



H3 upload speed

Table of contents

1. Background
2. HTTP/2 upload speed
3. HTTP/3 upload speed

Mozilla Necko

- Focus on Security, Privacy and Performance
- HTTP, DNS, WebSocket, WebTransport in Firefox Browser
- Caching and Proxies



Networking Performance

- Time to display a page
- How long it takes to upload large files like videos
- Focus on upload in past two years
- In this talk: HTTP/2 and HTTP/3 and their difference



HTTP/2 upload

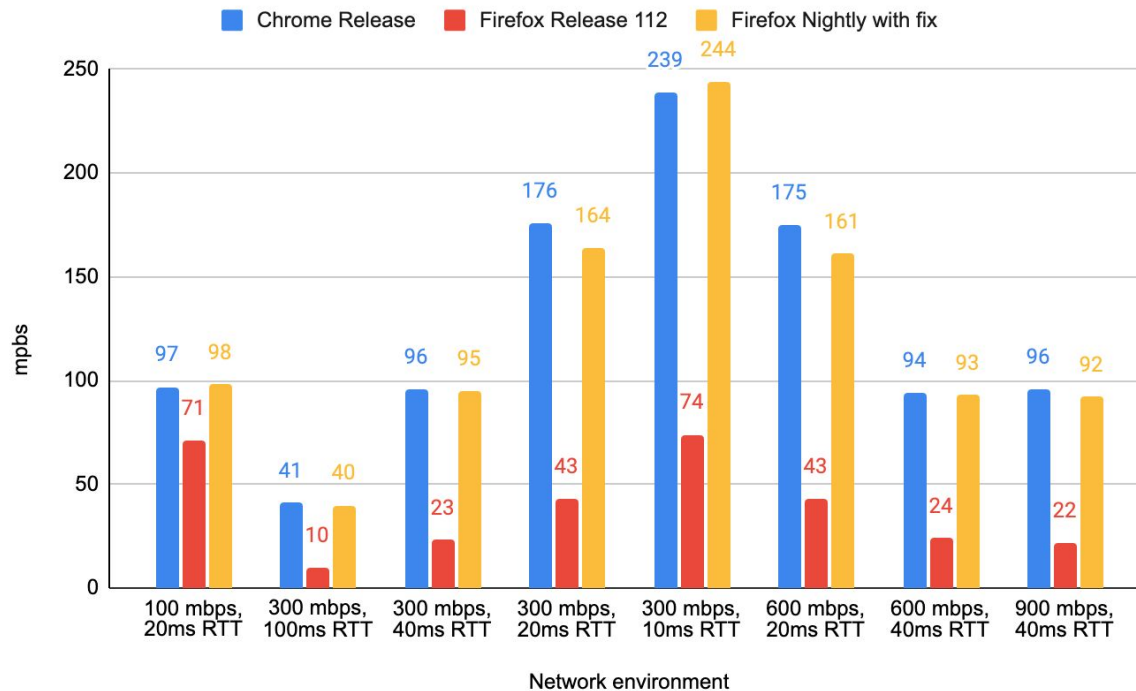
- HTTP/2: multiple HTTP requests over one TCP connection
- Operating system responsible for send performance in TCP
- Bug: Fixed size send buffer in operating system
- We just need to provide enough data to the operating system



HTTP/2 upload payoff

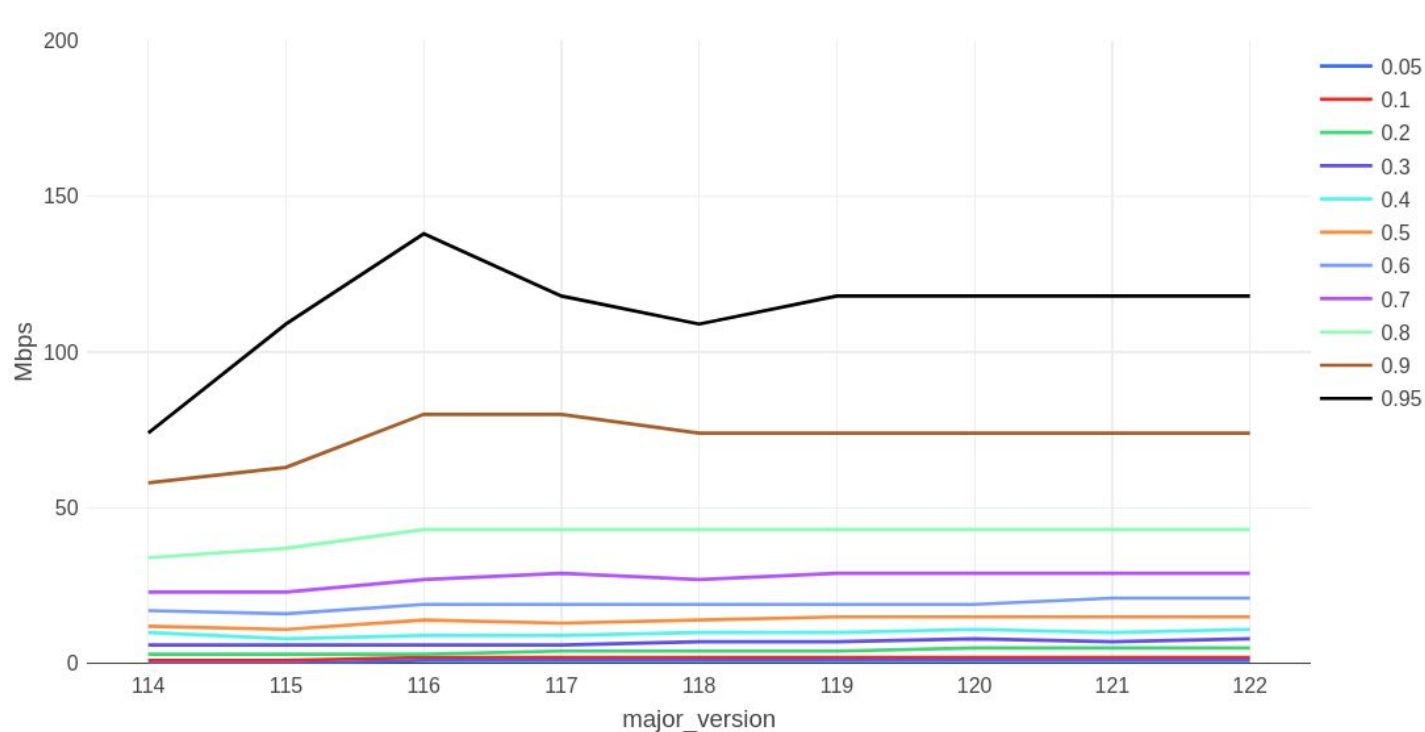
In some Network conditions upload took 4 times as long

HTTP/2 Upload throughput - Windows



HTTP/2 upload payoff

95 percentile doubled from 74 Mbps to 138 Mbps

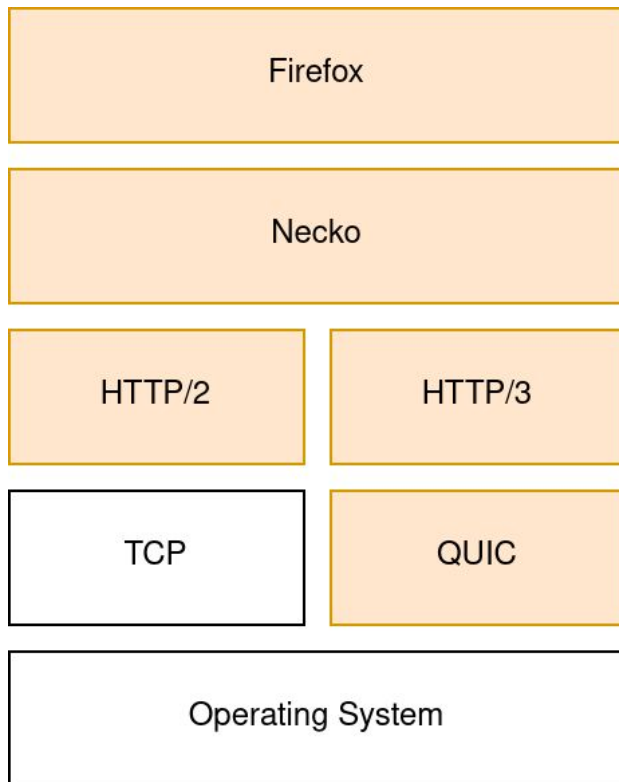


Differences HTTP/3 and HTTP/2

- HTTP/3: using QUIC as transport layer instead of TCP
- QUIC got standardized alongside HTTP/3 in 2021
- HTTP/3 in Firefox started in 2019, enabled in 2021
- Operating system responsible for send performance in TCP, Firefox in QUIC
- TCP got optimized for decades, QUIC in Firefox only for 2-4 years



Differences HTTP/3 and HTTP/2



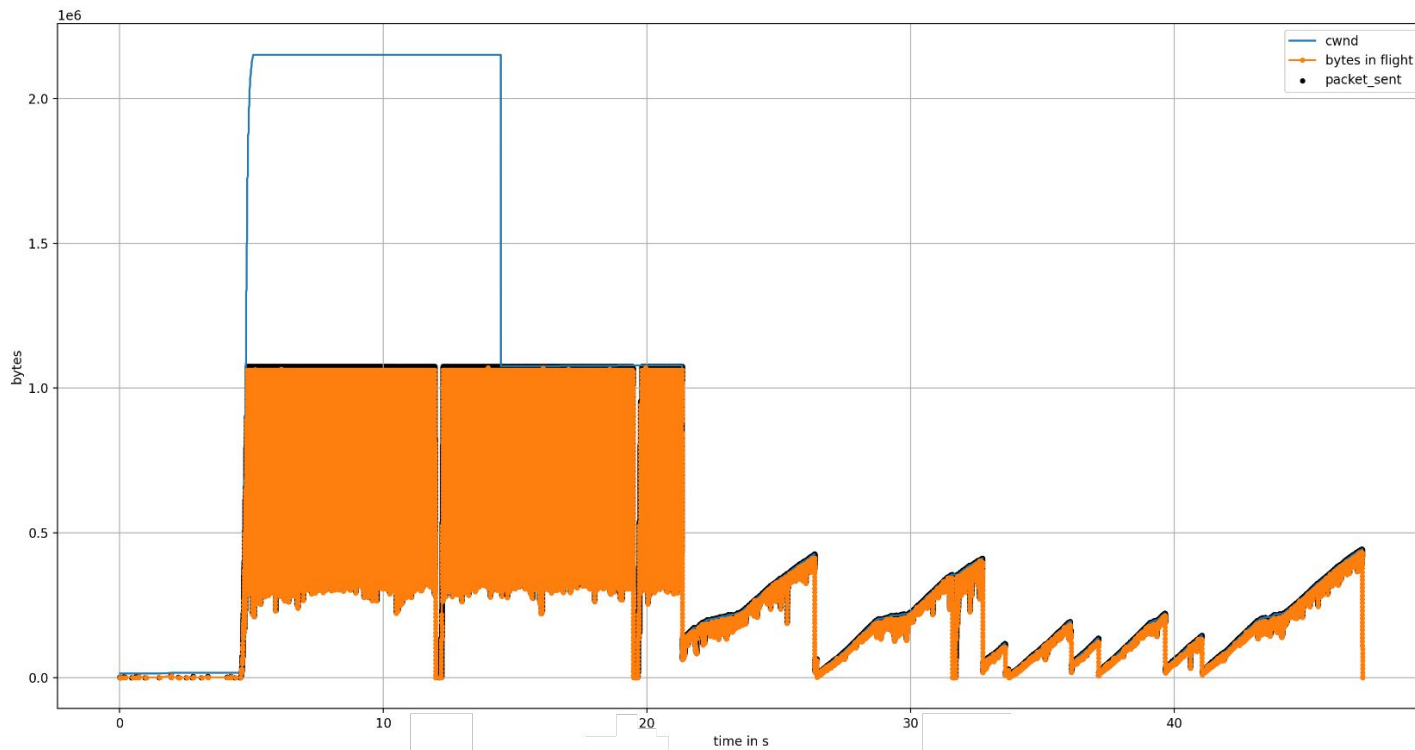
Concepts in QUIC (and TCP)

- Why: Multiple TCP connections over one QUIC
- TLS integrated → fewer round trip times Handshake
- Avoiding to overload the network: Congestion window (CWND)
- CWND is our estimate on our upload speed



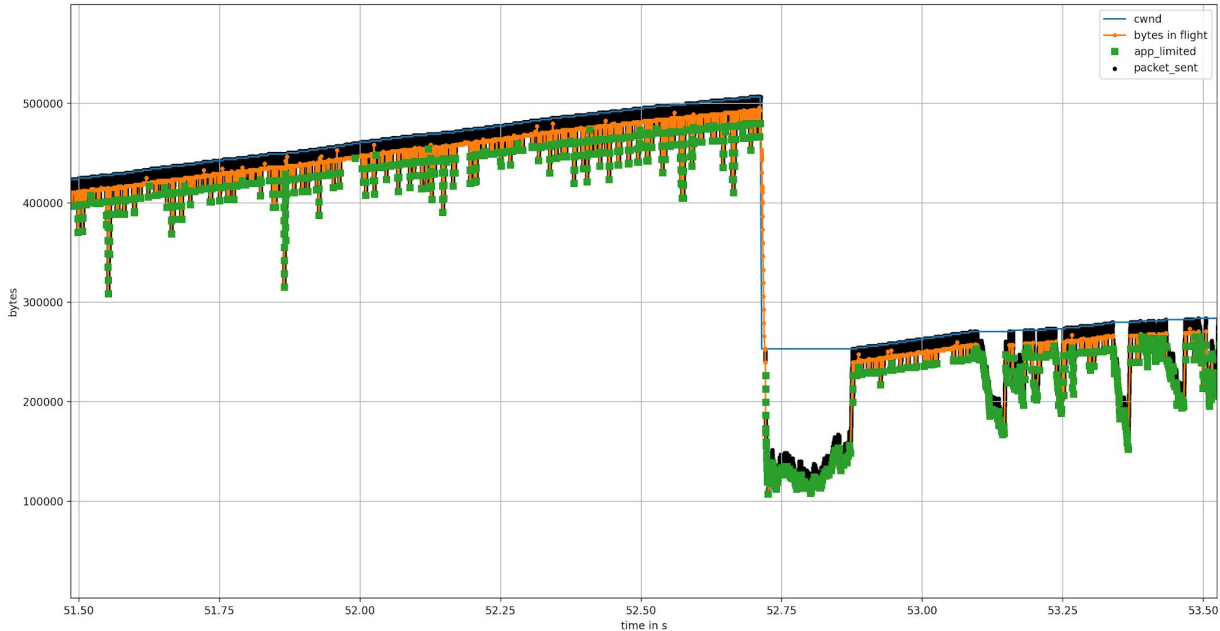
Congestion Window / Bytes in flight

IO-Graph from log: Congestion window drops to almost 0 instead of halving



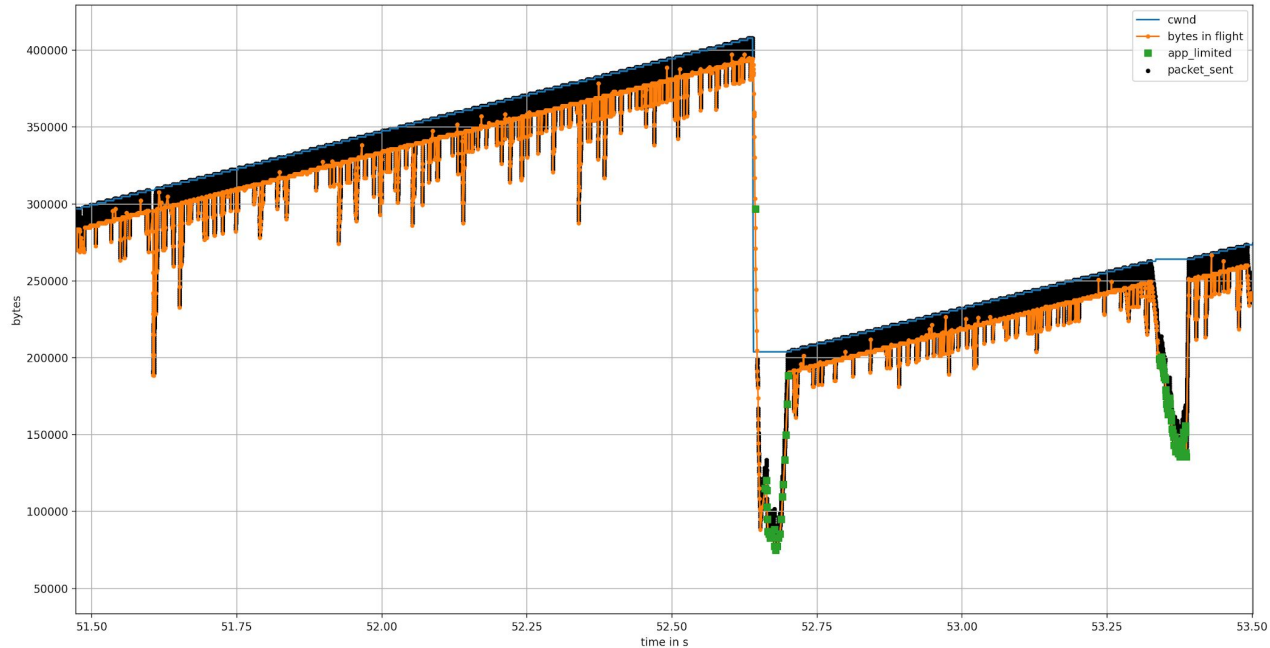
Congestion Window / Bytes in flight

New IO-Graph including app_limited packets in green



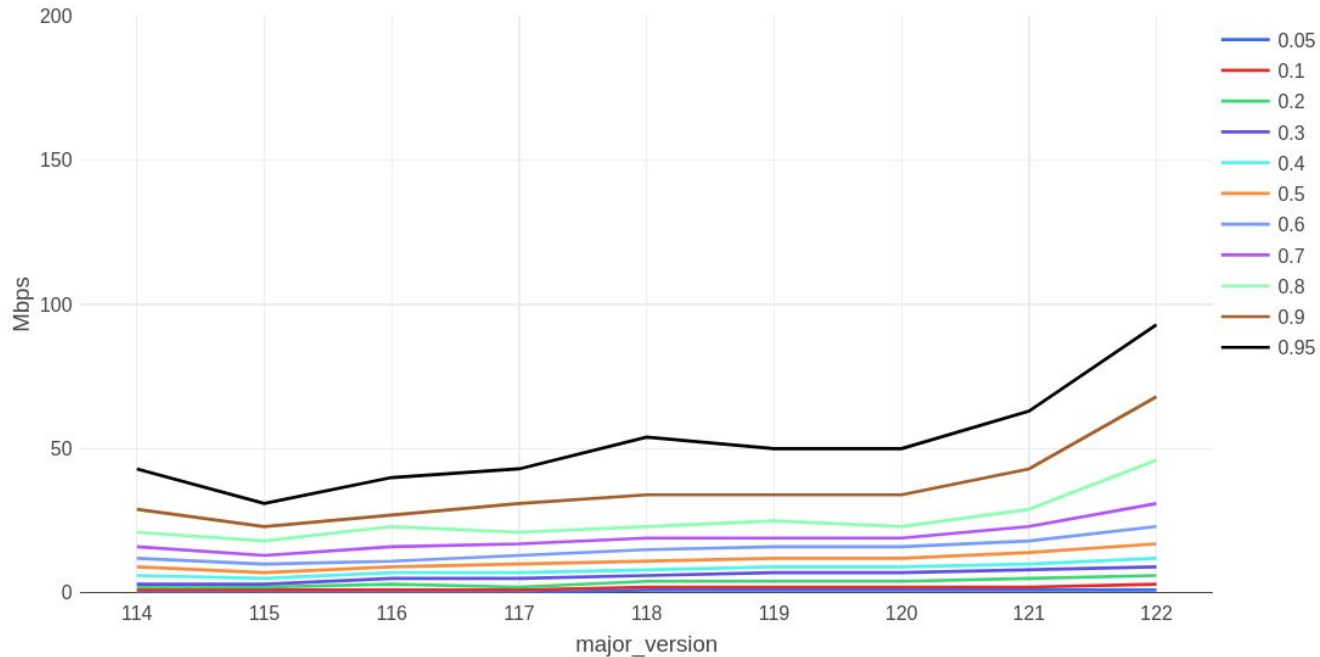
Congestion Window / Bytes in flight

Fixing app_limit detection allows CWND to increase 50-75% faster



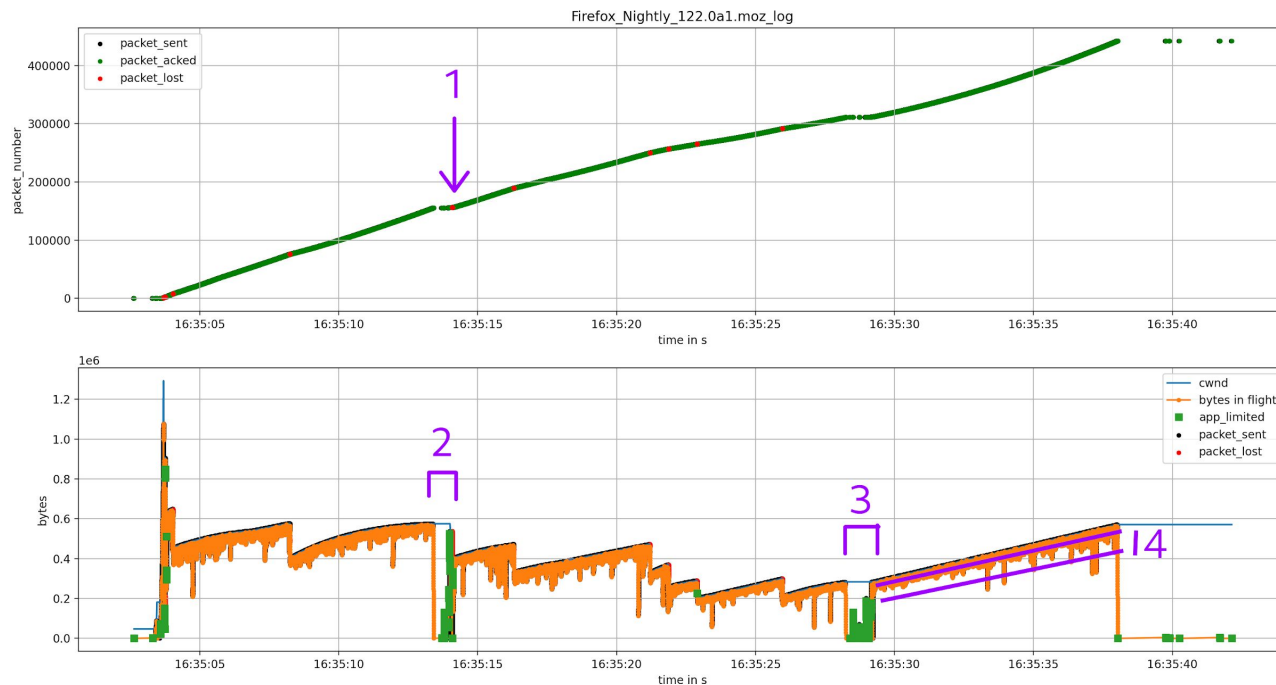
HTTP/3 sees steady increase

Tripled from 31 Mbps to 93 Mbps



Current state

3 more known problems that we are resolving



Further work

- CPU bottlenecks
- 1MB fixed size buffer
- Packet reordering causes too much congestion events
- Setting up CI to catch more regression
- Seen much improvements, but still a lot more potential



Thanks for raising bugs and contributing

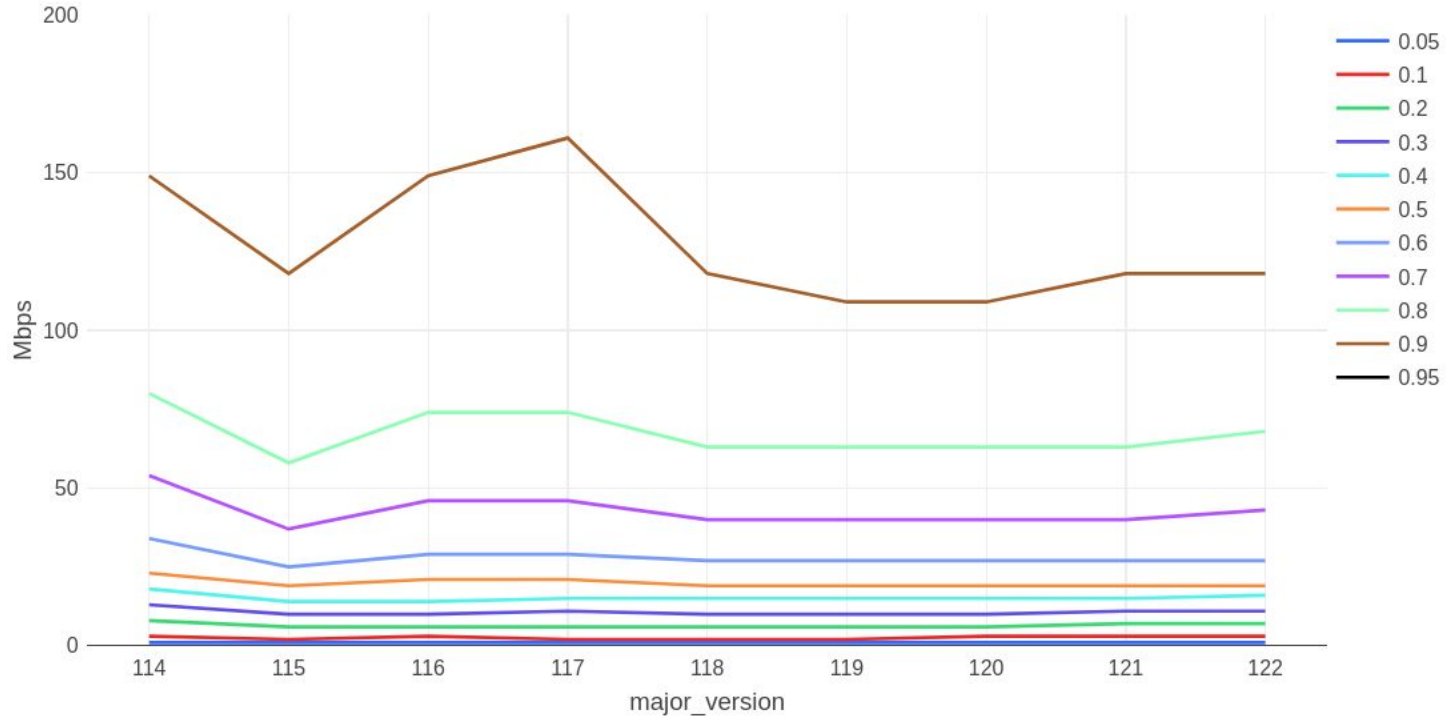
- <https://bugzilla.mozilla.org> (H3 upload: [Bug 1852924](#))
- Matrix: #necko:mozilla.org
- Necko specific documentation
 - <https://wiki.mozilla.org/Networking>
 - <https://firefox-source-docs.mozilla.org/networking>
- Blog: <https://mozilla-necko.github.io/>
- Meeting-notes: <https://mozilla-necko.github.io/meeting-notes>
- Office hours, where you can talk to us directly



Thank you!

HTTP/1 stayed more similar

95 percentile varies from 109 Mbps to 161 Mbps



Packet reordering

