Firefox

The Fast and the Spurious

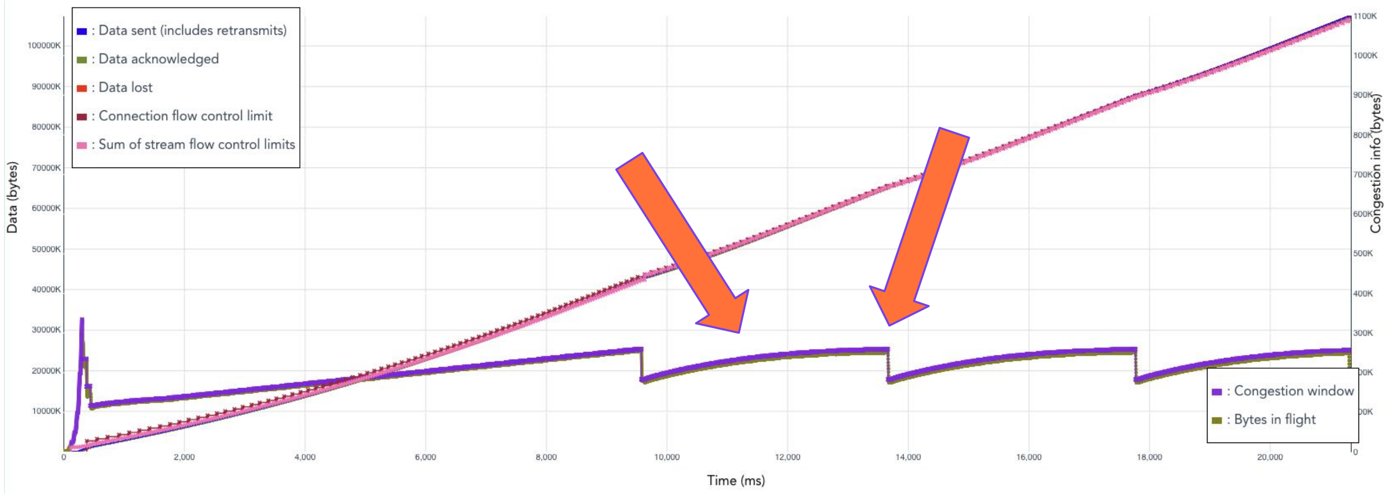
Congestion Control Experimentation in Firefox’s QUIC stack

2026-01-31 Oskar Mansfeld — FOSDEM

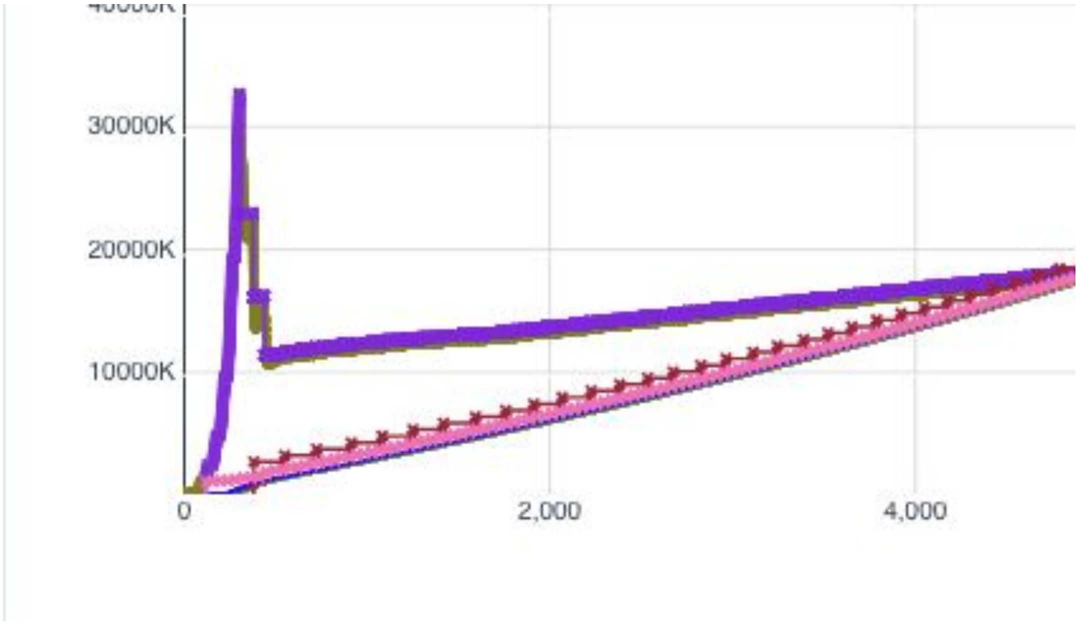
What is Congestion Control?

What is Congestion Control?

- Algorithmically finding ideal send rate
- Neqo implements [RFC 9438 – CUBIC]
- Additive increase and multiplicative decrease



Another Building Block: Slow Start

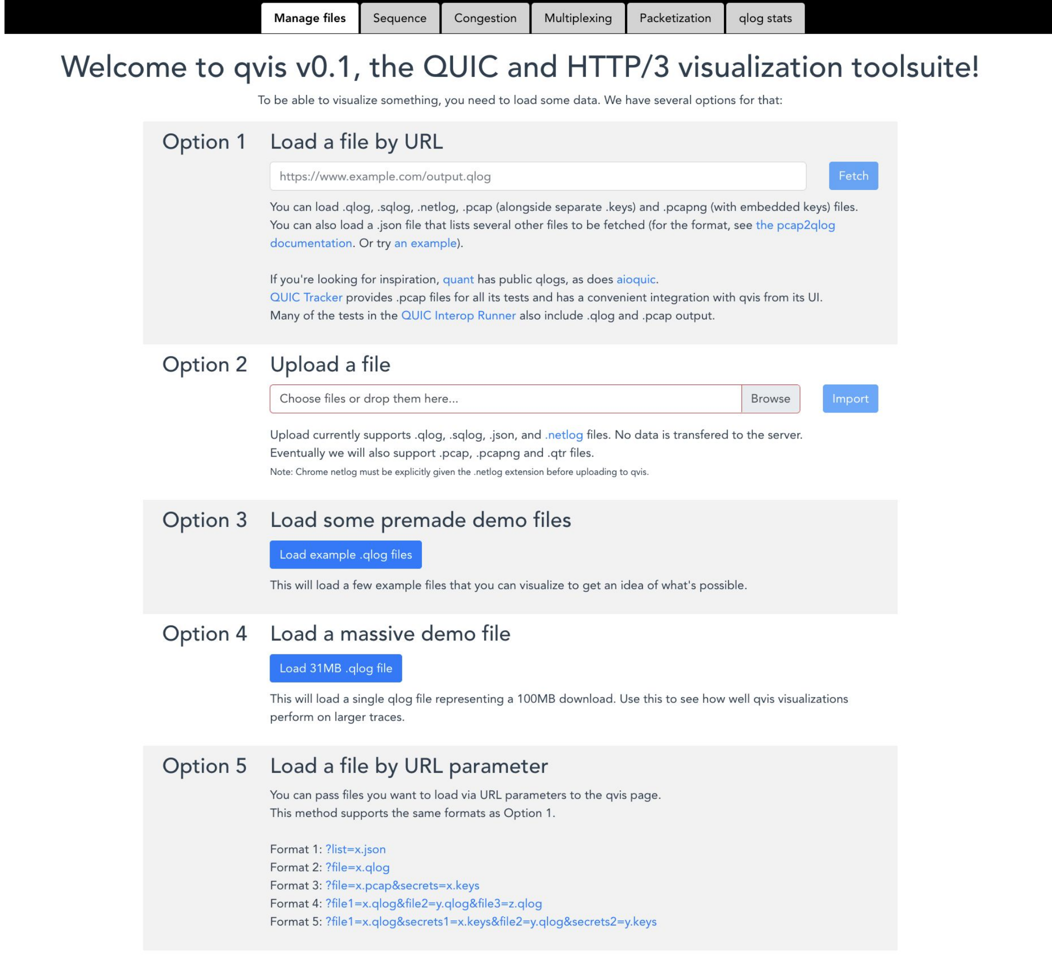


Interlude: Shoutout to qvis

Great tool to interactively visualize qlogs!

All Congestion Window plots in this presentation are screen captures from https://qvis.quictools.info/

Give them a star on https://github.com/quiclog/qvis



Welcome to qvis v0.1, the QUIC and HTTP/3 visualization toolsuite!

To be able to visualize something, you need to load some data. We have several options for that:

Option 1 Load a file by URL

You can load .qlog, .sqlog, .netlog, .pcap (alongside separate .keys) and .pcapng (with embedded keys) files.
You can also load a .json file that lists several other files to be fetched (for the format, see the pcap2qlog documentation. Or try an example).

If you're looking for inspiration, quant has public qlogs, as does aioquic.
QUIC Tracker provides .pcap files for all its tests and has a convenient integration with qvis from its UI.
Many of the tests in the QUIC Interop Runner also include .qlog and .pcap output.

Option 2 Upload a file

Upload currently supports .qlog, .sqlog, .json, and .netlog files. No data is transfered to the server.
Eventually we will also support .pcap, .pcapng and .qtr files.
Note: Chrome netlog must be explicitly given the .netlog extension before uploading to qvis.

Option 3 Load some premade demo files

This will load a few example files that you can visualize to get an idea of what's possible.

Option 4 Load a massive demo file

This will load a single qlog file representing a 100MB download. Use this to see how well qvis visualizations perform on larger traces.

Option 5 Load a file by URL parameter

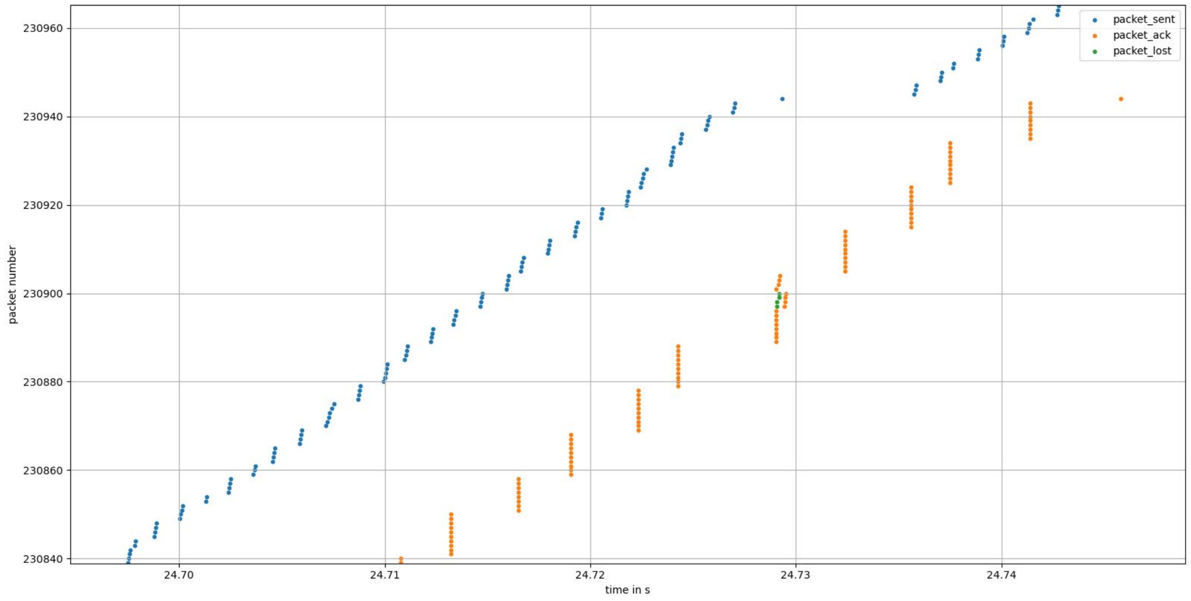
You can pass files you want to load via URL parameters to the qvis page.
This method supports the same formats as Option 1.

Format 1: ?list=x.json
Format 2: ?file=x.qlog
Format 3: ?file=x.pcap&secrets=x.keys
Format 4: ?file1=x.qlog&file2=y.qlog&file3=z.qlog
Format 5: ?file1=x.qlog&secrets1=x.keys&file2=y.qlog&secrets2=y.keys

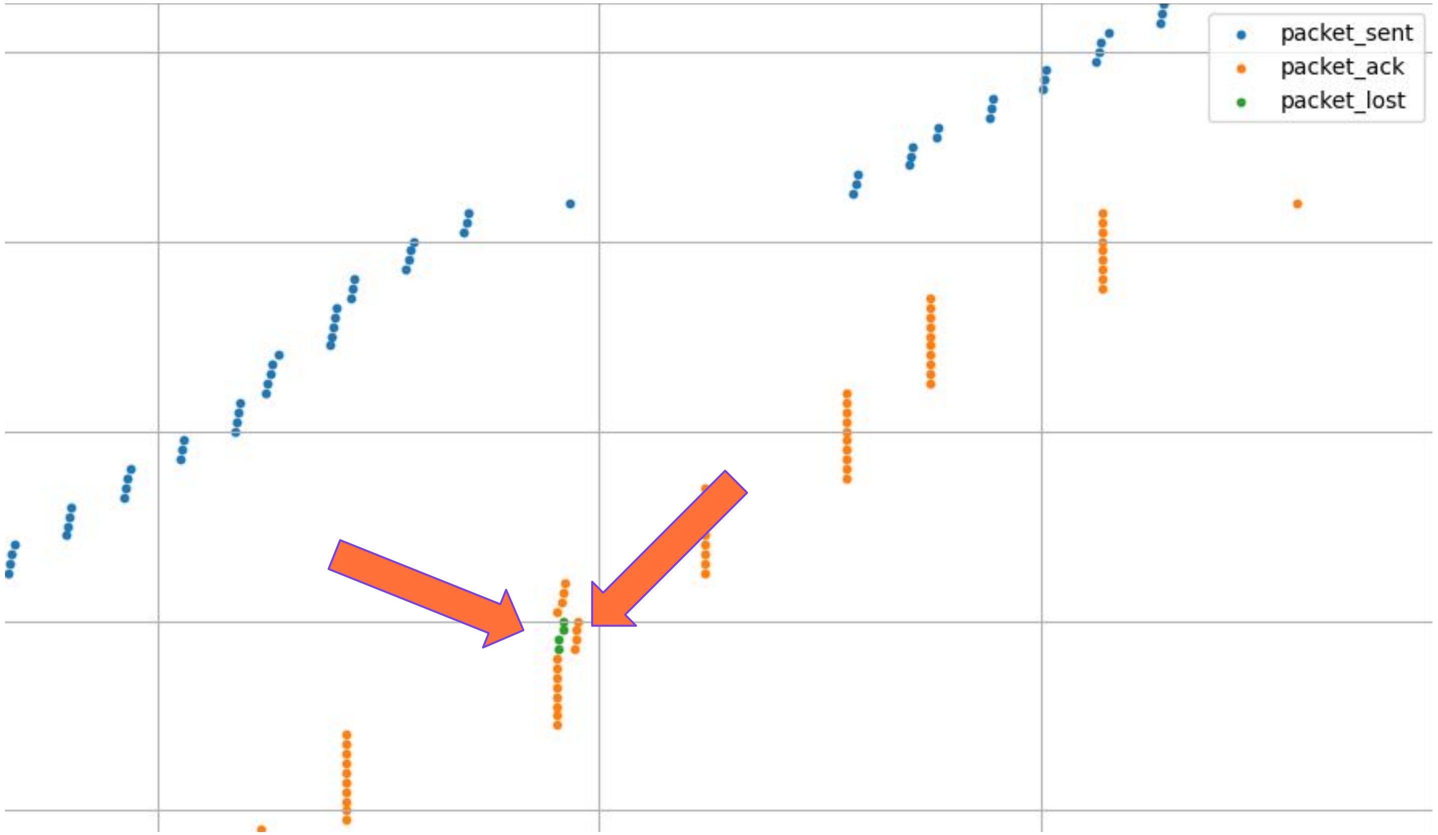
Spurious Congestion Event Recovery

Going back to FOSDEM 2024

- Manuel Bucher (today @ Firefox Privacy) held a talk "H3 upload speed" where he showed this graph:



Spurious Loss



Spurious Loss

- Leads to **Spurious Congestion Events**
- Window is reduced despite there not being real congestion
- Heavy and unnecessary performance degradation
- RFC 9348 section 4.9.2 suggests a mechanism for detection and recovery

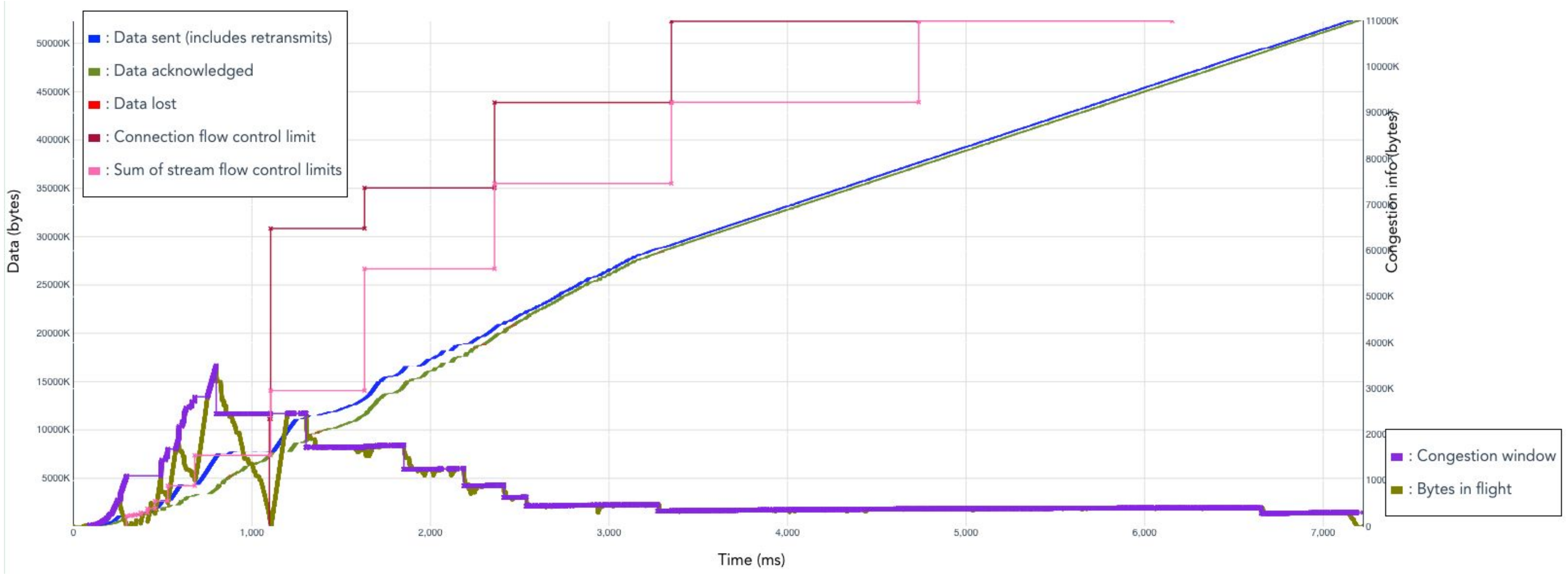
Implementation

- First implemented the detection logic and exposed metrics to gauge impact
- ~5% of connections see Spurious Congestion Events, but those that see them see a lot of them
- Recovery merged in #3298 and is now in Firefox Nightly 149

→ let's look at some measurements!

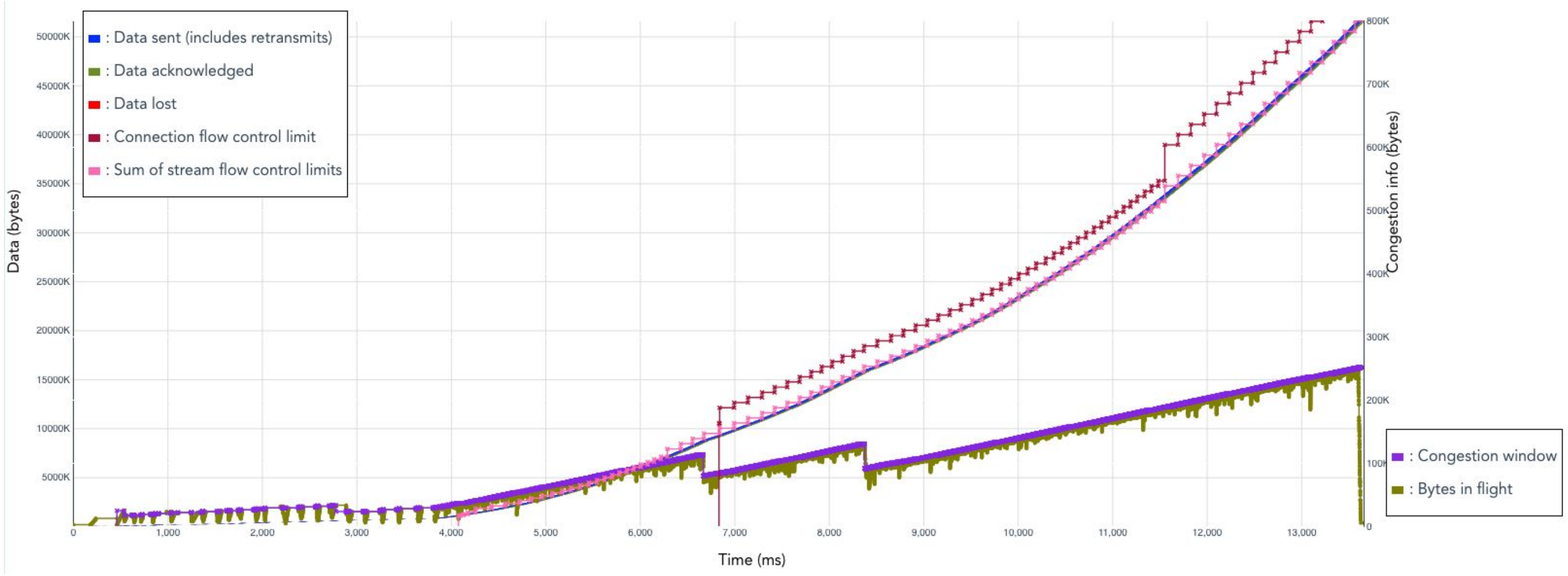
Measurements

No spurious congestion events ~7s completion time



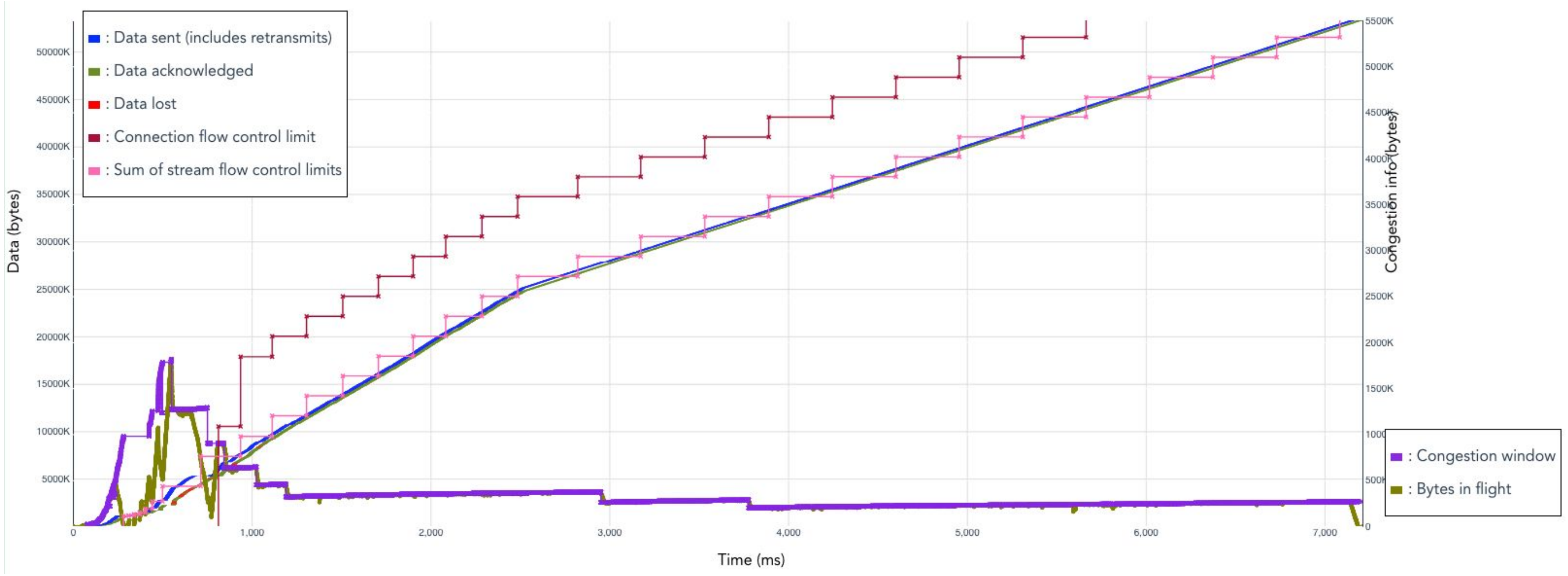
Measurements

Spurious congestion events ~13s completion time

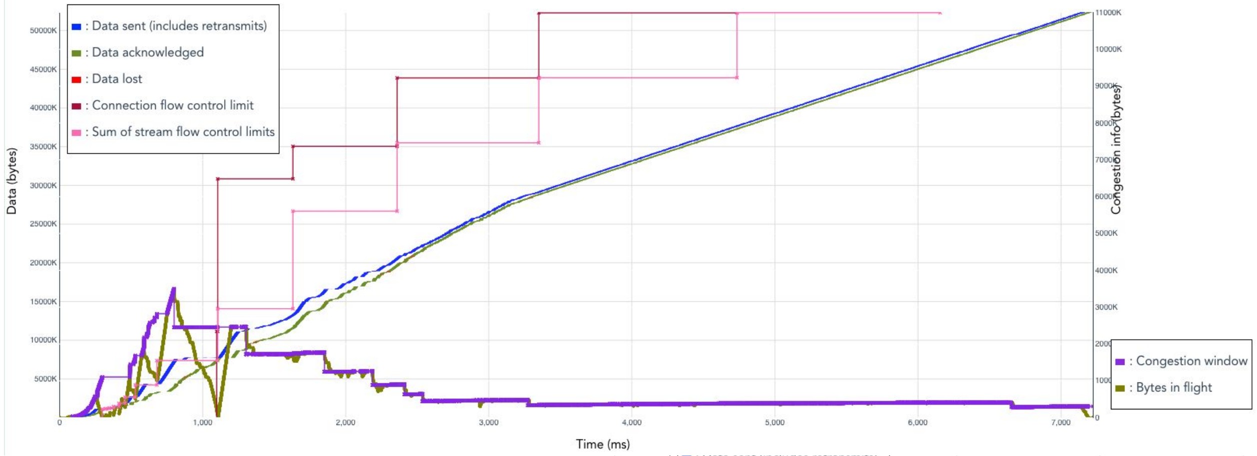


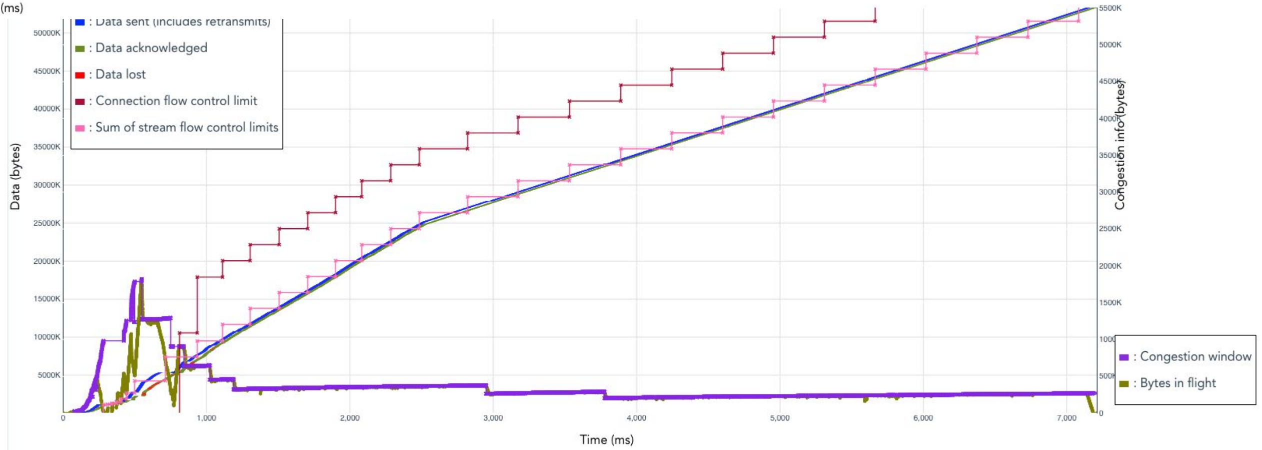
Measurements

With recovery patch ~7s completion time



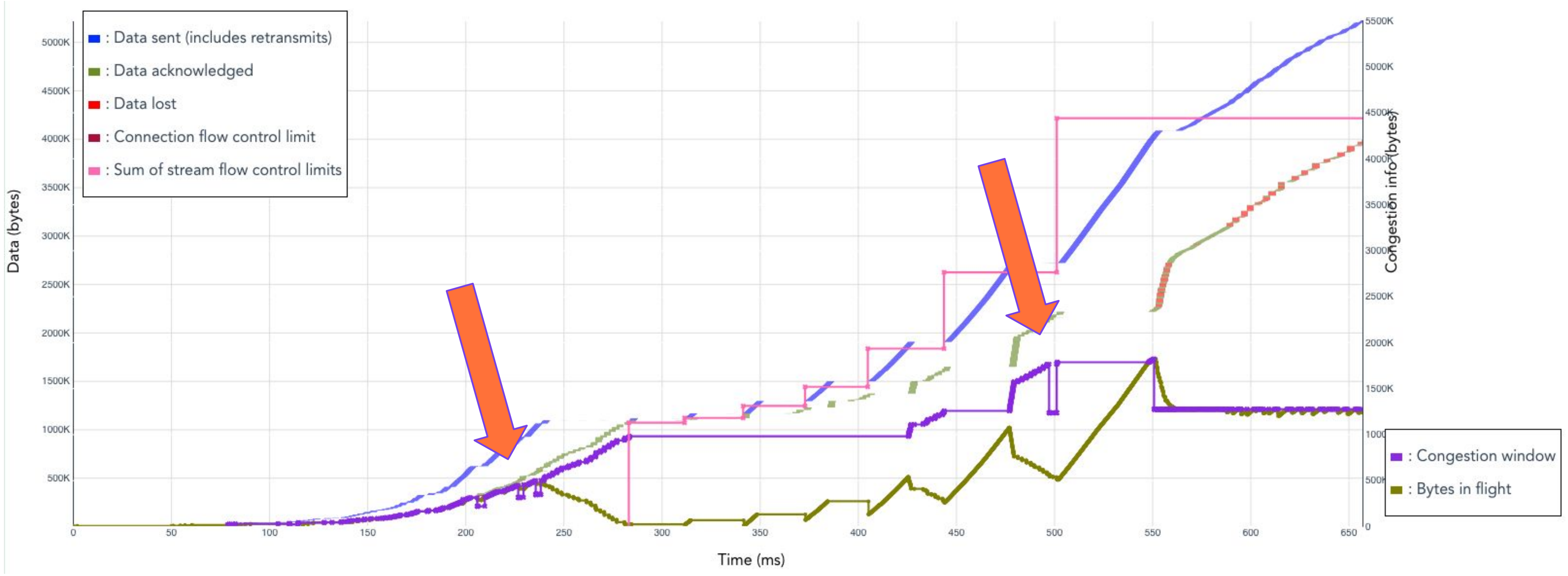
Measurements





Measurements

Recovery in action (zoomed in)



Measurements

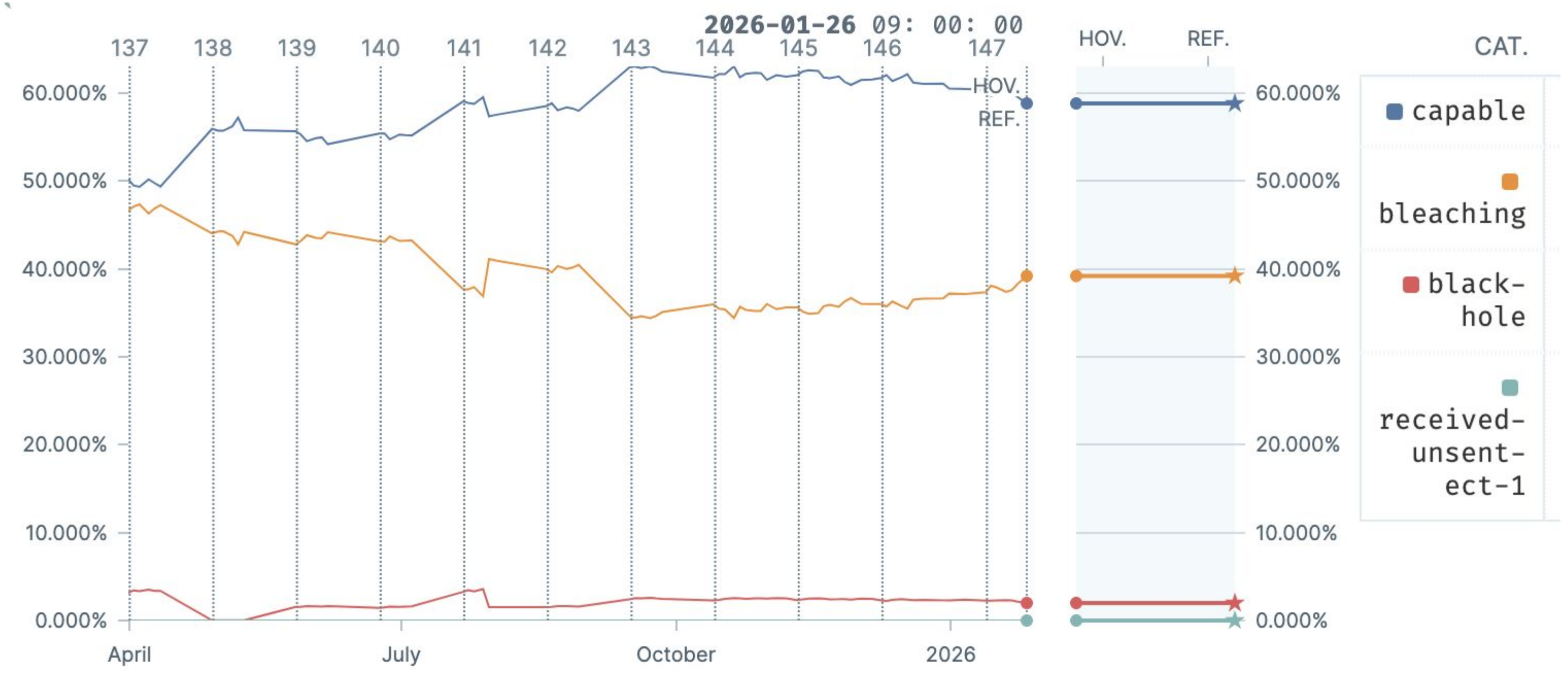
- We see strong anecdotal evidence and signal from micro-benchmarks
- But what about real world data?
- High level metrics very noisy due to heterogeneous environments
- So especially changes like this one that only apply to some subset of users are hard to see in metrics

Alternative Backoff with ECN

What is ECN?

- Explicit Congestion Notification (RFC 3168)
- Adjust rate without relying on packet loss as a signal
- Middlebox notifies sender if queue buildup starts by setting IP header value
- BUT path has to be capable of marking and passing along ECN signal

State of ECN



https://glam.telemetry.mozilla.org/fog/probe/networking_http_3_ecn_path_capability/explore?activeBuckets=%5B%22capable%22%2C%22bleaching%22%2C%22black-hole%22%2C%22received-unsent-ect-1%22%5D&app_id=beta&timeHorizon=ALL

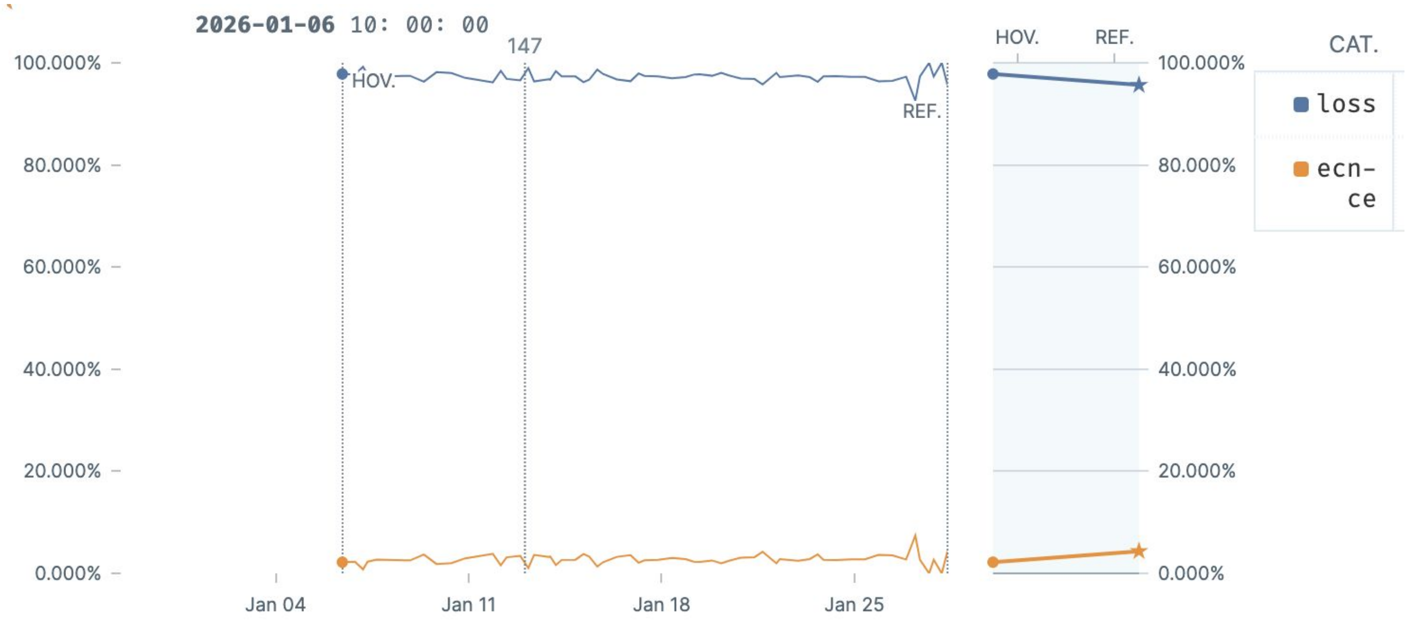
Alternative Backoff with ECN

- Usually ECN triggers the same window reduction as loss
- RFC 8511 suggests using a smaller decrease because ECN is an earlier signal than loss
- This would lead to higher overall utilization
- Looked good in simulations and merged in #3233
- But again, how does it behave in reality?

Analysis

- As usual, higher level metrics are too noisy
- Lower level metrics could have been influenced by other things bundled into the Neqo release
- Makes it hard to isolate feature impact

State of ECN



https://glam.telemetry.mozilla.org/fog/probe/networking_http_3_congestion_event_reason/explore?activeBuckets=%5B%22loss%22%2C%22ecn-ce%22%5D

How to fix the data problem?

Specialized metrics

- High level metrics with a built-in filter
- E.g. throughput for connections that see at least one spurious congestion event

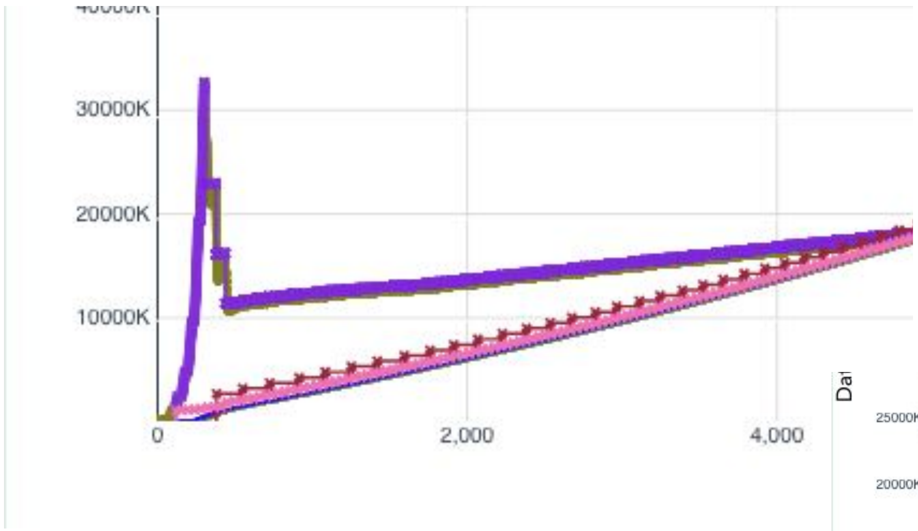
→ “see a% throughput increase for connections that see spurious loss, which is b% of connections”

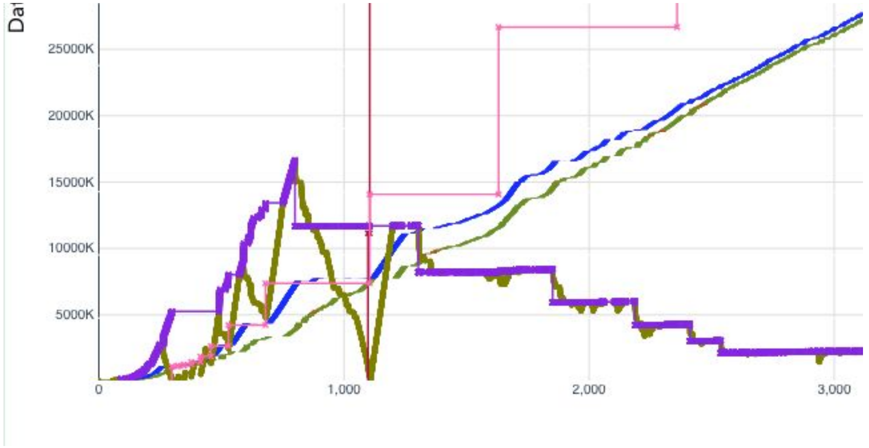
Run A/B experiments

- Isolate features
- Needs code instrumentation and takes time to set up and evaluate
- But gives most accurate signal

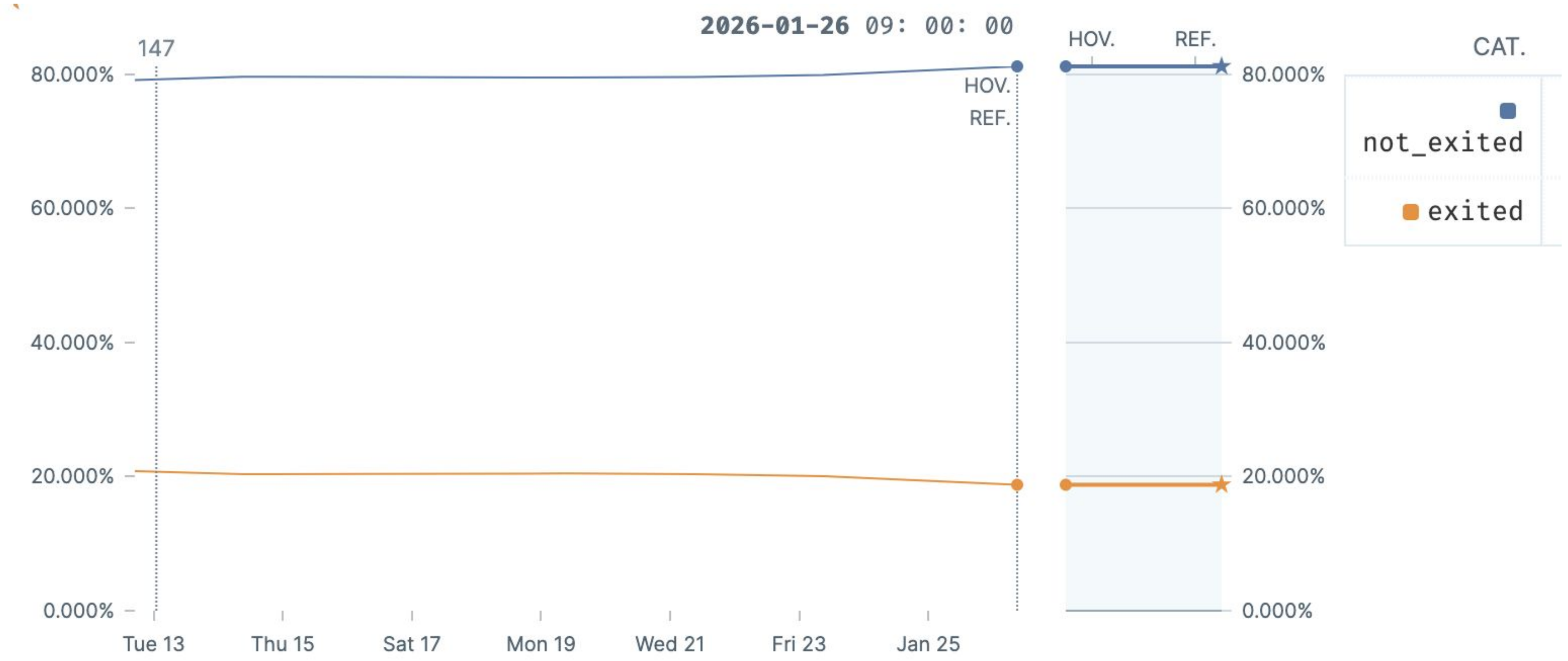
What’s Next: Slow Start Exit

The problem


40000K
30000K
20000K
10000K
0
2,000
4,000


Da
25000K
20000K
15000K
10000K
5000K
0
1,000
2,000
3,000

Worth optimizing?



https://glam.telemetry.mozilla.org/fog/probe/networking_http_3_slow_start_exited/explore?activeBuckets=%5B%22not_exited%22%2C%22exited%22%5D&app_id=beta&timeHorizon=ALL

What's next

- Working on a research project to compare slow start heuristics (HyStart++, SEARCH, maybe more) that aim to exit without packet loss
- Taking learnings, designing specialized metrics and scheduling time for experimentation
- Hopefully publish results in a paper to help advance emerging standards like the SEARCH project with real world data

Thank you!

Questions?

- FOSDEM hallway
- Mail: omansfeld@mozilla.com
- Matrix: @omansfeld:mozilla.org

- Check out our data!
  https://glam.telemetry.mozilla.org/
- Check out qvis! https://qvis.quictools.info/