On the Interplay between TLS Certificates and QUIC Performance

Published in: CoNEXT ‘22

Summarized by

Sangwon Lim (sangwonlim@snu.ac.kr)

2023-06-08

Contents

Introduction

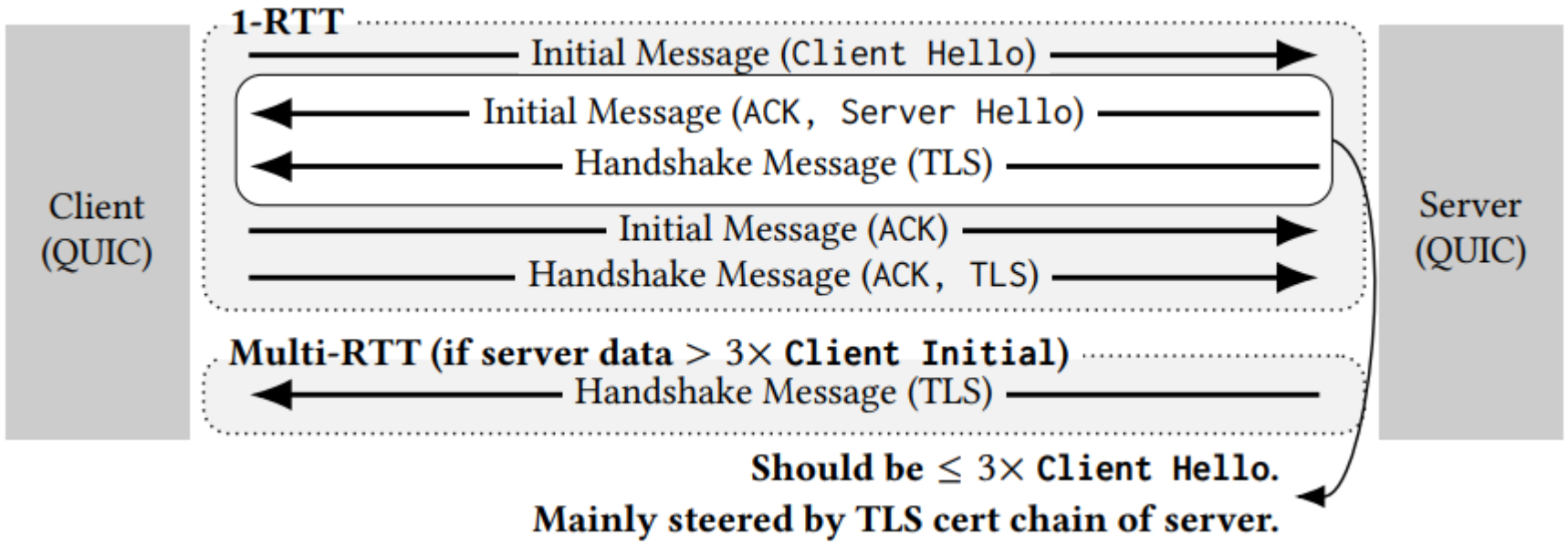
- The QUIC protocol (RFC 9000) was designed to improve Web performance and reduce access latency while keeping communication confidential

- A key approach is **the reduction of initial round trip times** by integrating the QUIC handshake with the TLS 1.3 handshake and coalescing multiple QUIC packets into one UDP datagram

- The authors revisited QUIC connection setup performance

Background

- In QUIC handshakes, server replies are limited to 3×the size of the client Initial[1)] until the client is verified



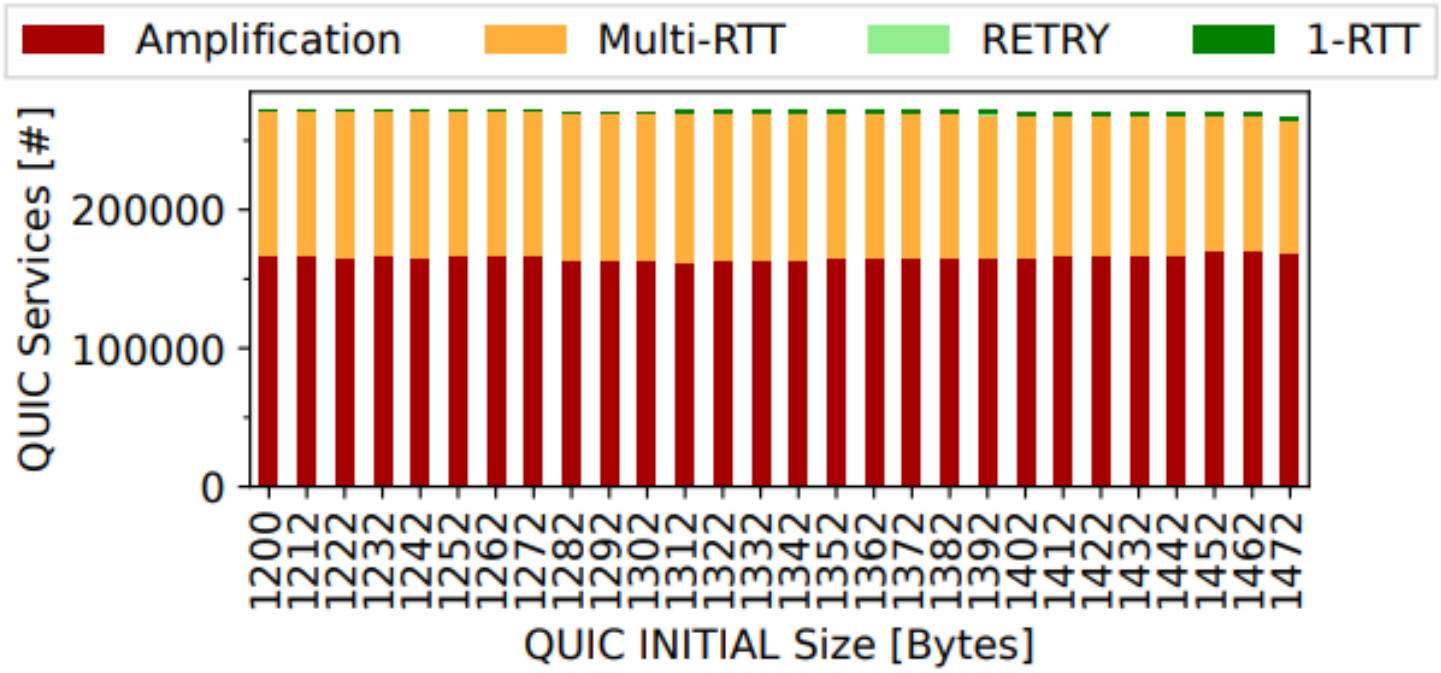
1) *Common client initial packet sizes are 1,250 bytes for **Chromium**-based browsers and 1,350 bytes for Firefox*

QUIC Scan method

- Target: all QUIC-reachable names from the Tranco 1M top domain list
  - ✓ 272k QUIC-enabled services
- Tool: quicreach, which does not support certificate compression
- Scenarios:
  - Complete handshake
    - 1-RTT (optimal)
    - RETRY (less efficient)
    - Multi-RTT (unnecessary)
    - Amplification (not RFC-compliant)
  - Incomplete handshake

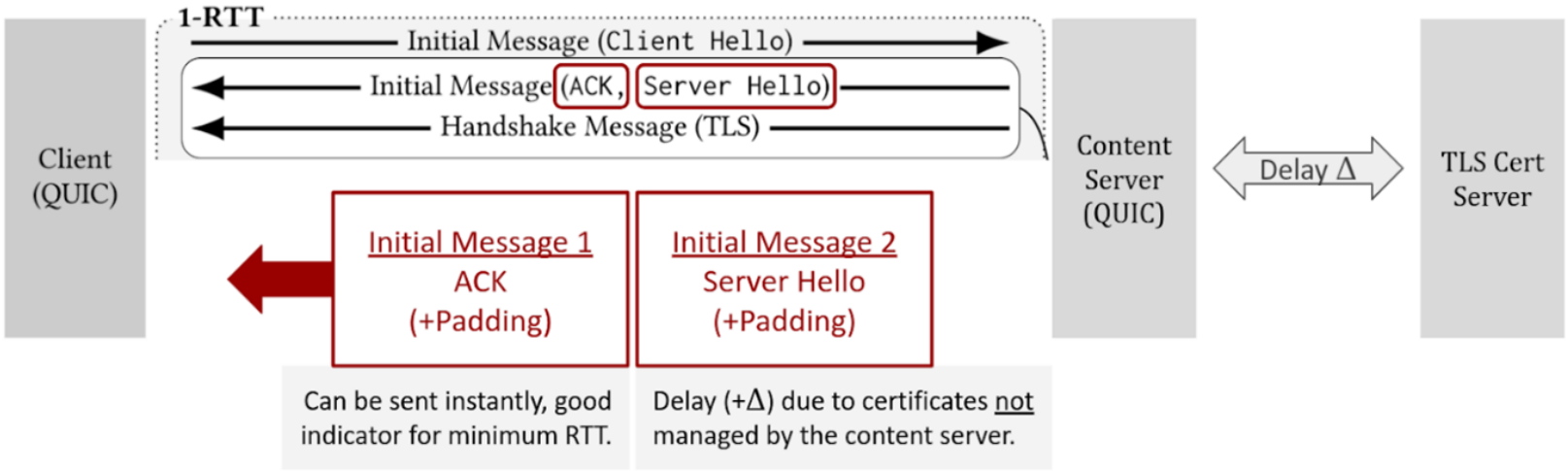
QUIC handshakes in the wild (complete)

- Almost no ideal handshakes (1-RTT) in different initial sizes
  - ✓ Amplification: 61%, Multi-RTT: 38%, and RETRY&1-RTT: very rare for Initial size of 1362 bytes (similar to Firefox)



> Coalescence

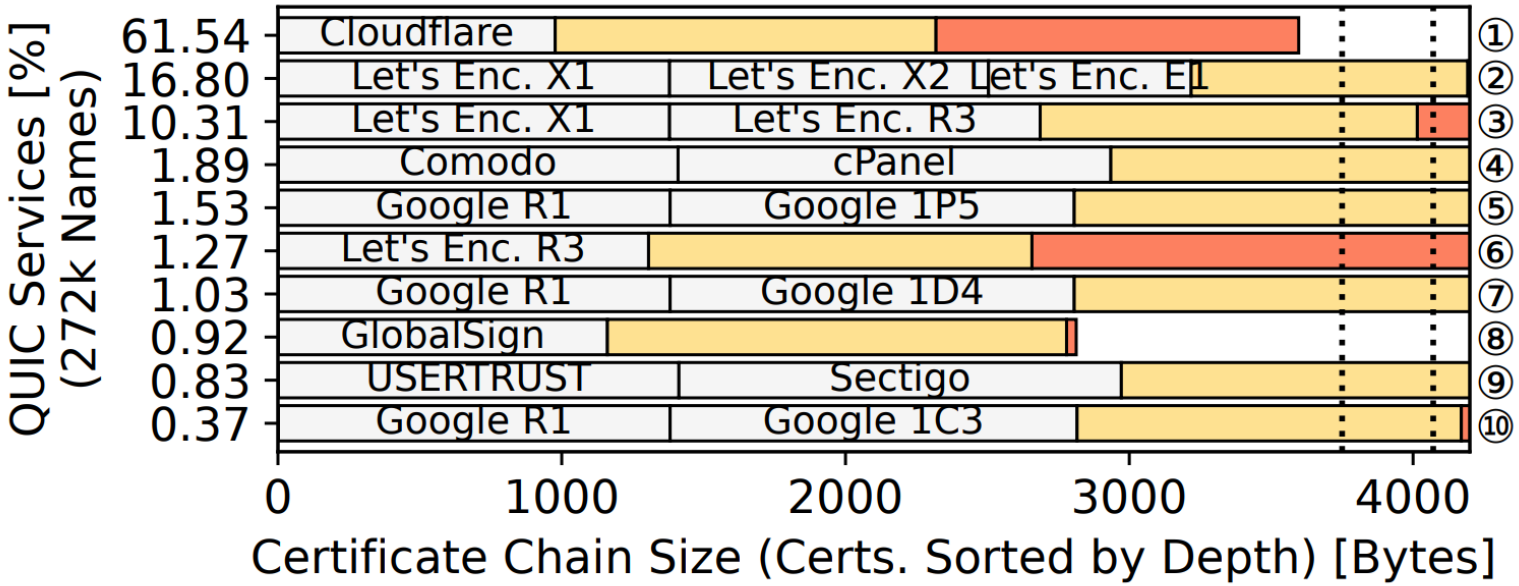
- Cloudflares missing coalescence explains amplification
  - ✓ 96% of the amplification handshakes are completed with Cloudflare
  - ✓ Initial flags are sent separately, leading to two UDP datagrams.



This picture was taken from another article.
https://blog.apnic.net/2023/01/16/on-the-interplay-between-tls-certificates-and-quic-performance/

> Non-leaf TLS certificates size

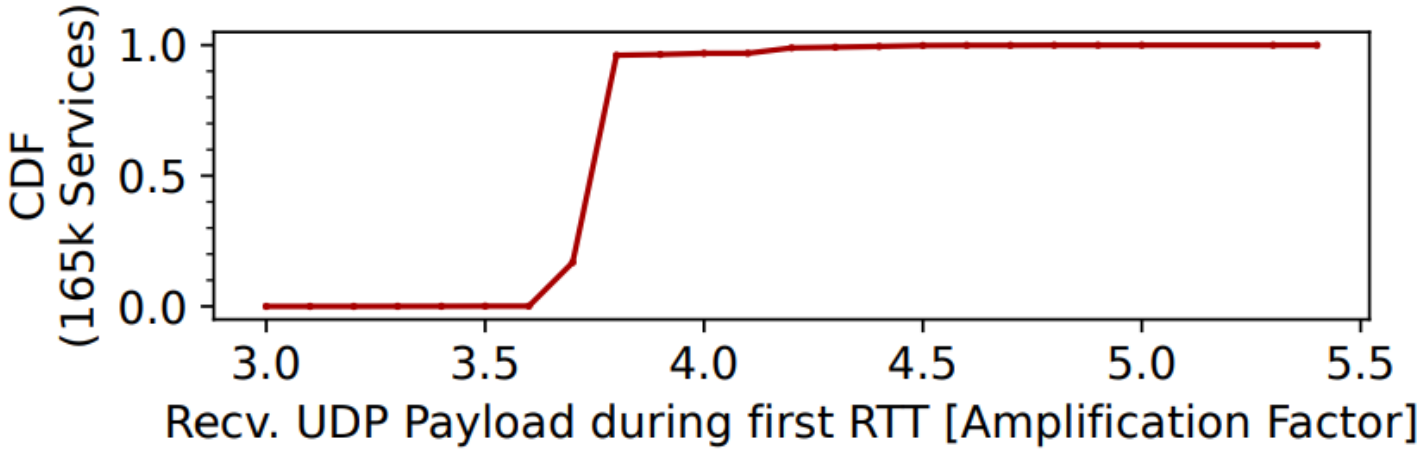
- Large non-leaf TLS certificates impede QUIC performance



Yellow boxes( ): median sizes, orange boxes ( ):the largest leaf certificate
Dotted lines represent the max allowed reply sizes of a server given common client Initial sizes

> Amplification factor

- Although exceeded, it remains relatively small below 6x

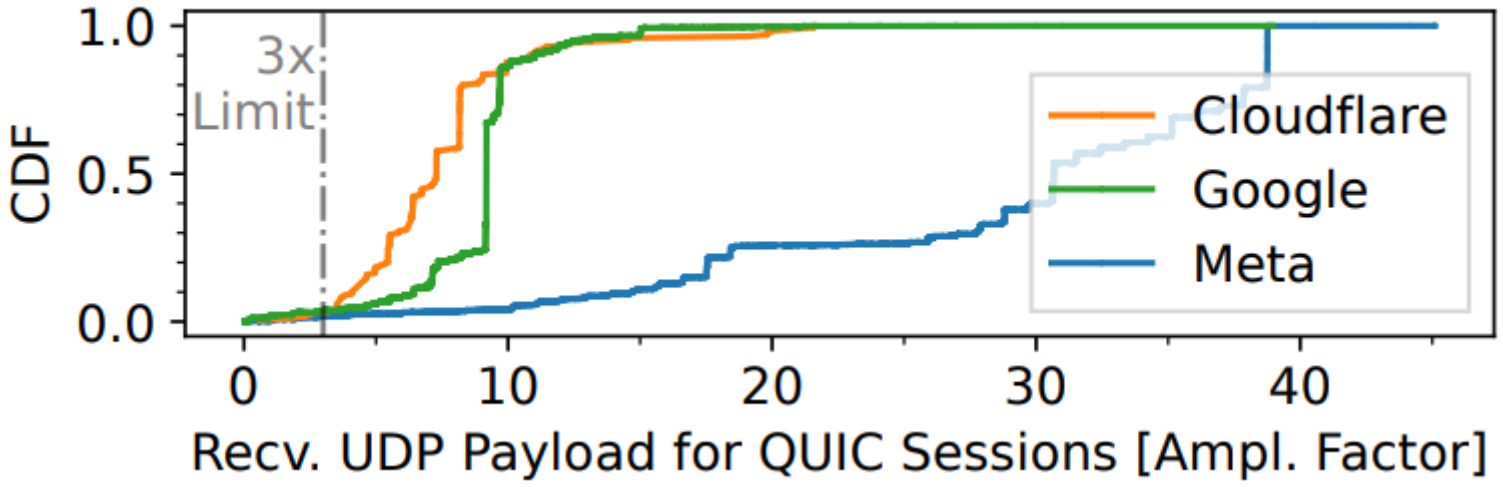


✓Compression helps

- Median compression rate of ≈ 65%, this can keep the size under the amplification limits for 99% of TLS certificate chains

QUIC handshakes in the wild (incomplete)

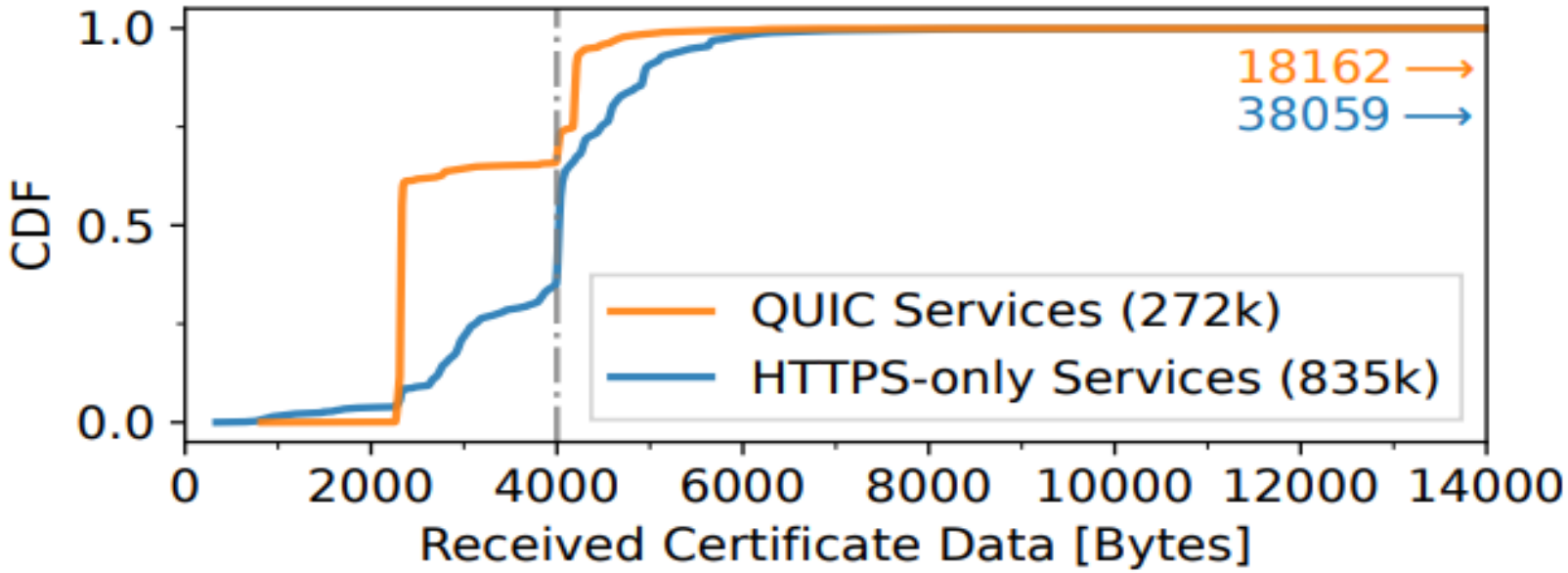
- Examining Amplification Potential
  - ✓ Responses (together with retransmissions) from Meta servers[1)] lead to amplification factors of up to 28×



1) *Instagram and WhatsApp exhibit the highest amplification factors*

TLS certificates vs. QUIC certificates

- Distribution of certificate chain sizes



TLS certificates vs. QUIC certificates

- Relative ratio of crypto algorithms and key lengths in use

Service	Certificate	RSA		ECDSA	
		2048	4096	256	384
QUIC	Non-leaf	15.1%	22.4%	40.4%	22.1%
	Leaf	19.2%	1.4%	78.9%	0.0%
HTTPS-only	Non-leaf	63.3%	32.1%	2.7%	1.6%
	Leaf	81.4%	8.1%	7.8%	1.9%

Guidance

- Certificate compression can compensate for large TLS certificates
- Carefully created TLS certificates and certificate chains can positively influence QUIC protocol performance
- At the QUIC server-side implementation, bytes that result from padding or retransmissions must be included in anti-amplification limit checks

Conclusion

- The authors measured and analyzed the QUIC handshake processes in the wild
- Large portions of QUIC connection setups are either multi-RTT, do not comply with the amplification limit, or both
- This paper gives guidance for stakeholders

Critique

- The authors utilized the quicreach tool, which does not support certificate compression. This can make the QUIC ecosystem look worse than it actually is.

- Certificate compression is makeshift since Post-Quantum Cryptograph has longer key lengths.
  - ✓I will present another approach independent of key length in the next FI.