Misunderstanding Throughput and Latency in Residential Networks

What Residential Traffic Actually Looks Like
Some Speed^H^H^H^H^HCapacity Test Flaws & Improvements
Latency and Jitter Nightmares in the field

Dave Taht <dtaht@libreqos.io>
Chief Science Officer
LibreQos.io
https://blog.cerowrt.org #libreqos:matrix.org @dtaht:matrix.org

LibreQoS.io
Fast, Flexible QoE

About Dave Taht

- Co-Founded Bufferbloat project w/Jim Gettys
  Run the Make-Wifi-Fast and LibreQos projects
  Co-author of fq_codel, cake
  Embedded Linux Developer since 1998
- Most Recent Podcast: "Heavy Networking 666"
- Dave's Funniest Network Videos:
  - People As Packets at Apnic
  - Explaining Wifi Aggregation
  - Juggling Packets w/TTI Vanguard

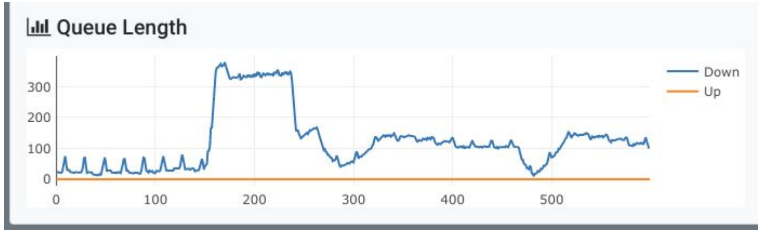
Dave's Serious Stuff:

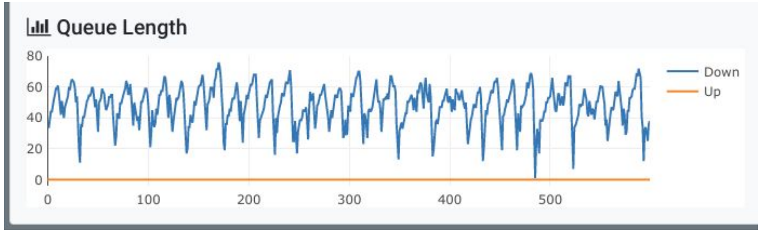
RFC8290

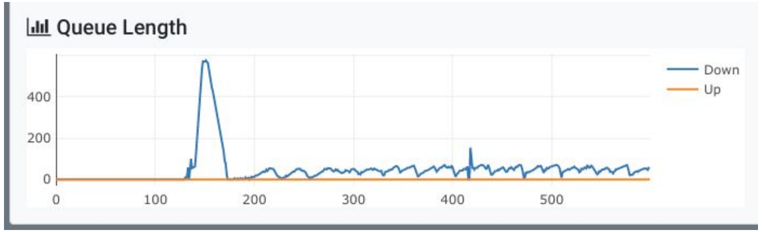
BITAG Latency report

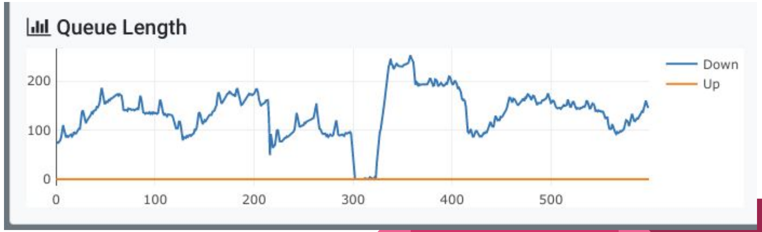
Bufferbloat.net

Example TCP behaviors on a Single Queue





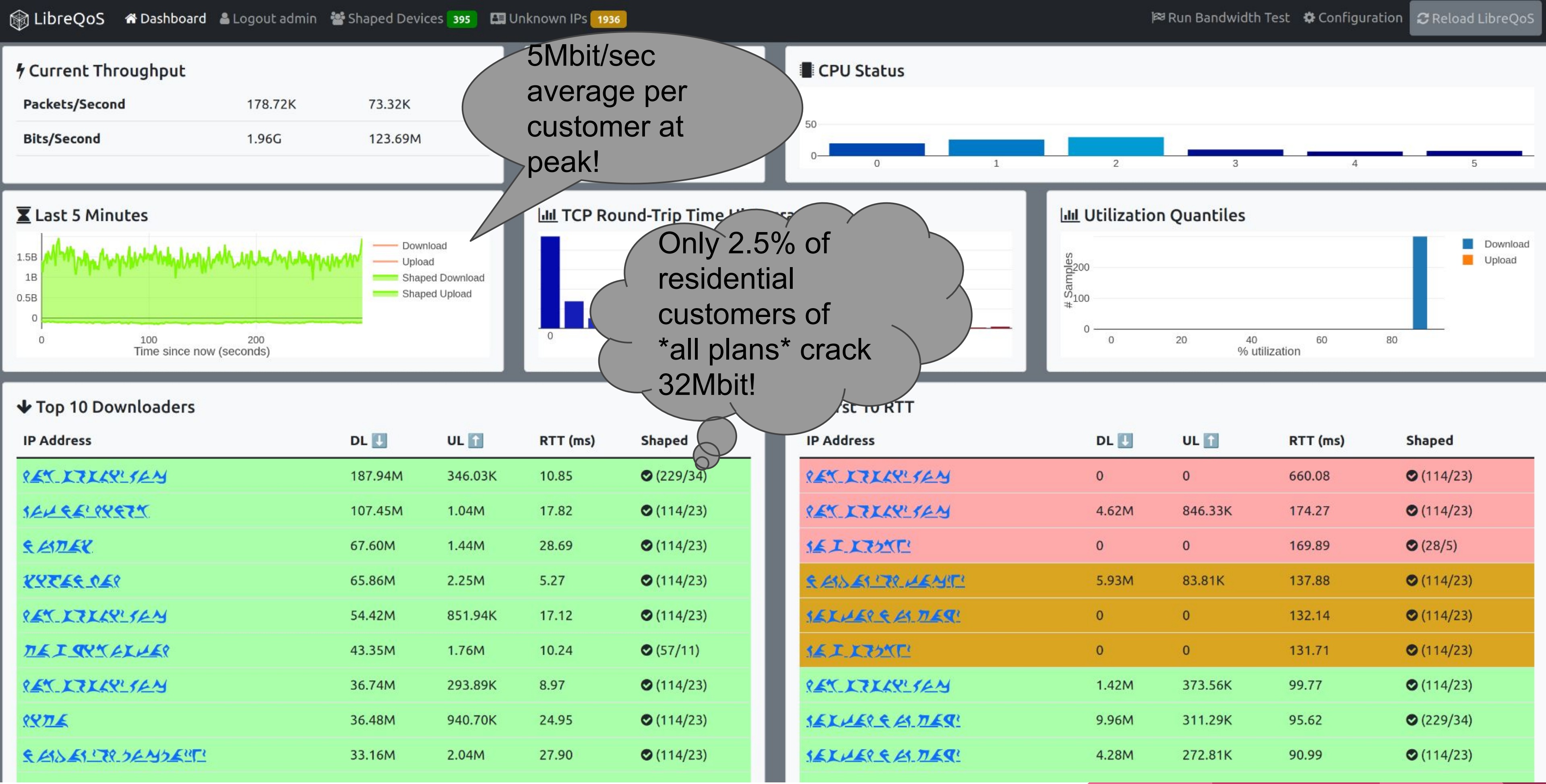




There will be a quiz later...

Actual Residential Network Loads Today

- Web Page Usage: typically 3 second bursts
- Downloads and applications: infinite bandwidth
- Movie Streaming: 25Mbit, 4 second bursts)
- Chat: Kilobits!
- Videoconferencing: 1-4Mbit
- Gaming: Kilobits
- BUT Lots of little things all the time!


LibreQoS
Dashboard
Logout admin
Shaped Devices 395
Unknown IPs 1936
Run Bandwidth Test
Configuration
Reload LibreQoS
Current Throughput
Packets/Second 178.72K 73.32K
Bits/Second 1.96G 123.69M
5Mbit/sec average per customer at peak!
CPU Status
Last 5 Minutes
Download
Upload
Shaped Download
Shaped Upload
Time since now (seconds)
TCP Round-Trip Time
Only 2.5% of residential customers of *all plans* crack 32Mbit!
Utilization Quantiles
Samples
% utilization
Top 10 Downloaders
IP Address DL UL RTT (ms) Shaped
187.94M 346.03K 10.85 (229/34)
107.45M 1.04M 17.82 (114/23)
67.60M 1.44M 28.69 (114/23)
65.86M 2.25M 5.27 (114/23)
54.42M 851.94K 17.12 (114/23)
43.35M 1.76M 10.24 (57/11)
36.74M 293.89K 8.97 (114/23)
36.48M 940.70K 24.95 (114/23)
33.16M 2.04M 27.90 (114/23)
IP Address DL UL RTT (ms) Shaped
0 0 660.08 (114/23)
4.62M 846.33K 174.27 (114/23)
0 0 169.89 (28/5)
5.93M 83.81K 137.88 (114/23)
0 0 132.14 (114/23)
0 0 131.71 (114/23)
1.42M 373.56K 99.77 (114/23)
9.96M 311.29K 95.62 (229/34)
4.28M 272.81K 90.99 (114/23)

On Residential Networks

- Once you have “enough” bandwidth, you hardly ever use more
- Very small increase per person in household
- 50Mbit Plan – 26-34Mbit

See also the BITAG Latency Report!

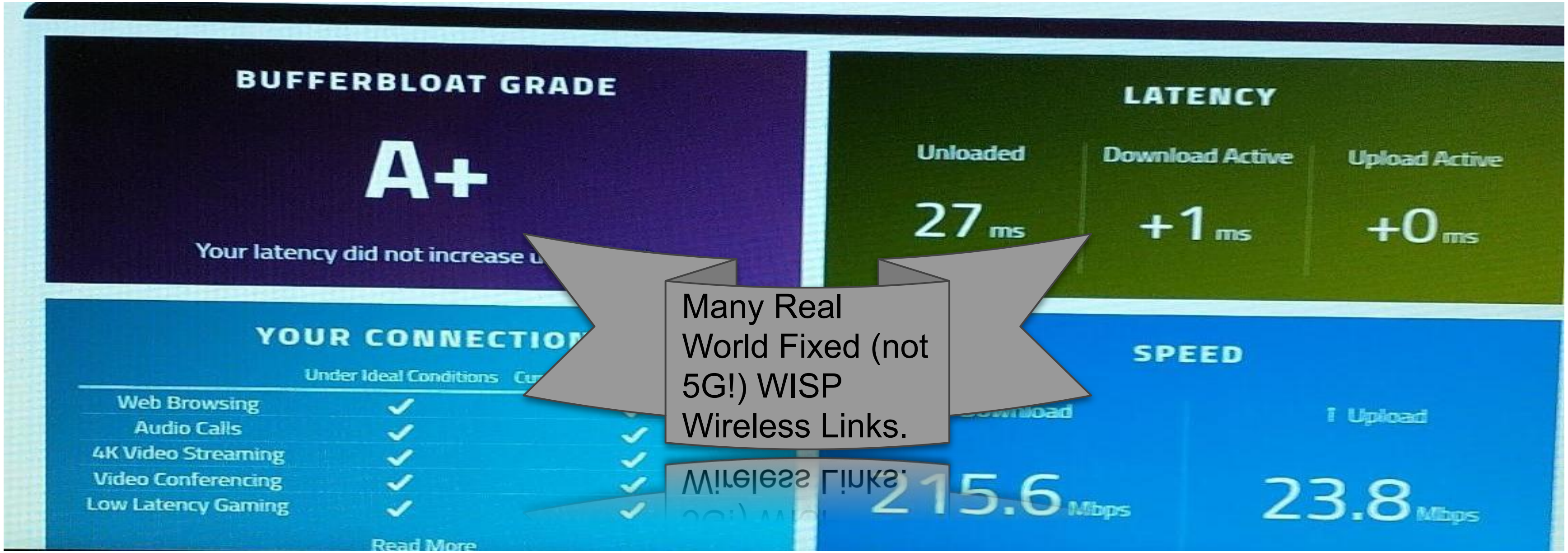
- 1 Gbit Plan -- 26-34Mbit

For customer satisfaction - for the gamers, videoconferencers -

What matters more is **consistently low latency**, low MTBF, low MTTR, and something that just works, all the time.

Bandwidth is a Lie!

My goal for all networking technologies



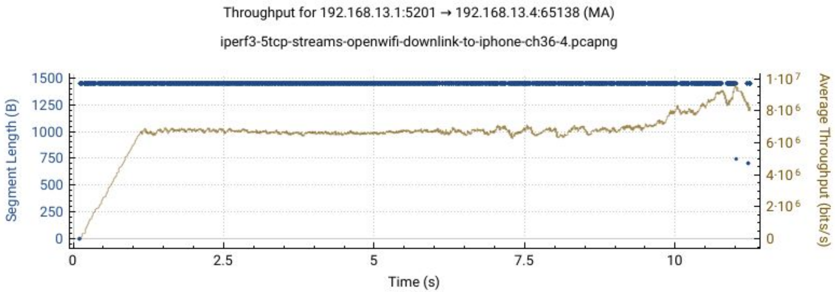
Many Real World Fixed (not 5G!) WISP Wireless Links.

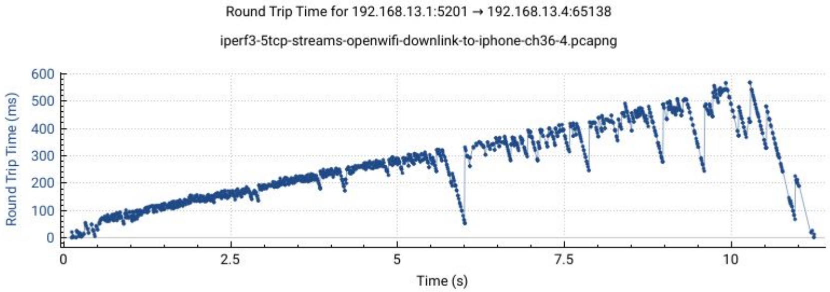
The tools are out there!

WiFi issues

- 97% of Residential access ends in WiFi
- 46% of ISPs say WiFi is their biggest problem
- 18B WiFi users!
- Yet:
  - No interop events!
  - 6 different versions in the field
  - OpenWrt and the Make-wifi-fast projects NEED YOU!

What is “Average” Latency?





[OpenWifi Test 2/23]

Speedtest Persistent Measurement Bugs

- Too many averages over too short or too long intervals
- Survivorship Bias in the FCC broadband maps
  - How many tests did not complete, and why?
  - What is the ratio of subscribers per BGP AS to testers? (more testers indicates more problems)

Most tests run for 20s or less!

  - Users use the network all day, a videoconference runs for an hour

Feature Requests for all Future Speedtesters

- Networks behave very differently when under a load going in both directions simultaneously.
- I would really like the fancy tests that are now testing up + ping, then down + ping… to ALSO test up + down + ping + AQM.
  - Ideally with staggered starts.
- The results are very different, the side effects often devastating.
- Only the flent RRUL test series does this today.
- And… May I benchmark your benchmarks?

Speedtest vs Starlink comparison

https://www.speedtest.net/result/14438617607

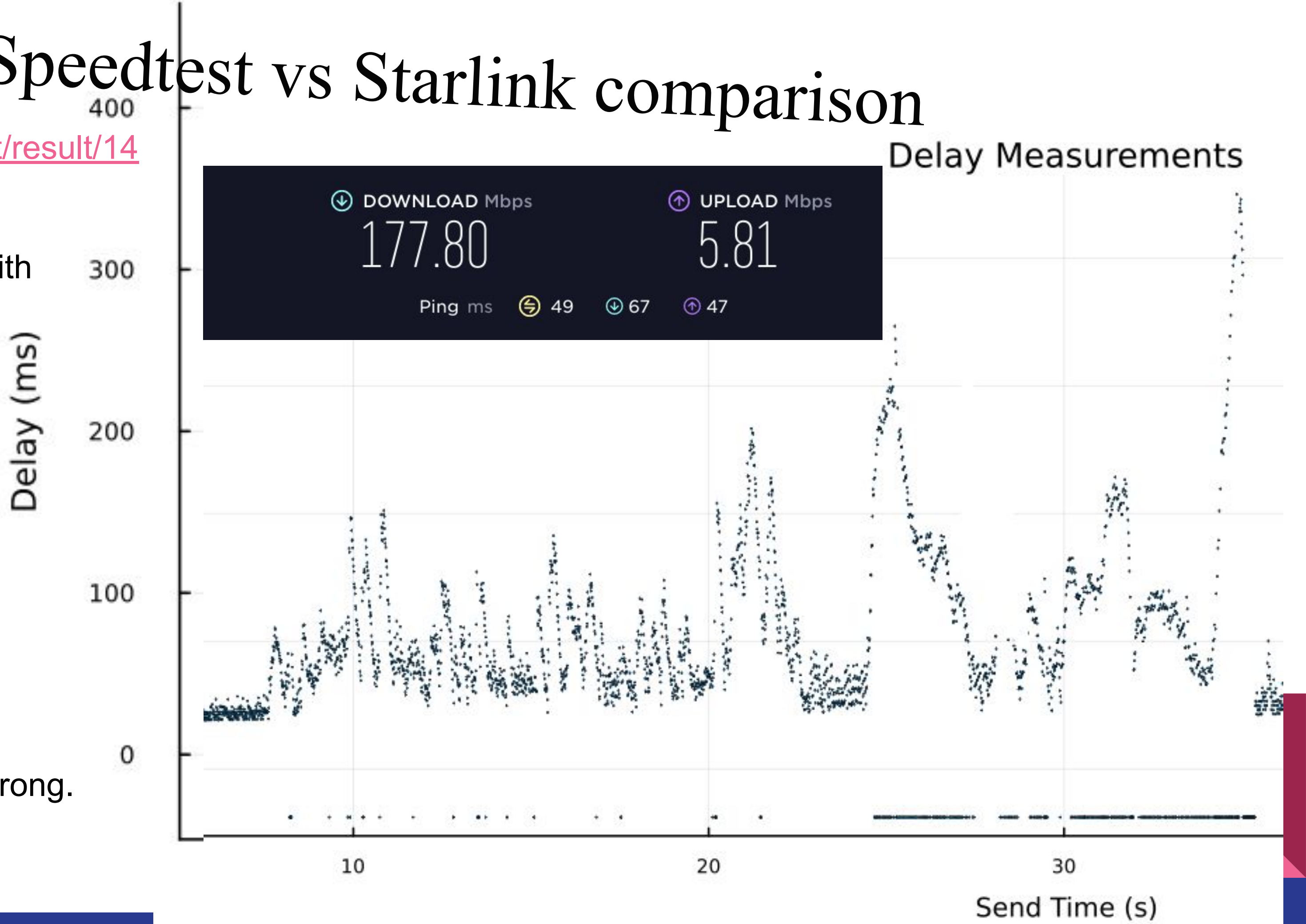
IRTT test simultaneous with Speedtest.net Test
IRTT shows 400ms peak Latency.
Speedtest result (100ms intervals?)

Doesn't.

This is a repeatable test.

Please try it at home.

One of us is measuring wrong.
Interquartile mean?



Please use the Nyquist Theorems?

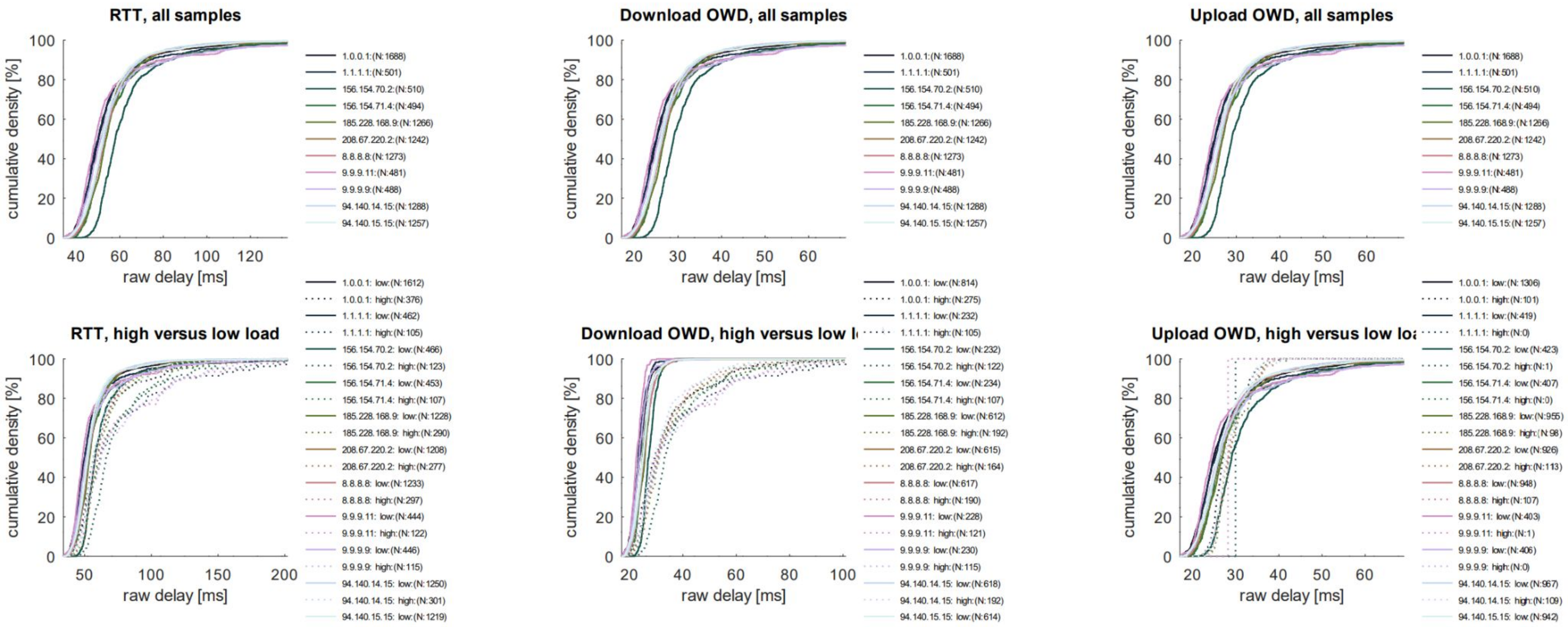
- In signal processing, the Nyquist frequency (or folding frequency), named after Harry Nyquist, is a characteristic of a sampler, which converts a continuous function or signal into a discrete sequence. For a given sampling rate (samples per second), the Nyquist frequency (cycles per second), is the frequency whose cycle-length (or period) is twice the interval between samples, thus 0.5 cycle/sample. For example, audio CDs have a sampling rate of 44100 samples/second. At 0.5 cycle/sample, the corresponding Nyquist frequency is 22050 cycles/second (Hz). Conversely, the Nyquist rate for sampling a 22050 Hz signal is 44100 samples/second.[1][2][A] - Wikipedia
- The Nyquist Rate for VOIP is **10**ms. Videoconferencing, also.
- For the MetaVerse, it's **2**ms.
- For the Data Center, 10us is about as low as you can measure (ns is better)
- Not 100ms! Not 1s! Not 5 minutes! Not hourly! as is all too common today

Thank you!

Some Potentially Truthful Metrics

- DPH – Distortions per Hour
  - Any loss, jitter or delay in a stream of over 60ms away from the natural rate of 20ms is a “glitch”, a distortion, for VOIP, videoconferencing and gaming
- SPOM – Steady Packets over Milliseconds
  - Ingress rate of 1 130b packet per 10ms, egress within that 10ms = 1. Perfect delivery of those within that deadline is a 1.
  - With, and without load. Over periods as long as an hour or more.
- Web Page Load Time
  - With and without bidirectional load
- Continuous in-band measurement at the ISP and CPE

CDF plots I find very useful



My Stone Knives and Bearskins

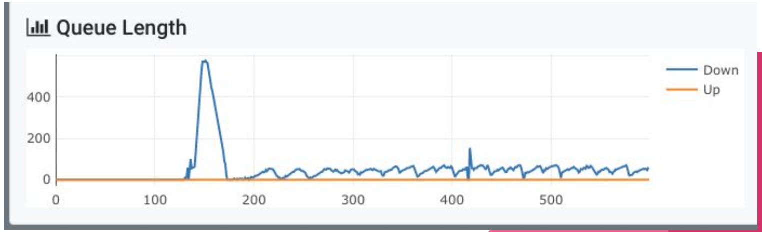
- Diagnostic Tools
  - IRTT and Flent (IRTT: STAMP superset)
  - OBUDPST (TR-471) and Crusader
  - IPERF2 --bounceback
  - TREXX
- Analysis Tools
  - Flent
  - Wireshark
  - TCPTrace

- Device Solutions
  - FQ_Codel & CAKE SQM
  - CAKE-Autorate
  - fq_codel on wifi
  - Whatever else you can get!

- ISP Solutions
  - RFC7567 everywhere
  - Ship better gear to customers
  - Shaping middleboxes everywhere else, like Preseem and LibreQos

Overthinking it – TCP Basics

- A single TCP cubic/reno/etc flow have **well defined behaviors** at any given RTT and bandwidth - Slow Start and Congestion Avoidance.
- Testing long enough - til time of first drop is good, a few drops better! (You might have to wait a looooong time)
- Staggered start testing is simple and revealing, also.
- Try those?



Some Tech Reports Worth Reading

- Analyzing the latency of sparse flows in FQ_CODEL
- SFQ_CODEL, PIE & Taildrop (Cablelabs)
- FQ_PIE
- 3 L4S RED TEAM REPORTS - Key Findings
- 3 Staggered Start TCP flows through PIE, CODEL, FQ_CODEL
- Updating the theory of buffersizing
- FQ_Codel Worldwide Report 2022
- Bufferbloat & Beyond FIXME: BJORN's Thesis is great
- Or "Bufferbloat" on google scholar!

Suggestions

- Build Relatable tools
- Measure at 10ms or less intervals (Nyquist Rate)
- Measure the needs of each application
- Put in better transports and FQ+AQM algorithms
- Focus on reducing glitches for a better user experience
- And validate your benchmarks against the others.

Summary

- Bandwidth is not speed. Latency matters to get good bandwidth.
- Network transfers are bound by physical round trip time, serialization delay, queuing delay, and retransmits
- Today's internet does congestion control (responses to overload) via packet loss and RFC3168 ECN marking.
- Every potential bottleneck link can use a fq + aqm queuing algorithm to reduce queuing delay, absorb transient shocks, and reduce or eliminate packet loss. See: RFC7567, RFC8289, RFC8033, RFC8290. But it doesn't help enough.

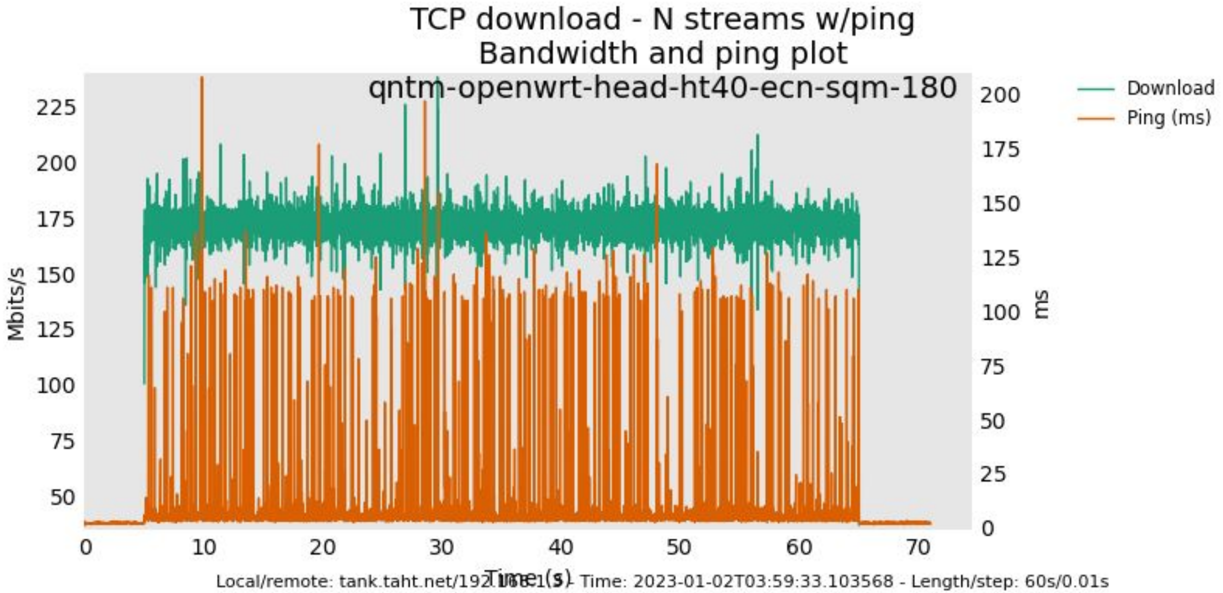
Some Network Nightmares
… the too many causes of latency and jitter

https://blog.cerowrt.org

Packets on LTE going 6 times around the planet…

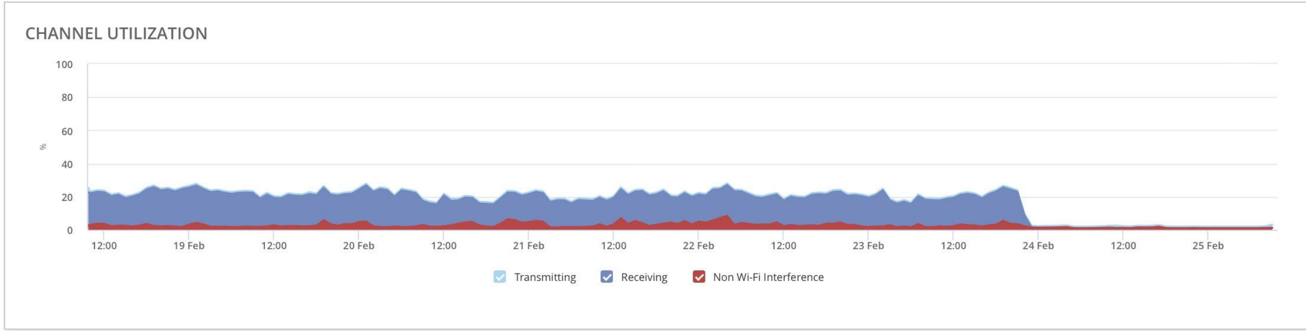
```
64 bytes from 8.8.8.8: icmp_seq=155 ttl=111 time=1459 ms
64 bytes from 8.8.8.8: icmp_seq=156 ttl=111 time=1071 ms
64 bytes from 8.8.8.8: icmp_seq=157 ttl=111 time=1055 ms
64 bytes from 8.8.8.8: icmp_seq=158 ttl=111 time=1160 ms
64 bytes from 8.8.8.8: icmp_seq=159 ttl=111 time=1344 ms
64 bytes from 8.8.8.8: icmp_seq=160 ttl=111 time=1464 ms
64 bytes from 8.8.8.8: icmp_seq=162 ttl=111 time=1718 ms
64 bytes from 8.8.8.8: icmp_seq=163 ttl=111 time=1005 ms
64 bytes from 8.8.8.8: icmp_seq=164 ttl=111 time=1442 ms
64 bytes from 8.8.8.8: icmp_seq=166 ttl=111 time=1340 ms
64 bytes from 8.8.8.8: icmp_seq=167 ttl=111 time=1027 ms
64 bytes from 8.8.8.8: icmp_seq=168 ttl=111 time=1404 ms
64 bytes from 8.8.8.8: icmp_seq=169 ttl=111 time=1412 ms
64 bytes from 8.8.8.8: icmp_seq=170 ttl=111 time=937 ms
64 bytes from 8.8.8.8: icmp_seq=171 ttl=111 time=1576 ms
64 bytes from 8.8.8.8: icmp_seq=173 ttl=111 time=1366 ms
64 bytes from 8.8.8.8: icmp_seq=174 ttl=111 time=1104 ms
64 bytes from 8.8.8.8: icmp_seq=175 ttl=111 time=1216 ms
64 bytes from 8.8.8.8: icmp_seq=176 ttl=111 time=1284 ms
64 bytes from 8.8.8.8: icmp_seq=178 ttl=111 time=1533 ms
64 bytes from 8.8.8.8: icmp_seq=179 ttl=111 time=1572 ms
64 bytes from 8.8.8.8: icmp_seq=180 ttl=111 time=1326 ms
64 bytes from 8.8.8.8: icmp_seq=181 ttl=111 time=940 ms
64 bytes from 8.8.8.8: icmp_seq=182 ttl=111 time=993 ms
```

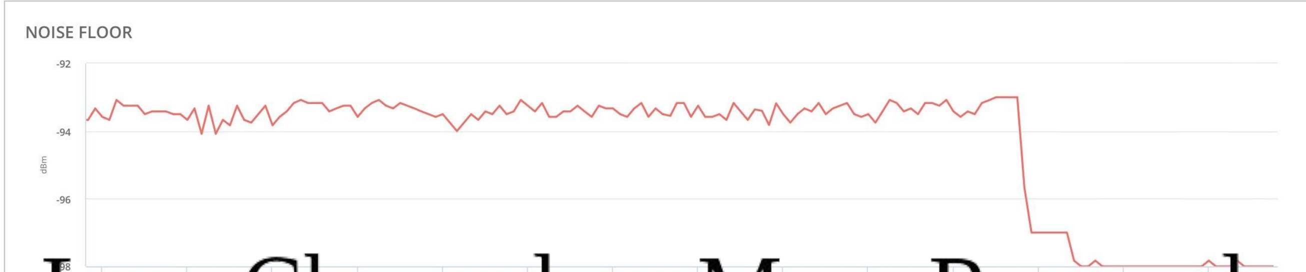
Great Bandwidth on Wifi…



Unusable jitter for VOIP!!

WiFi 80Mhz vs 40Mhz Noise Floor



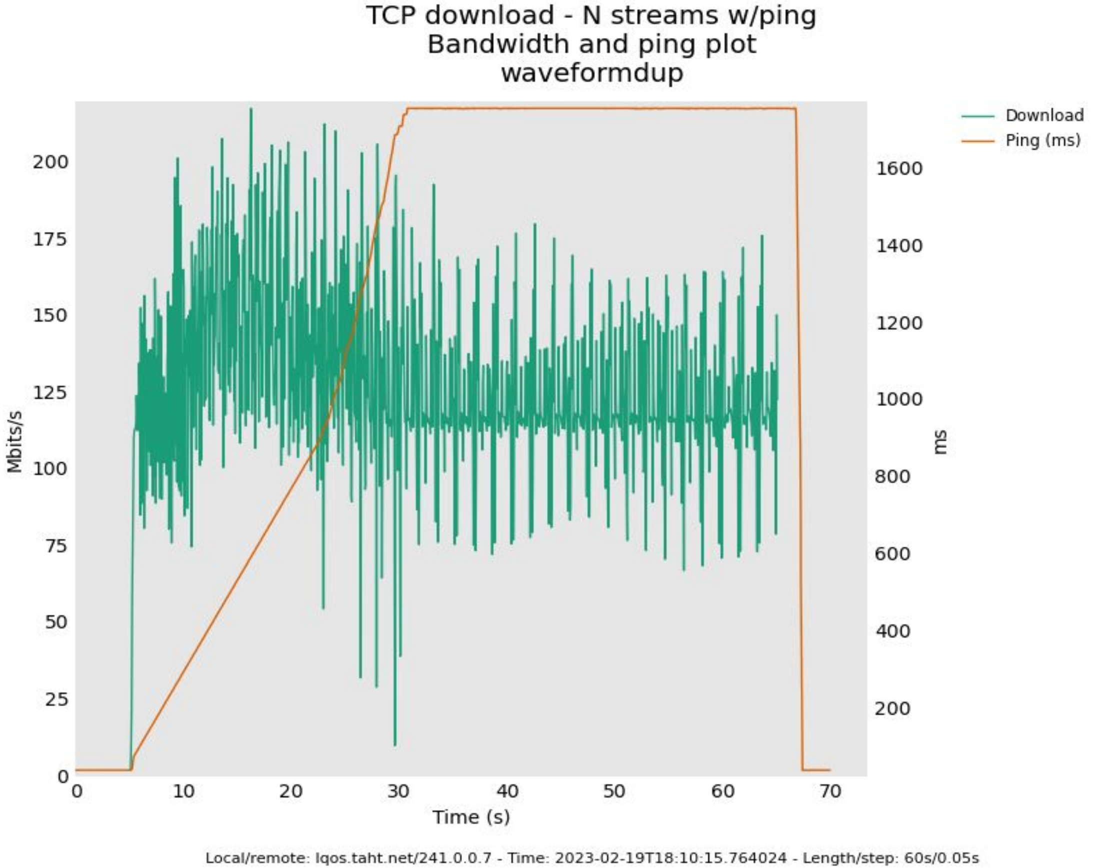


WiFi – Less Channels = More Range, less interference

Too long queues without signaling or Flow Queuing

Needs: 1024 Checkers, 64 bytes each

2 sec single flows w/infinite queues at 100Mbit



Billionaires building

Home routers

Out of 12 year old OpenWrt

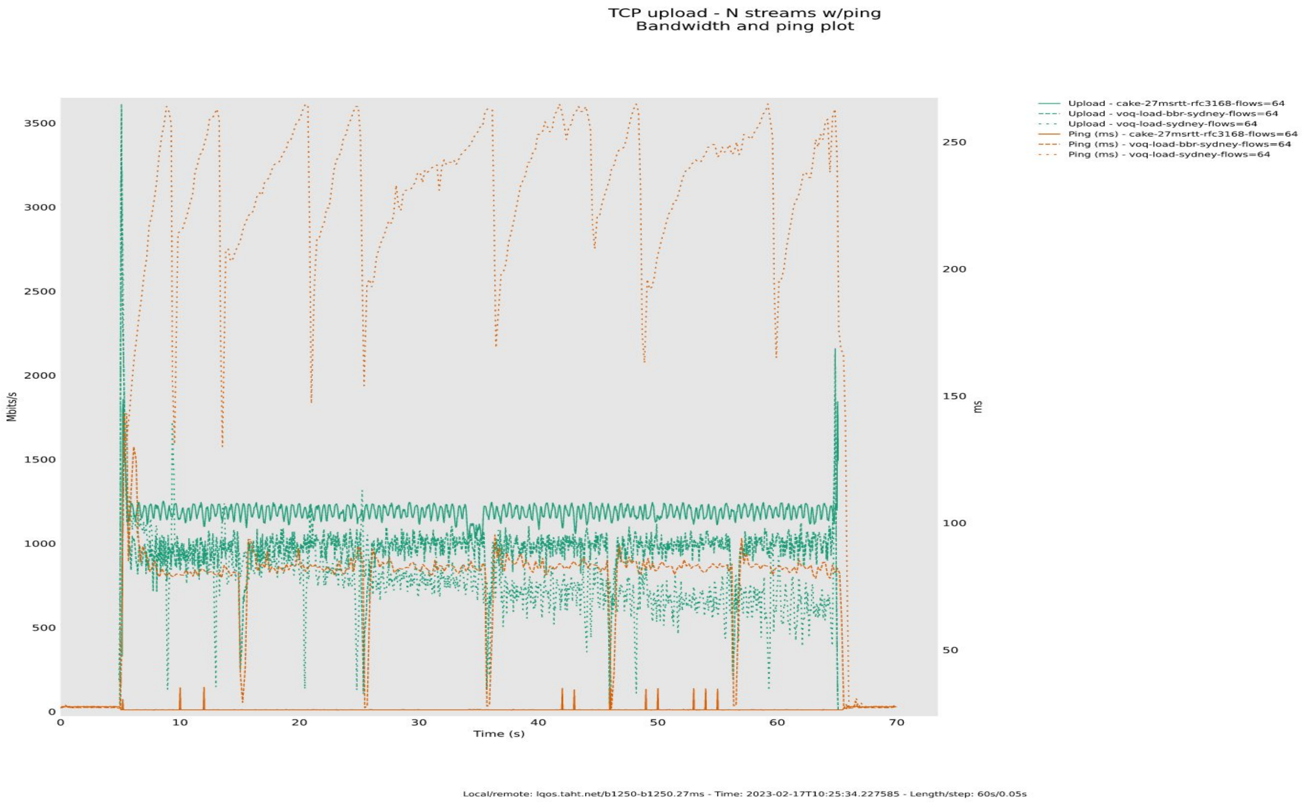
obsolete Software

Rife with bufferbloat…

CVEs…

And unable to handle IPv6

Network collapses over time and distance

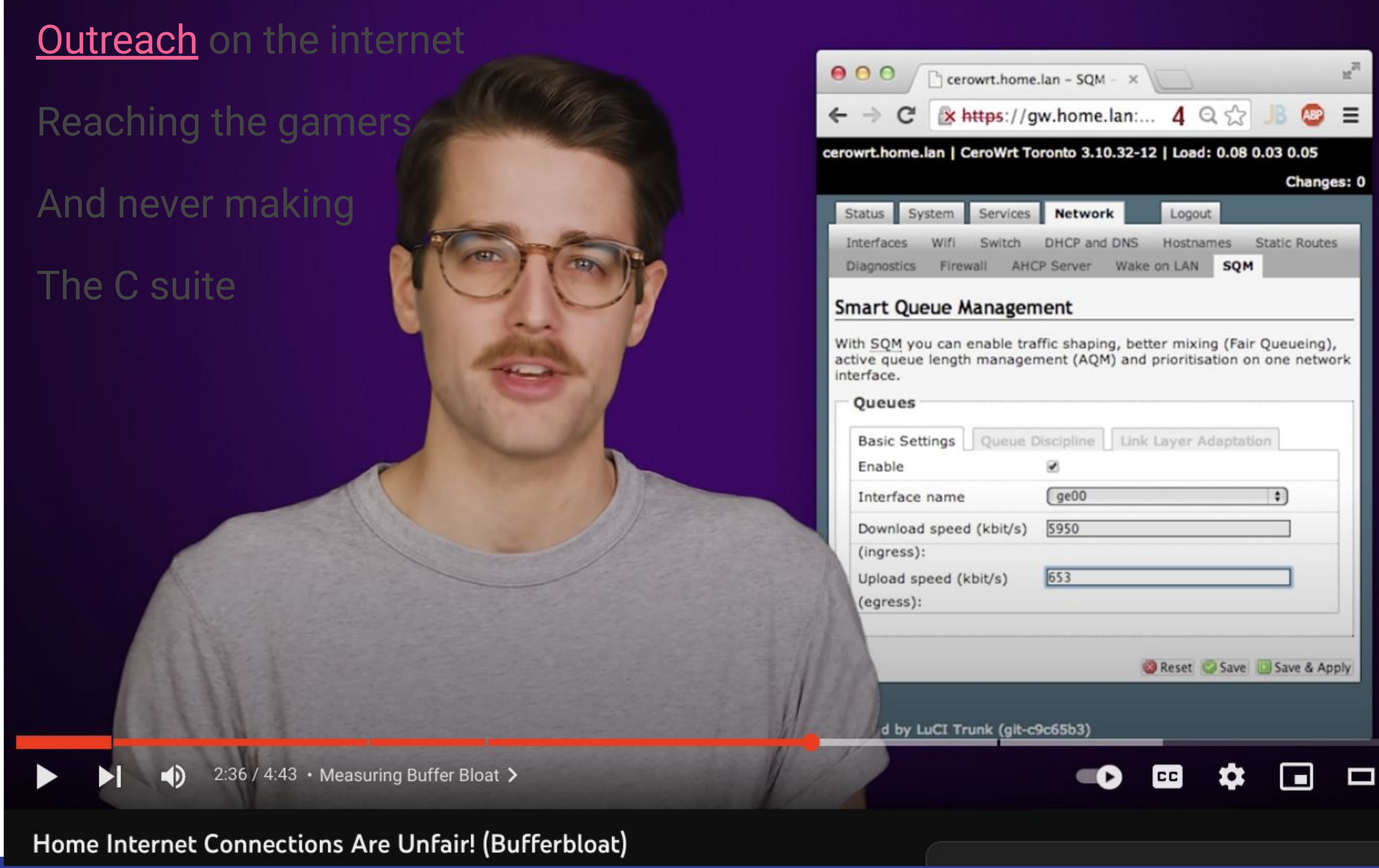


Outreach on the internet

Reaching the gamers

And never making

The C suite



Multiple specifications lacking implementations

A Non-Queue-Building Per-Hop Behavior (NQB PHB) for Differentiated Services

Abstract

This document specifies properties and characteristics of a Non-Queue-Building Per-Hop Behavior (NQB PHB). The purpose of this NQB PHB is to provide a separate queue that enables smooth, low-data-rate, application-limited traffic flows, which would ordinarily share a queue with bursty and capacity-seeking traffic, to avoid the latency, latency variation and loss caused by such traffic. This PHB is implemented without prioritization and can be implemented without rate policing, making it suitable for environments where the use of these features is restricted. The NQB PHB has been developed primarily for use by access network segments, where queuing delays and queuing loss caused by Queue-Building protocols are manifested, but its use is not limited to such segments. In particular, applications to cable broadband links, Wi-Fi links, and mobile network radio and core segments are discussed. This document recommends a specific Differentiated Services Code Point (DSCP) to identify Non-Queue-Building flows.¶

Internet Engineering Task Force (IETF)
Request for Comments: 9331
Category: Experimental
ISSN: 2070-1721

K. De Schepper
Nokia Bell Labs
B. Briscoe, Ed.
Independent

January 2023

The Explicit Congestion Notification (ECN) Protocol for Low Latency, Low Loss, and Scalable Throughput (L4S)

Abstract

This specification defines the protocol to be used for a new network service called Low Latency, Low Loss, and Scalable throughput (L4S). L4S uses an Explicit Congestion Notification (ECN) scheme at the IP layer that is similar to the original (or 'Classic') ECN approach, except as specified within. L4S uses 'Scalable' congestion control, which induces much more frequent control signals from the network, and it responds to them with much more fine-grained adjustments so that very low (typically sub-millisecond on average) and consistently low queuing delay becomes possible for L4S traffic without compromising link utilization. Thus, even capacity-seeking (TCP-like) traffic can have high bandwidth and very low delay at the same time, even during periods of high traffic load.

The L4S identifier defined in this document distinguishes L4S from 'Classic' (e.g., TCP-Reno-friendly) traffic. Then, network bottlenecks can be incrementally modified to distinguish and isolate existing traffic that still follows the Classic behaviour, to prevent it from degrading the low queuing delay and low loss of L4S traffic. This Experimental specification defines the rules that L4S transports and network elements need to follow, with the intention that L4S flows neither harm each other's performance nor that of Classic traffic. It also suggests open questions to be investigated during experimentation. Examples of new Active Queue Management (AQM) marking algorithms and new transports (whether TCP-like or real time)

Low Latency, Low Loss, and Scalable Throughput (L4S) Internet Service: Architecture

Abstract

This document describes the L4S architecture, which enables Internet applications to achieve low queuing latency, low congestion loss, and scalable throughput control. L4S is based on the insight that the root cause of queuing delay is in the capacity-seeking congestion controllers of senders, not in the queue itself. With the L4S architecture, all Internet applications could (but do not have to) transition away from congestion control algorithms that cause substantial queuing delay and instead adopt a new class of congestion controls that can seek capacity with very little queuing. These are aided by a modified form of Explicit Congestion Notification (ECN) from the network. With this new architecture, applications can have both low latency and high throughput.

The architecture primarily concerns incremental deployment. It defines mechanisms that allow the new class of L4S congestion controls to coexist with 'Classic' congestion controls in a shared network. The aim is for L4S latency and throughput to be usually much better (and rarely worse) while typically not impacting Classic performance.

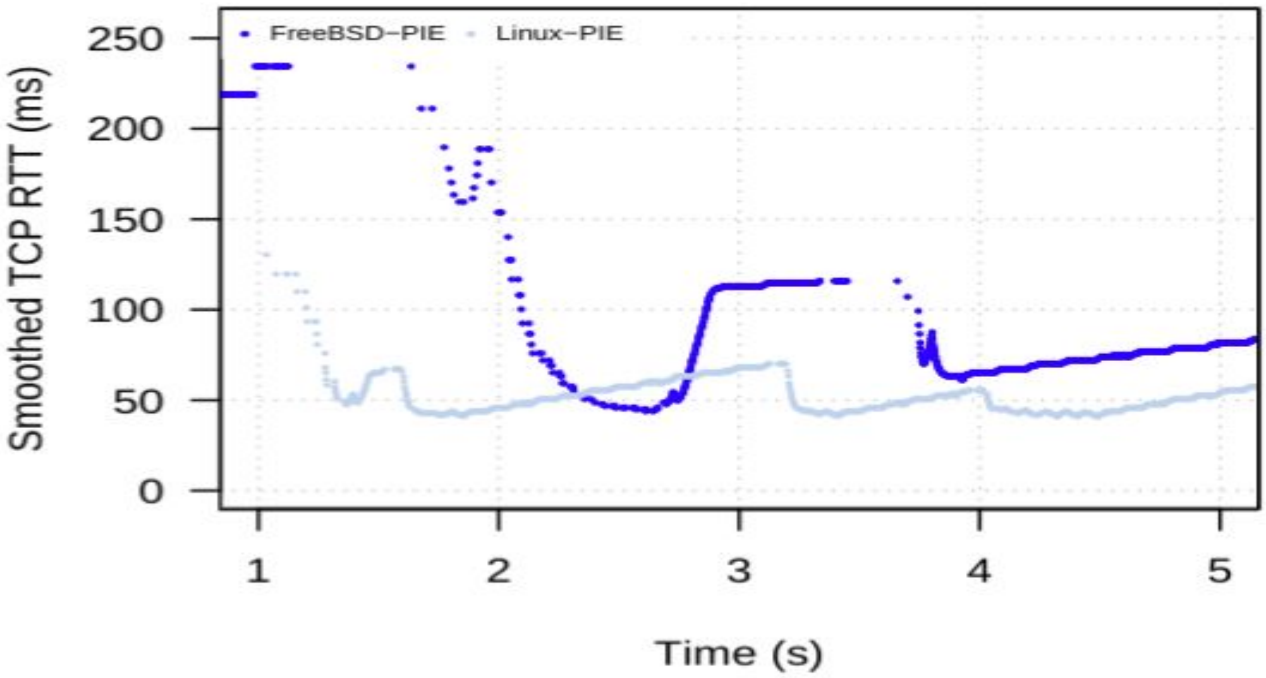
Pie AQM, BSD vs Linux

(b) CWND vs. Time from t=0 to t=5

BROKEN

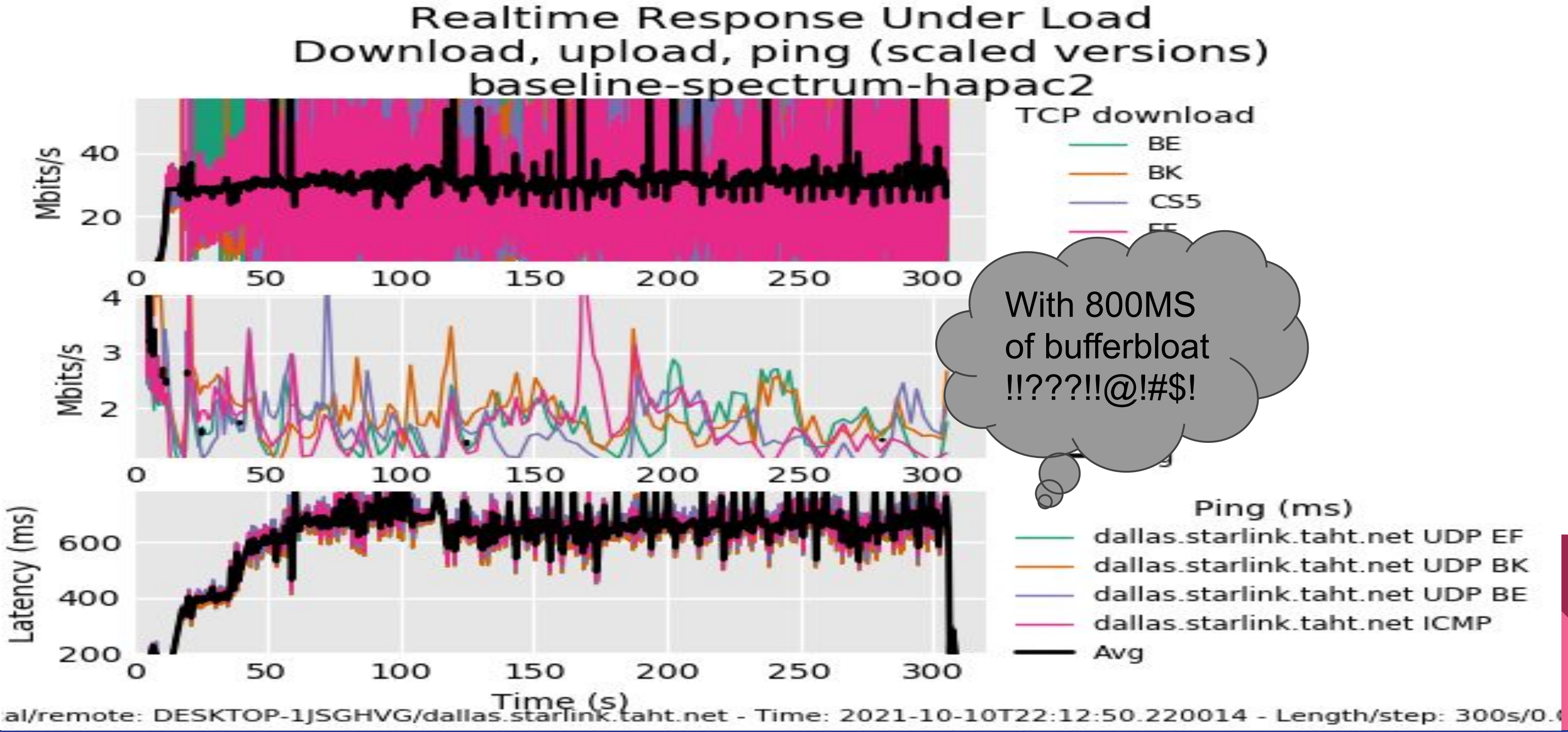
AQM

CODE

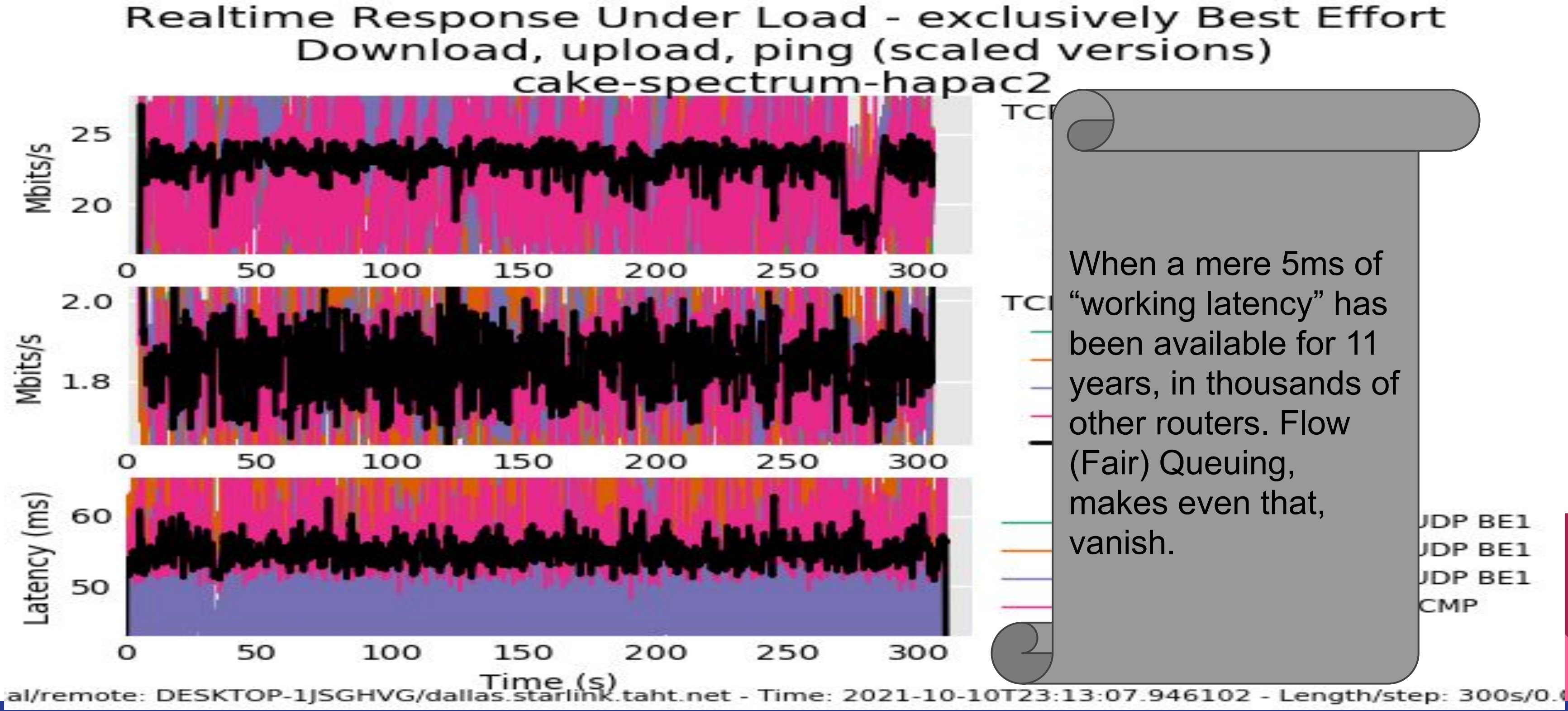


(d) smoothed TCP RTT vs. Time from t=1 to t=5

100/20 Service mandated by FCC



100/20 w/cake, Spectrum Cable



When a mere 5ms of "working latency" has been available for 11 years, in thousands of other routers. Flow (Fair) Queuing, makes even that, vanish.

Still...

All these things, will be

fixed

with Nyquist sampling...

Flow Queuing

AQM

Packet Captures

hard work...

more eyeballs, and

[more tears on the train.]()

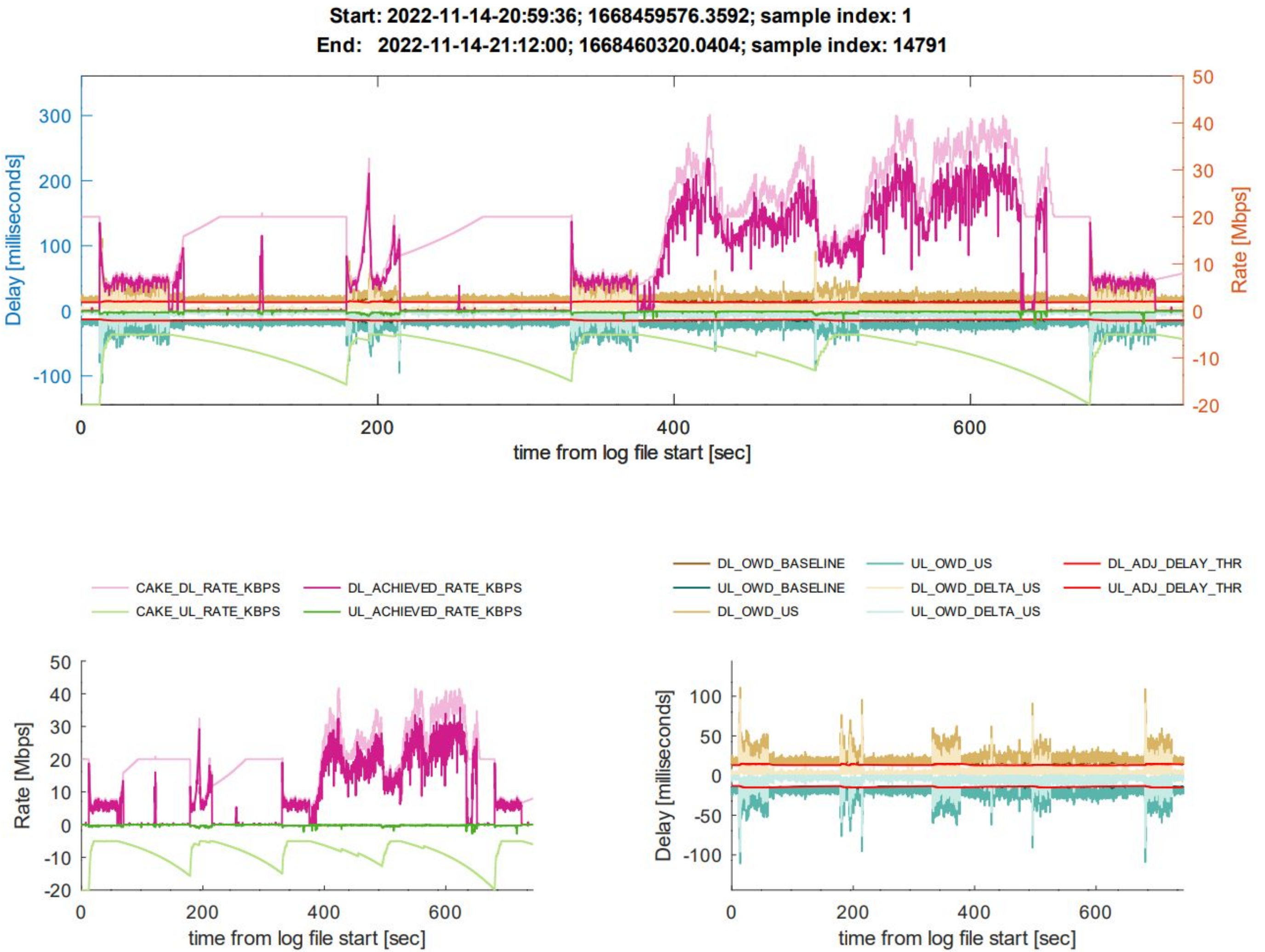
Thank You

We can get out of this bloat together!

LibreQoS.io
Fast, Flexible QoE

SQM AUTO RATE

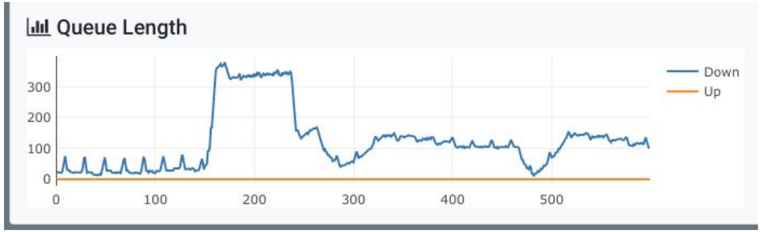
LTE Active Sensing

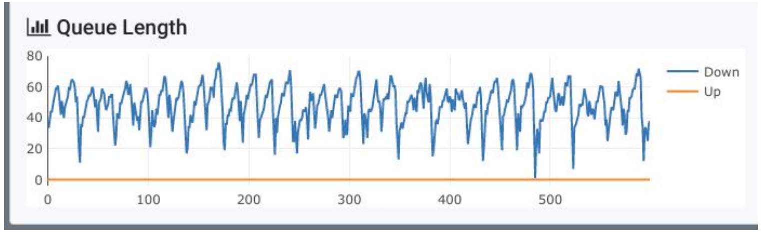


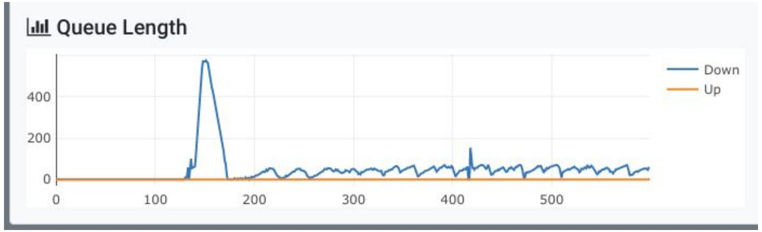
Worries - UnderBloat!

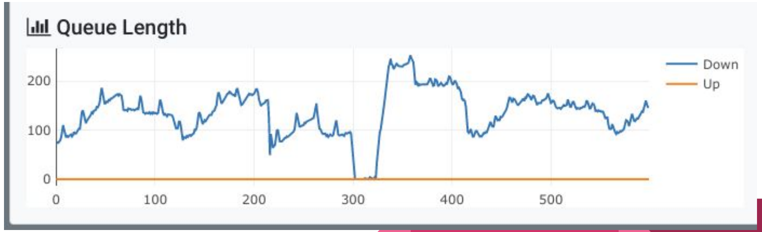
- All our speedtests start up 16 or more flows to get a result in under 10 seconds.
- I have now seen multiple fiber links with only 5ms of buffering instead of a BDP.
- This kind of indirectly proves that Nick Mckeown's theory of buffersizing is correct!
- But it doesn't help to have such small buffers for one flow!
- While 1 BDP is no longer needed for paced transports (linux mostly), an outer limit (lacking quality AQM) of 20-60ms for BFIFOs would be helpful going forward to realize true bandwidth gains from more bandwidth.

What TCP behaviors do these traces describe?

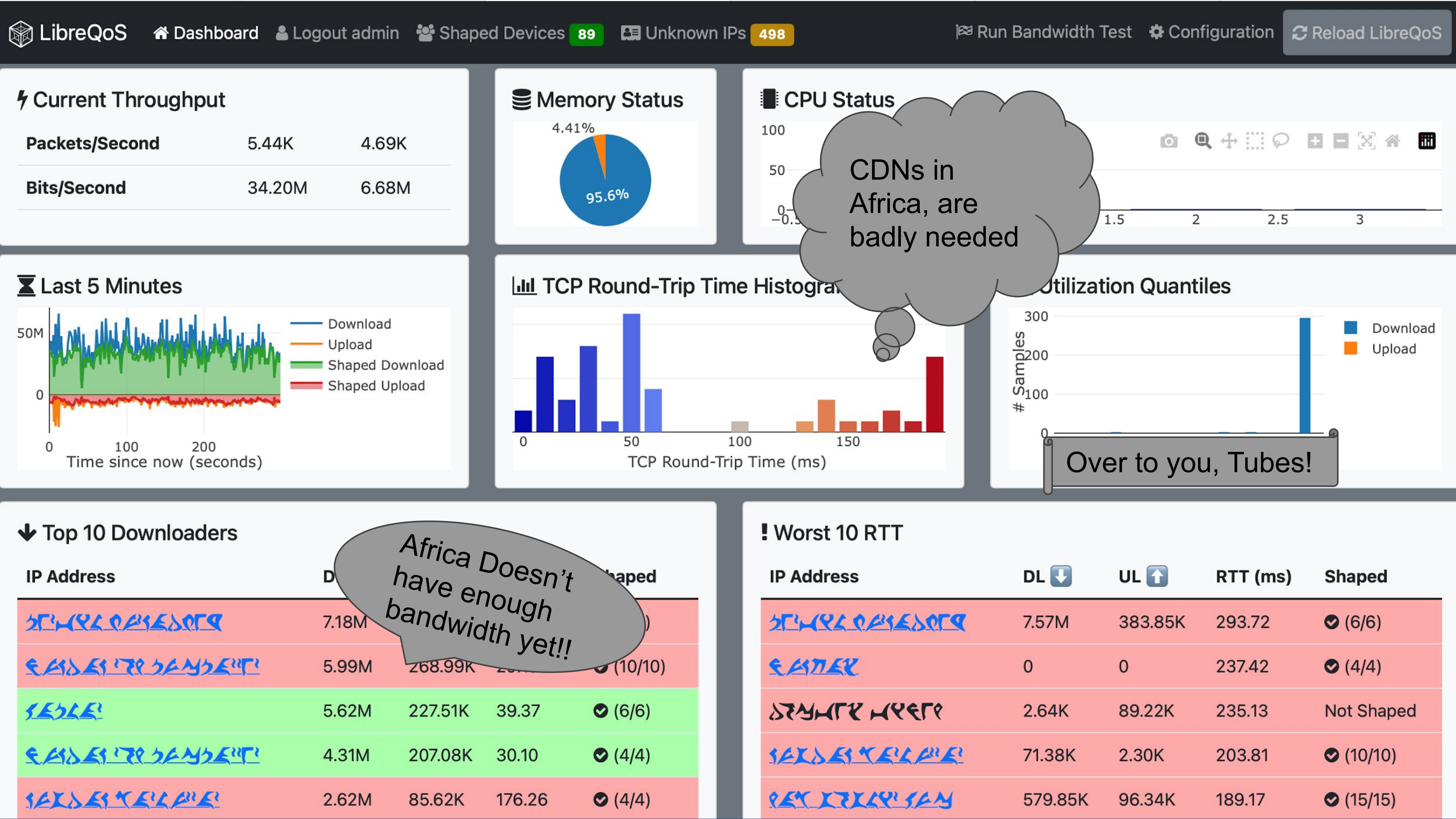








All these bloats are fixed with Flow Queueing (FQ)...


LibreQoS
Dashboard
Logout admin
Shaped Devices 89
Unknown IPs 498
Run Bandwidth Test
Configuration
Reload LibreQoS
Current Throughput
Packets/Second 5.44K 4.69K
Bits/Second 34.20M 6.68M
Memory Status
4.41%
95.6%
CPU Status
CDNs in Africa, are badly needed
Last 5 Minutes
Download
Upload
Shaped Download
Shaped Upload
Time since now (seconds)
TCP Round-Trip Time Histogra
TCP Round-Trip Time (ms)
Utilization Quantiles
Samples
Over to you, Tubes!
Top 10 Downloaders
IP Address
Africa Doesn't have enough bandwidth yet!!
7.18M
5.99M 268.99K (10/10)
5.62M 227.51K 39.37 (6/6)
4.31M 207.08K 30.10 (4/4)
2.62M 85.62K 176.26 (4/4)
Worst 10 RTT
IP Address DL UL RTT (ms) Shaped
7.57M 383.85K 293.72 (6/6)
0 0 237.42 (4/4)
2.64K 89.22K 235.13 Not Shaped
71.38K 2.30K 203.81 (10/10)
579.85K 96.34K 189.17 (15/15)