On the benefits of reduced HoL blocking in QUIC

(via gQUIC, by Charles ‘Buck’ Krasic)

Background and Approach

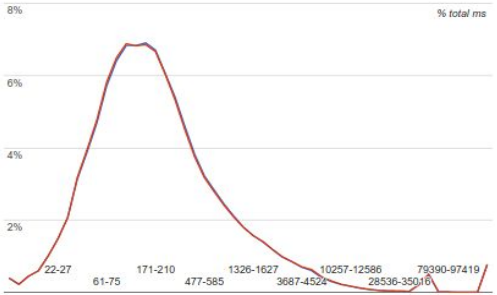
- Multiplexing is a core feature of HTTP/2
  - But layering over TCP vulnerable to HoL blocking
    - Not so for QUIC streams
  - No data until now
- Forced HoL blocking (FHOL)
  - Tweak gQUIC in Chrome to support tunneling all HTTP body data through a single stream
    - Uses HTTP/2 DATA frames on stream 3 (gQUIC headers stream)
    - FHOL writes are buffered, reduces or eliminate interleaving between sequential HTTP transactions

Experimental Setup

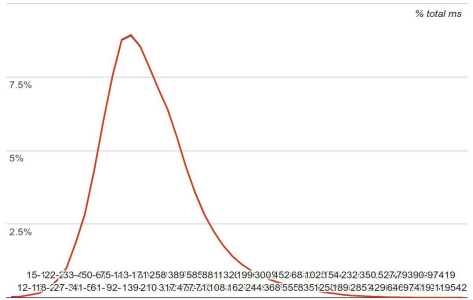
- Compare gQUIC to gQUIC+FHOL
- Measures latency impacts, YouTube QoE effects of (HTTP body) HoL blocking only
  - HPACK HoL header blocking not addressed here.
- Factors out other QUIC improvements: Zero-RTT, loss recovery, etc.
- Experimental data from Chrome Stable over two weeks (June 15th-29th)
  - Results from Chrome client metrics, Search and YouTube QoE pipelines

Latency Results (QUIC+FHOL vs QUIC)

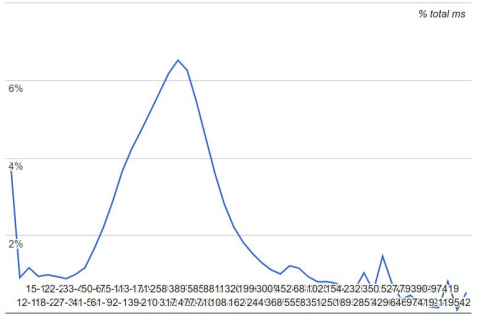
- Latency of individual HTTP transactions
  - Negligible at median
  - 95’th percentile
    - 98ms (3%) worse on Windows, 16ms (1%) worse on Android
    - Windows is 3.2s, Android is 1.6s
- Population differences
  - Suspect tail differences due to hanging gets, etc.



Windows QUIC



Android QUIC



Android (not QUIC)

QoE Results (QUIC+FHOL vs QUIC)

- Search Latency
  - FHOL on Windows 0.16% slower, 0.82 fewer events. Android 0.13% slower.
    - Fewer events suggests increased user abandons due to poorer experience.
- YouTube QoE
  - Windows: Mean Time Between Rebuffers down 2%, Rebuffer Rate increased 2.5%, Mean Client Bandwidth down 2.32%
  - Android: Mean Client Bandwidth down 2.98%
    - Far smaller population for Chrome based YT playbacks

Summary

- HoL blocking is negligible at the median, but adds 1-3% latency at the tail
- Tail performance is impactful on higher level QoE
  - We A/B test constantly, *FHOL was a top mover of YouTube QoE* of recent experiments.
- ...and what about HPACK?
  - Measured max HoL blocking time on headers stream per connection
  - Android: Median - 157 ms, 95’th percentile 2.6 seconds
  - Suggests that QoE improvements for fixing HPACK HoL blocking could be comparable, probably *somewhat greater* than FHOL deltas.