Alcatel·Lucent

Bufferbloat

Dark Buffers in the Internet

Why You Should Care

Jim Gettys

Bell Labs

August 31, 2012

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

Common Laments of the 2012 Era Internet....

"Daddy, the Internet is Slow Today!"

"Junior, stop what your are doing on your computer so I can make a phone call!!!"

"Boy, this conference's wireless network was working fine before everyone sat down, but now it's horrible!"

"My cell phone's 3g network is incredibly slow here, though my signal strength is good!"

"Sorry, I didn't understand what you said, Skype/Vonage is having problems right now and you dropped out for an instant!"

"My home network is useless whenever I backup my computer!"

"This motel's network is unusable; even Google Search is glacial!"

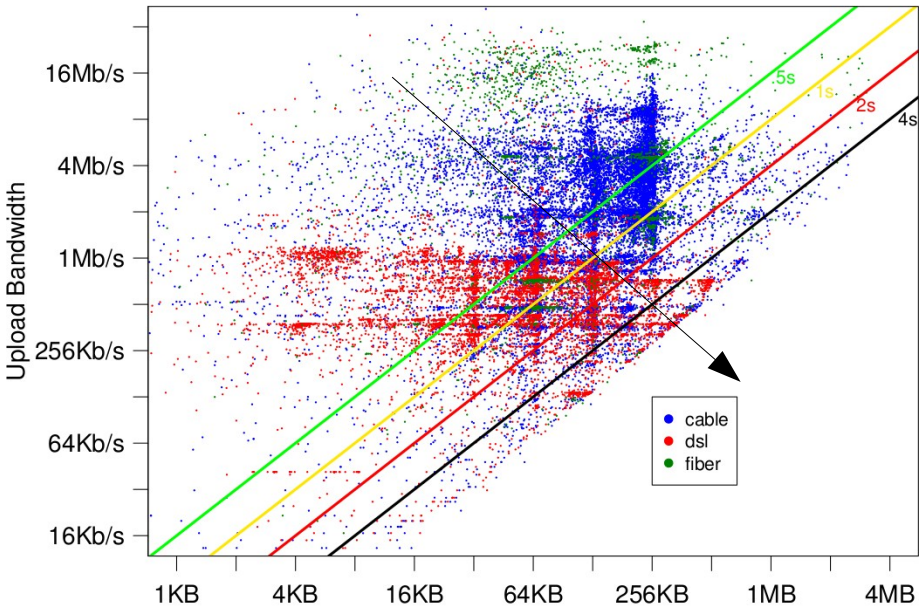
What do all of these laments have in common?

Bufferbloat!

Demonstration video

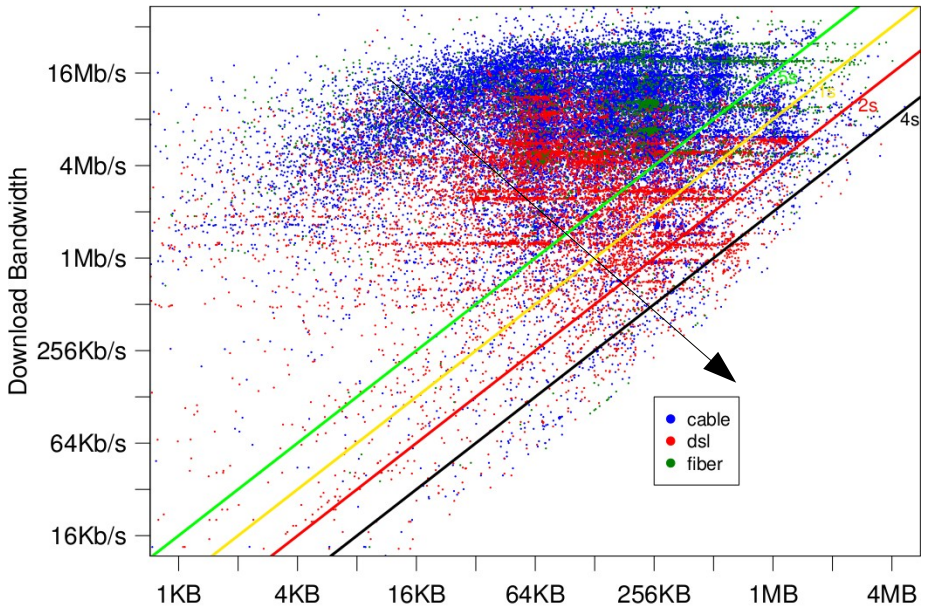
http://www.youtube.com/watch?v=npiG7EBzHOU

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



Green diagonal line == .5 second latency

Uplink



black diagonal line == 4 second latency

Downlink

Arrow direction is increasing latency

Note: telephony standards for latency are maximum of 150ms!!!

This data is a *lower* bound on the severity of the broadband bufferbloat problem.

What happens when a network is slow due to bufferbloat? - protocols fail due to both packet loss & high latency timeouts

Once a network/link exhibits high latency and bad packet loss, other critical, 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 on your network
- DHCP - if these packets are lost or excessively delayed, machines can't get on the network
- RA and ND - essential for IPv6 functioning
- VOIP/teleconferencing- 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 network applications, web or otherwise, suffers

Protocols can fail entirely with timeouts & excessive packet loss

Buffers only fill when they are **next** to a saturated bottleneck – at all other times they are “dark”

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

- Your MAC itself may have packet buffers internally;
- Network device drivers themselves
- Your network interface's ring buffer potentially buffers **thousands** of packets
- And the VM system your OS may be running on top of may add yet more layers

Your wireless access, in **both** directions

- Cellular Wireless Systems have major problems: it's why your cell phone may be very slow
- 802.11 has similar issues: long packet delays destroy timely notification

Your switch fabric (8 ms/switch at 1GBPS): 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.)

Each and every router and switch in your path, and the line cards in those routers

Bufferbloat Situation

Buffers only fill before a bottleneck. But those bottlenecks are now routinely next to any wireless device

Hypothesis: most (but not all) bufferbloat locations we personally experience are in the edge: e.g. home & cellular networks

Home routers *and* hosts are at least as bad as broadband

Many problems all over the Internet: the edge is likely the most severe, though it is endemic in hosts, home routers, broadband gear, 3g, some switches, overloaded routers.... Be paranoid!

Reminder: there are *two* bottlenecks are in play in the home

Broadband hop (single bloated queue!)

Wireless hop (potentially four HW queues in 802.11)

Transient Bufferbloat

Web browsers and Web sites are doing ***Evil*** together

- Web browsers no longer limit the number of TCP connections
- Web sites are now often “sharded”: split across a number of different names
- Has destroyed any congestion avoidance when Web surfing

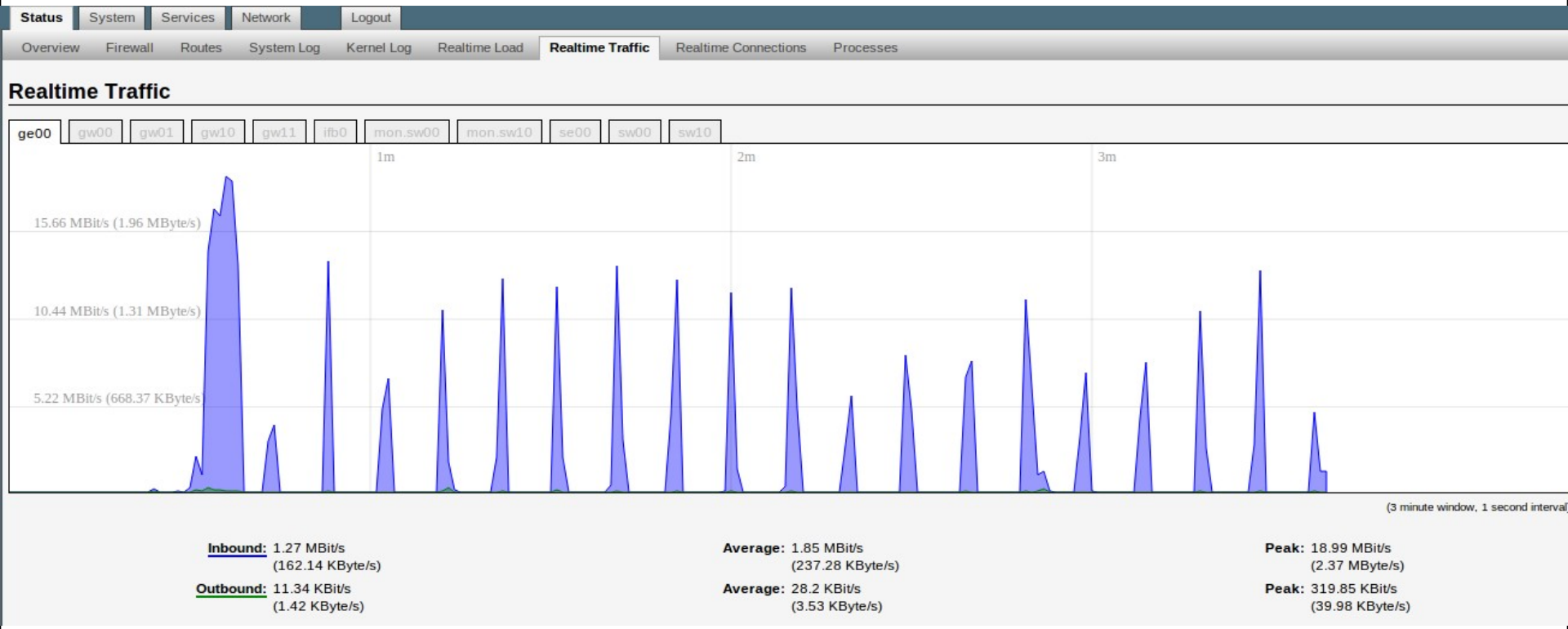
Extreme example

CNN.com-> induces 60 simultaneous TCP connections!

- @ IW 4, that is 240 packets, or 360K bytes (3Mb) in flight to your customer's broadband connection, *which has a single grossly overbuffered so from the server side you see little packet loss!*
- What do you think happens to your customer's VOIP/Skype conversation?

Is this what you want to do to your customer?

Netflix “streaming” Behaviour



Host bufferbloat, and your home router

Since today's home routers are usually using general purpose operating systems (usually Linux), the problem is on both sides of your wireless link

Buffers hide in multiple places in modern OS's + hardware
(*Linux, Macintosh, Windows alike*)

Let's do a simple calculation, presuming 10Mbps actual data transfer rate:

- 256 packets is of order 3,000,000 bits == 1/3 of a second (one way)

What happens at a busy conference, where your “fair share” might be 100Kbps? 30 seconds: applications (and people) timeout entirely....

Fixed size buffering is usually ***nonsense***

Buffering must always be dynamically managed!

Automatic Queue Management is a necessity, not a nice to have!

But wasn't AQM solved in the 1990's?

The classic AQM algorithm is RED, (Floyd and Jacobson, 1993); there are others

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

- Kathie & Van twice tried to publish papers explaining RED's flaws
- A 1999 draft of *RED in a Different Light* draft did escape

RED requires tuning, and the 100 or more papers about RED tuning in the last decade confirm this. Ergo, network operator's reluctance to enable RED is understandable, even if their fears are (usually, but not completely) excessive: RED must be used carefully!

And RED can't work at all in the face of variable bandwidth, such as found in broadband, and wireless

So we need something better: enter the CoDel (“Constant Delay”) algorithm

CoDel Status

See Van Jacobson's presentation at the Vancouver IETF of Kathie Nichols and Van Jacobson's new CoDel (Constant Delay) AQM algorithm: published in the CACM, July, 2012, available on the ACM Queue website: *Controlling Queue Delay*

Linux 3.5 has codel and fq_codel queue disciplines: fq_codel combines CoDel with SFQ: fq_codel by Eric Dumazet, SFQ by Paul McKenney

We really, really, really like fq_codel

- Fair queuing is only 2% of CPU on 10GigE: proof point that smart queuing is feasible on today's systems
- Should replace PFIFO_FAST as the default queue discipline

Work on refining the Codel's design and implementation continues.

There is No Single Bullet for a Low Latency Internet

Multiple queues are essential in broadband equipment and WiFi; but today we have only one bloated queue in broadband!

AQM needed to avoid elephant flows and queues from filling: TCP's responsiveness is *quadratic* in the delay

Smart Queuing is also needed

- "Fair" depends on where you are: I don't mean simply TCP fair queuing but smart queueing among TCP flows, among devices, among customers, among policies; we must become much smarter than dumb FIFO queues
- TCP fair queuing helps RTT fairness, ack compression, interactive versus non-interactive bulk transfers, etc.

Port based Classification & diffserv with multiple queues essential: one 1500 byte packet @ 1Mbps == >13ms

Really Big Headaches

Current broadband has a single bloated queue

- The technologies admit to additional queues, but these are today only available to the ISP's telephony services

How to communication the customer's classification preferences? (At least two possibilities...)

Broadband splits the diffserv domain between the customer & the broadband head end

- A explicit protocol
- Andrew McGregor's idea to infer incoming classification & marking from outgoing marking on flows

Wireless is **hard**!

Bandwidth is extremely variable:

Remember, there can be **multiple** buffers in an system!

BQL (mostly) solves the driver bloat problem for Ethernet

802.11n aggregation requires the driver to have access to many, many packets, underneath Linux's queue disciplines; our system interfaces need rethought; how and where do we run (fq)CoDel?

Before CoDel, we had no hope

Commercial Home Router Disaster

Commercial home routers are broken in 4 major ways

- Firmware is horribly antique and insecure; today's latest commercial home routers usually ships (at least) 5 year old software on new hardware, which seldom if ever is updated once “stable”, which then rots for years after that without update
- Decent IPv6 deployment is now gated by the home routers
- Extreme bufferbloat in all its forms
- Tragedy of the Commons: Funding model of the home router market is broken; there is next to no funding toward engineering to fix problems today: this means that little will happen without community participation

Time to roll up your sleeves and get your hands dirty...
OpenWrt is already years ahead of what you can buy at Best Buy.

In Disaster, There is Opportunity: CeroWrt

CeroWrt is an advanced build of OpenWrt, using WNDR 3700v2 and WNDR3800 routers for more flash, Atheros radios, and fast CPU

Every line of code is available to modify; changes that work go upstream to OpenWrt and Linux as fast as are validated

Today running Linux 3.3.8 release with CoDel, BQL. Running fq_codel on WiFi, which is today only partially effective due to buffering in the drivers due to 802.11n aggregation

Current Bind & DNSsec in chroot jail; dnsmasq also available

Routes, not bridges; 6 networks in the box

Real web server, proxy, IPv6 support, mesh networking, extensive network test tools, etc.....

Come help test, develop, and improve

Demonstrate your heretical ideas with running code!

There is hope! But much work left to do...

You can suffer **much** less at home **immediately**, if you understand bufferbloat

Bufferbloat is now understood to be a serious problem in the technical community, but problems are all over the Internet, from end-to-end

DOCSIS (cable) will improve greatly very soon, with the deployment of a new DOCSIS buffer control amendment, starting market place pressure

We (now, with CoDel) have all the pieces required to build a low latency Internet, but it requires many tools

Linux has made the most progress of any operating system to date with:

- BQL (Byte Queue Limits)
- TCP small queues
- Codel/FQ Codel

But wireless is **hard**, and there is much, much, much more to do.

Remember, we are all in this bloat together!

Please come help before we sink!

My Blog – http://gettys.wordpress.com

Other Information

http://www.bufferbloat.net/projects/bloat