Honouring the life of Dave Täht

Toke Høiland-Jørgensen - Red Hat

Who was Dave Täht?

- Bufferbloat project co-founder and “nagger in chief”
- Space nerd and all-round dreamer
- Musician and songwriter

Dave passed away on April 1st, 2025, aged 59.

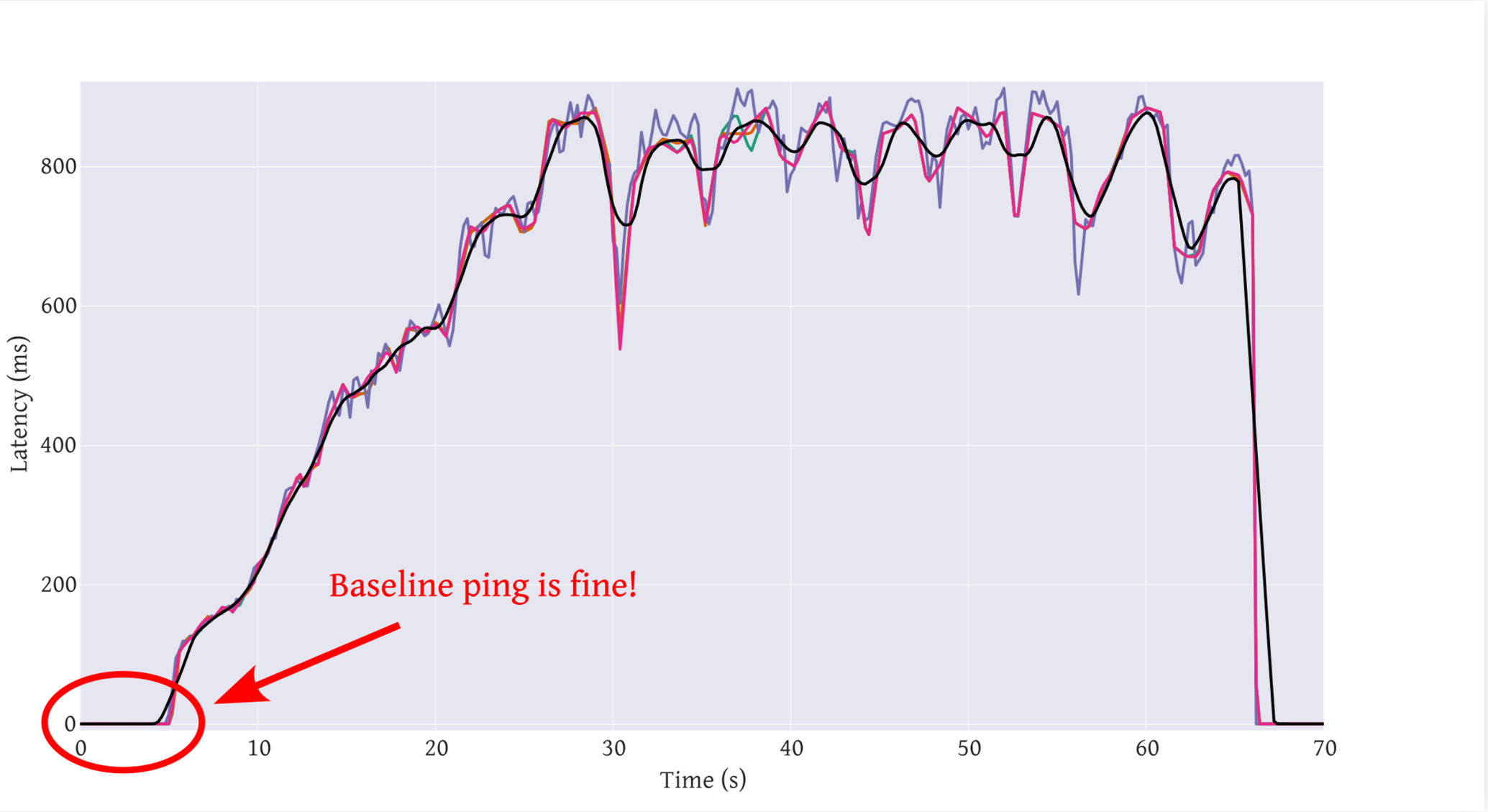
How I met Dave

So sure, come on down. The only thing I have on my plate is defining a spec for a bufferbloat test - and raising awareness of fq_codel and BQL while I'm here. So our interests are congruent.

The food at the reception was good… and free.

Collaborating with Dave

A brief history of Bufferbloat



Dark buffers in the Internet (2011)

acmqueue

Bufferbloat: Dark Buffers in the Internet

Networks without effective AQM may again be vulnerable to congestion collapse.

Jim Gettys, Bell Labs, Alcatel-Lucent; and Kathleen Nichols, Pollere Inc.

Today's networks are suffering from unnecessary latency and poor system performance. The culprit is bufferbloat, which is the existence of excessively large and frequently full buffers inside the network. Large buffers have been inserted all over the Internet without sufficient thought or testing. They damage or defeat the fundamental congestion-avoidance algorithms of the Internet's most common transport protocol. Long delays from bufferbloat are frequently attributed incorrectly to network congestion, and this misinterpretation of the problem leads to the wrong solutions being proposed.

Congestion is an old problem on the Internet, appearing in various forms with different symptoms and causing major problems. Buffers are essential to the proper functioning of packet networks, but overly large, unmanaged, and uncoordinated buffers create excessive delays that frustrate and baffle end users. Many of the issues that create delay are not new, but their collective impact has not been widely understood. Thus, buffering problems have been accumulating for more than a decade. We strive to present these problems with their impacts so that the community can understand and act upon the problem and, we hope, learn to prevent future problems.

This article does not claim to be the first to identify the problems of excessive buffering, but it is an attempt to create a wider understanding of the pervasive problem and to give a call to action.

https://dl.acm.org/doi/10.1145/2063166.2071893

The CeroWrt project

CeroWrt 3.10 Release Notes

CeroWrt 3.10 Beta Test Release Notes

Current version is **3.10.50-1**, built on 28 July 2014. The current build can be downloaded from: http://snapon.lab.bufferbloat.net/~cero2/cerowrt/wndr See the **Status** section (below) for more information.

About CeroWrt

CeroWrt is a wireless router OS built on the OpenWrt firmware . It is a research project intended to resolve the bufferbloat epidemic in home networking today, and to push forward the state of the art of edge networks and routers. Sub-projects include proper IPv6 support, tighter integration with DNSSEC, and most importantly, reducing bufferbloat in both the wired and wireless components of the stack.

Features

- High performance routing in a relatively inexpensive “home” router - the Netgear WNDR3800.
- A major improvement to the problem of bufferbloat. VoIP, Skype, gaming, and other latency-sensitive applications continue to work well even during heavy up/download.
- IPv6 support. Another major goal of CeroWrt is to make IPv6 networking in the home as simple as IPv4. IPv6 subnet assignment and other features are enabled, and have been extensively tested on comcast’s deployment.
- Linux 3.10.x kernel. Many of the fixes for bufferbloat have been implemented in mainline Linux. This means that bufferbloat is improving for the rest of the world. http://kernel.org
- CeroWrt defaults to the fq_codel queueing discipline that implements the Codel algorithm from Kathie Nichols and Van Jacobson along with Eric Dumazet’s adaptation of Fair Queueing (fq_codel) on top.
- CeroWrt also includes these queueing disciplines for experimentation: fq_codel, efq_codel, nfq_codel, sfq, codel, ns2_codel, RED, ARED, SFQRED, QFQ and Cisco’s PIE.
- Babel routing protocol with source specific routing support (babels). Source Sensitive Routing allows for multiple exit nodes on IPv6 and IPv4 among other things.
- Improved DNS handling by incorporating dnsmasq for both DNS and DHCP support. CeroWrt 3.10 enables DNSSEC by default, but see note in the Status section.
- Incorporates Best Common Practices 38 (BCP38) to defeat Denial of Service attacks which employ IP Source Address Spoofing.
- Adequate entropy for the random number generators, for better encryption (WPA, SSL), ethernet drivers, etc.
- All the features expected from a modern small office/home (SOHO) router:
  - Dynamic DNS to establish a static DNS name even if the IP address from your IPS changes
  - UPnP (Universal Plug and Play) and SSDP proxy that allows DLNA discovery across CeroWRT’s routed (not bridged) interfaces
  - mDNS (multicast DNS) that both allow other computers on the local network to find each other
  - Polipo caching web proxy
  - All the features of the OpenWrt distribution, including the attractive LuCI web GUI for configuration. We track the OpenWrt development code base (“Barrier Breaker”) and incorporate the capabilities of that distribution. http://openwrt.org and http://wiki.openwrt.org/doc/howto/luci.essentials
- CeroWrt has a broad set of useful packages built-in or optionally loaded. See the list of Major Packages below.

Make-wifi-fast

Make Wi-Fi Fast Project

This project focuses on reducing latency throughout the wifi stack, firmware, and hardware.

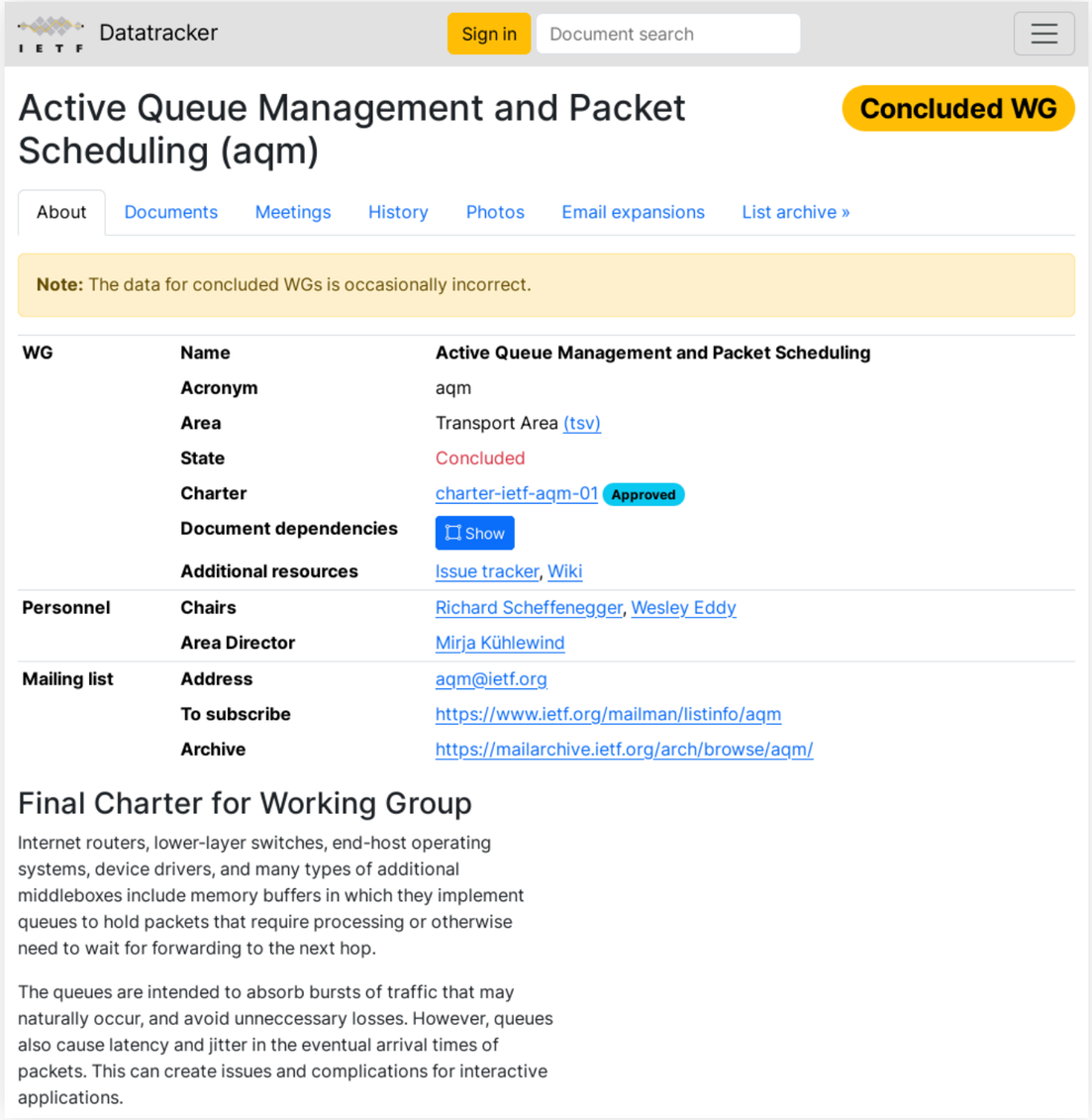
> **The Make Wi-Fi Fast Manifesto - Wi-Fi does not need to be slow!**
>
> The hardware now available for Wi-Fi can accomplish tremendous performance, but it is hobbled by software designs that guarantee high latency under load. This, in turn, dramatically lowers performance in real-world settings (multiple users, home routers, commercial access points) leading to the *myth* that "Wi-Fi is slow."
>
> We believe that the same sort of systems thinking that went on in the Bufferbloat Project can lead to performance improvements of an order of magnitude or more in Wi-Fi.

Current Status

In late 2024, we have decreased Wi-Fi latency by at least an order of magnitude, with fair sharing of airtime across fast and slow devices.

- Working software is available in the Linux kernel, as well as the OpenWrt firmware that runs on off-the-shelf routers, x86 boxes, and embedded systems.
- An academic paper describing the mechanism has been published at the 2017 USENIX Annual Technical Conference: Ending the Anomaly: Achieving Low Latency and Airtime Fairness in Wifi.
- Ongoing work was discussed at the Linux network developers conference (NetDev 2.2) in November 2017. The session was recorded and is available from the NetDev web site.

IETF AQM working group



Datatracker Sign in Document search

Active Queue Management and Packet Scheduling (aqm)

Concluded WG

About | Documents | Meetings | History | Photos | Email expansions | List archive »

Note: The data for concluded WGs is occasionally incorrect.

WG	**Name**	**Active Queue Management and Packet Scheduling**
	Acronym	aqm
	Area	Transport Area (tsv)
	State	Concluded
	Charter	charter-ietf-aqm-01 Approved
	Document dependencies	Show
	Additional resources	Issue tracker, Wiki
Personnel	**Chairs**	Richard Scheffenegger, Wesley Eddy
	Area Director	Mirja Kühlewind
Mailing list	**Address**	aqm@ietf.org
	To subscribe	https://www.ietf.org/mailman/listinfo/aqm
	Archive	https://mailarchive.ietf.org/arch/browse/aqm/

Final Charter for Working Group

Internet routers, lower-layer switches, end-host operating systems, device drivers, and many types of additional middleboxes include memory buffers in which they implement queues to hold packets that require processing or otherwise need to wait for forwarding to the next hop.

The queues are intended to absorb bursts of traffic that may naturally occur, and avoid unneccessary losses. However, queues also cause latency and jitter in the eventual arrival times of packets. This can create issues and complications for interactive applications.

LibreQoS


LibreQoS
Product
LTS
Company
Chat
Social
News
GitHub
Contact
LibreQoS Reshapes Networks
Fix Latency Issues
Mitigate both latency and latency-under-load for all congestion points on your network fabric.
Stop Excessive Customer Calls
"All our tech support calls about 'speed' just vanished." — New Zealand ISP Operator
Solve Customer Dissatisfaction
Solve issues before they impact customers - Take a proactive approach with the industry-leading QoE solution.

Bufferbloat mitigation innovations in Linux

Codel and PIE and their FQ variants

FIGURE 2 TCP Connection After One RTT

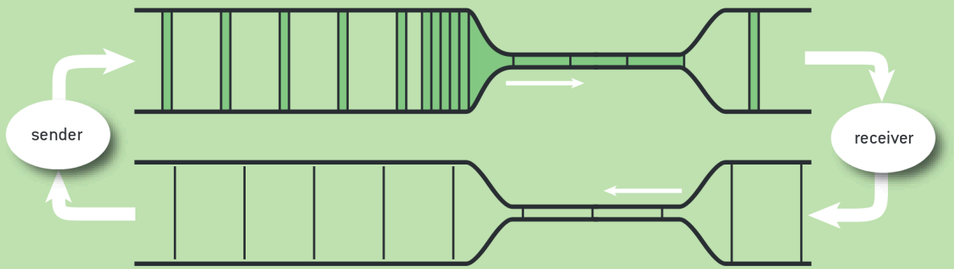
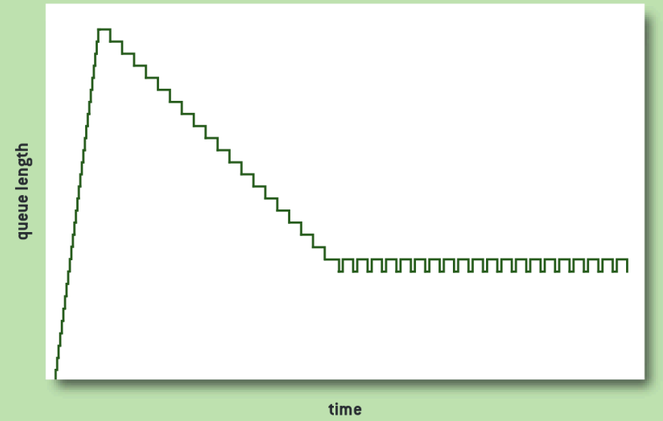


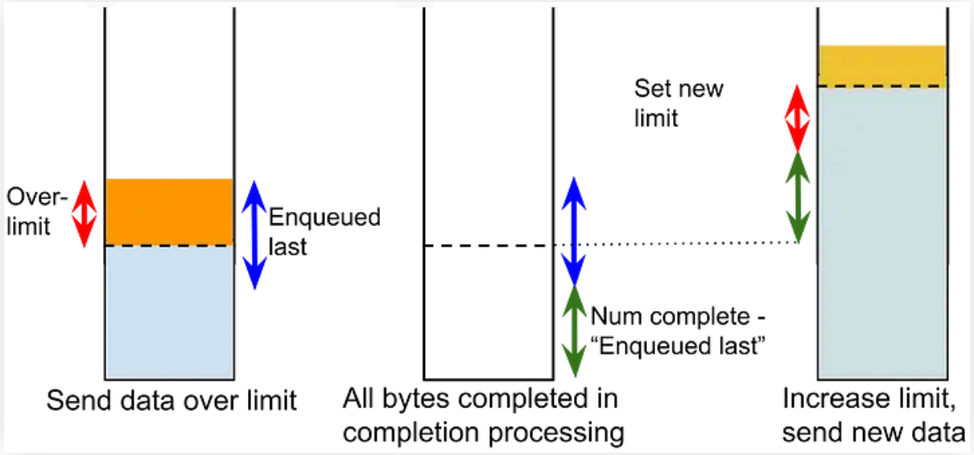
FIGURE 3 Queue Size vs. Time

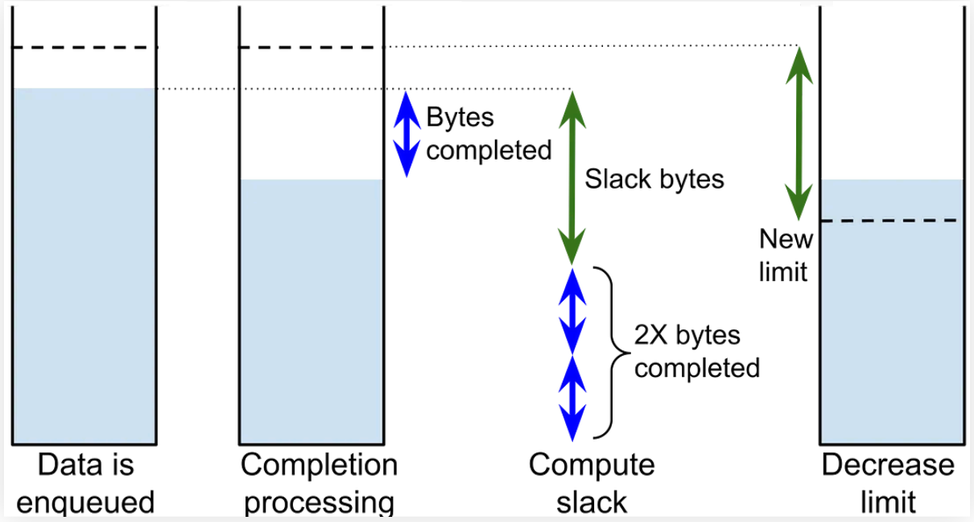


https://queue.acm.org/detail.cfm?id=2209336

RFC 8289 and 8290

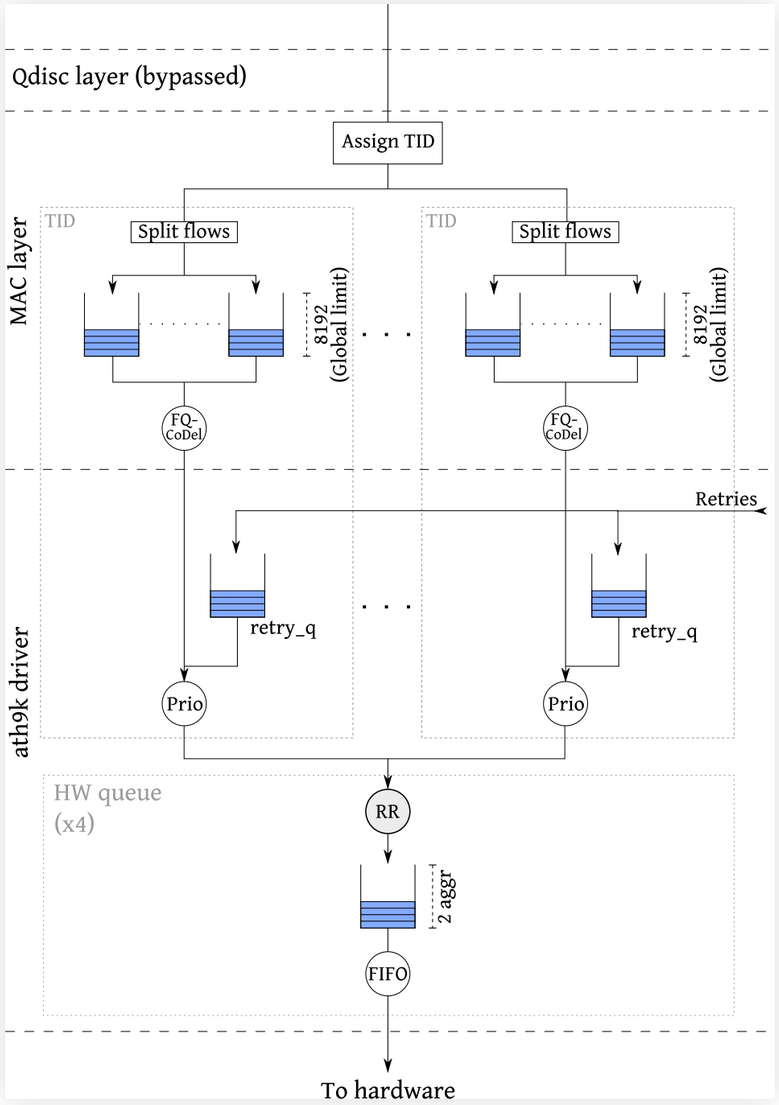
BQL





https://medium.com/@tom_84912/byte-queue-limits-the-unauthorized-biography-61adc5730b83

WiFi queueing and AQL



https://www.usenix.org/conference/atc17/technical-sessions/presentation/hoilan-jorgesen

TCP small queues and pacing

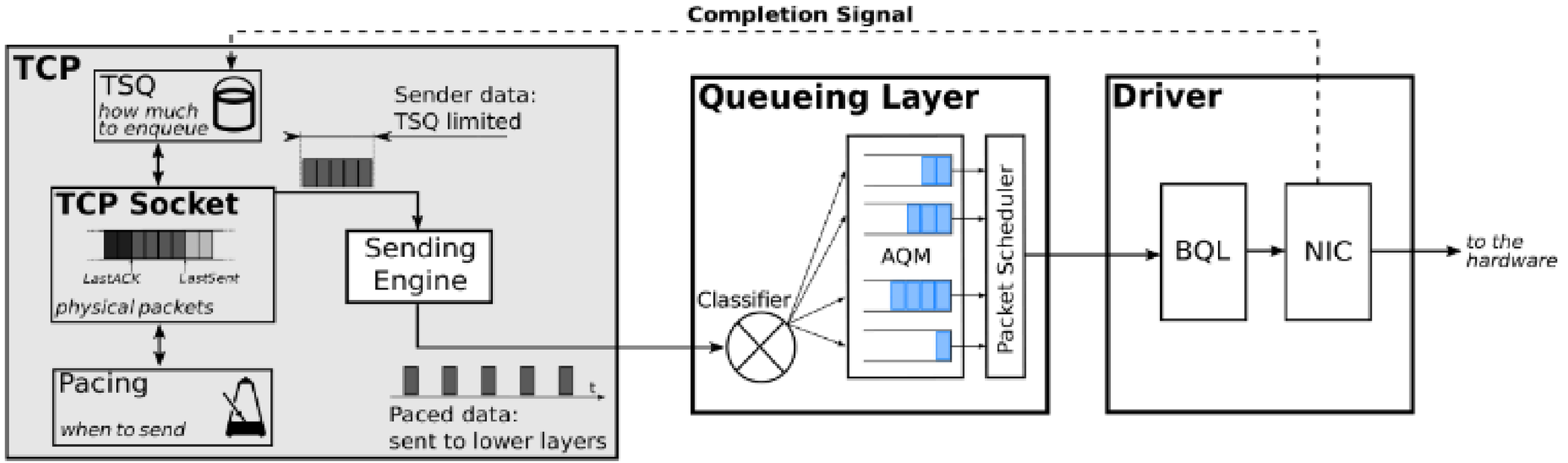


FIGURE 1. TCP-IP linux stack.

https://doi.org/10.1109/ACCESS.2021.3113891

BBR and other congestion controls

BBR
Congestion-Based
Congestion Control

MEASURING BOTTLENECK BANDWIDTH AND ROUND-TRIP PROPAGATION TIME

NEAL CARDWELL
YUCHUNG CHENG
C. STEPHEN GUNN
SOHEIL HASSAS YEGANEH
VAN JACOBSON

By all accounts, today's Internet is not moving data as well as it should. Most of the world's cellular users experience delays of seconds to minutes; public Wi-Fi in airports and conference venues is often worse. Physics and climate researchers need to exchange

CAKE

Piece of CAKE: A Comprehensive Queue Management Solution for Home Gateways

Toke Høiland-Jørgensen
Dept. of Computer Science
Karlstad University, Sweden
toke.hoiland-jorgensen@kau.se

Dave Täht
Teklibre
Los Gatos, California
dave.taht@gmail.com

Jonathan Morton
Somero, Finland
chromatix99@gmail.com

***Abstract*—The last several years has seen a renewed interest in smart queue management to curb excessive network queueing delay, as people have realised the prevalence of bufferbloat in real networks.**

However, for an effective deployment at today's last mile connections, an improved queueing algorithm is not enough in itself, as often the bottleneck queue is situated in legacy systems that cannot be upgraded. In addition, features such as per-user fairness and the ability to de-prioritise background traffic are often desirable in a home gateway.

In this paper we present Common Applications Kept Enhanced (CAKE), a *comprehensive network queue management system* designed specifically for home Internet gateways. CAKE packs several compelling features into an integrated solution, thus easing deployment. These features include: bandwidth shaping with overhead compensation for various link layers; reasonable DiffServ handling; improved flow hashing with both per-flow and per-host queueing fairness; and filtering of TCP ACKs.

Our evaluation shows that these features offer compelling advantages, and that CAKE has the potential to significantly improve performance of last-mile internet connections.

compelling benefit of CAKE is that it takes state of the art solutions and integrates them to provide:

- a high-precision rate-based bandwidth shaper that includes overhead and link layer compensation features for various link types.
- a state of the art fairness queueing scheme that simultaneously provides both host and flow isolation.
- a Differentiated Services (DiffServ) prioritisation scheme with rate limiting of high-priority flows and work-conserving bandwidth borrowing behaviour.
- TCP ACK filtering that increases achievable throughput on highly asymmetrical links.

CAKE is implemented as a *queueing discipline* (qdisc) for the Linux kernel. It has been deployed as part of the OpenWrt router firmware for the last several years and is in the process of being submitted for inclusion in the mainline Linux kernel.[1]

The rest of this paper describes the design and implementation of CAKE and is organised as follows: Section II outlines

Honouring Dave's memory

"A man is not dead while his name is still spoken."

-Going Postal, Chapter 4 prologue

Open and *repeatable* research

“Reproducible” vs “Repeatable” Experiments

- Researcher MIGHT be able to repeat experiment in light of new data.
- Reviewer typically lacks time to reproduce.
- Reader MIGHT be able, from the descriptions in the paper, reproduce the result, by rewriting the code from scratch.

- Researcher MUST be easily repeat the experiment in light of new data.
- Reviewer SHOULD be able to re-run the experiment and inspect the code.
- Reader SHOULD be able from the code and data supplied by the research, repeat the experiment quickly and easily.

Relentless focus on latency

Every moment of our lives is precious, and every moment spent waiting on a computer, wasted. We optimize our lives, almost subconsciously, in a quest for lower latency between what we want and how fast we get it.

[…]

In improving mankind's interpersonal latency, I'd like us to always aim at the speed of light, and ever closer to the speed of thought.

https://www.internetsociety.org/wp-content/uploads/2013/09/28_towards_imperceptible_latency.pdf

Further reading

- Dave's Wikipedia entry: https://en.wikipedia.org/wiki/Dave_Taht
- Bufferbloat web site: https://bufferbloat.net
- My own PhD: https://bufferbloat-and-beyond.net/

“You know they’ll never really die while the Trunk is alive[…] It lives while the code is shifted, and they live with it, always Going Home.”

-Moist von Lipwig, Going Postal, Chapter 13