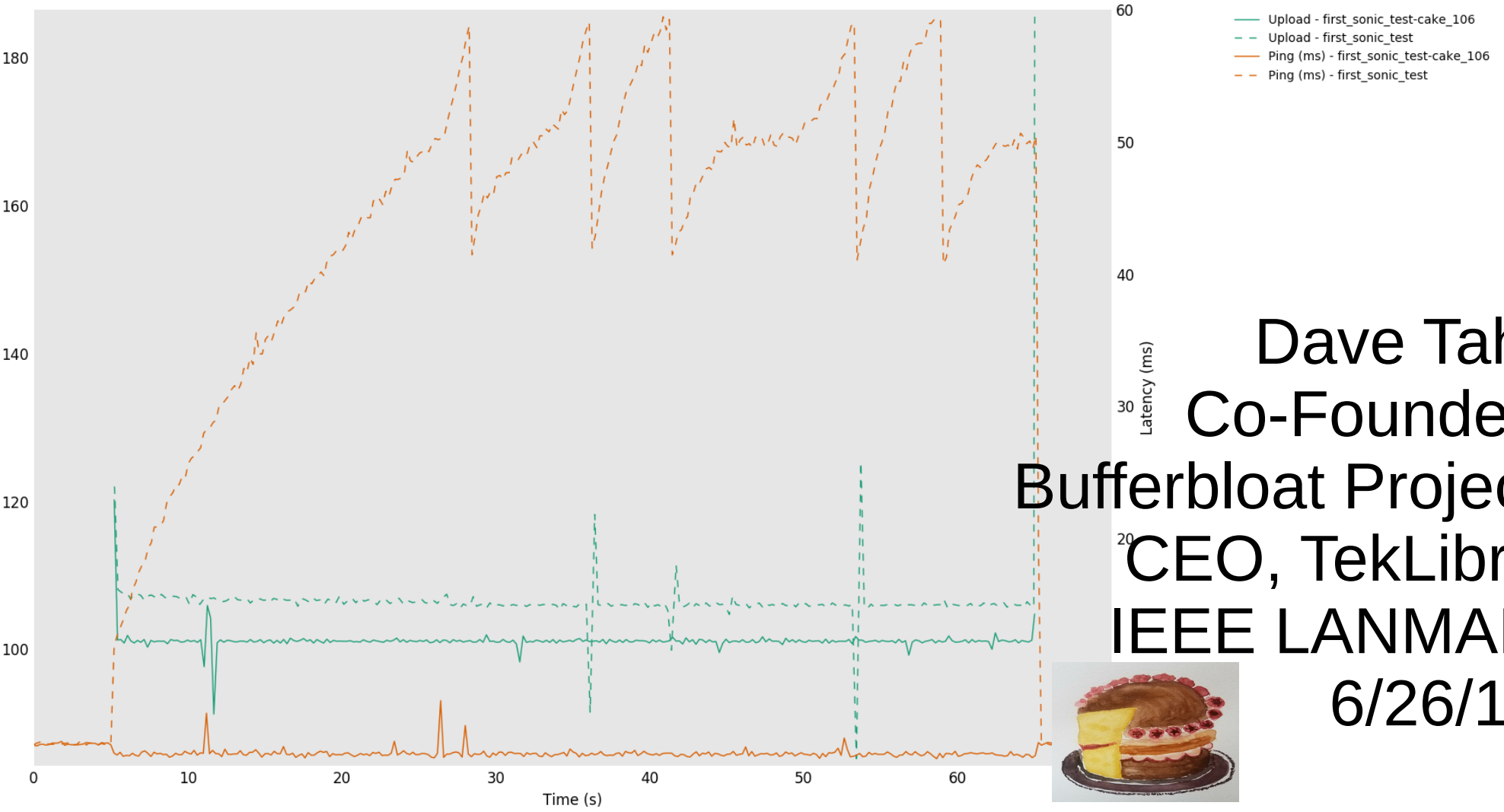
sch_cake

Comprehensive smart queue management for Network Gateways

TCP upload stream w/ping
Bandwidth and ping plot



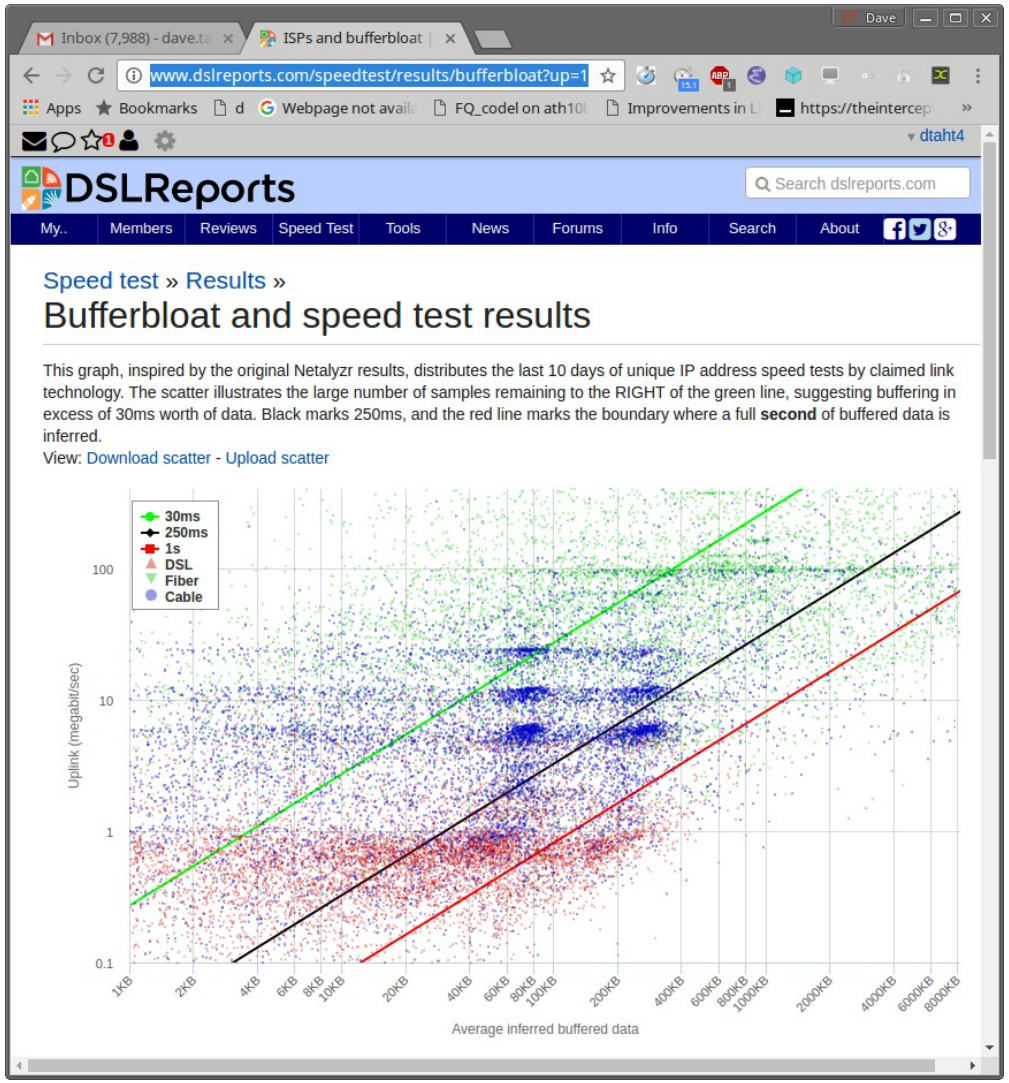
Dave Taht
Co-Founder, Bufferbloat Project
CEO, TekLibre
IEEE LANMAN
6/26/18

Local/remote: dancer/flent-fremont.bufferbloat.net - Time: 2017-01-17T12:57:50.157122 - Length/step: 60s/0.20s

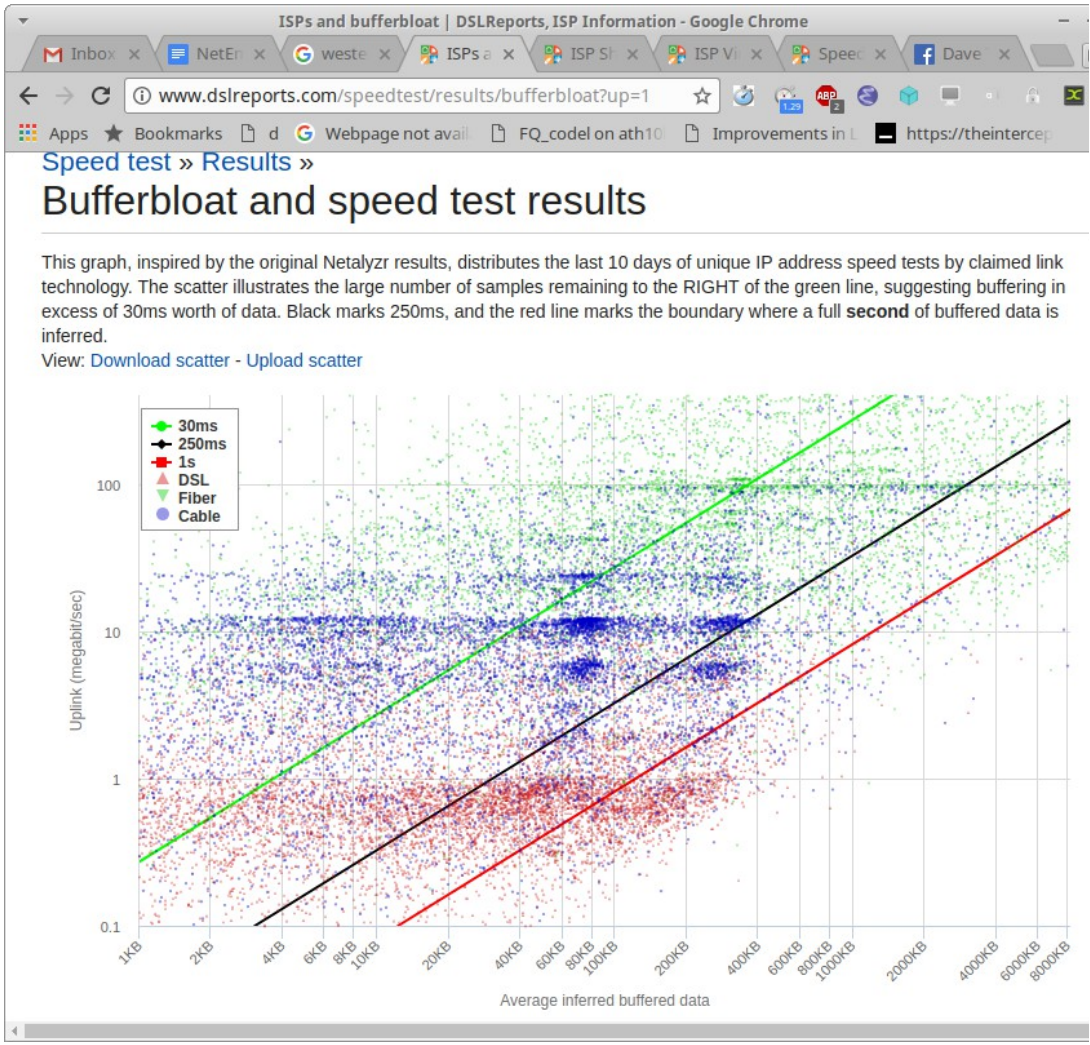
Bufferbloat

- “The undesirable latency that comes from a router or other network equipment buffering too much data. It is a huge drag on Internet performance created, ironically, by previous attempts to make it work better. “
- Where < 30ms queuing delays under load are desirable
  - Often 2+ real world seconds on cable modems and DSL
  - 10 seconds or more on Wifi
  - 600+ seconds on gogo-in-flight
- “Bloated buffers lead to network-crippling latency spikes.”

Measuring network latency with load



Jan 31, 2017



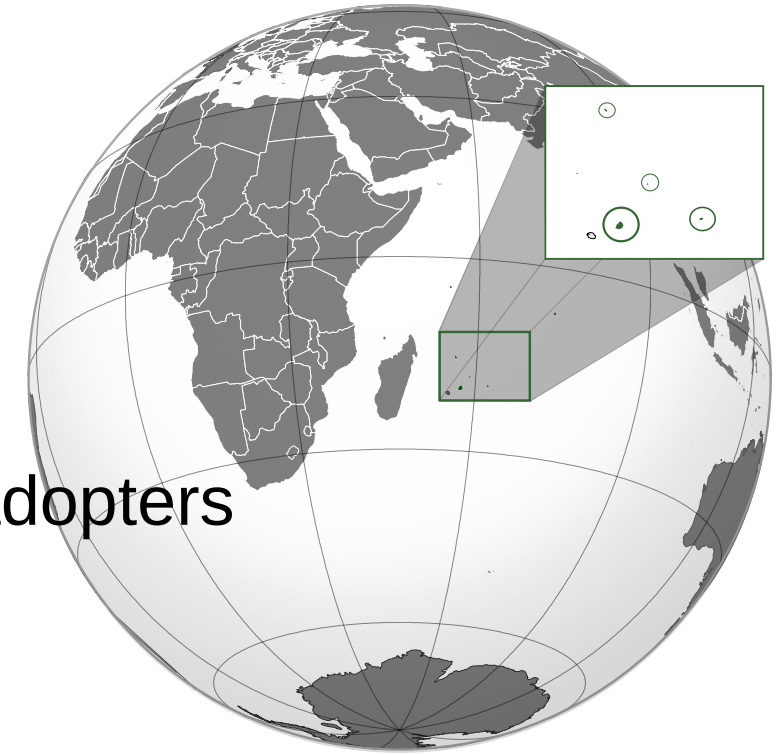
June 14, 2018

The Fair Queuing odyssey

- 1985 RFC970
- 1990: SFQ approximated hash per packet scheduler
- 1995: DRR approximated byte fair scheduler
- 1998: DRR++ approximated byte fair scheduler with QoS
- 2002: SQF – small flows gain priority
- 2012: Codel – Active queue management/drop head queueing
- 2012: fq_codel added to Linux Kernel
- 2012-2018 – BQL, Pacing, SQM deployments
- 2018: RFC8290 published

Bufferbloat.net engineering methods

- Labs in the UK, Denmark, Sweden, Germany, USA
- Extensive simulation at a wide variety of RTTs and B/Ws
- Actual deployment as part of multiple open source router projects, worldwide, 5+ years.
- Iterative development
- 100% Open source
- No patents
- Open Access publications
- Enthusiastic volunteers and early adopters



End to end approaches don’t suffice

- Pacing, TCP cubic, BBR, L4S… best case only work within an RTT, usually many...
  - With IW10, Interrupt bulking (NAPI), TSO/GRO “superpackets”, buffering in the device, driver, main queue, stack and application. Bursts caused by congestion
- FQ works between packets -
  - Typical CDN internet RTT = 20ms
  - Per 1500 byte packet time at: 1Gbit = 13us
  - 64 bytes at 10Gbit = 6ns
- Fair Queueing on gateways **is necessary to break up bursts, to mediate bad behaviors, and to return flows to being individual packets.**
- **Bonus: the smoothness between flows makes all the E2E approaches work better, more closely approximating poisson models.**
- **Network congestion control moves into the network, where it belonged in the first place.**

FQ + AQM is now everywhere

- sch_fq (w/tcp pacing) default at google, BQL universal across ethernet device drivers, innumerable other bufferbloat related tcp improvements in the stack
- fq_codel is now the default queue management system in most Linux distributions.
- Also available on BSD, ns2, ns3, click, openflow, and several proprietary versions
- “fq_codel for wifi” is now in most third party router firmware for ath9k and ath10k, and shipping for multiple commercial products like eero, evenroute, turris omnia, meraki, and google wifi. See “Ending the Anomaly” for details. Entered Linux kernel mainline early 2017.
- “Smart Queue Management” (SQM) (htb + fq_codel) or something like it,

  swept the third party router firmware market, marketed as “streamboost”, Adaptive QoS, and other brand names in COTS gear. Called “SQM” on openwrt, edgerouters and eero.
- Cake (deficit shaper + fq_codel w/8-way set associative queuing) is part of lede/openwrt and other products downstream
  - Shipping on evenroute, turris omnia, other openwrt derived products, and hopefully entering the Linux kernel mainline this month
  - Available as backports as far back as Linux 3.10
  - Current branch now scales past 40GigE:

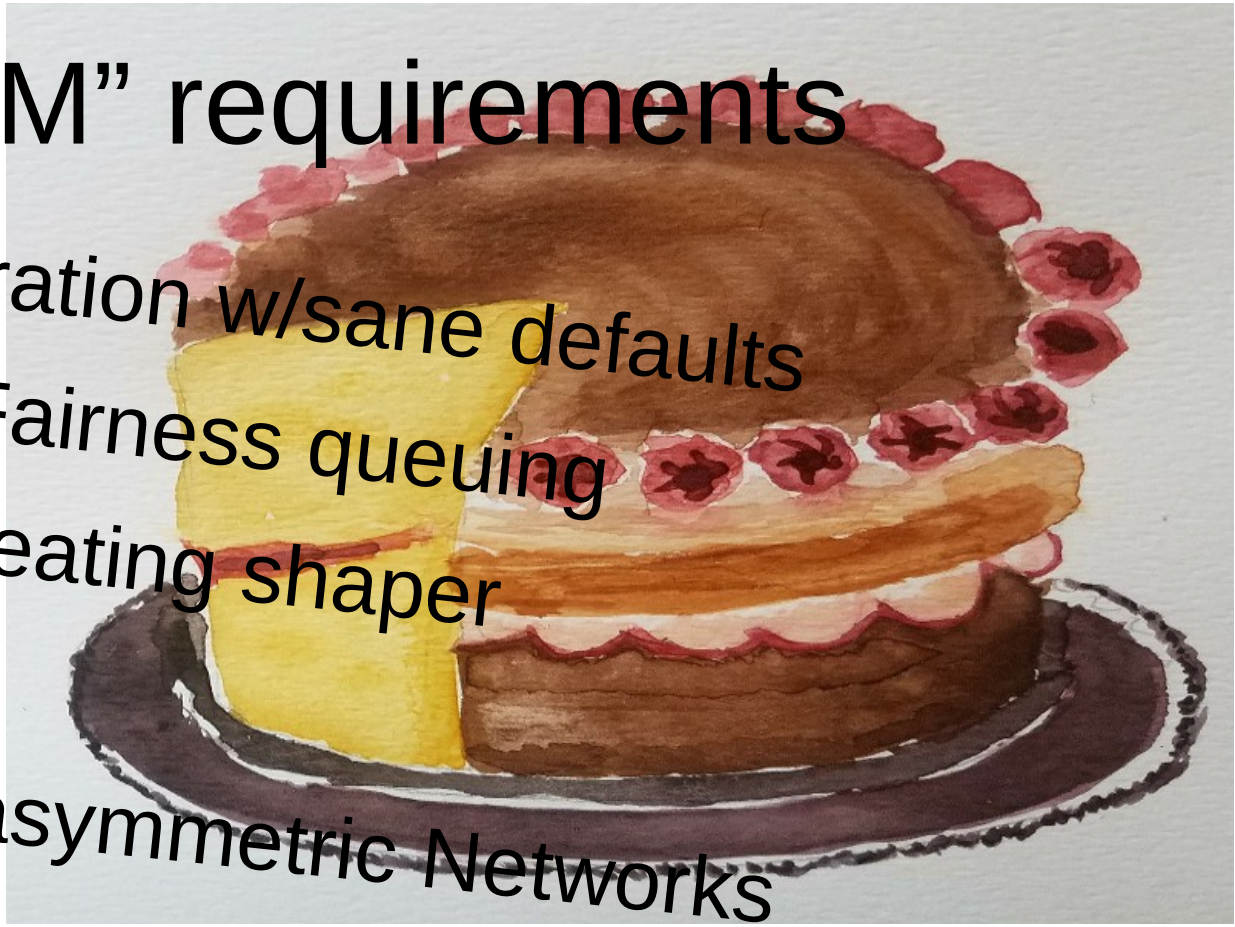
https://github.com/dtaht/sch_cake/

The case for per host fair queuing

- E2E approaches don’t handle
  - Malicious hosts
  - Bursty hosts (wifi/LTE aggregates)
  - Misbehaving hosts (steam/bittorrent)
  - Low rate/high priority hosts (voip/videoconferencing)
  - Bulk flows (uploading photos to facebook, streaming movies)
  - Load spikes (slashdotting your web server)
- With FIFO on an overbufferred and congested link, any one device can destroy the network for everyone else. And does.
- With fq_codel abusive hosts running torrent or steam can take more bandwidth than “fair”.
- With sch_cake, all active hosts can share in the bandwidth equally, with exceptions for diffserv (de)prioritization

Final “SQM” requirements

- Simple Configuration w/sane defaults
- Host and Flow Fairness queuing
- Robust HTB-defeating shaper
- Diffserv Support
- Ack Filtering for asymmetric Networks

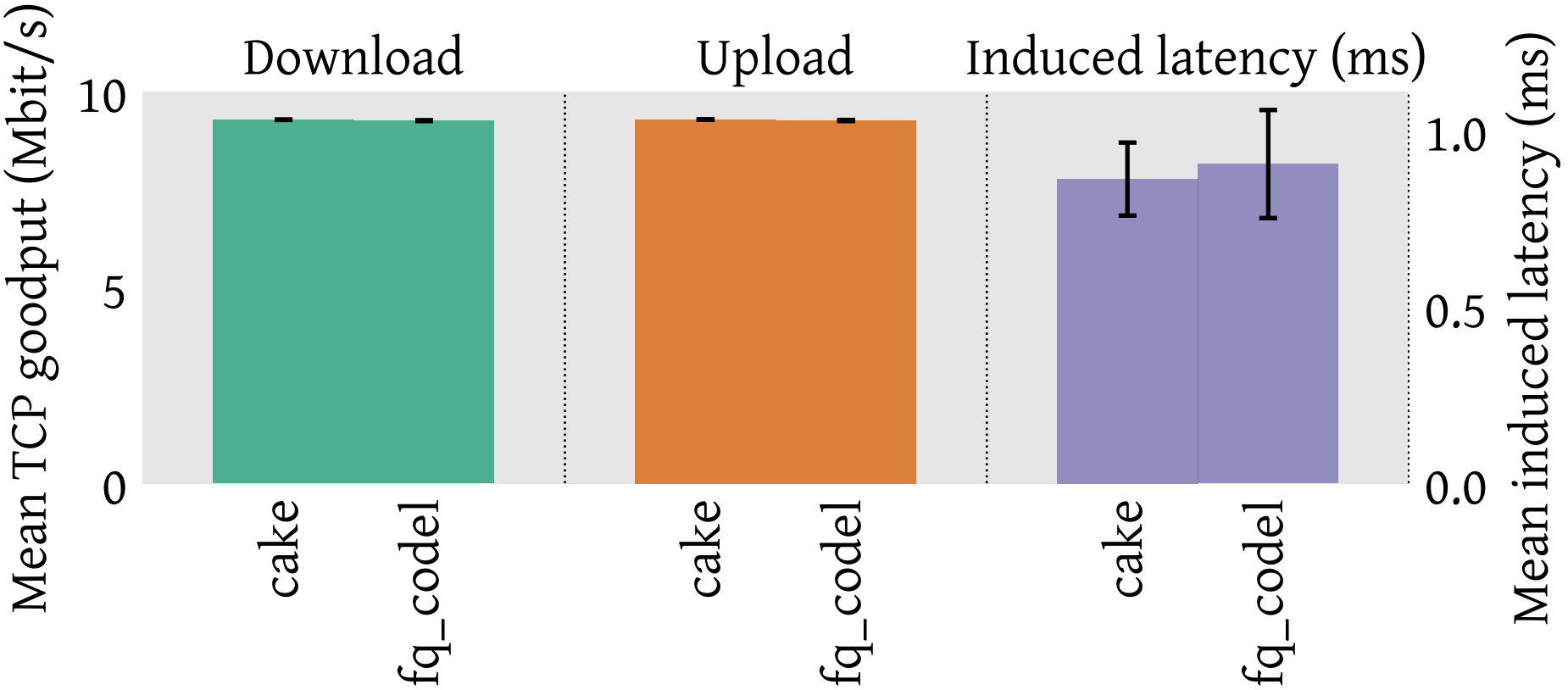


“Common Applications Kept Enhanced”

sch_cake: Simple configuration

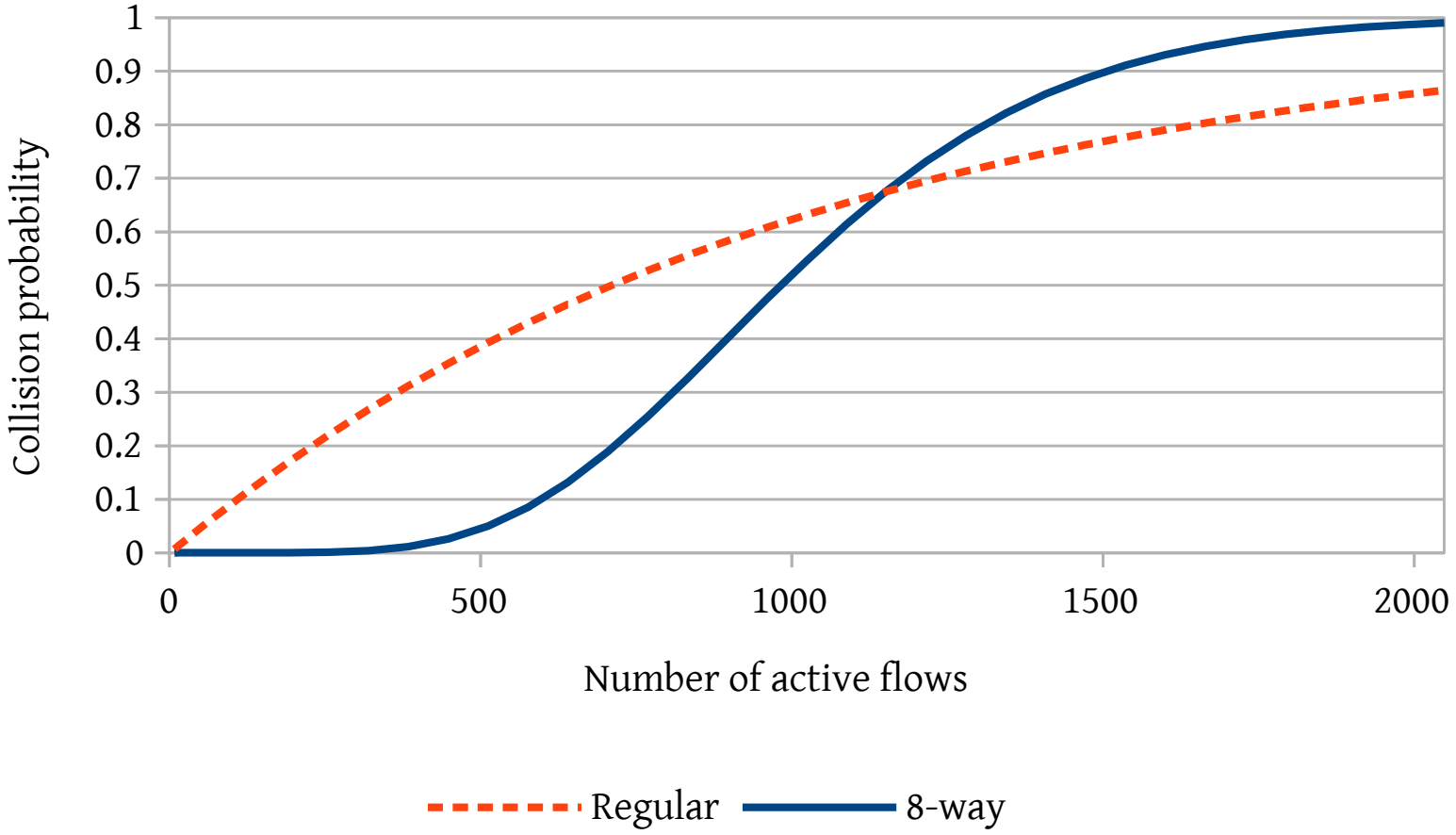
- Outbound shaping for docsis:
  - tc qdisc add dev eth0 root cake bandwidth 10Mbit docsis ack-filter nat
- Inbound shaping
  - Ip link set ifb0 up
  - tc qdisc add dev ifb0 root cake bandwidth 100mbit docsis nat ingress besteffort wash
  - tc filter replace dev eth0 ingress prio 1 handle 12 u32 action mirred egress redirect dev ifb0
- Similar setup for DSL and ethernet framing types
- Can also run at line rate, no shaping, w/BQL

Comparable to htb+fq_codel



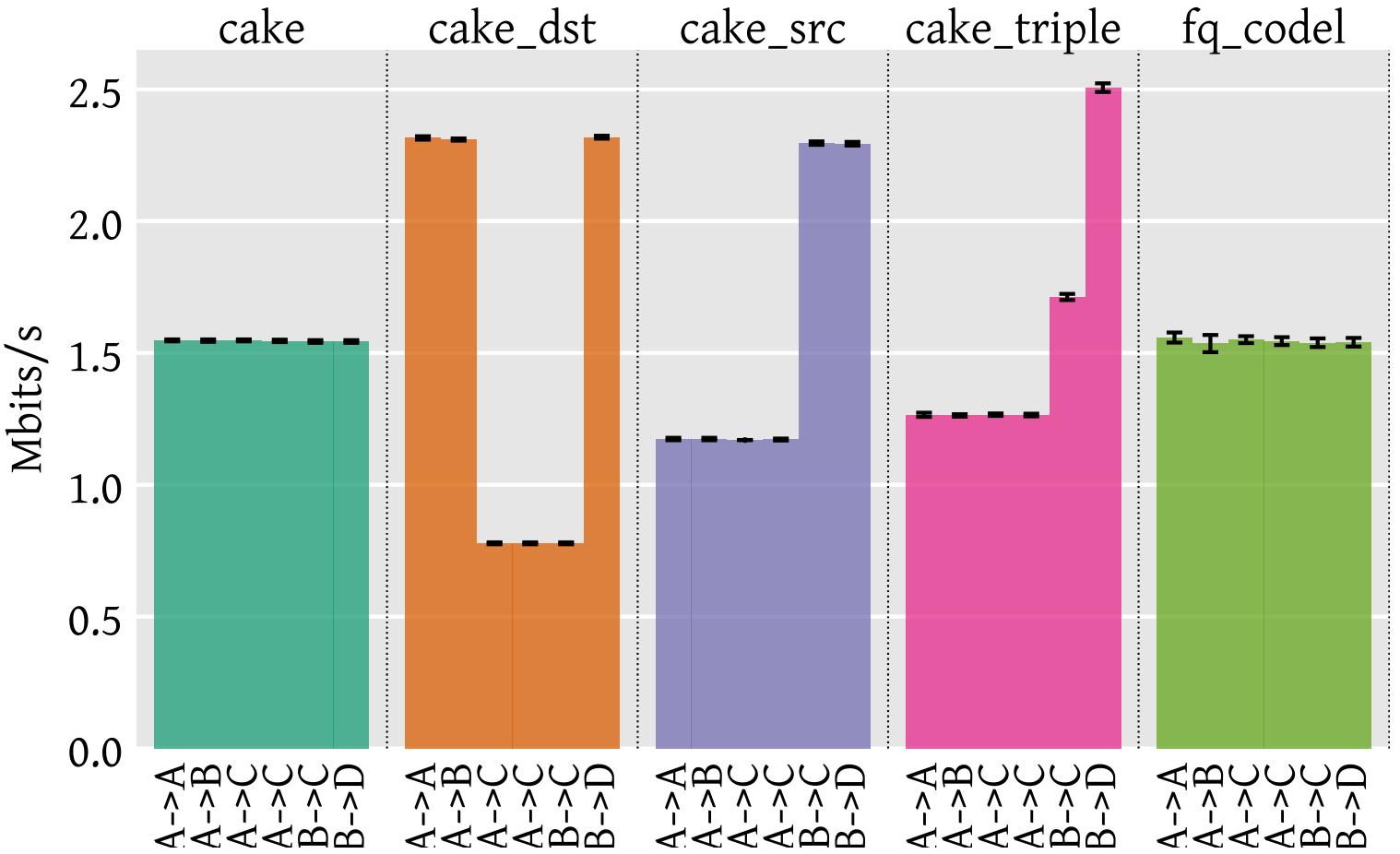
But: stores one less packet.
barely visible at 10mbit, very visible at 1mbit

8 way set associative queue hashing



Near perfect flow isolation

Per host FQ modes (Even through nat)

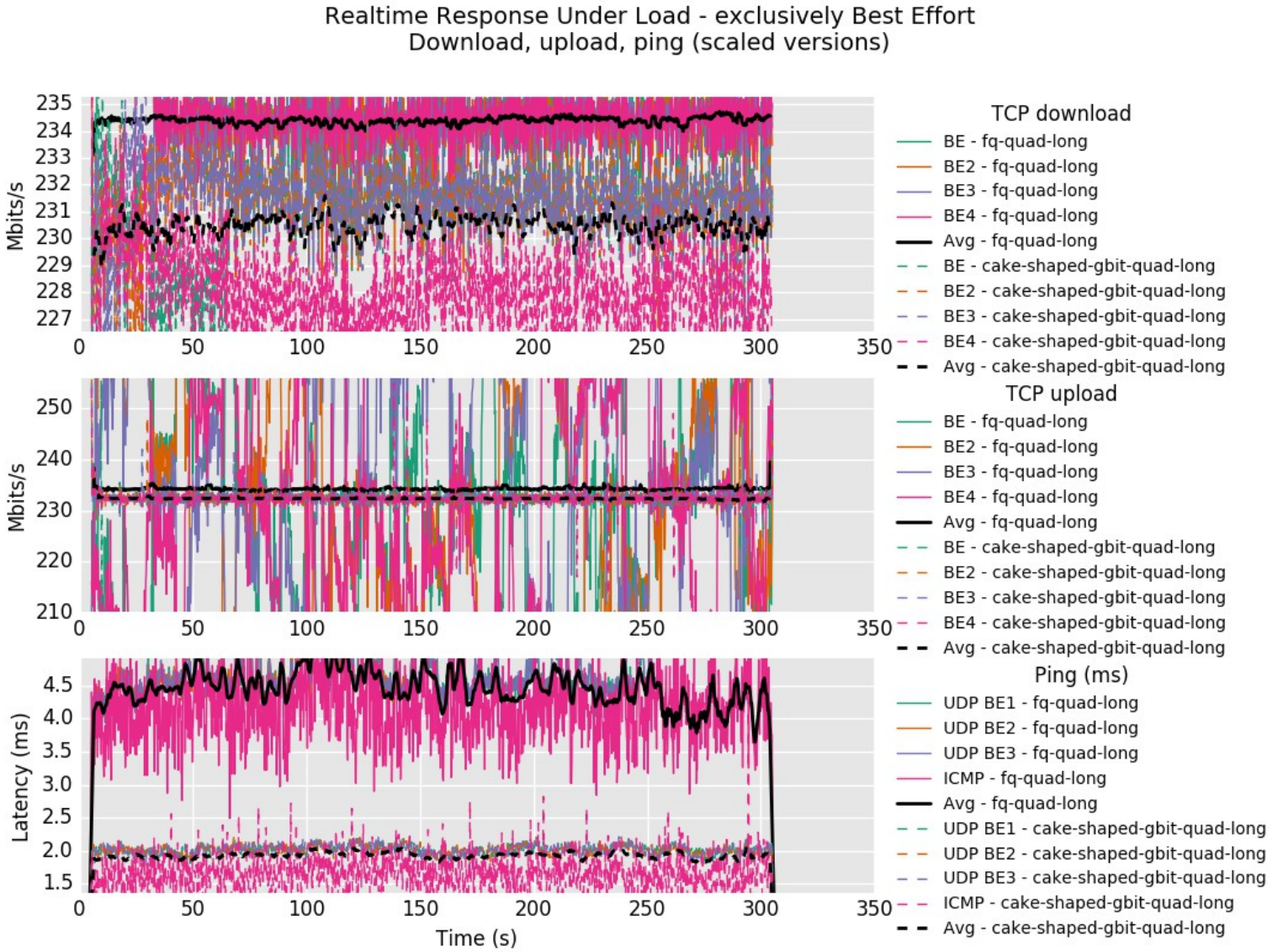


SCH_CAKE
Deficit based shaping

- Use case: Downstream from a token bucket shaper
- HTB + bloated FIFO ↔ HTB + fq_codel
  - Requires a lower set bandwidth to take control back
  - Fiddly – 85% - 95% of the upstream setpoint
- sch_cake can (with perfect framing) be at the same setpoint as the upstream, while still taking back control of the link, with vastly better queuing.
- Can subvert every TB based shaper out there.
- Or: Shape to line rate. And win:

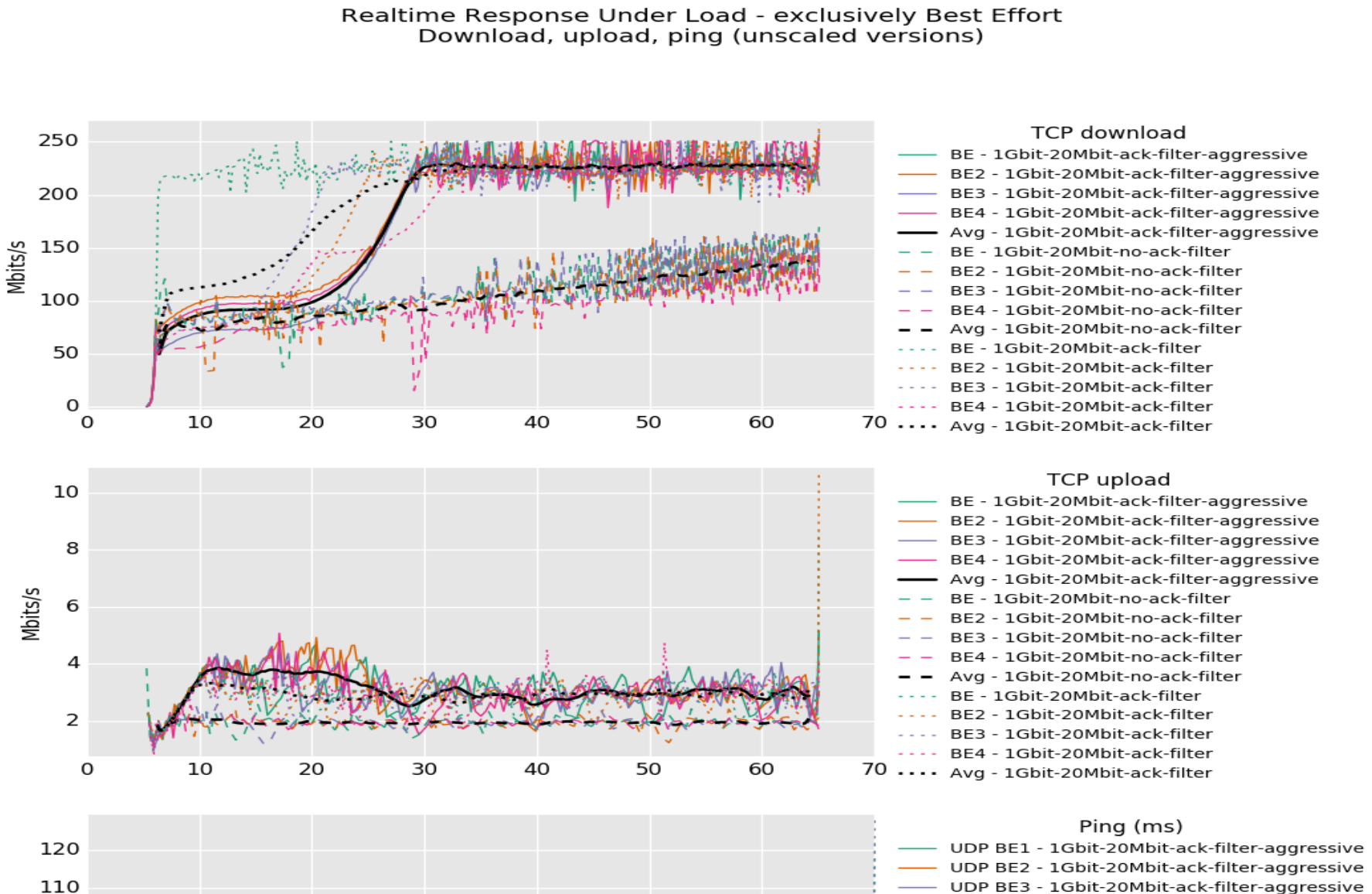


sch_cake (shaped) v sch_fq + BQL
1 Gbit, quad core atom



Ack filtering

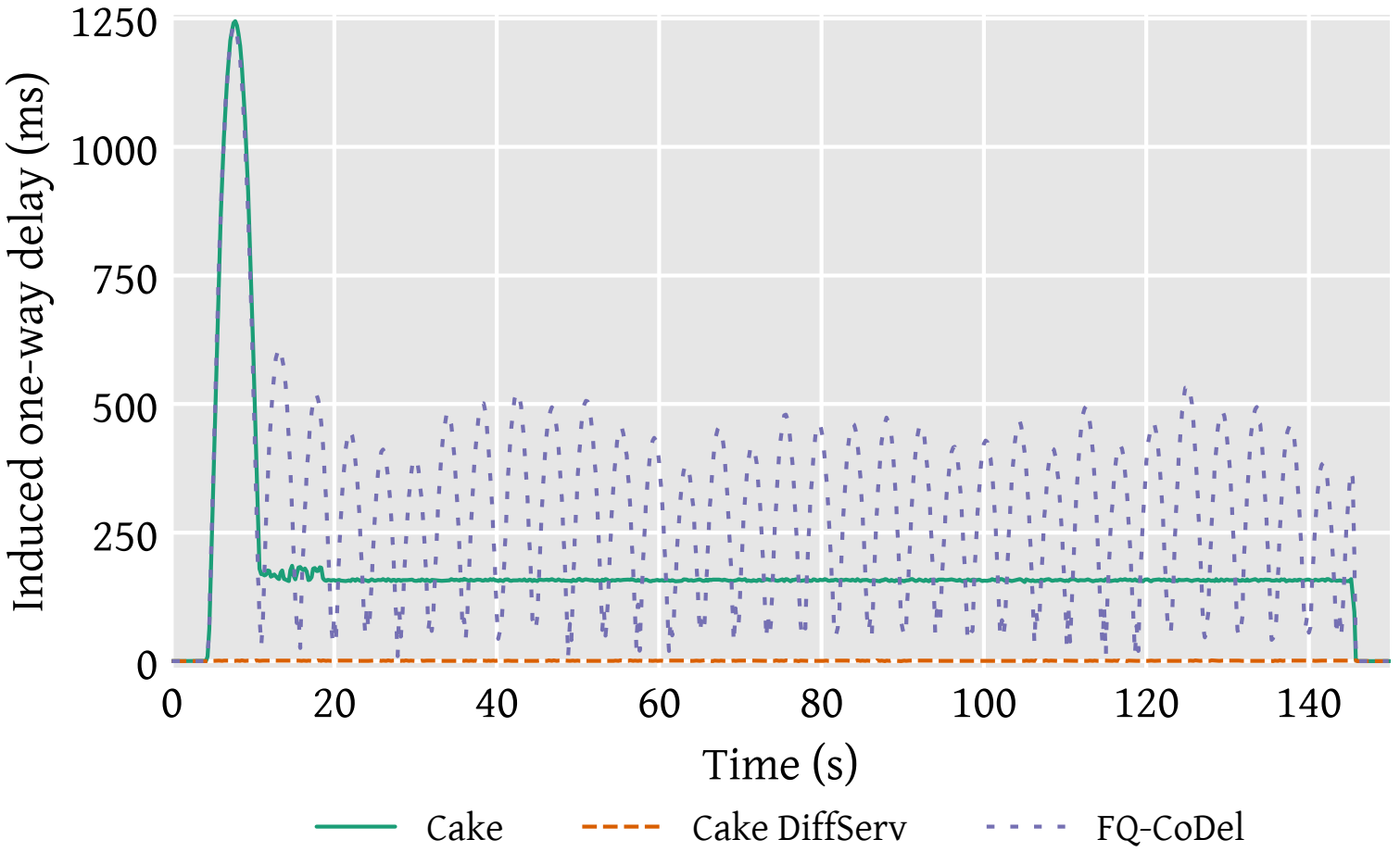
“My god, you’ve created a monster” - Eric Dumazet



Diffserv Support

- Diffserv: marking QoS fields within a packet with a specific 6 bit code to express more or less prioritization.
- Conventional diffserv support is generally thought of increased drop probability… which doesn’t work.
- Cake treats it as bandwidth reservation
  - Max of 1/4 for interactive flows
  - Minimum of 1/16 for background flows (CS1)
  - The rest for best effort
  - All tiers can borrow from the others
  - fq_codel is applied to all flows to hold latency and queue lengths low
    - Exceeding your allocation increases drop probability via fq_codel

A CS6 marked flow vs 32 tcp flows



Future sch_cake work

- GPON framing
- Ack-compression in the tcp stack
- Scalability to 40+Gige at the core
- L4S – style, more aggressive ECT(1) handling
- DPDK, NS2, NS3, P4?
- Multi-core support
- Hardware Implementation
- Evaluation of cobalt
- IDS rules for deprioritizing attack traffic

Cake was *not* sponsored by:

- O2, UPC, T-Mobile, Zyxel, BT OpenReach, BT Wholesale, TalkTalk, Virgin Media, Vodafone, Telia, Elisa, NTT DoCoMo, Rogers, Nokia, Deutsche Telekom, or Telstra.
- Nor Comcast, Time Warner, AT&T, Google Fiber, Google Access, Free.fr, France Telecom, Cisco, Huawei, Emerson, Linksys, ASUS, TP-Link, Eero, Ubiquiti, Netgear, Intel, ARM or AMD.
- Nor the KGB, NSA, NSF, ITU, IETF, McDonalds, Netflix, FCC, FTC, NIST, Shuttleworth Foundation, Wells Fargo bank, the Mafia, or Wall Street...
- Nor SavetheNet, Public Knowledge, the EFF, or any other org we approached except NLNET.

  THX, NLNET!
- Cake was developed by the users, for the users, based on extensive feedback from the field on the “smart queue managment” (SQM) system shipped as part of openwrt, dd-wrt, lede, pfsense, ubiquiti’s edgerouters, netduma, evenroute, and other open source router projects.
- Cake is open source, patent free, widely available, and with fully documented behaviors. It applies the best of modern fair queueing, aqm, and shaping techniques to achieve the lowest latency, maximum fairness, and highest utilization on the edge routers we’ve yet achieved in the bufferbloat project.
- Now everyone can have cake.

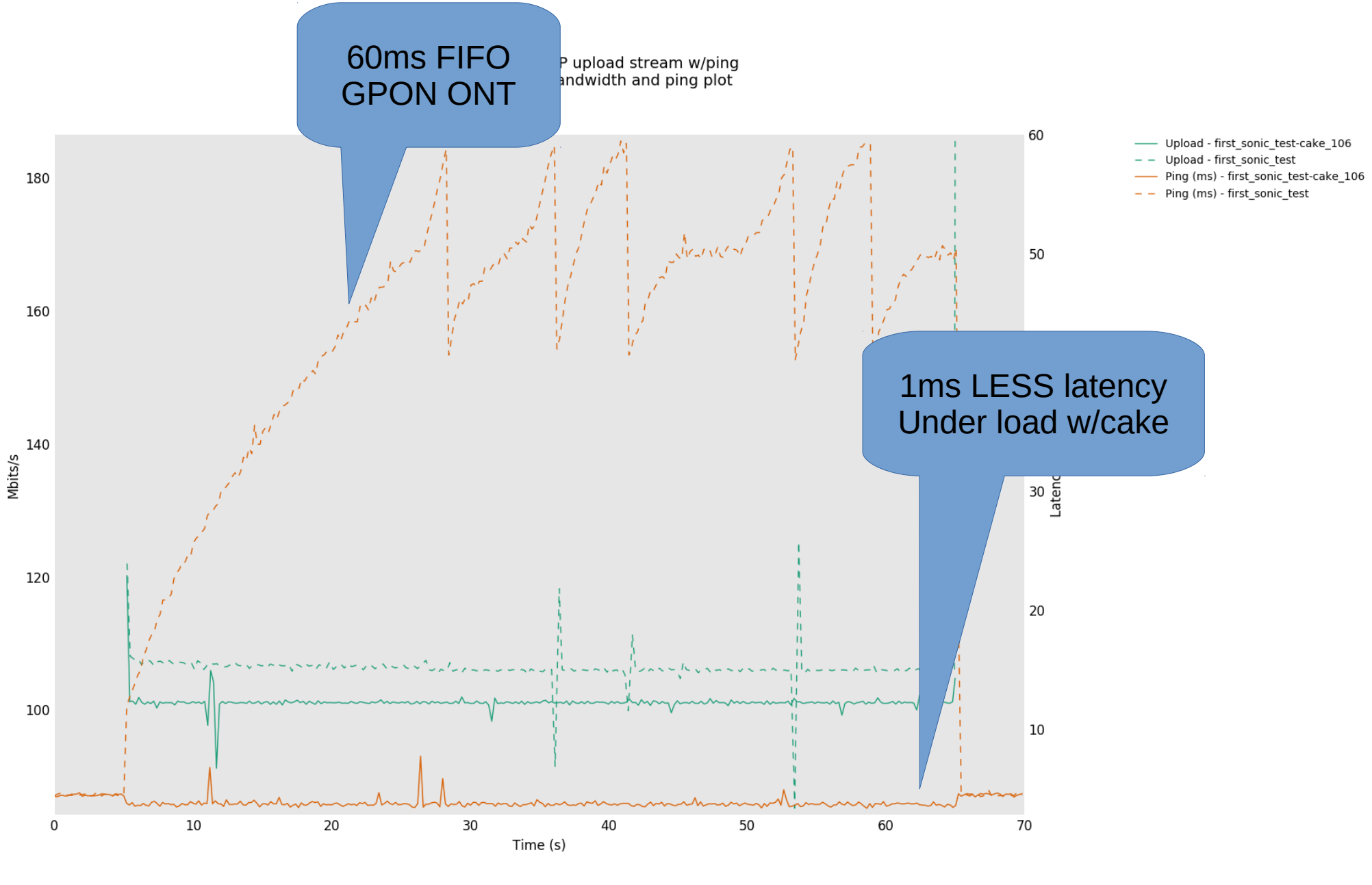
Questions?

https://github.com/dtaht/sch_cake

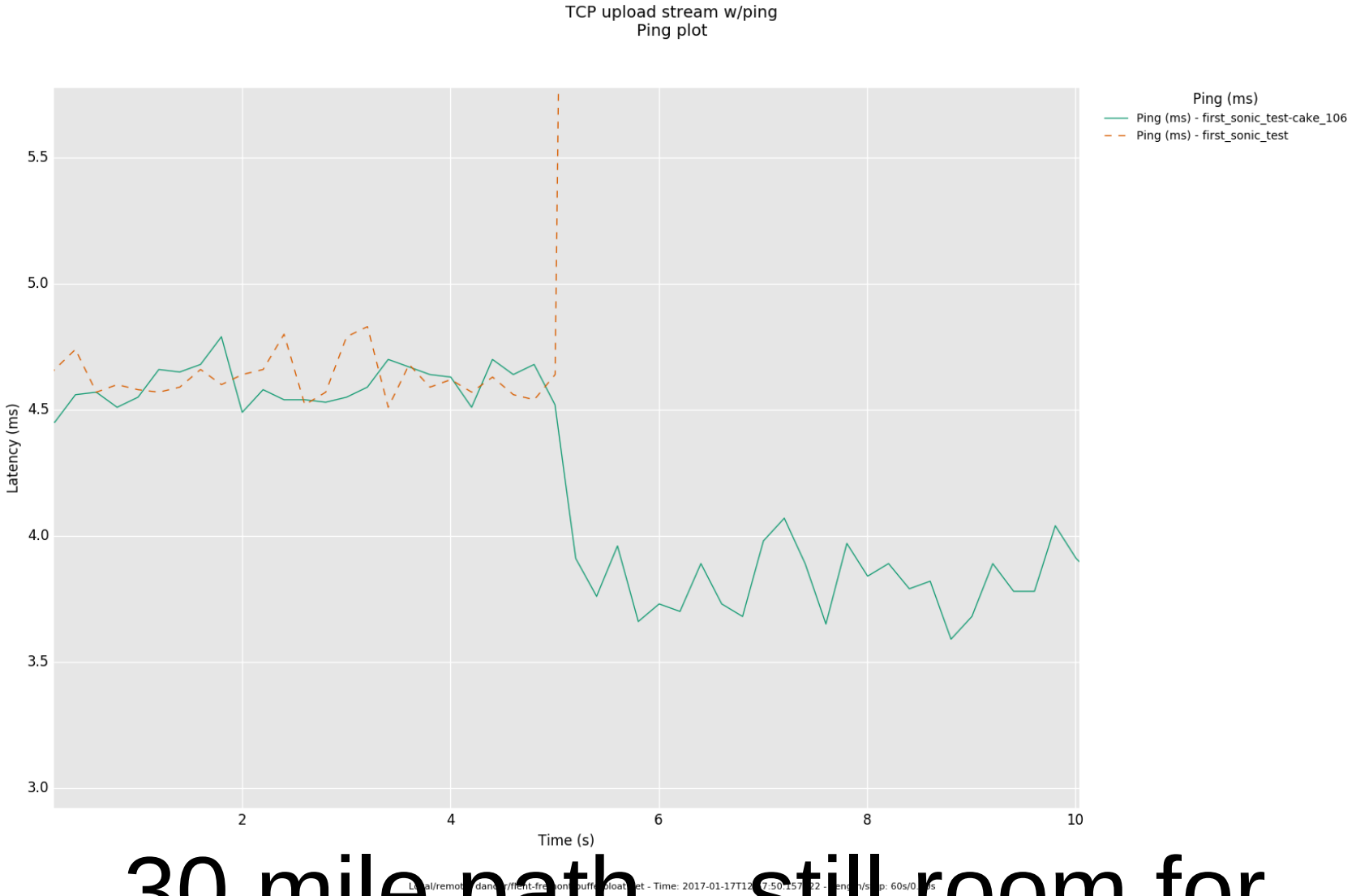
Extra Slides

- Cake on a GPON fiber network
- Cake native, shaped, v sch_fq and fq_codel

Cake v Sonic Fiber @100Mbit

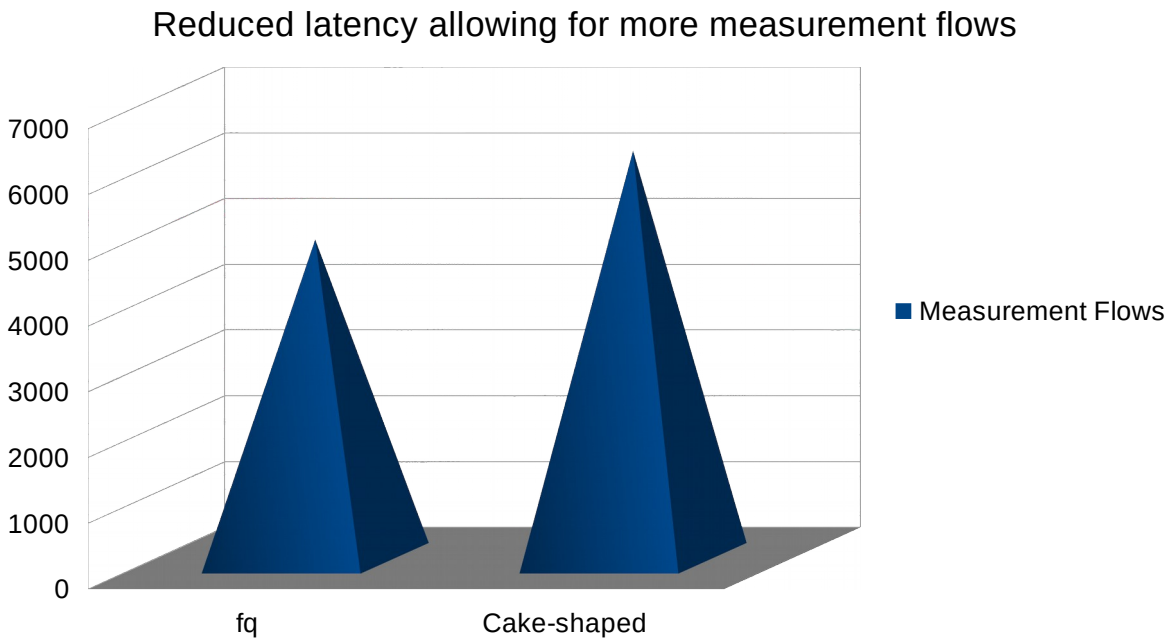


Cake gets inside the GPON request/grant loop



30 mile path – still room for improvement!

Not a bandwidth hit… we just fit in 20% more measurement flows



sch_fq v sch_cake
shorter RTTs, smaller packet trains