

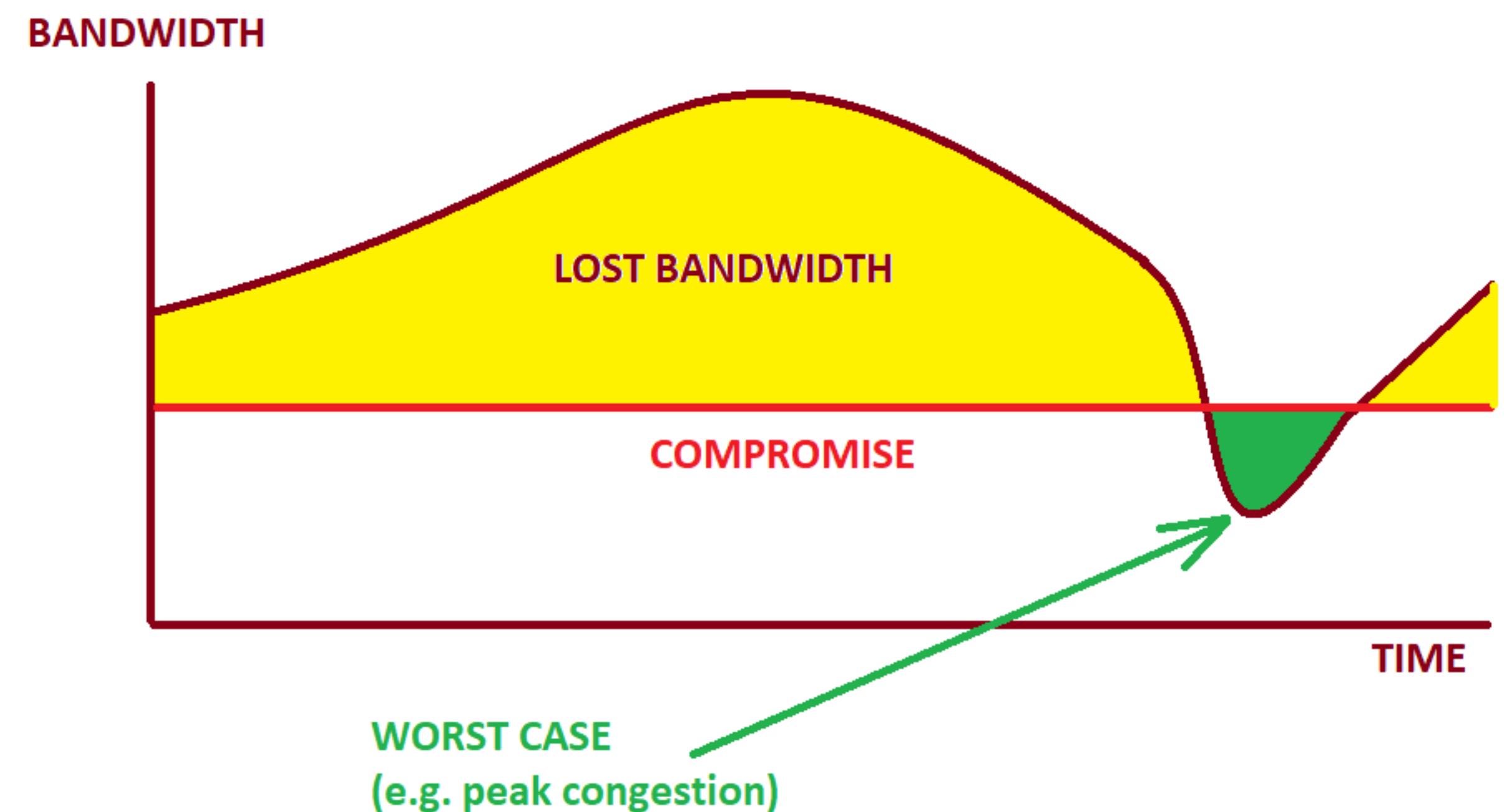
Characterising Latency

And why we need to

Duncan Cameron | Fri 24 Jan 2025

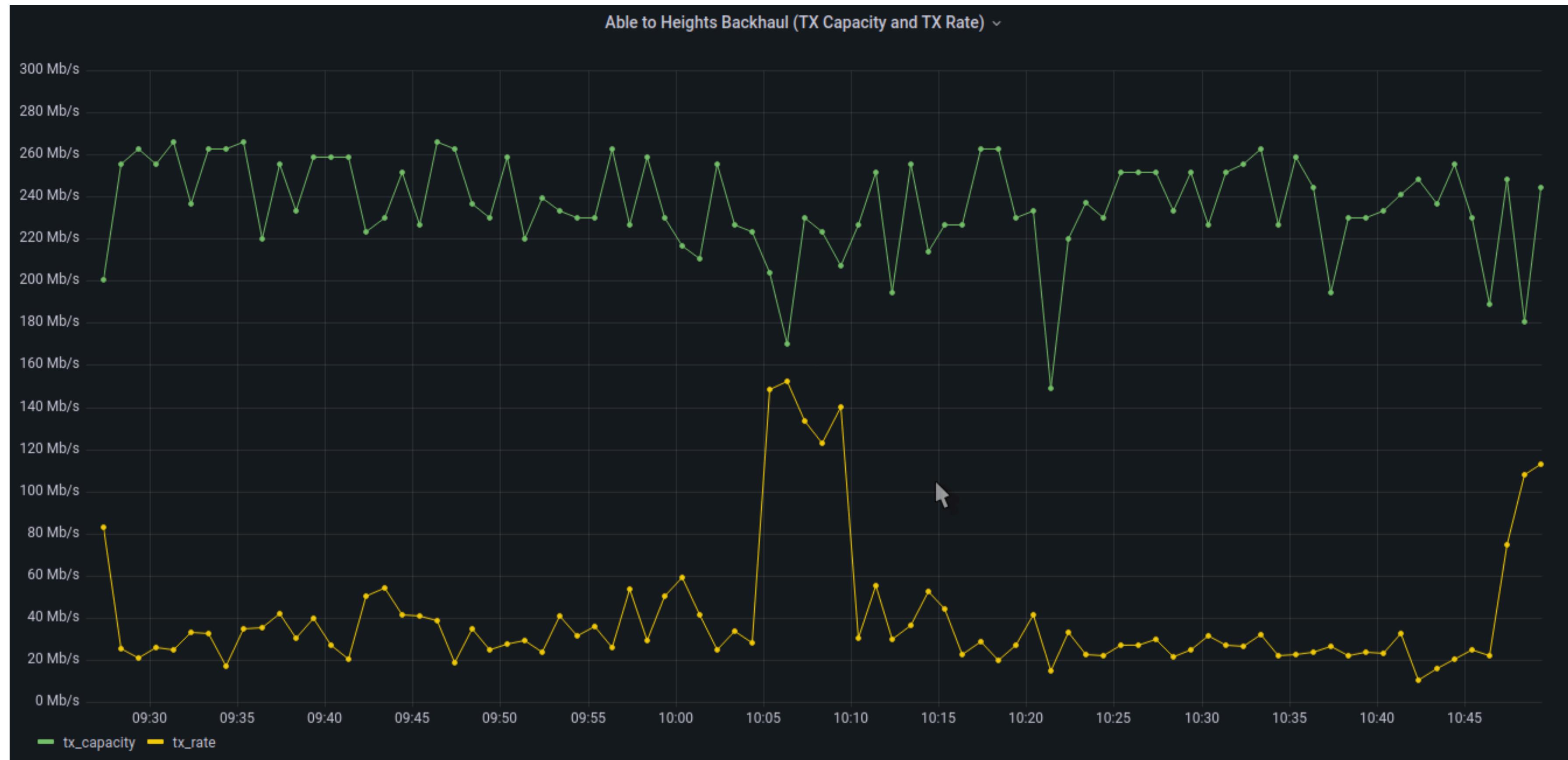
What's the problem?

- Active Queue Management solutions (FQ-CoDel, CAKE), keep latency low by flow-queuing and dropping packets from disruptive flows.
- Works fine, right?
- Yes, if you have consistent link capacities!
- Not so much on variable links. Compromise rates are set, resulting in lost bandwidth.

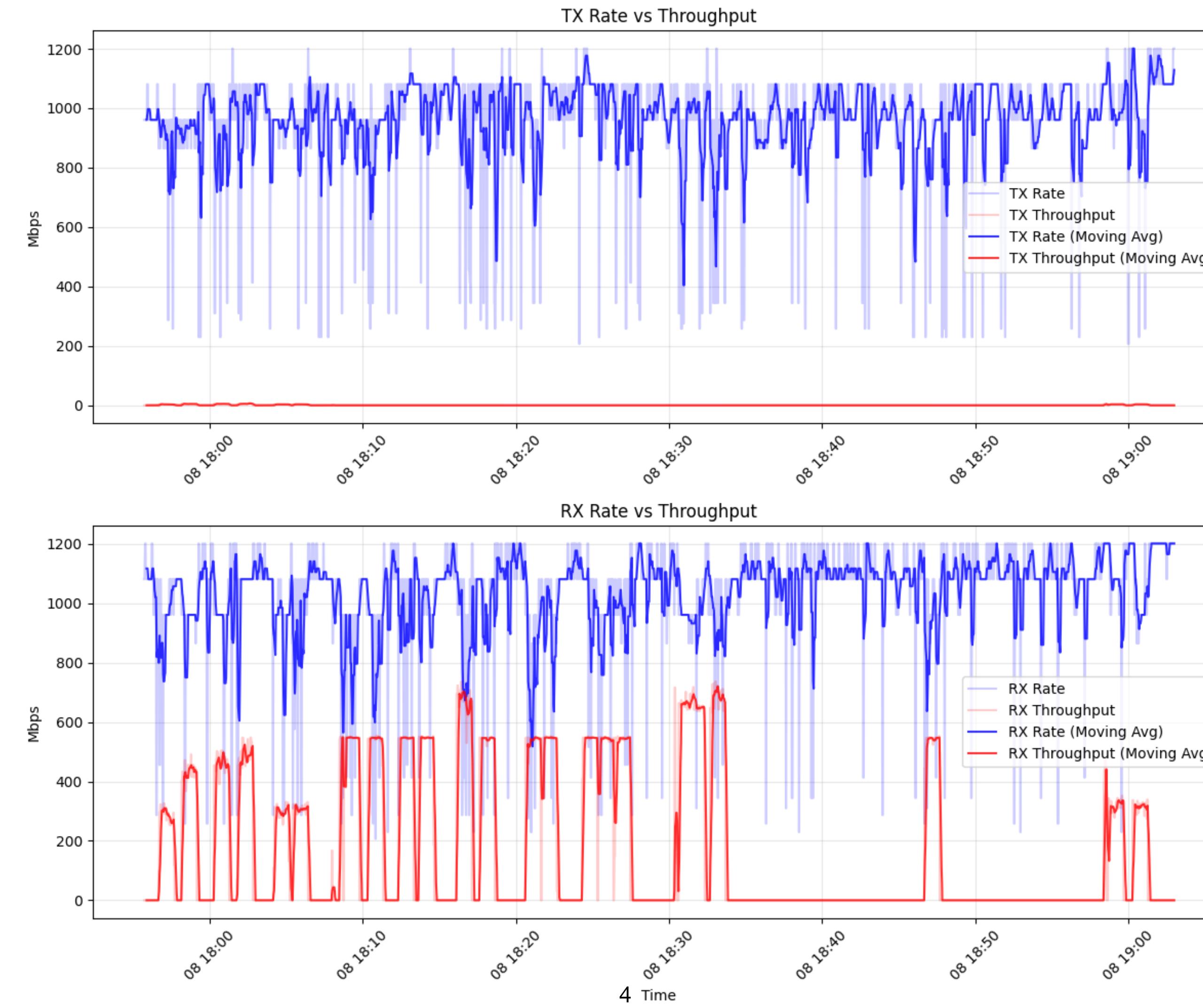


From: <https://github.com/lynxthecat/cake-autorate?tab=readme-ov-file>

Show me the figures

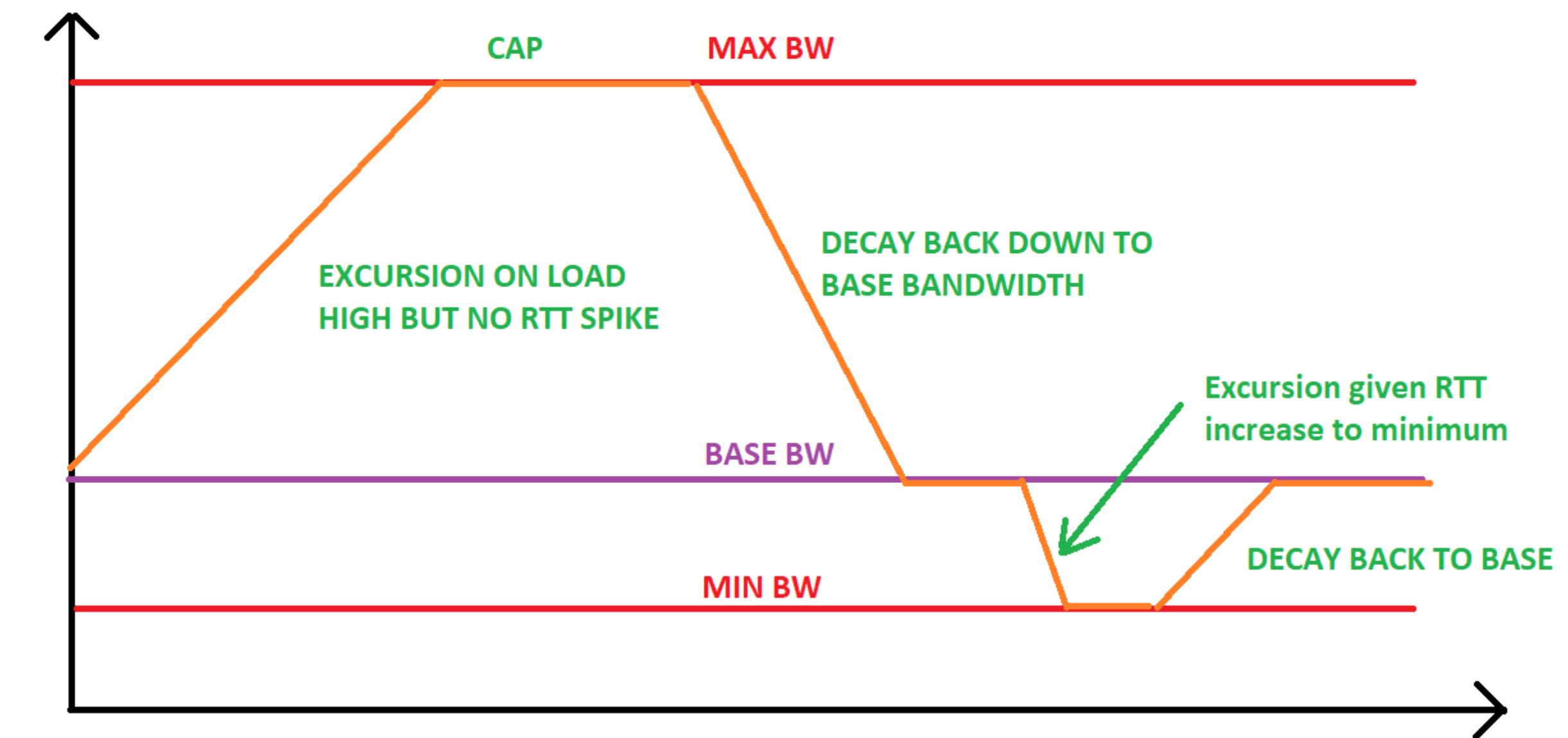


Show me the figures



Background

- The cake-autorate project aims to solve this same issue.
- “... monitors load (receive and transmit utilization) and ping response times from one or more reflectors (hosts on the internet), and adjusts the download and upload rate (bandwidth) settings for CAKE.”



From: <https://github.com/lynxthecat/cake-autorate?tab=readme-ov-file>

Background

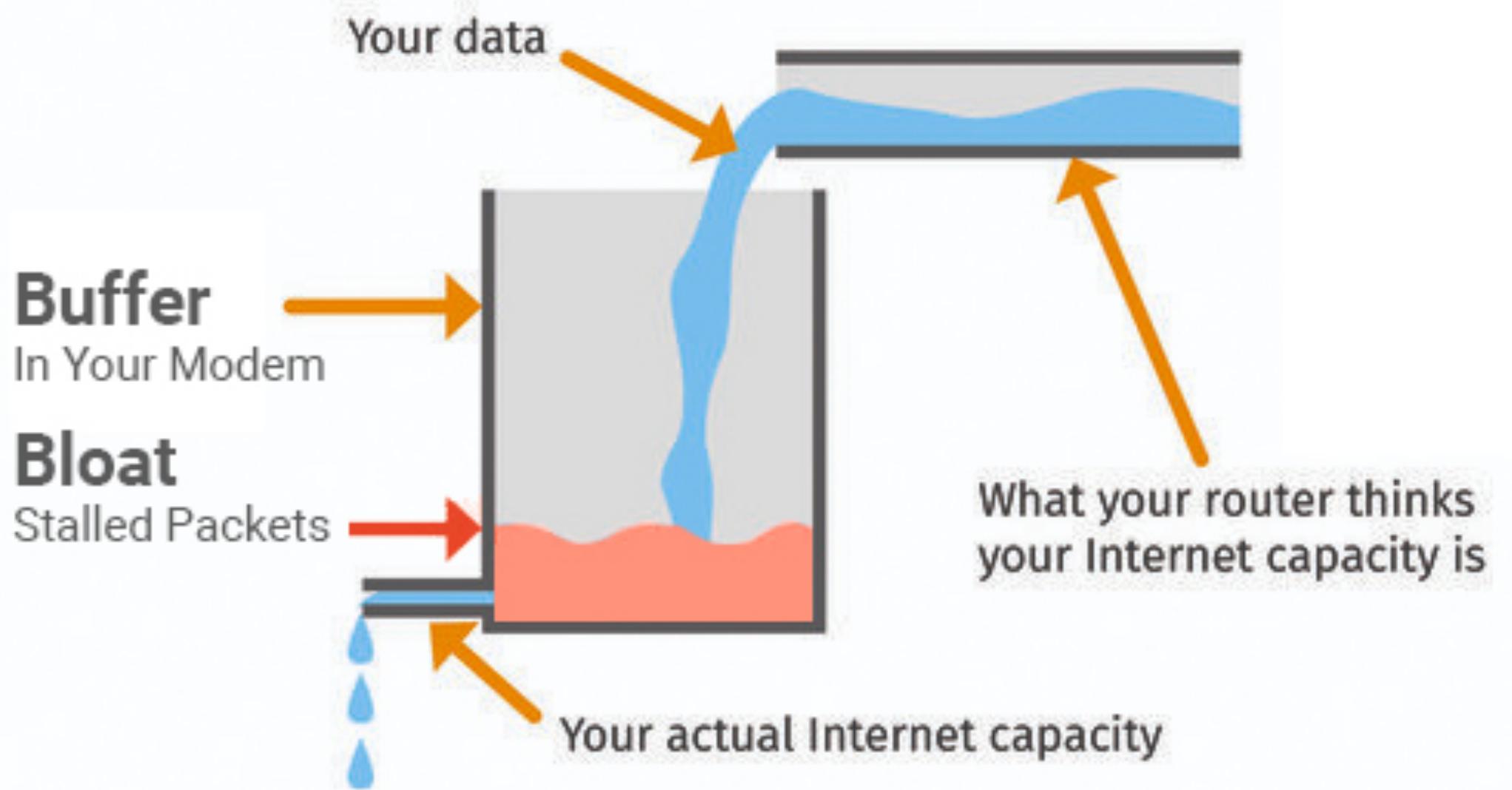
- Jon Brewer and Ulrich Speidel asked about this at APNIC 58.
- Jon pointed out there is potentially a lot of lost capacity with fixed link shaping rates.
- Later pointed out the BLU project (very large Italian Wireless ISP) that uses radio stats as-is for traffic engineering (but we know these values are often wrong, per other slides)



From: <https://www.youtube.com/watch?v=KwkSlxl00Dg>

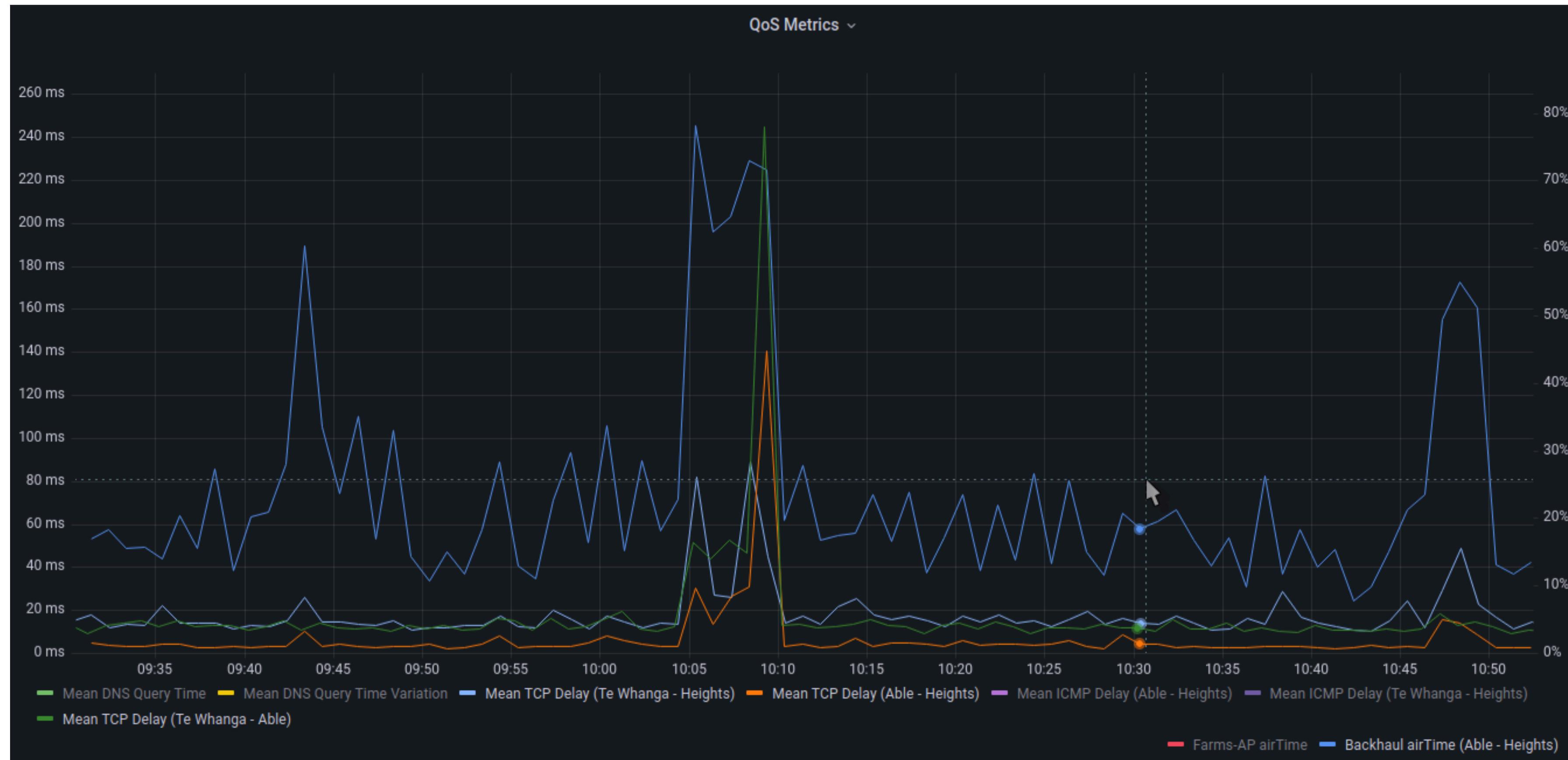
Wait... what about latency?

- What happens if we avoid using “safe capacities” for FQ/AQM and other traffic engineering purposes?
- Under low-load, probably nothing bad.
- Under high-load, bloated buffers and angry customers.

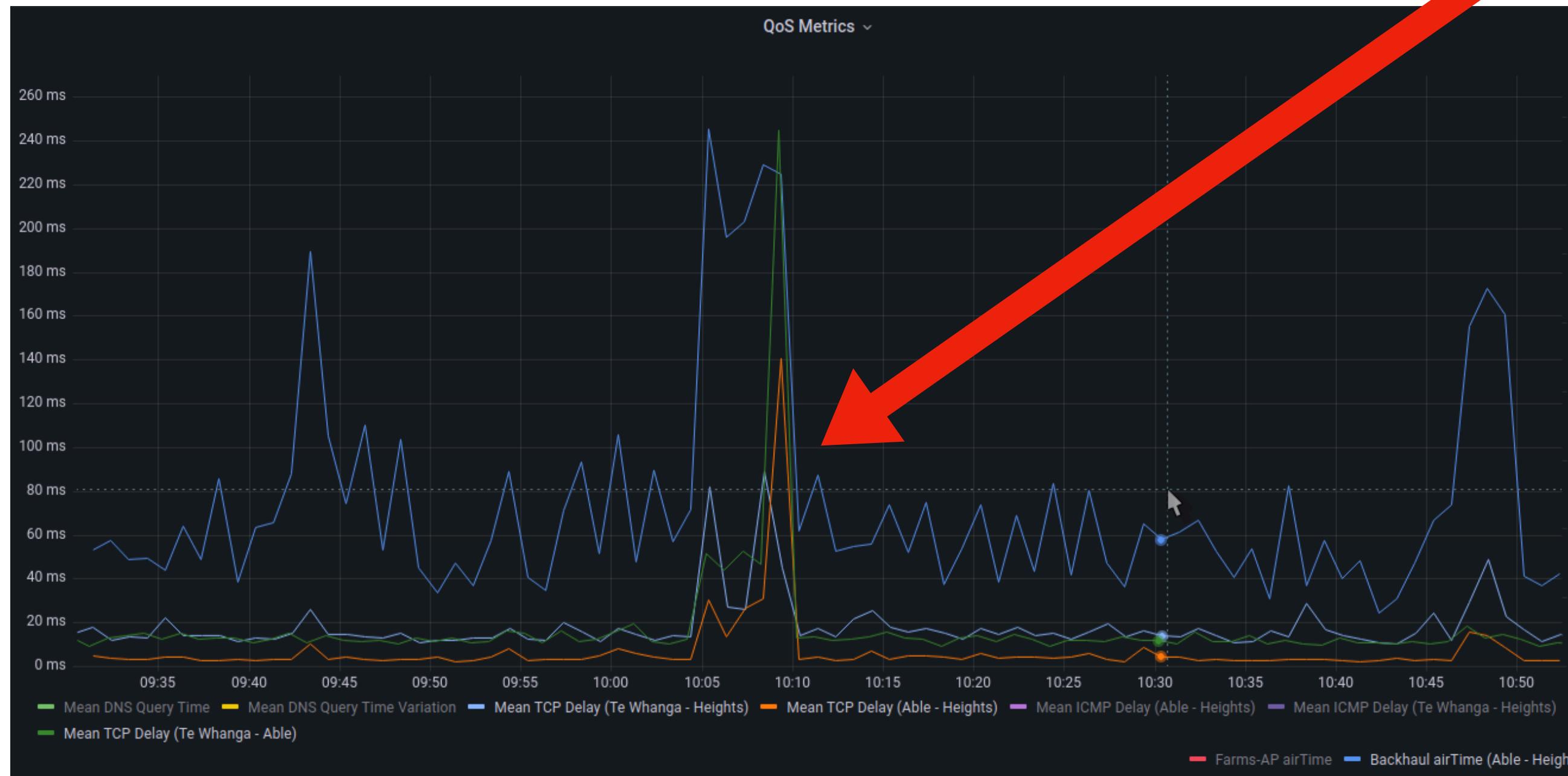
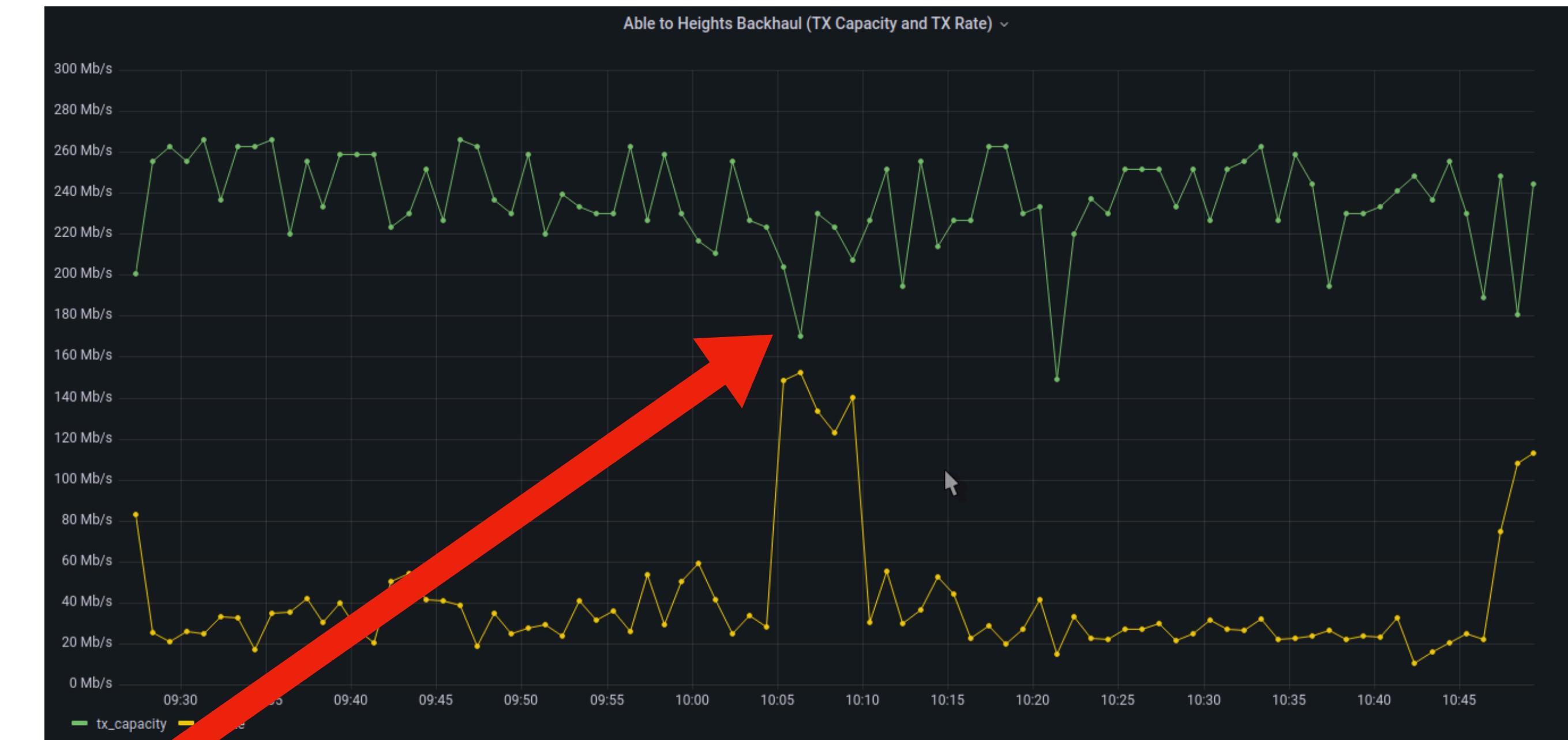


From: <https://blog.erik.is/posts/bufferbloat/>

Bloat happens...

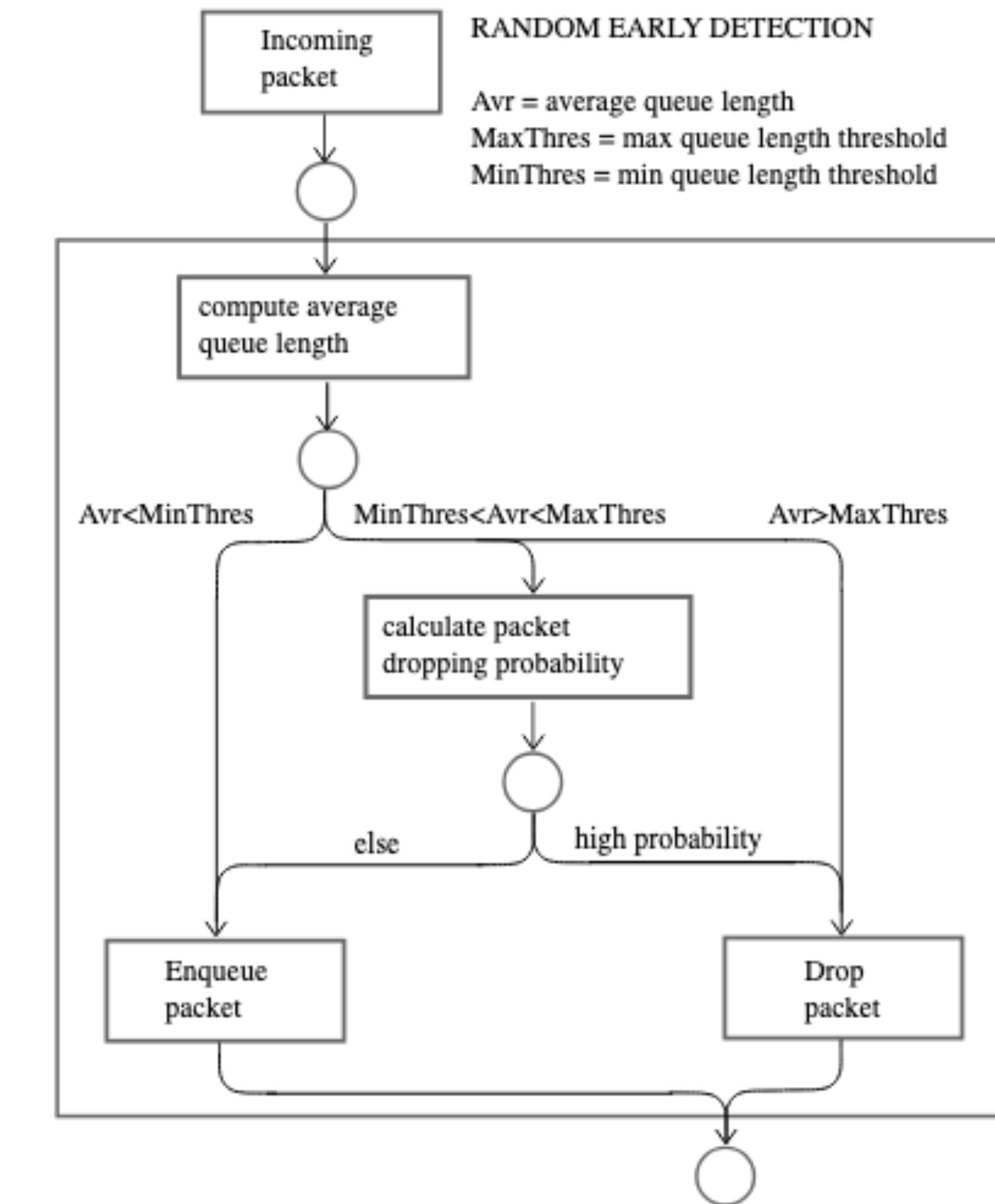


Bloat happens...



Next steps

- We know the real TCP/IP throughput from ElasticRoute, not the unreliable radio values.
- We can get queue backlog from Linux-based radios using netlink.
- We can then use queuing theory to estimate how congested our links are.
- The Random Early Discard (RED) qdisc did this for dropping packets.



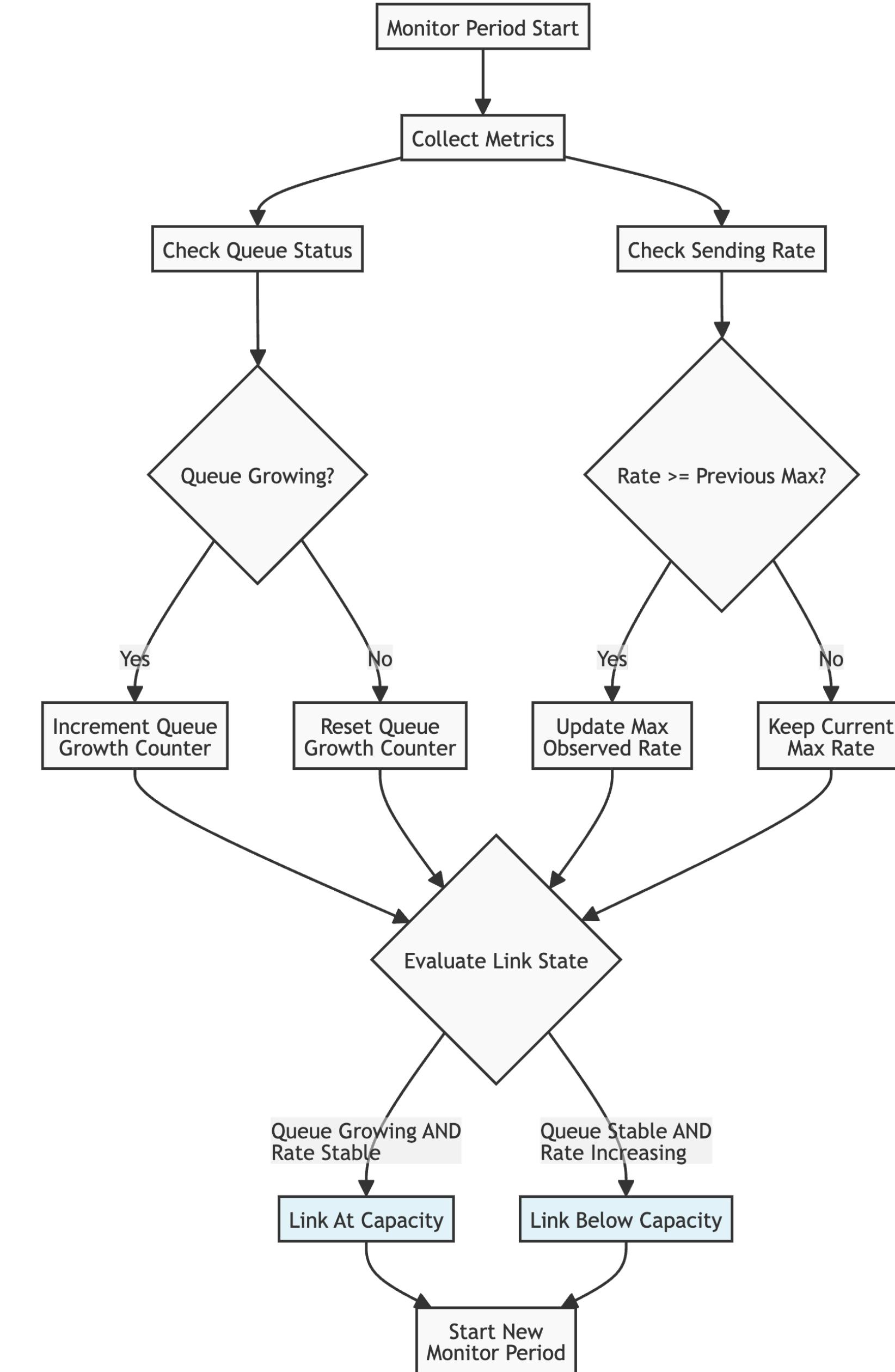
Next steps

- The BLUE qdisc built upon RED, and uses packet loss and link utilisation history to manage congestion.
- BLUE maintains a single probability, which it uses to mark (or drop) packets when they are queued.
- If the queue is continually dropping packets due to buffer overflow, BLUE increments the marking probability, thus increasing the rate at which it sends back congestion notification.



Next steps

- Today, better qdiscs exist for FQ/AQM (FQ-CoDel, CAKE).
- However, RED and BLUE have concepts that we can directly apply to reliably infer if links are at capacity.
- Using Ubiquiti radios running the Ubiquiti-build of OpenWrt, we can retrieve queue backlog and drop statistics without replacing or re-inventing the qdisc.
- Finally, we can test if this works well in practice.
- More to come soon!



Next

Thanks for listening!