (and better AQM algorithms)
- ECN
- Fair queuing and diffserv at the edge
- Get the operating systems fixed: they make you look bad when it is their fault rather than the network

Remember: you take in in the ear or the inbox even when it's someone else's bloat!

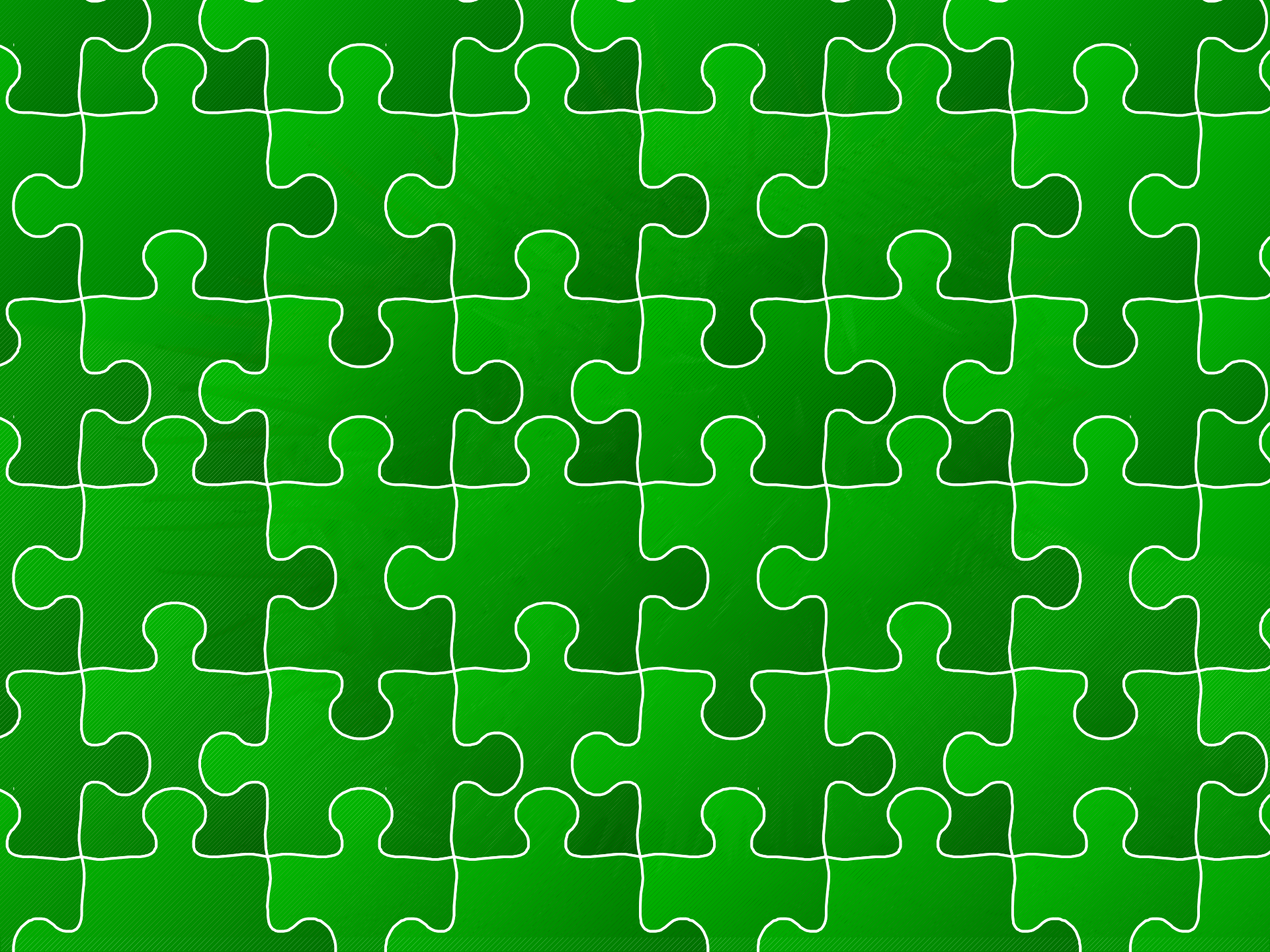
NANOG has given me hope!

The first time I've ever seen a “green” netalyzr result **anywhere** in the world is at NANOG52.

Testing latency upstream under load from my laptop using eBDP and a decent txqueuelen I retained good latency.

Thank you!

Now on to the rest of the planet....



Questions?

Remember, we are all in this bloat together!

Please come help before we sink!

My Blog – http://gettys.wordpress.com

Bufferbloat.net – http://bufferbloat.net

http://lists.bufferbloat.net

http://trac.tools.ietf.org/group/irtf/trac/wiki/ICCRG

This talk – http://mirrors.bufferbloat.net/Talks/NANOG11

Longer YouTube Talk video

http://www.youtube.com/watch?v=qblozKVz73g