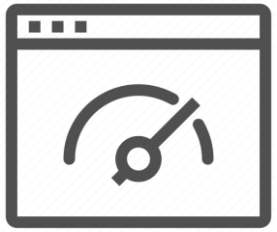
Dissecting Performance of Production QUIC

Published in: the Web Conference 2021

Summarized by
Sangwon Lim (sangwonlim@snu.ac.kr)

Contents

Introduction



- Web performance has become a crucial concern for online services
- One of the core protocols, QUIC, has gained increased adoption
- QUIC is currently the foundation for emerging protocols, e.g., HTTP3

QUIC overview

- QUIC is a new transport protocol designed to replace TCP + TLS

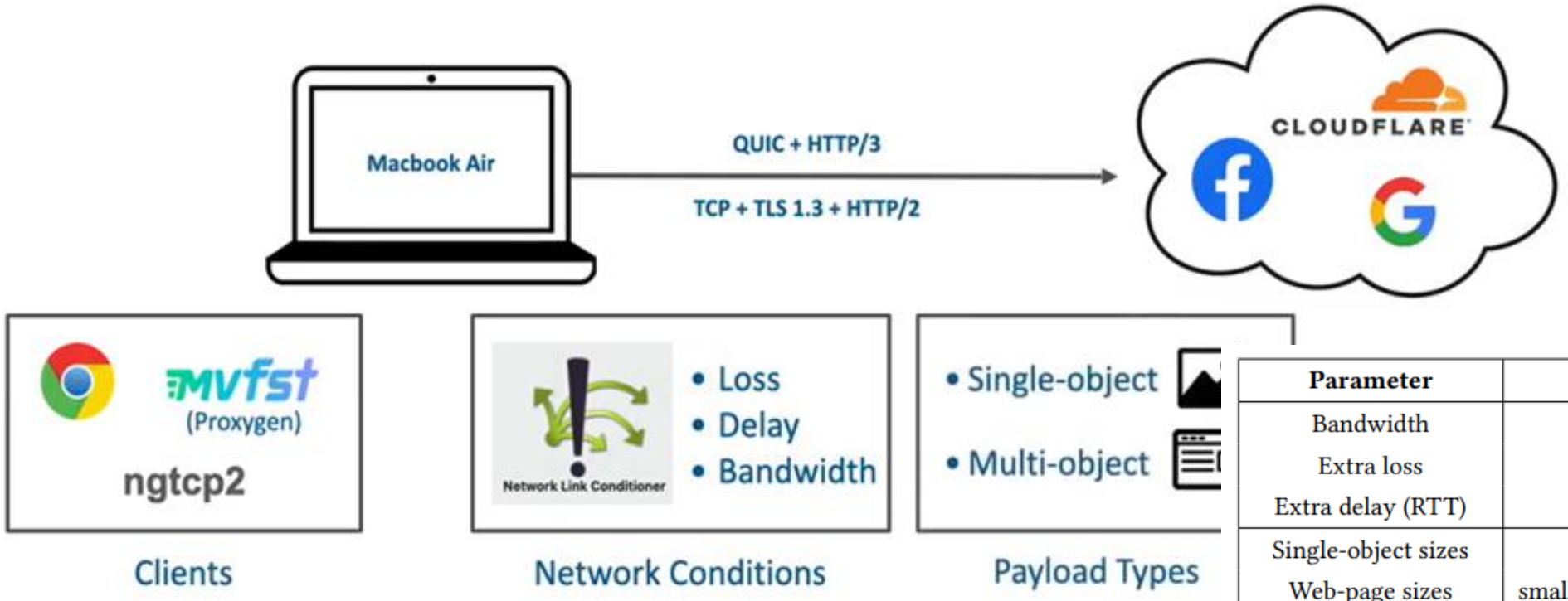
- QUIC's primary objective is improved latency over TCP.

  - ✓ User-space Implementation
  - ✓ Faster Connection Establishment
  - ✓ Removal of TCP's Head-of-Line (HOL) Blocking Problem
  - ✓ Improved and Simplified Recovery Mechanism

Related works

Author	QUIC Version	Prod-uction end-points	Root Cause Analy-sis	Results	Explanation
Google [27]	h3-29	Yes	No	QUIC improved client desktop throughput by 3%, decreased search latency by 2%, and decreased video rebuffer rates by 9%.	None provided.
Facebook [35]	N/A[2]	Yes	No	QUIC reduced request errors by 6%, reduced tail latency by 20%, and reduced mean-time-between-rebuffering (MTBR) by 22%.	QUIC's state-of-the-art recovery mechanisms improves tail metrics.
Cloudflare [50]	h3-27	Yes	No	H3's performance was 1-4% worse than H2's across multiple POP locations.	Different congestion control used for TCP and QUIC.
Saif [45]	h3-27	No	No	In a local environment, QUIC performed worse for all network conditions except high loss.	Code churn and unoptimized open-source implementation.
Codavel [10]	h3-20	No	No	During packet loss, QUIC performed better for small payloads (250KB) but substantially worse for large payloads (17MB)	Unoptimized open-source implementation and different congestion control.
This work	h3-29	Yes	Yes	QUIC performed favorably for Google under all network scenarios but had mixed results for Facebook and Cloudflare.	QUIC's performance is largely determined by its congestion control implementation.

Testbed setup



Parameter	Values Tested
Bandwidth	10mbps
Extra loss	0%, 0.1%, 1%
Extra delay (RTT)	50ms, 100ms
Single-object sizes	100KB, 1MB, 5MB
Web-page sizes	small (≤ 0.47MB), medium (≤ 1.54MB), large (≤ 2.83MB)
Content Providers	Facebook, Google, Cloudflare

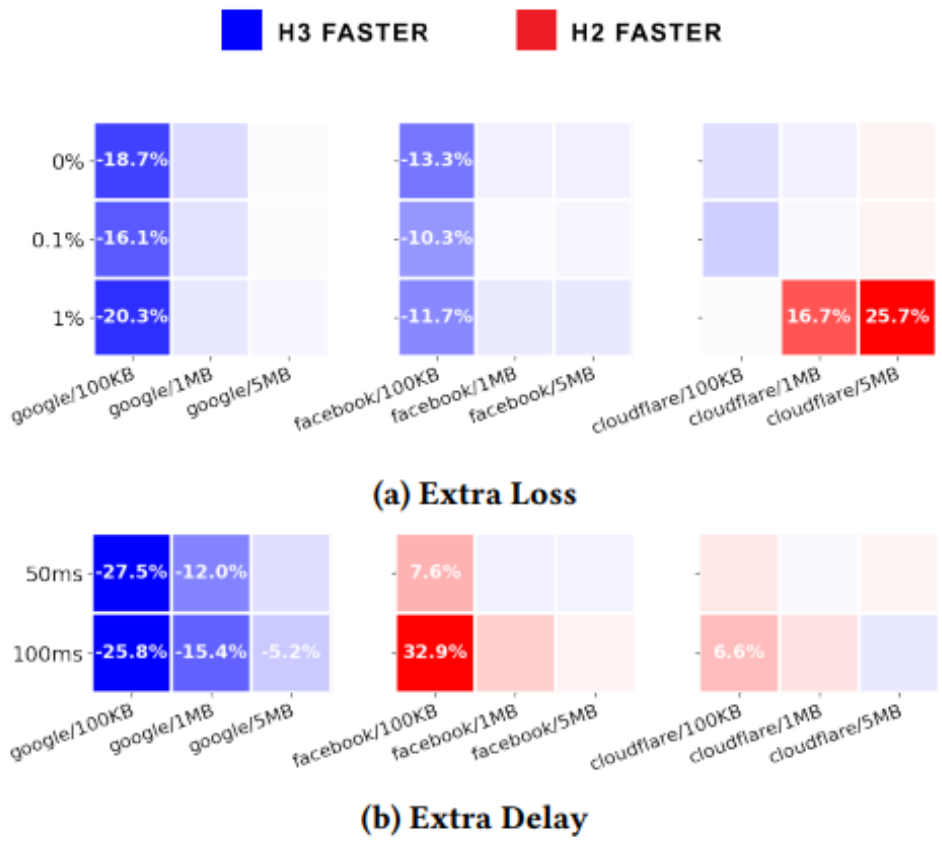
Insight #1

- QUIC's faster connection establishment has a noticeable but limited impact

✓ H3 (QUIC) was faster for 100KB in most cases

✓ For 1MB or 5MB endpoints, performance differences between H2 and H3 were negligible

✓ In some cases, QUIC performed much worse, but this behavior was inconsistent across content providers



** Less than 5% are not labelled*

Insight #2 (1/2)

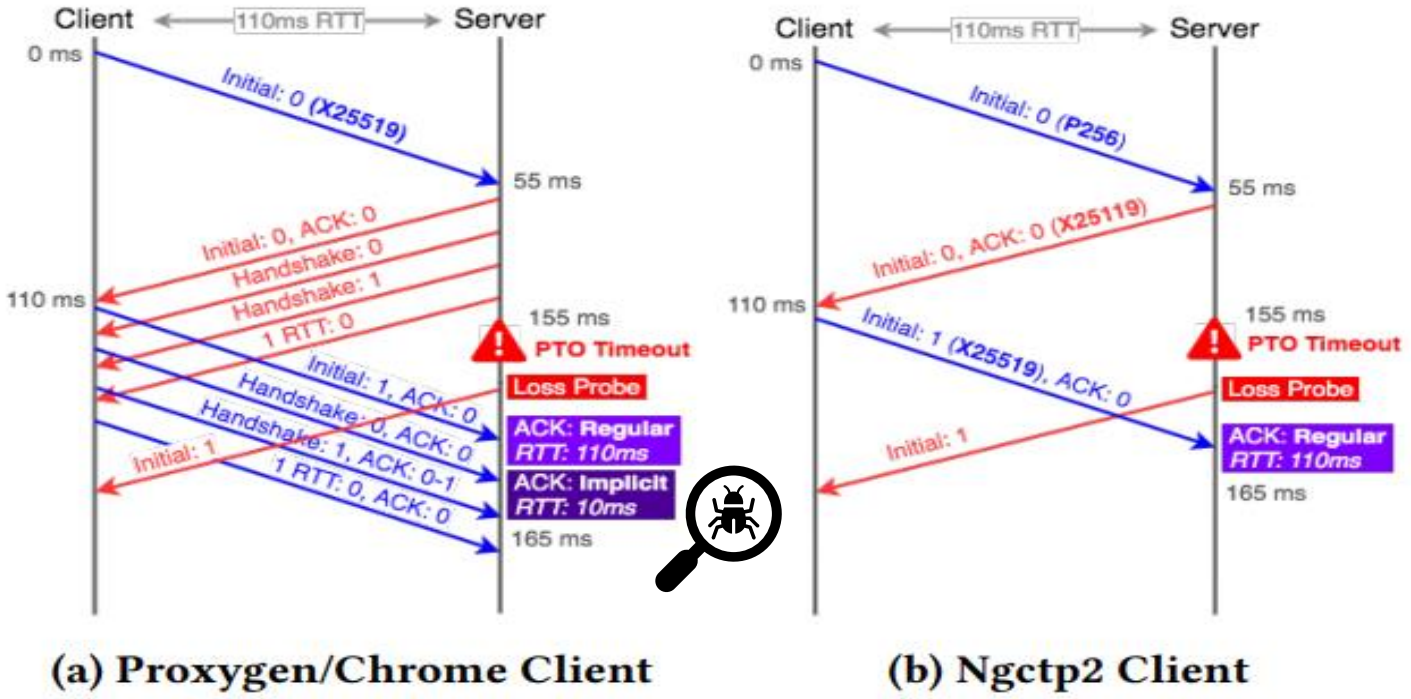
- QUIC clients are not always configured similarly, which can lead to performance problems when servers are configured with specific clients in mind

(a) 0% Loss (b) 1% Loss (c) 100ms Delay

Performance comparison between H3 clients / dark colors means worse performance

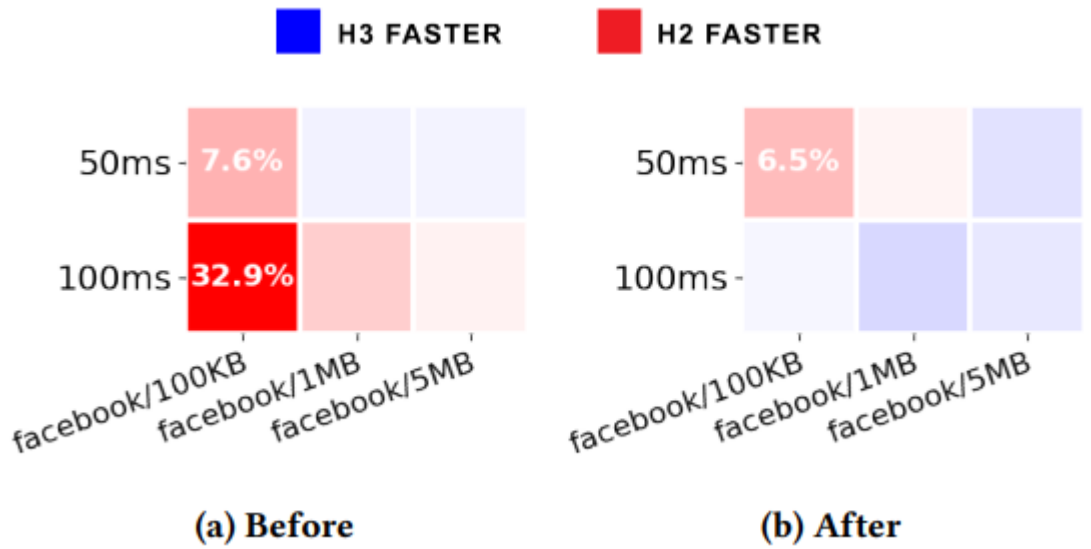
Insight #2 (2/2)

- QUIC clients are not always configured similarly, which can lead to performance problems when servers are configured with specific clients in mind



Insight #3

- Congestion control implementation quality matters a lot for QUIC performance

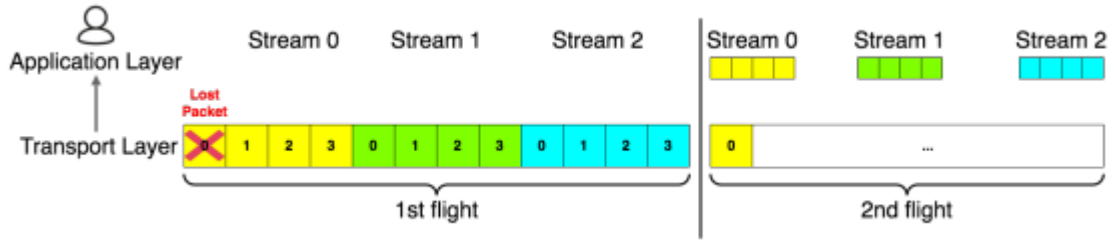


Before and after implicit ACK patch

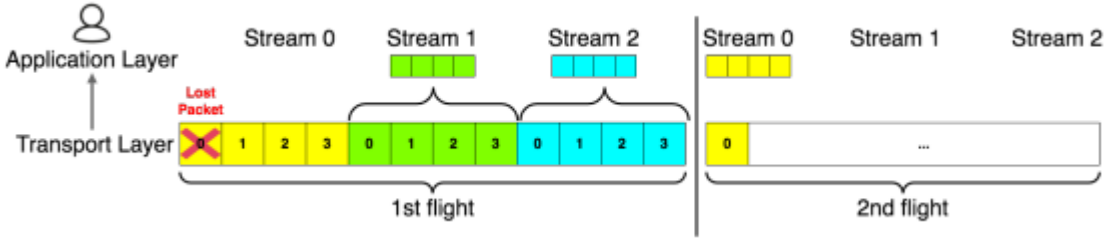
Insight #4 (1/3)

- QUIC's removal of HOL blocking has little impact on web-page performance relative to congestion control and QUIC's faster connection establishment

TCP

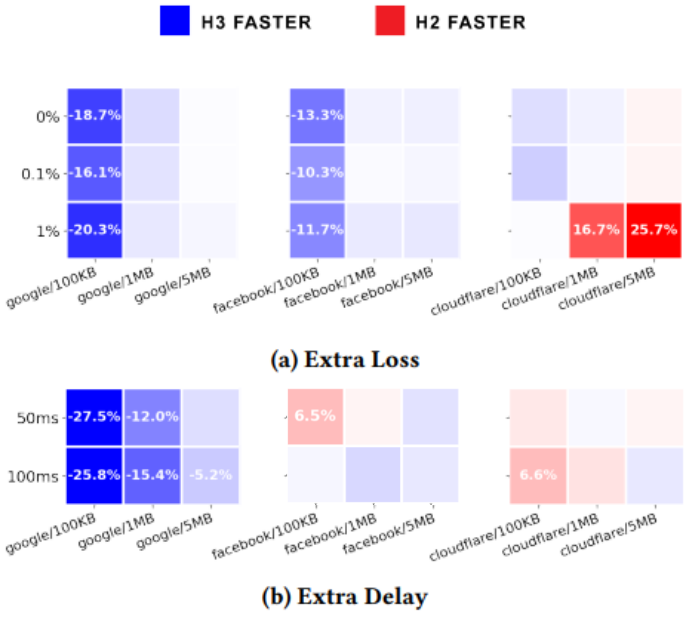


QUIC

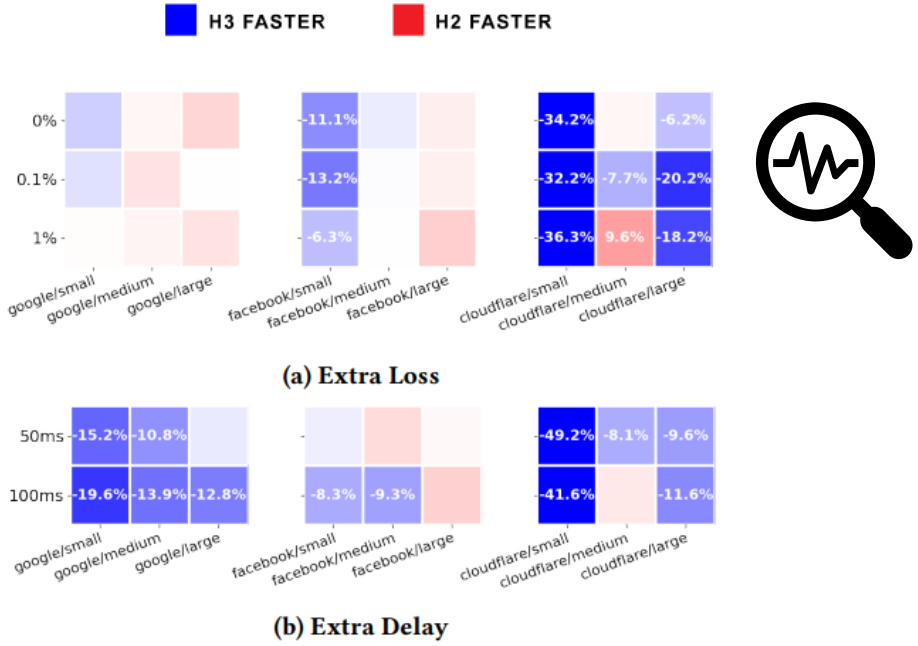


Insight #4 (2/3)

- QUIC's removal of HOL blocking has little impact on web-page performance relative to congestion control and QUIC's faster connection establishment



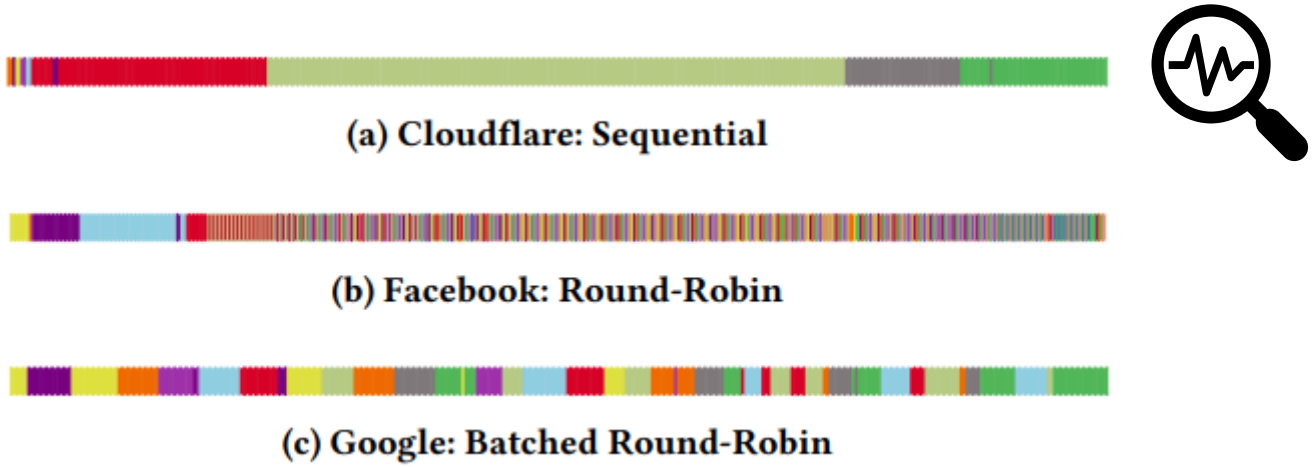
Single Object



Multi Object

Insight #4 (3/3)

- QUIC stream multiplexing strategies



(a) Cloudflare: Sequential

(b) Facebook: Round-Robin

(c) Google: Batched Round-Robin

Each color represents a unique HTTP stream and each segment represents consecutive frames from the same stream.

Conclusion

- Existing studies comparing QUIC and TCP performance have flaws
- Implementation quality and operator configuration > protocol design
- Optimizing QUIC in practice can be difficult and time-consuming