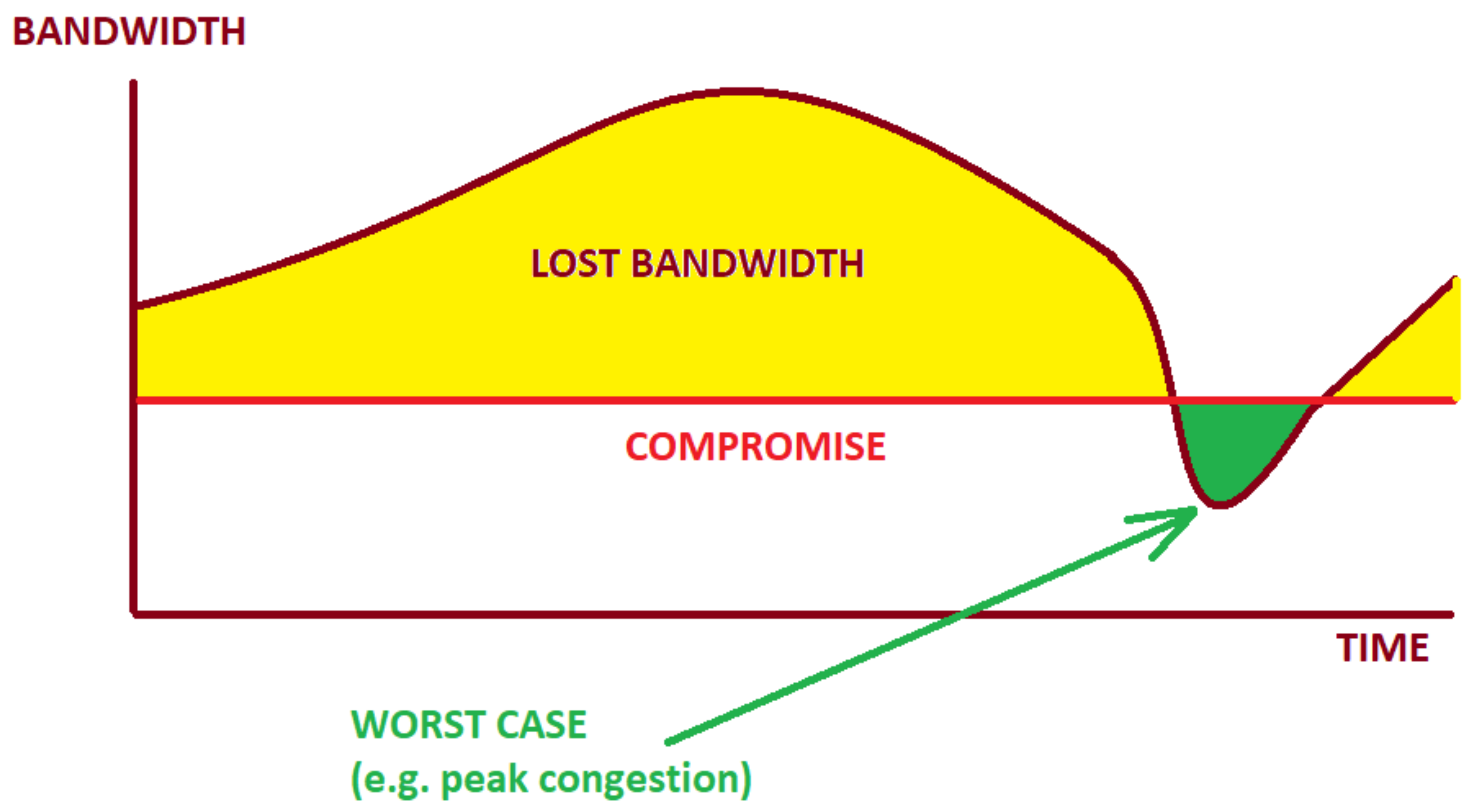
# PURPLE

## DYNAMIC CONTROL OF CAKE

Duncan Cameron | Victoria University of Wellington | APNIC 60 | 10 September 2025

## What's the problem?

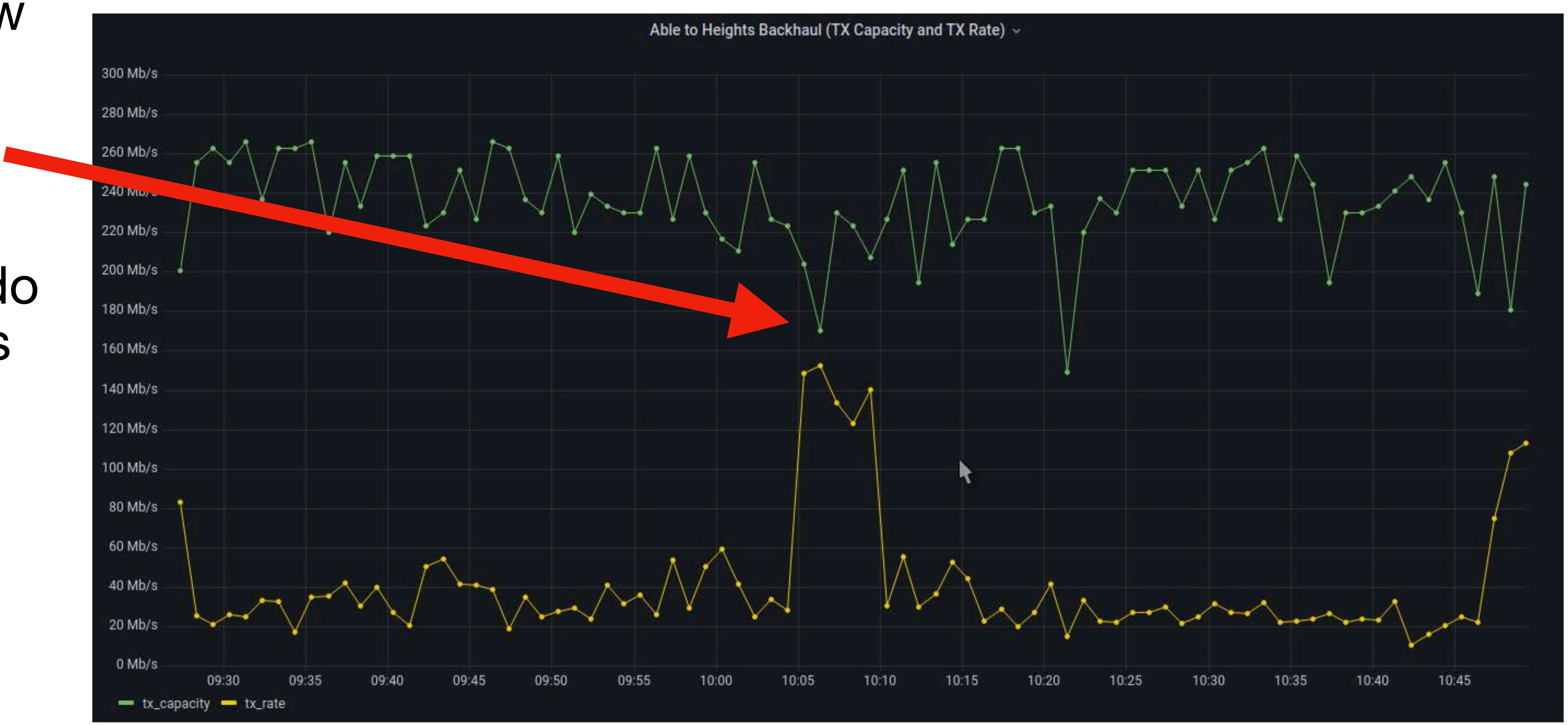
- Active Queue Management (**AQM**) solutions, e.g., Fair/Flow Queue Controlled Delay (**FQ-CoDel**) and Common Applications Kept Enhanced (**CAKE**), control latency by flow-queuing and dropping packets from disruptive flows
- Works fine, right?
- Yes, if you have consistent link capacities!
- Not so much on variable links. Compromise rates are often set, resulting in lost bandwidth



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

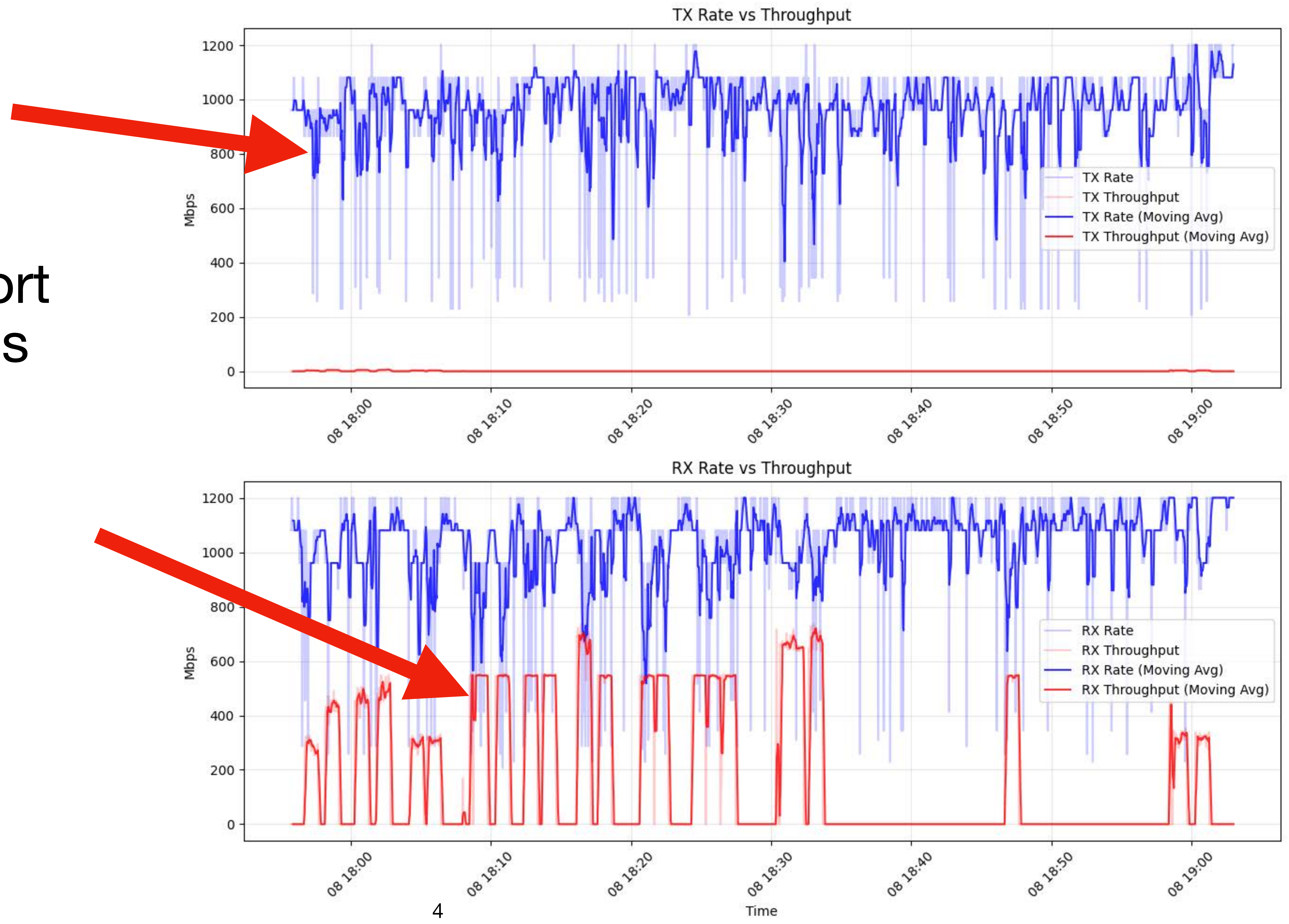
# Show me the figures

- Rate drops below the anticipated average are common
- Modern qdiscs do not deal with this well



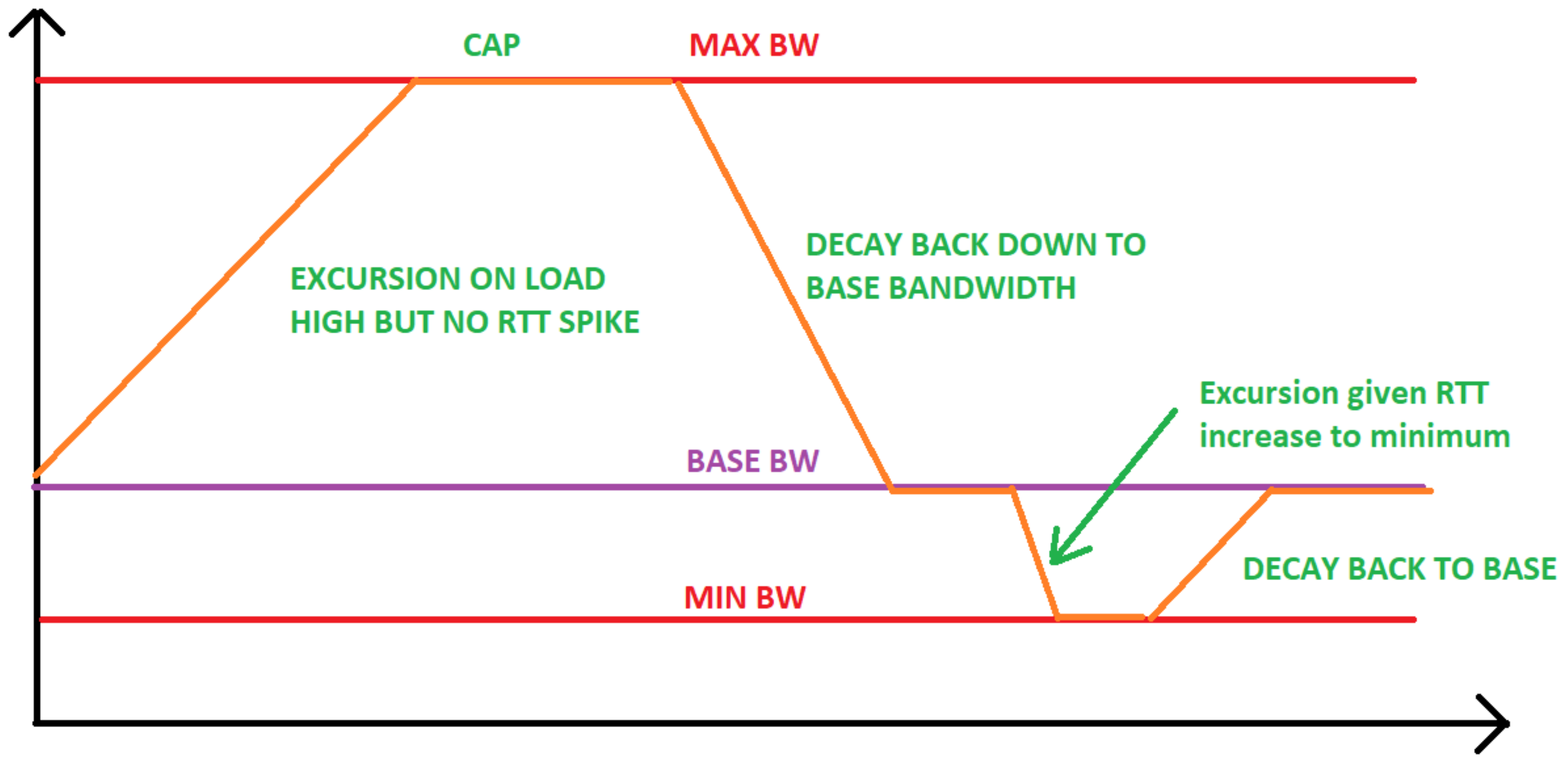
# Show me the figures

- Radio rate (capacity) estimation is also inconsistent
- Sometimes lacks any sort of sanity check, e.g., this figure
- Estimated rate drops below the achieved throughput at times
- What do we trust?



## 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

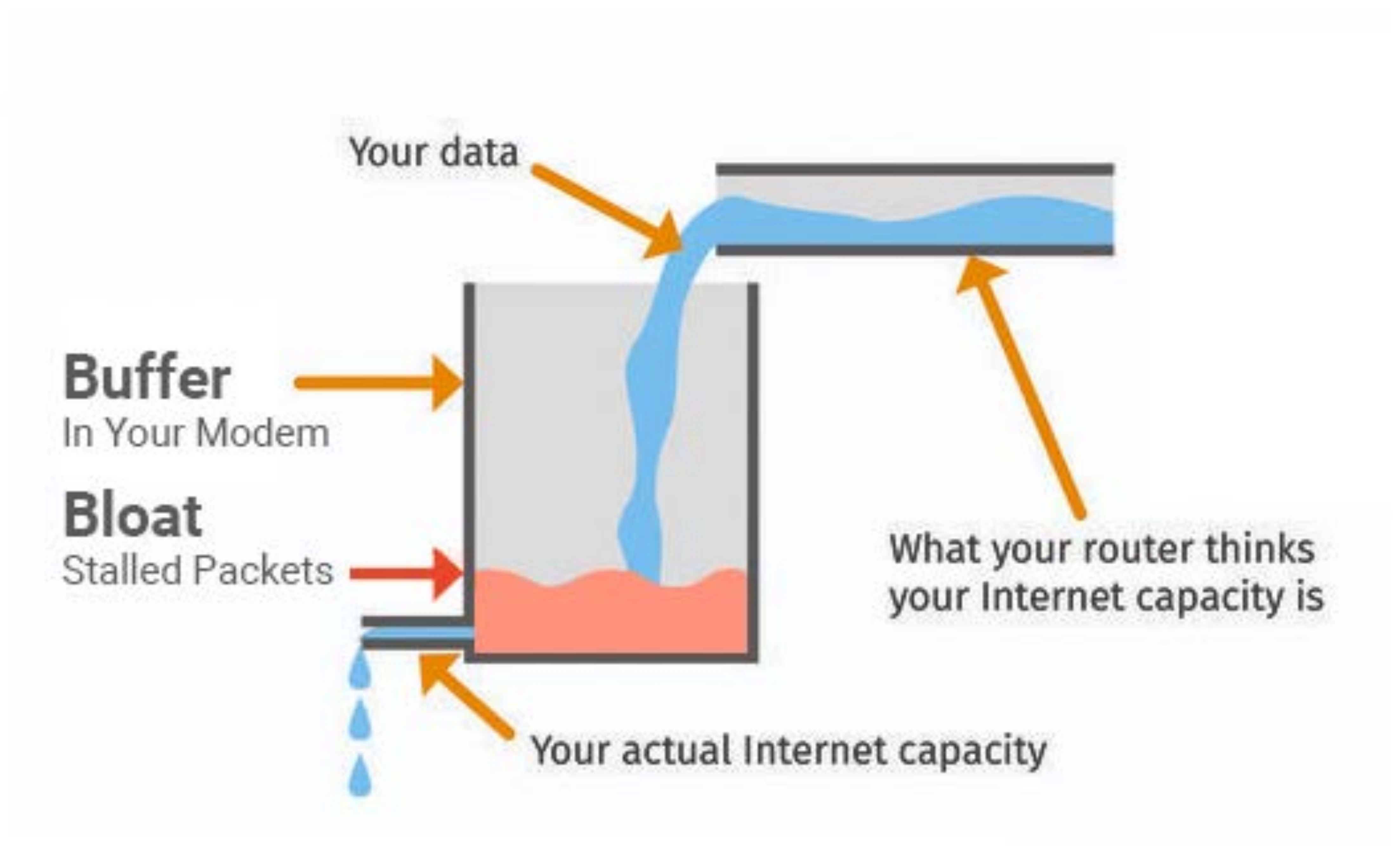
## Background

- Jon Brewer and Ulrich Speidel asked about this at APNIC 58
- Jon pointed out there is potentially a lot of lost capacity with fixed/static shaping rates
- Later pointed out the BLU project (very large Italian Wireless ISP) 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=KwkSIxl00Dg

## Wait… what about latency?

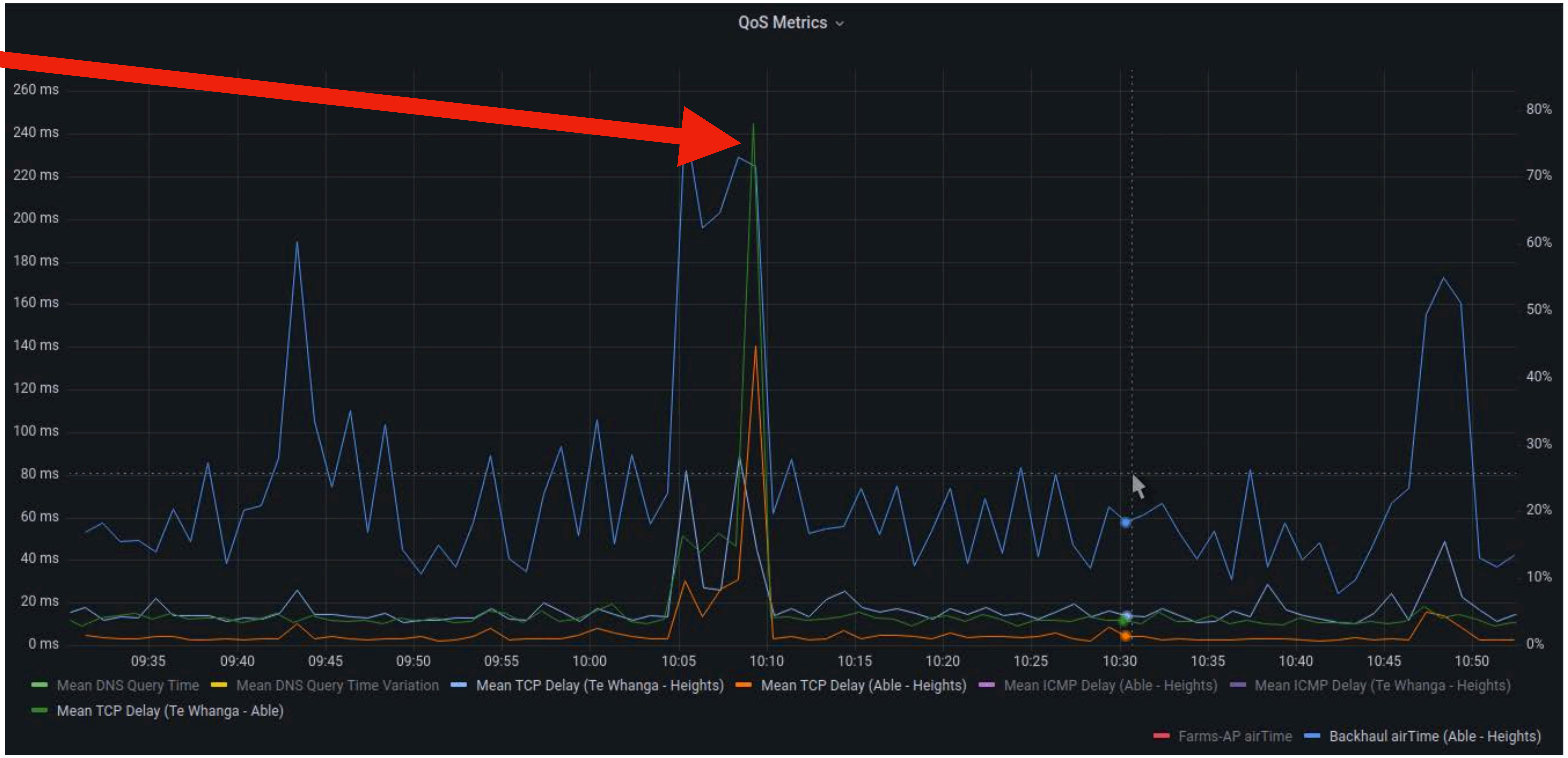
- What happens if we avoid using "safe capacities" for FQ/AQM and other traffic engineering purposes?
- Under low-load, probably nothing bad
- Under high-load, bloated buffers and angry customers


Your data
Buffer
In Your Modem
Bloat
Stalled Packets
What your router thinks your Internet capacity is
Your actual Internet capacity

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

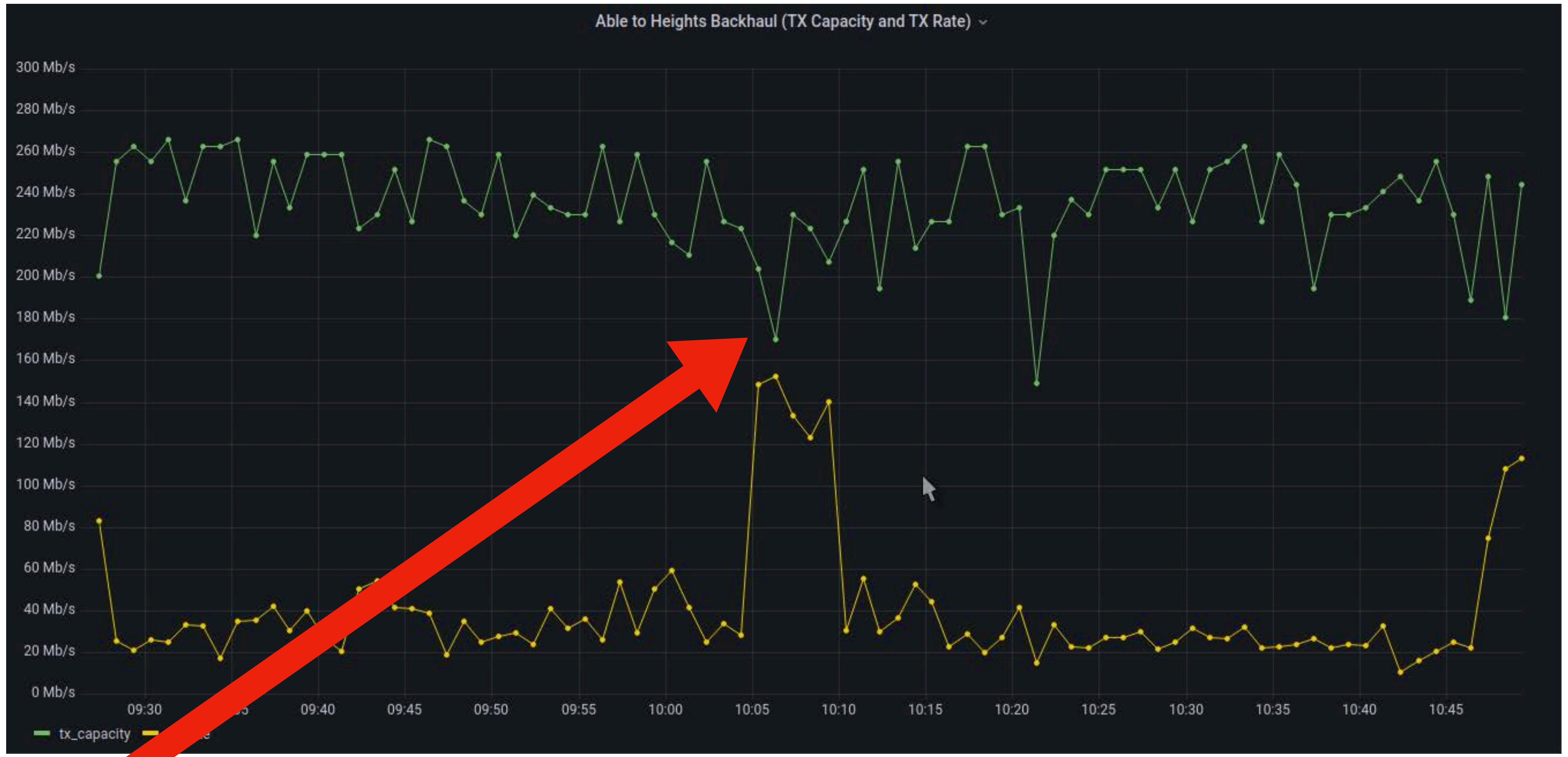
## Bloat happens…

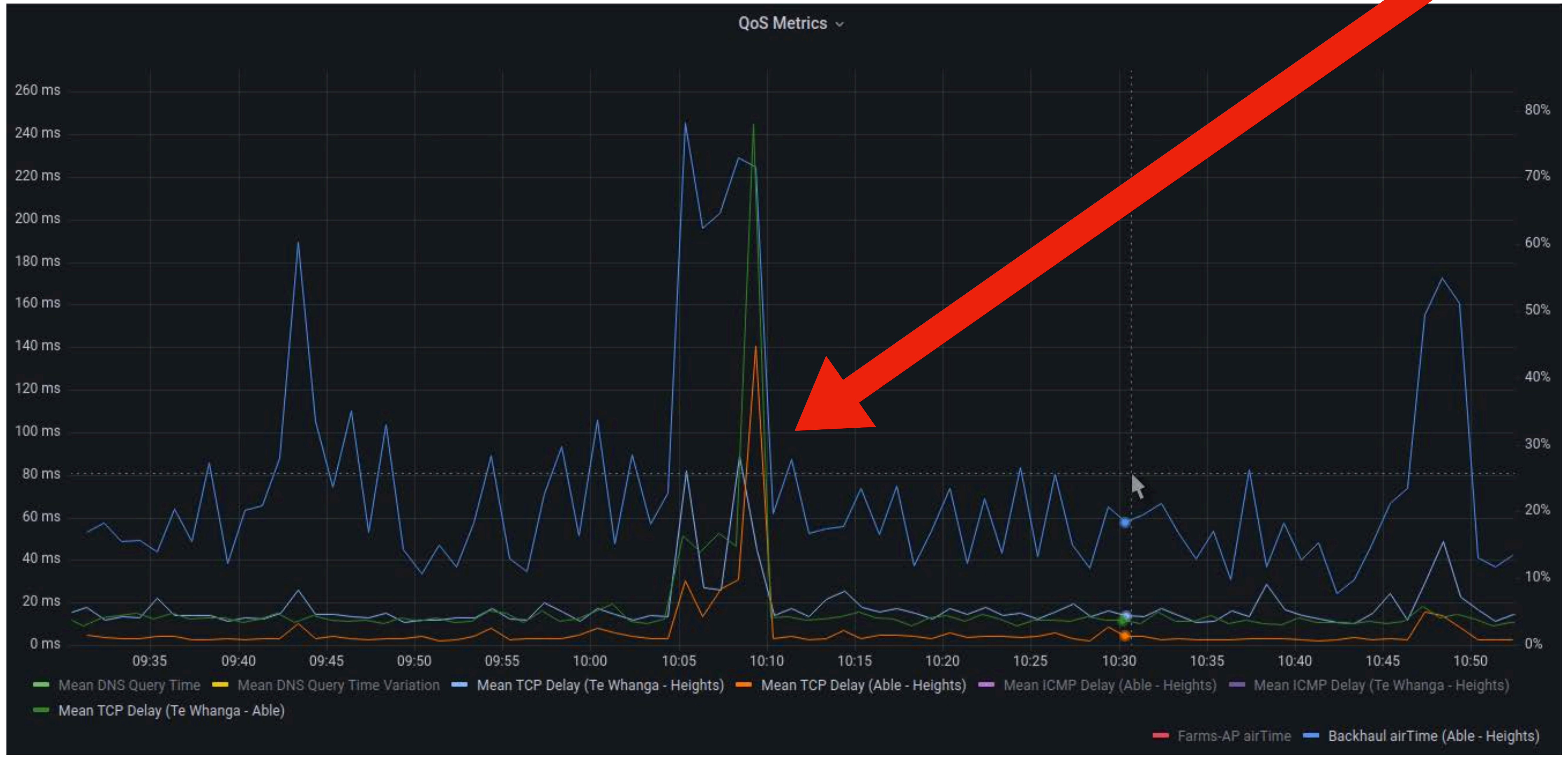
- Where did this come from?
- Brief instances of very high latency are not okay!
- Not captured well by looking at long-term averages



## Bloat happens…

- Throughput increases, estimated capacity drops
- Queuing delay increases, “bloat” happens
- Static qdisc configuration fails

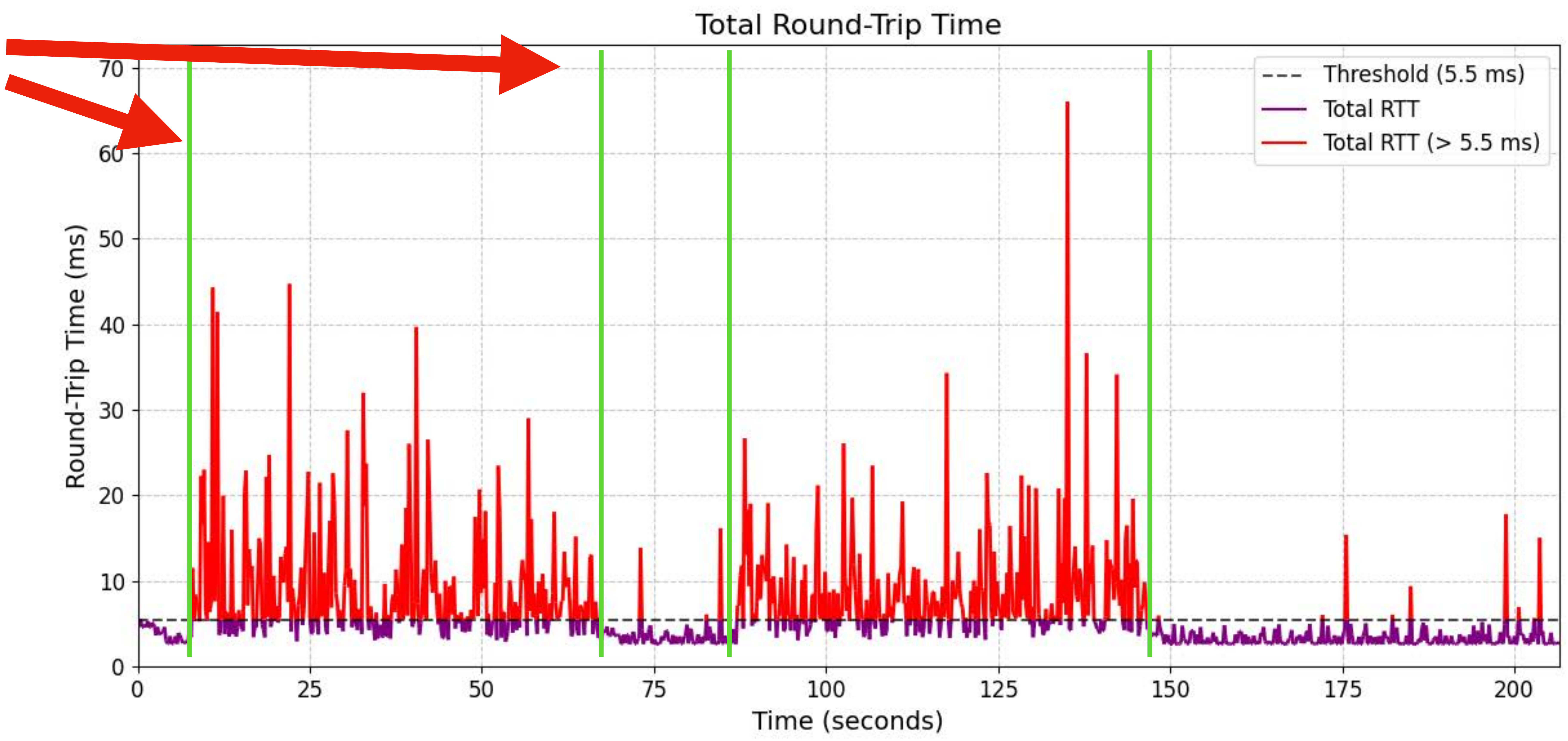




- Best to avoid utilisation near the max radio capacity - but how?

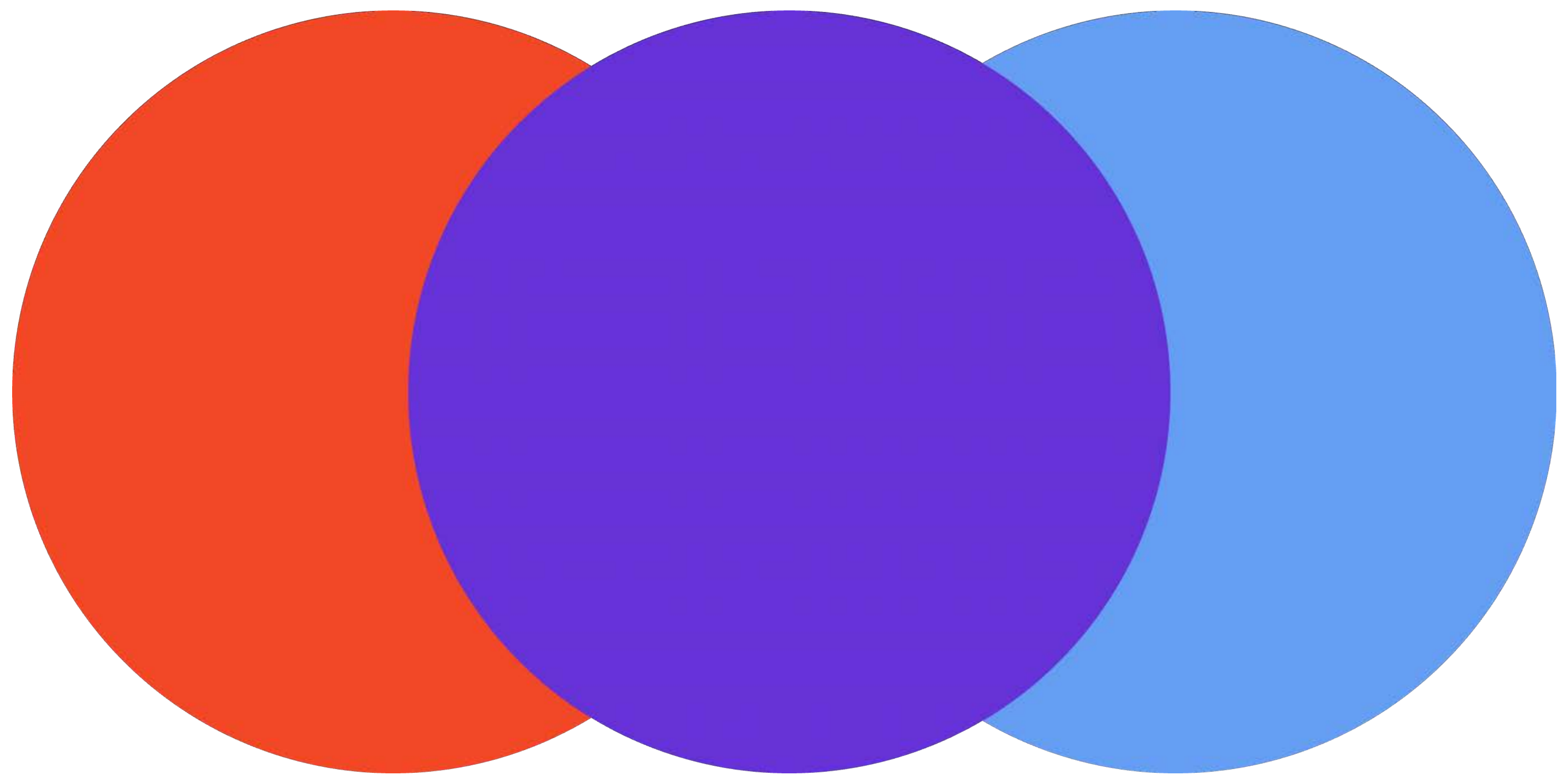
# Bloat happens…

- iPerf tests running (green), link fully saturated
- Significant increase to RTT visible



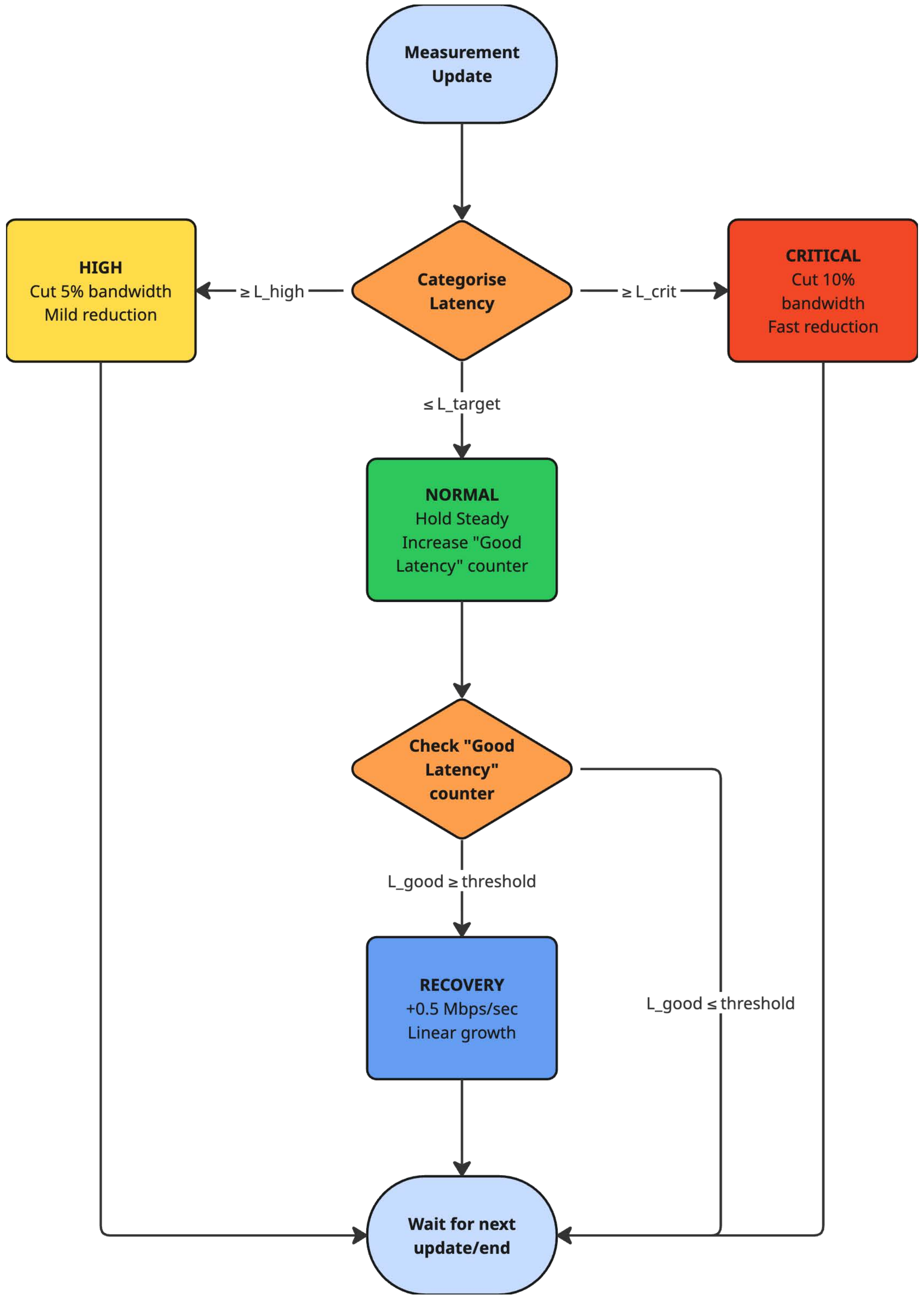
## Introducing PURPLE

- RED + BLUE (qdiscs) = PURPLE
- Estimates "useful capacity" using a given target latency
- Controls bloat through dynamic adjustment of CAKE's bandwidth parameter
- Uses precise one-way delay measurements for control input (from the *Polus* scheme, see APNIC 56 talk)

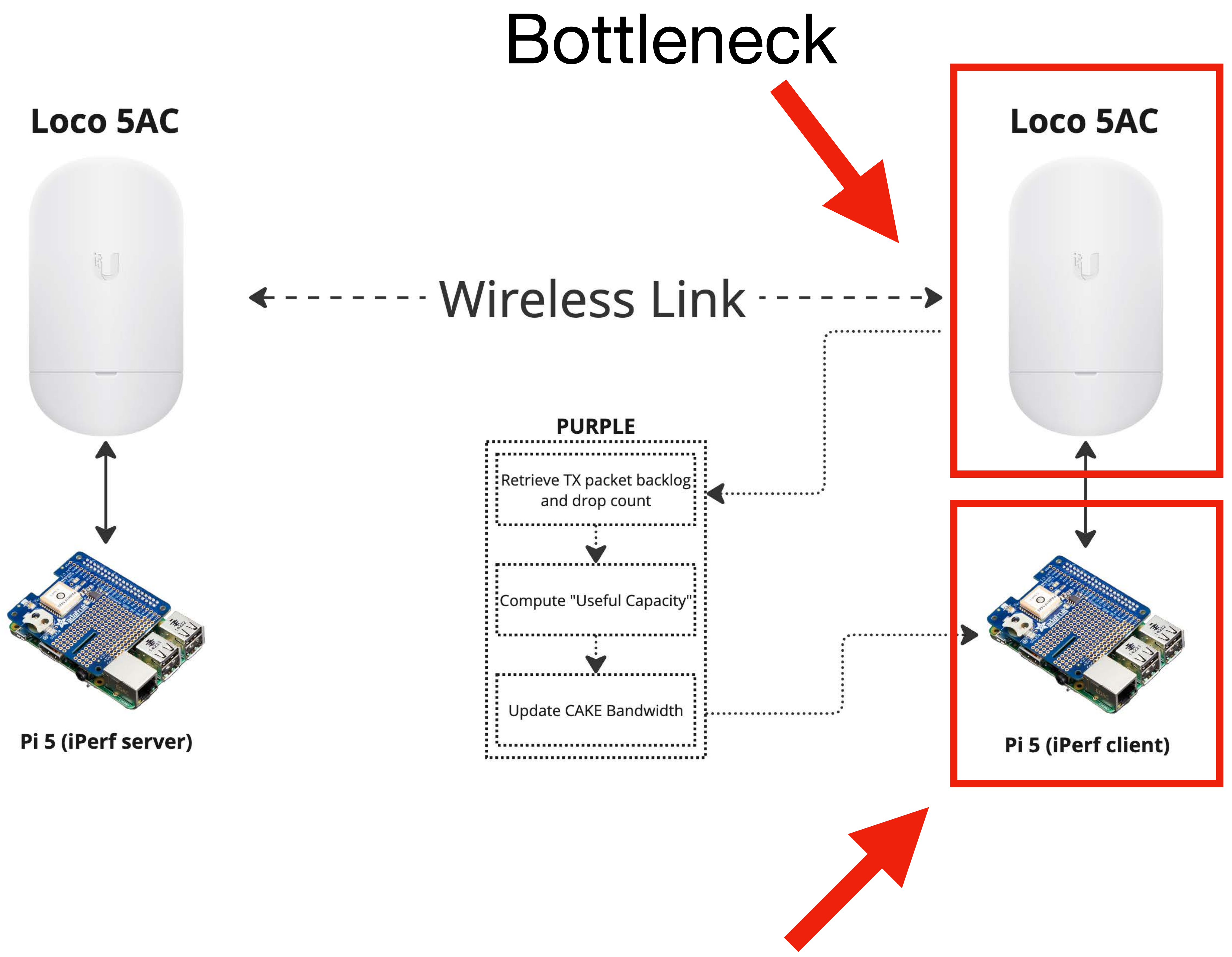


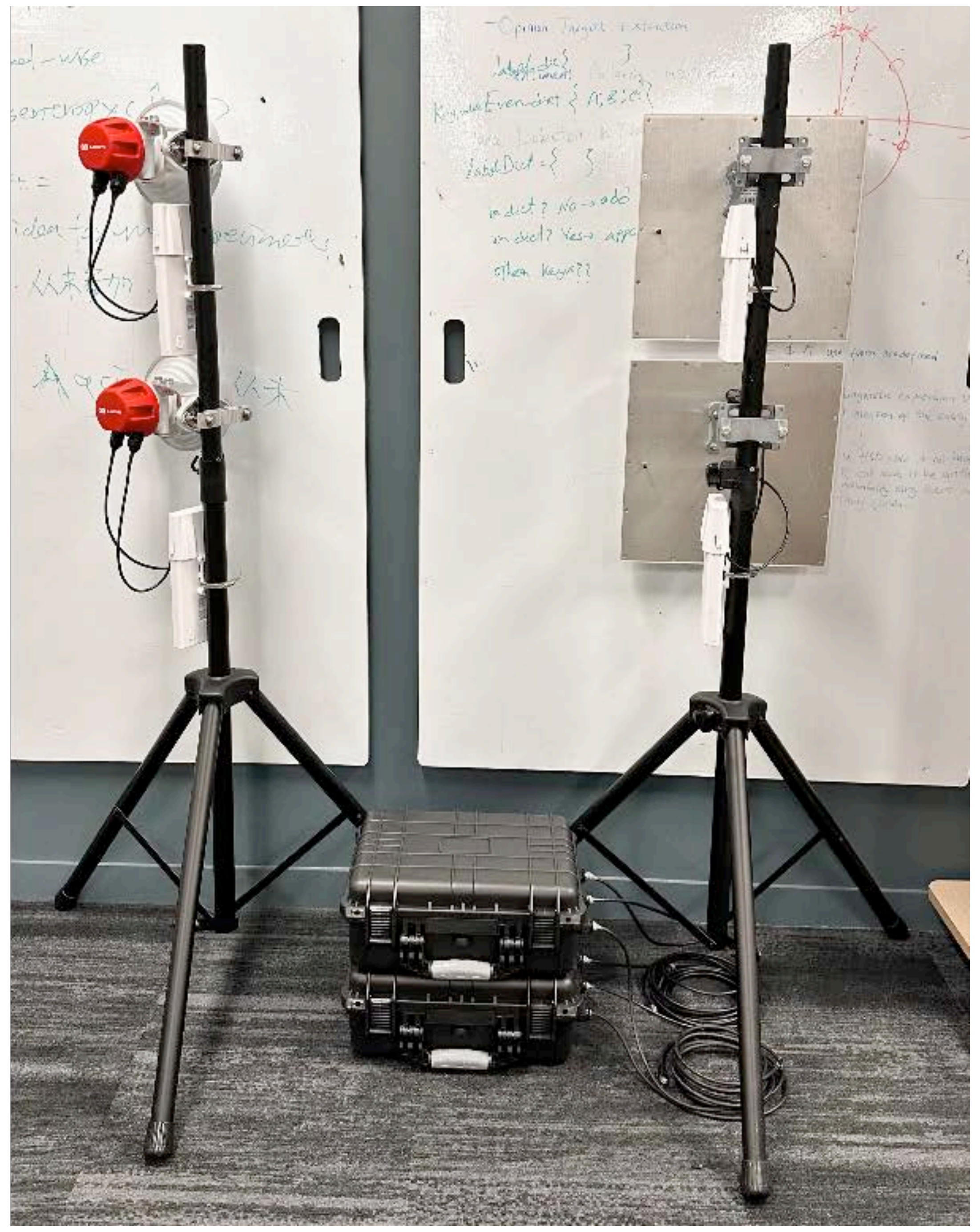
# PURPLE is SIMPLE

- Uses tried-and-tested Additive Increase, Multiplicative Decrease (AIMD) algorithm made famous by TCP
- Alternative (non-AIMD version) can use low-level queue metrics where they are available (requires open radio firmware)



# Evaluation Setup



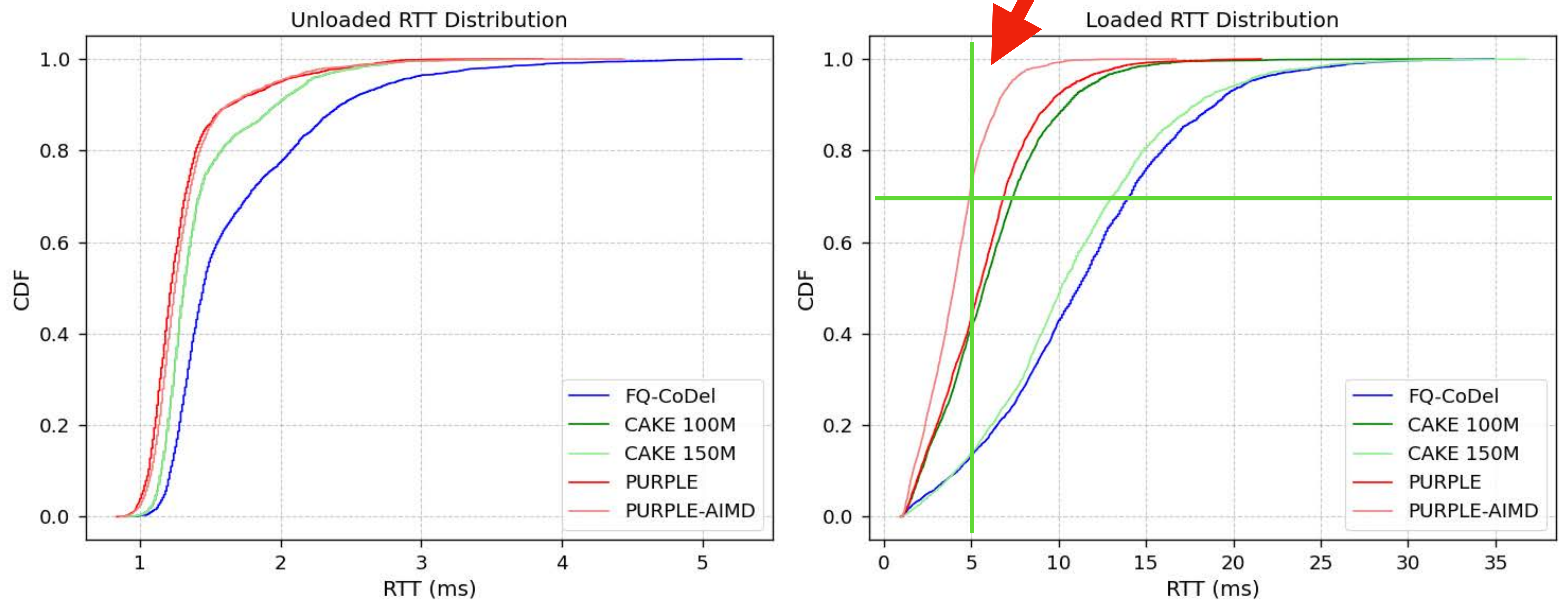


PURPLE controls the CAKE egress bandwidth directly behind the bottleneck

(we assume we cannot change the bottleneck qdisc)

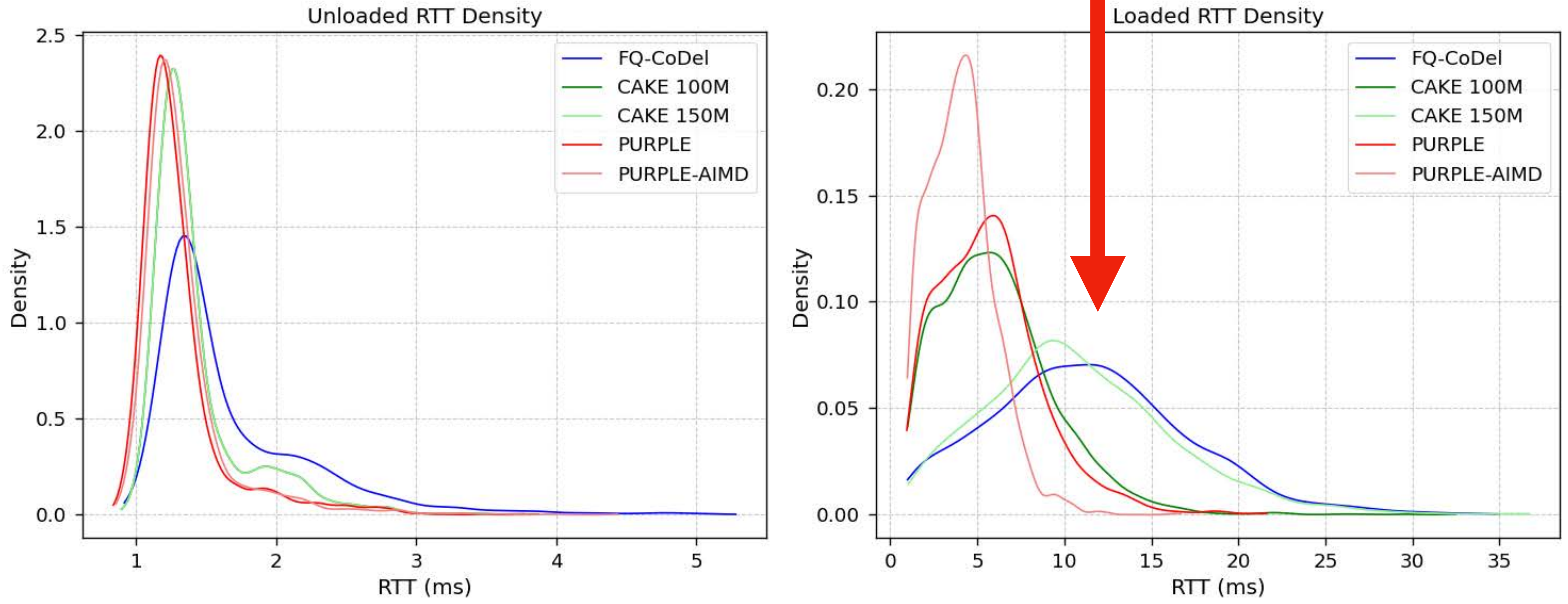
# Performance Results

- Significant improvement over “optimistic” static configurations



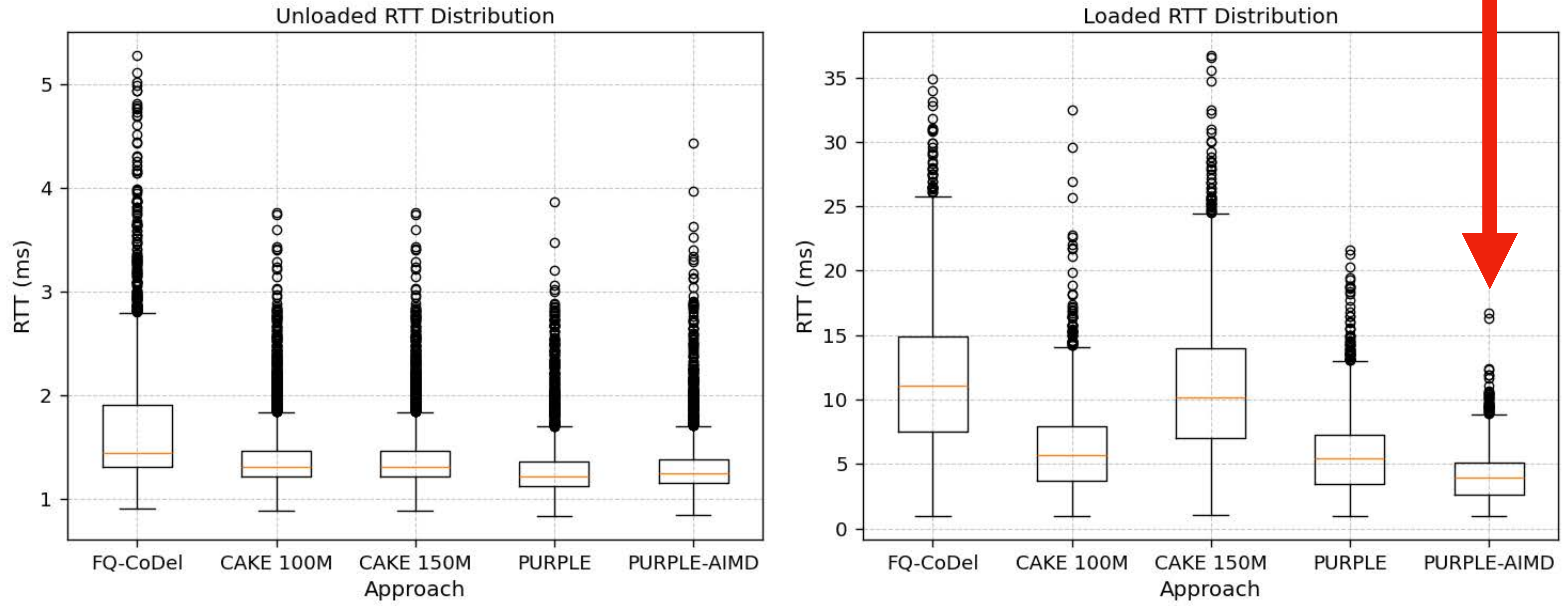
# Performance Results

- Setting the wrong “static” bandwidth is almost as bad as not setting it at all

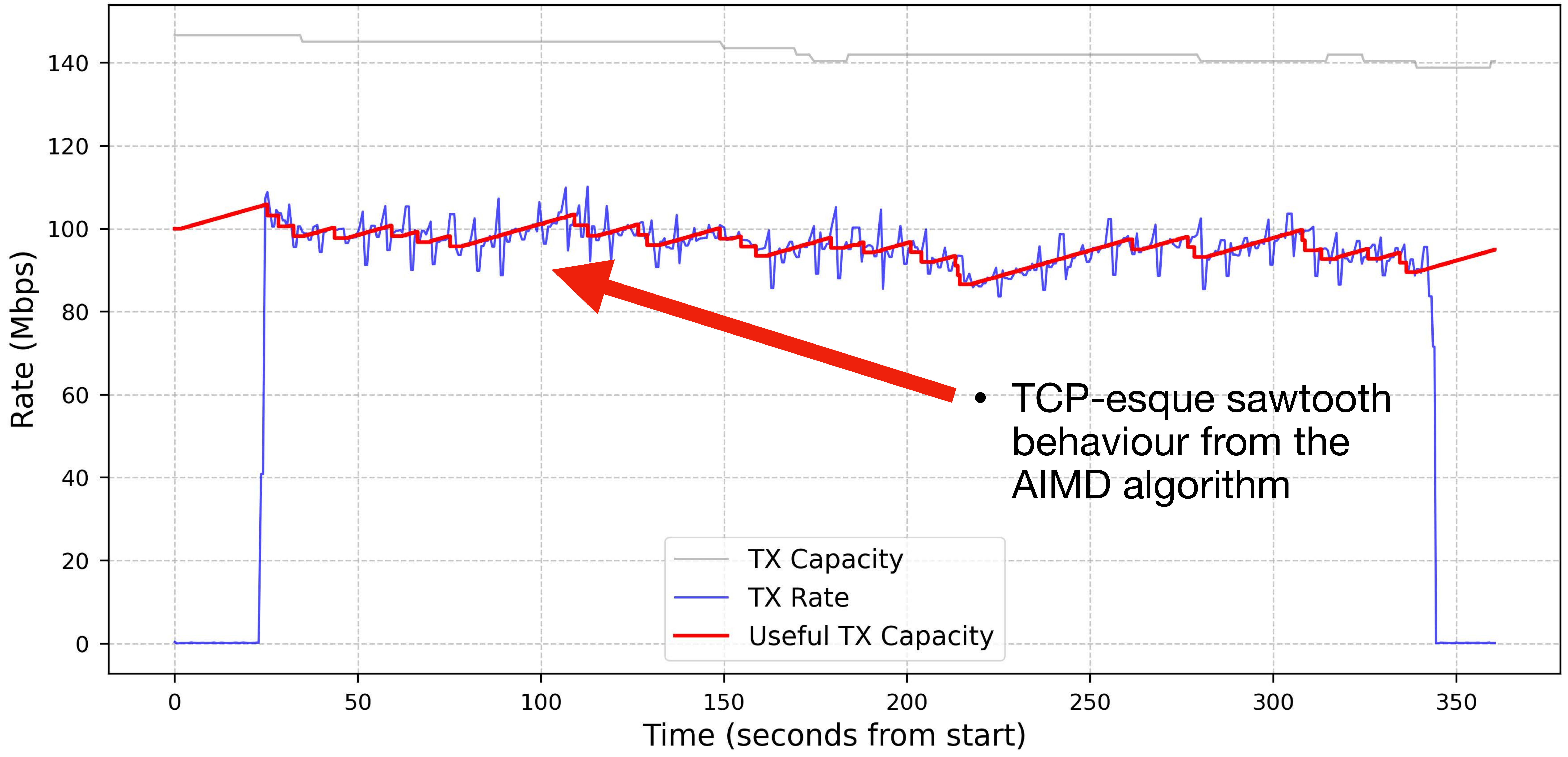


# Performance Results

- PURPLE-AIMD is the clear winner

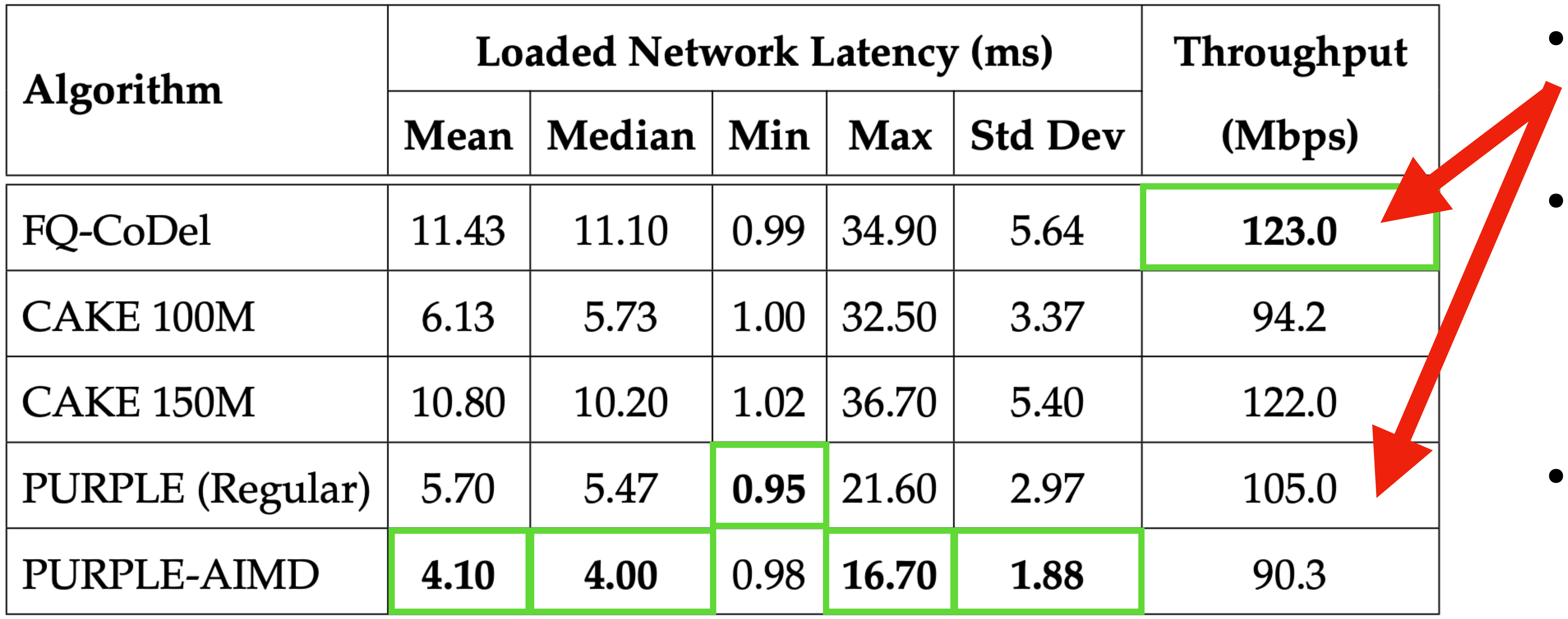


# Performance Results - AIMD Behaviour



- TCP-esque sawtooth behaviour from the AIMD algorithm

# Performance Results - Compromise

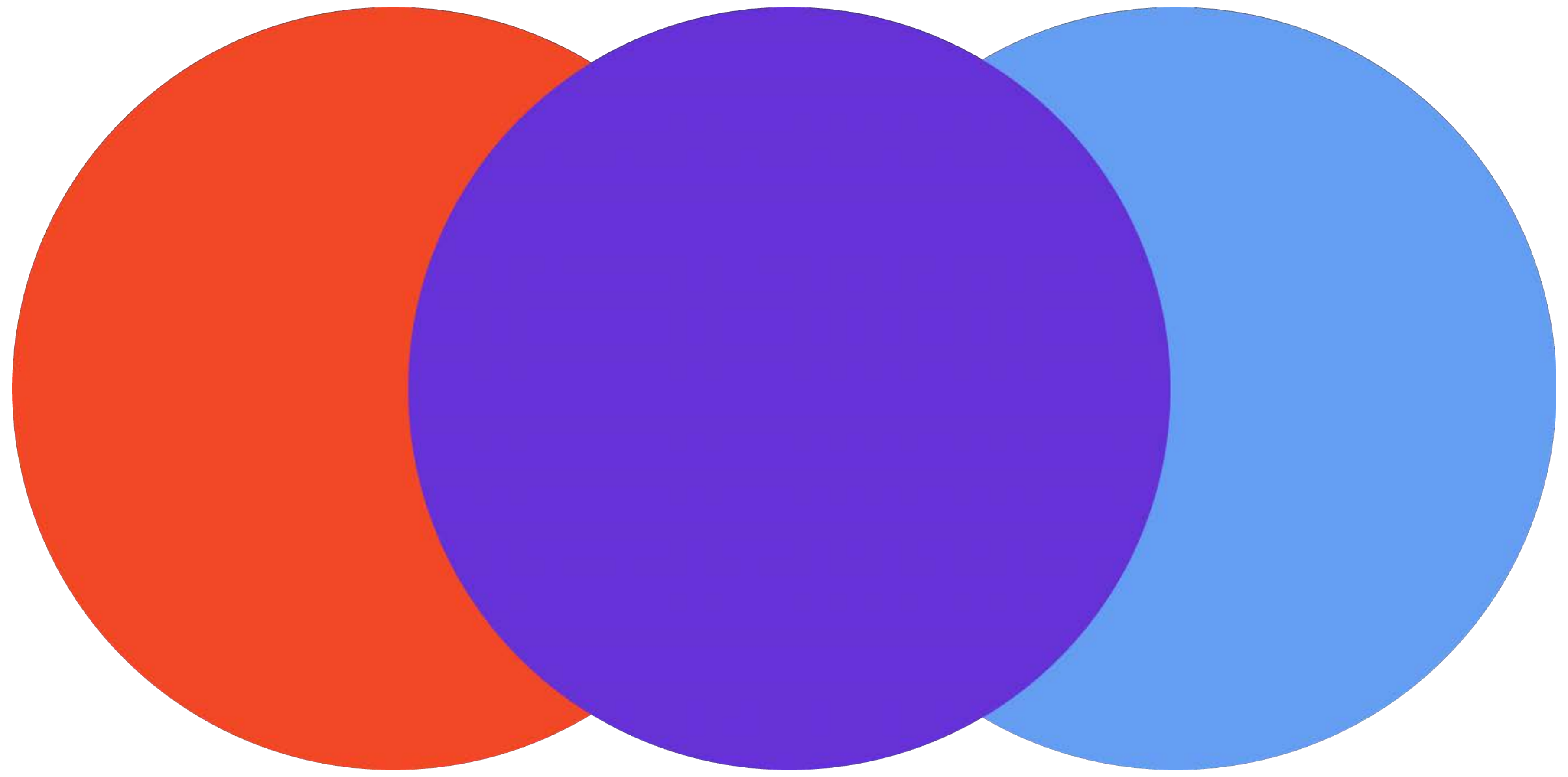


| Algorithm | Loaded Network Latency (ms) | | | | | Throughput |
|---|---|---|---|---|---|---|
| | Mean | Median | Min | Max | Std Dev | (Mbps) |
| FQ-CoDel | 11.43 | 11.10 | 0.99 | 34.90 | 5.64 | **123.0** |
| CAKE 100M | 6.13 | 5.73 | 1.00 | 32.50 | 3.37 | 94.2 |
| CAKE 150M | 10.80 | 10.20 | 1.02 | 36.70 | 5.40 | 122.0 |
| PURPLE (Regular) | 5.70 | 5.47 | **0.95** | 21.60 | 2.97 | 105.0 |
| PURPLE-AIMD | **4.10** | **4.00** | 0.98 | **16.70** | **1.88** | 90.3 |

- There is an inevitable latency-bandwidth tradeoff
- PURPLE achieves higher throughput over PURPLE-AIMD as it uses direct queue metrics
- Not available from all radios - active latency measurement not required where it is

## Enhancing PURPLE

- PURPLE can be passive where queue metrics can be retrieved from the bottleneck radio
- Supported by Ubiquiti airOS (OpenWrt) devices, but not many others
- More open-source support needed
- Alternatives to AIMD can be explored for improving the latency-bandwidth tradeoff



## Remembering Dave

- In loving memory of Dave Täht: 1965-2025
- FQ-CoDel, CAKE, and LibreQoS would not have been possible without him
- Before APNIC 56 in Kyoto, Dave told me: “If Geoff Huston holds up a LibreQoS tee-shirt, I’ll buy him a beer” - see next slide

## Geoff, enjoy your beer

## Thanks for listening!

WiNe Group Website

PURPLE GitHub Repo

duncan@wine.ac.nz

wine.ac.nz