Alcatel·Lucent

Bufferbloat and AQM

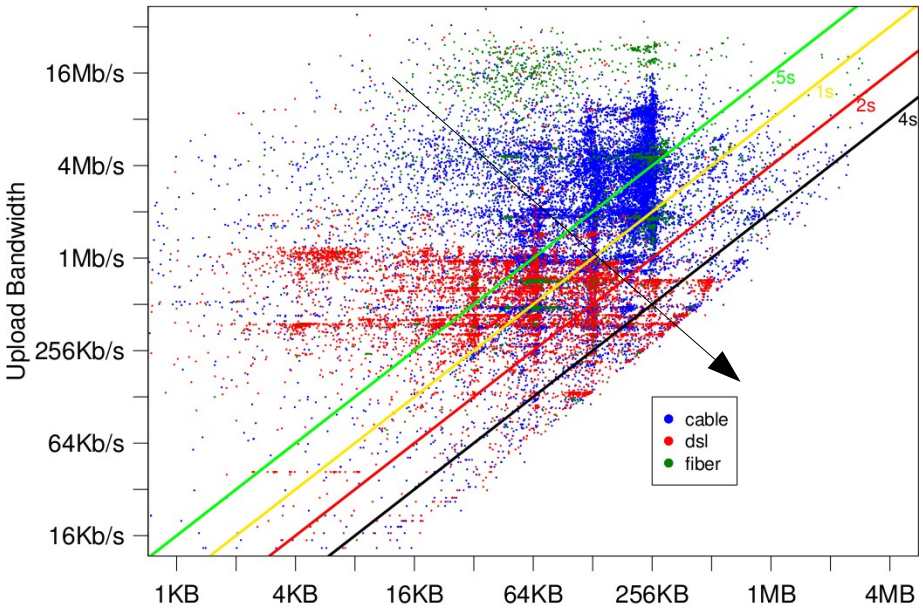
Jim Gettys

Bell Labs

July 29, 2012

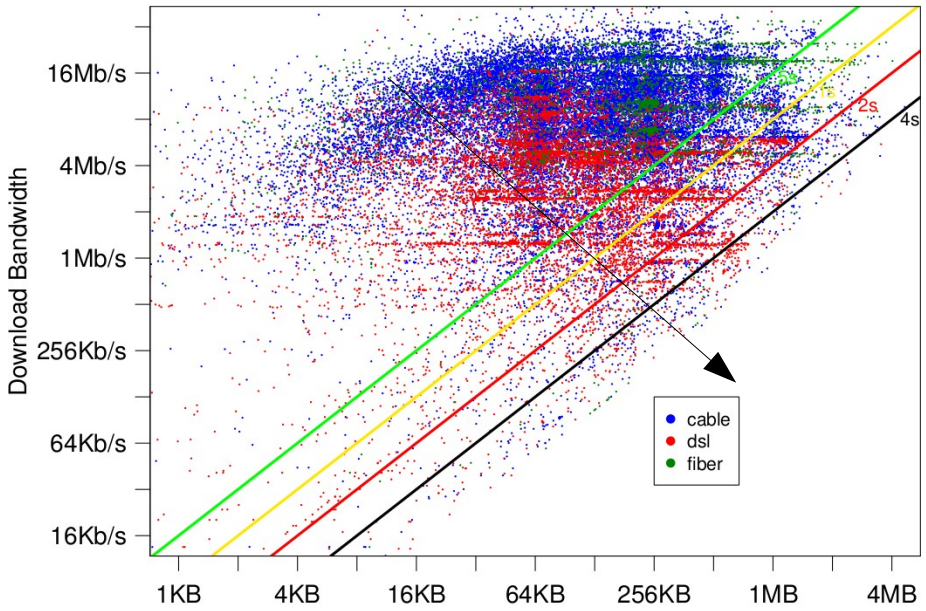
james.gettys@alcatel-lucent.com, jg@freedesktop.org

"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.

Bufferbloat Status

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

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

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

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

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)

DOCSIS (cable) change will deploy, which will help reduce cable bloat to ~100ms

There is No Single Silver Bullet

Smarter queuing is essential in broadband equipment: a single stupid drop tail bloated queue is a killer

AQM needed to avoid elephant flows:
TCP's responsiveness is quadratic in the delay, causing: "Wooley mammoth" bloated flows...

AQM is not just for routers anymore: hosts too!

Smarter Queuing is also needed everywhere

- "Fair" depends on where you are: I don't mean simply TCP fair queuing but "fair" among TCP flows, among devices, among customers, among policies, among processes, among traffic types...
- Fair/smart queuing helps TCP RTT fairness, ack compression, interactive versus non-interactive bulk transfers, DNS lookups...

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

AQM algorithms

At the edge of today's Internet, bandwidth is (very highly) variable on short timescales

Since there is sometimes but a single flow, TCP requires full bandwidth/delay product buffers: *but you don't know the bandwidth*, ***and*** *you don't know the delay*. . . .

No "correct" static drop tail buffer size can be possibly computed!

(W)RED and similar algorithms are based on queue length; not time in queue. They require careful tuning to be effective: bandwidth is an input to this tuning, and the tuning is different for different bandwidths. Tuning is anathema to (most) operators and impossible for home users.

Result: existing AQM not enabled in hosts, routers, broadband, and most routers

New AQM Algorithm: CoDel (“coddle”)

“Controlling Queue Delay”- Kathie Nichols & Van Jacobson, May, 2012; presented at this IETF in the transport area meeting

See July CACM article or May ACM Queue Article for details

CoDel is based on time in queue: not queue length: works over a wide range of varying bandwidths and RTT's experienced in the edge of the Internet w/o tuning

CoDel can work completely effectively across a set of queues

Fq_codel combines SFQ and CoDel; the combination is stunning, using 2% of a current Intel processor at 10G ethernet speeds!

It would be gravy if the same algorithm can be used elsewhere in the Internet so that AQM can become ubiquitous, as RFC 2309 (“The RED manifesto”) recommended

CoDel Status

Linux 3.5 has codel and fq_codel queue disciplines: you can play today! Please beat them to death and find problems and experiment

Ethernet was easy (with BQL): wireless is not so easy, due to driver buffering for aggregation, and will take more work

We really, really, really, really like fq_codel!

- Fair queuing is only 2% of CPU on 10GigE; unnoticeable in home router profiles: smarter queuing/scheduling of packets is very feasible

“Fair” is in the eye of the beholder

- Other forms of “fair” queuing yet to be done: e.g. “fair” wrt air time, “fair” wrt process groups, “fair” wrt. Device, etc.

A bug was recently discovered in CoDel under very high load with many flows; possible solutions are being simulated and implemented

Think Out of The Box – time for some heresy

You can't engineer around bufferbloat: we have to fix the network

Today's Internet edge: congestion is “normal” operation

We must have smarter queuing in the network edge. The same “one size fits all” packet marking/dropping algorithm for controlling TCP or other flows in the core of the network may not be the “right” answer for streaming real time audio and video flows in the edge of the network

Packets are not the logical units to drop; dropping a single packet out of a larger video frame may hurt; you may be better off dropping an entire frame at a time: codec expertise needs to meet congestion control

Transmitting packets that are “stale” any further than necessary which cannot meet real time deadlines only exacerbates congestion

Don't presume we can't change how our systems/routers work: few actually run AQM today: we have an opportunity to be novel

Home Router Disaster

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!

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