Post-Quantum

Cryptography Conference

Turning Quantum Threats into Opportunities: Modernizing WebPKI with QUIC and Metrics-Driven Insights

Muralidharan Palanisamy
CSO at AppViewX Inc

KEYFACTOR CRYPTO4A SSL.com ENTRUST HID

October 28 - 30, 2025 - Kuala Lumpur, Malaysia

PKI Consortium

Web PKI to PQC:

Modernizing WebPKI with QUIC Metrics Driven Insights

Presenter: Murali Palanisamy

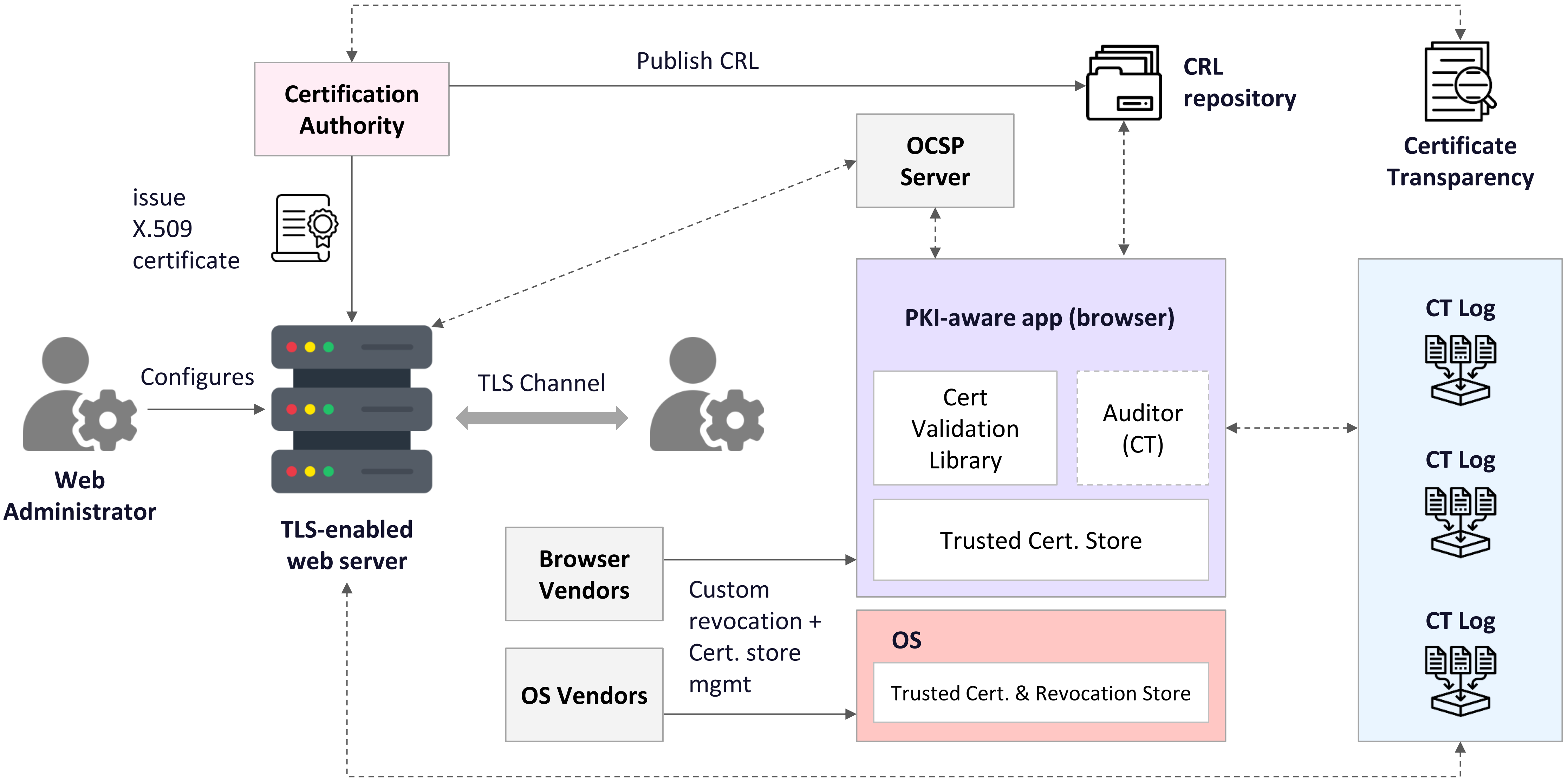
Speaker Introduction

Muralidharan Palanisamy

- Co-Founder and Chief Solutions Officer, AppViewX, driving innovation in enterprise cryptography and automation.
- Focused on PQC readiness, short-lived certificates, and crypto agility for large-scale infrastructures.
- Partners with enterprises and standards bodies to define PQC roadmaps and influence emerging crypto standards.

What is WebPKI?

Web Public Key Infrastructure



State of Enterprise Cryptography

July 2024

75% of F1000 support TLS1.3

4% use DNSSEC for DNS

74.85% use RSA

April 2025

89.14% of F1000 support TLS1.3

4% use DNSSEC for DNS

41%

Support X25519 MLKEM768 Hybrid Key Exchange

> *We're seeing a decisive shift in enterprise security strategies—encryption is now a first-class priority. Organizations are accelerating their adoption of TLS 1.3 and modern cipher suites, not just for compliance, but to future-proof their infrastructure against emerging threats.*

PQC Implementation Changes

Increased Handshake Latency

- PQ key shares and signatures enlarge TLS handshake packets.
- More bytes → higher round-trip time, especially over mobile or high-latency networks.

Higher Hardware & CPU Cost

- PQ operations (KEM, lattice math) demand more CPU cycles.
  Increases load on web servers, CDNs, and mobile clients during handshake bursts.
- Can reduce connection throughput under high concurrency.

Hybrid Key Exchange Complexity

- Combining classical (ECDH) + PQ KEM doubles key data and computation.
- Adds handshake overhead and increases risk of negotiation failure.
- Impacts first-byte latency and TLS session resumption efficiency.

Website Performance Effects

- Slower initial page loads due to longer TLS setup.
- Higher resource usage on edge nodes and browsers.

TLS with PQC

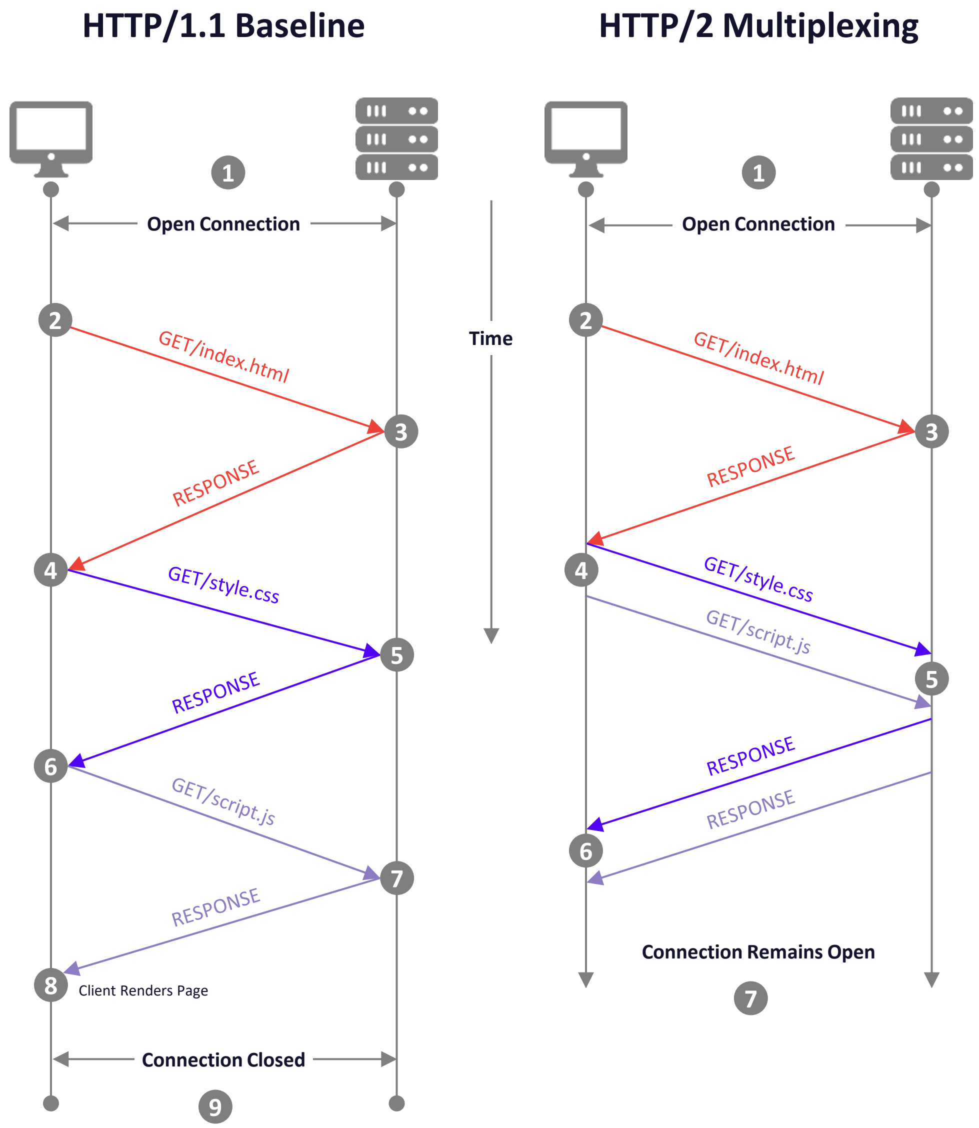
Algorithm	Public Key Size	Ciphertext Size	Notes
X25519 (ECDHE)	32 bytes	32 bytes	Current standard
Kyber-512 (ML-KEM-512)	800 bytes	768 bytes	NIST Level 1
Kyber-768 (ML-KEM-768)	1,184 bytes	1,088 bytes	NIST Level 3 (recommended)
Kyber-1024 (ML-KEM-1024)	1,568 bytes	1,568 bytes	NIST Level 5

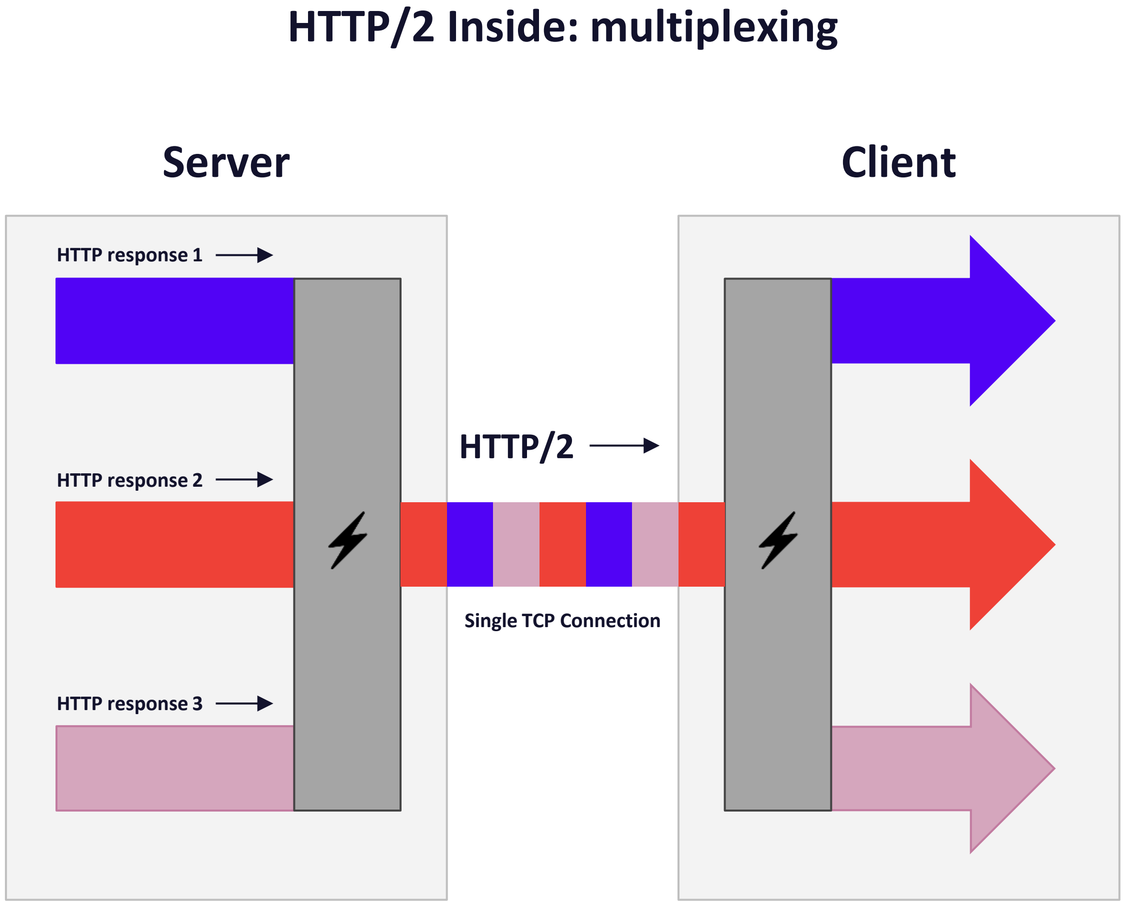
Handshake Mode	Total Extra Bytes vs Today	Notes
ECDHE + ECDSA (today)	~3–5 KB typical handshake	Browser + server with a short cert chain
Kyber-768 hybrid + Dilithium-3	+2.2 KB (KEX) + ~9 KB (certs) ≈ **+11 KB**	Total handshake 14–16 KB
Kyber-1024 hybrid + Dilithium-5	+3.1 KB (KEX) + ~12 KB (certs) ≈ **+15 KB**	Total handshake ~18–20 KB

Source: Cloudflare: A look at the latest post-quantum signature standardization candidates

Web HTTP 1.1/2/3 Evolution

HTTP 1.1 vs HTTP2





HTTP/1.1 and HTTP/2 - Evolution of the Web Transport Layer

HTTP/1.1 (RFC 2616, 1999 → Updated RFC 7230–7235, 2014)

- Based on **TCP**, using **one request per connection**.
- Introduced **persistent connections (keep-alive)** to reuse TCP sessions.
- Still **sequential and blocking** – each request waits for the previous one to complete.
- Suffers from **head-of-line blocking**, connection limits, and high latency for modern web apps.

HTTP/2 (RFC 7540, 2015)

- Still runs over **TCP**, but adds **multiplexing** – multiple streams over one connection.
- Uses **binary framing** instead of text-based messages → more efficient parsing.
- Supports **header compression (HPACK)** → smaller payloads, faster transfers.
- Enables **server push** to proactively send resources.
- Reduces connection overhead, but **still limited by TCP's head-of-line blocking**.

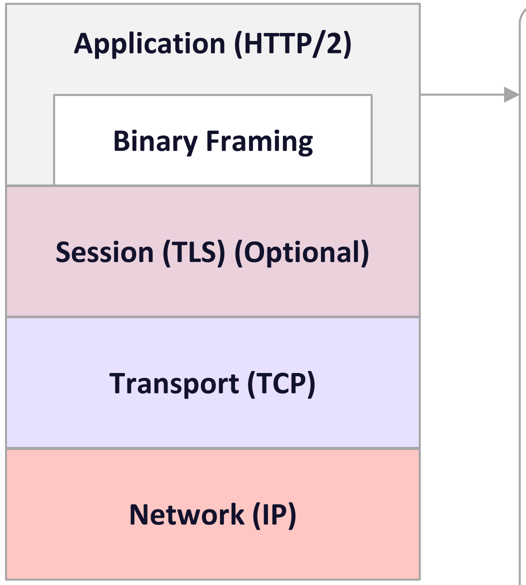
Evolution Summary

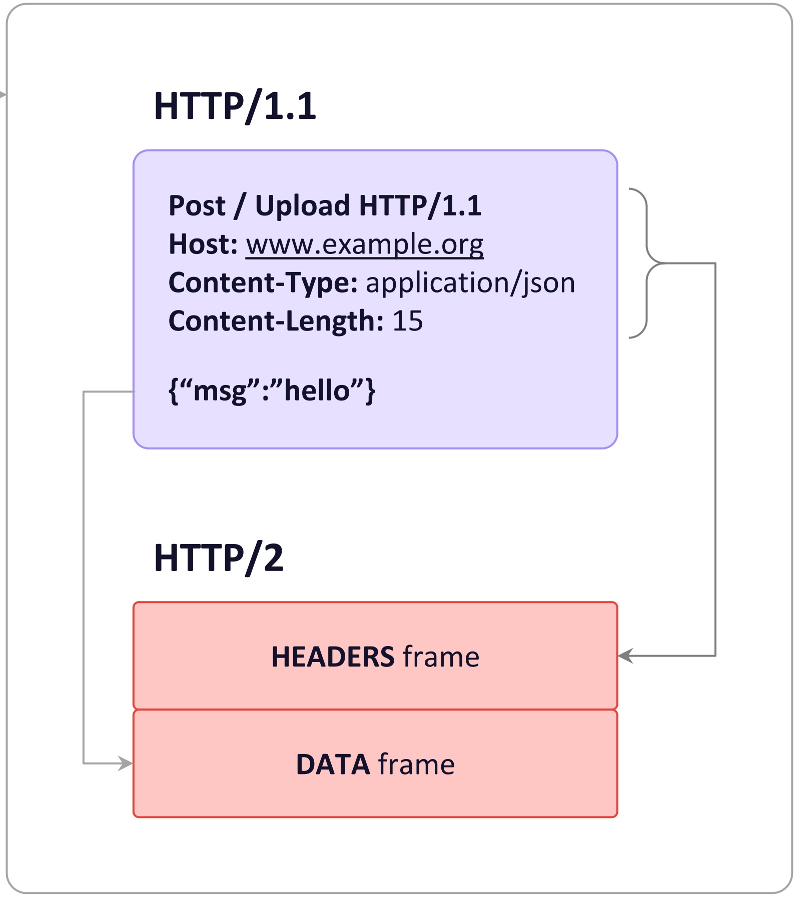
- HTTP/1.1 → Simplicity, wide support, but high latency.
- HTTP/2 → Efficiency and concurrency, yet constrained by TCP behavior.
- Set the stage for **HTTP/3/QUIC**, which replaces TCP to eliminate remaining bottlenecks.

HTTP/2 Summary

HTTP/2

1. **One TCP Connection**
2. **Request → Stream**
   Streams are multiplexed
   Streams are prioritized
3. **Binary Framing Layer**
   Prioritization
   Flow control
   Server push
4. **Header Compression (HPACK)**





HTTP/3 - QUIC Overview

- HTTP/3 is the latest version of the HTTP protocol, built on **QUIC**, a transport protocol running over **UDP** instead of TCP.
- Developed by Google, standardized by IETF to improve latency, reliability, and security. **QUIC:** *RFC 9000 (2021)* **HTTP/3:** *RFC 9114 (2022)*

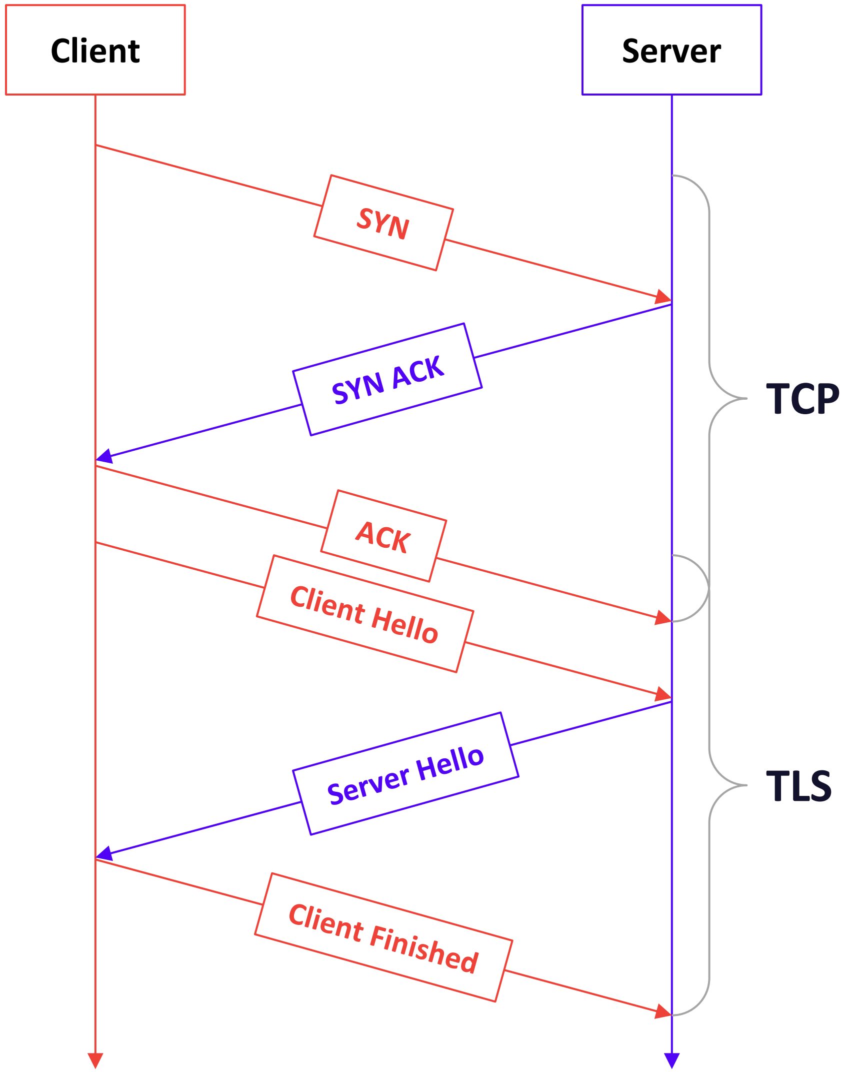
Key Features

- **Built-in Encryption** – QUIC integrates TLS 1.3 directly into the transport layer (no separate handshake).
- **Faster Connection Setup** – 0-RTT and 1-RTT handshakes reduce connection establishment delay.
- **Stream Multiplexing** – Independent streams eliminate TCP head-of-line blocking.
- **Connection Migration** – Seamless transition across networks (e.g., Wi-Fi → 5G) using connection IDs.
- **Better Loss Recovery** – Modern congestion control and packet-level ACKs for smoother throughput.

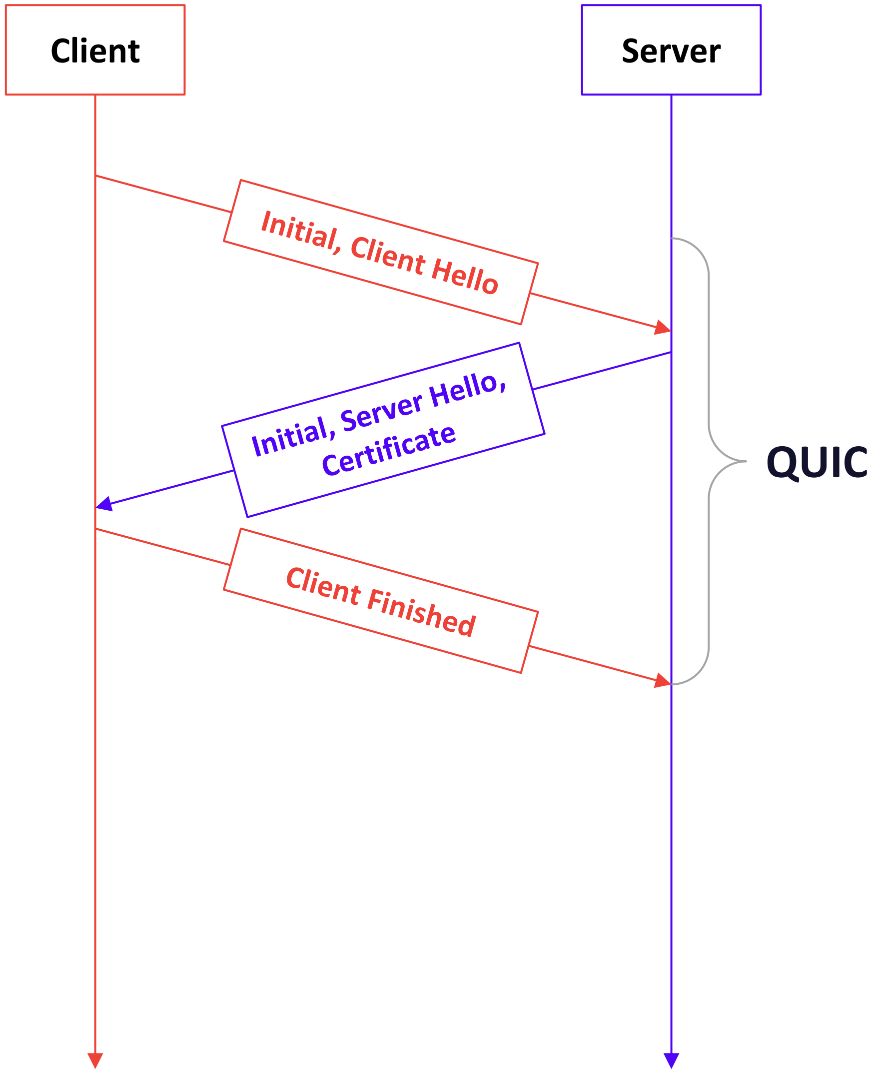
Performance Benefits

- 10–30% faster page loads on high-latency or mobile networks.
- Lower handshake latency and improved reliability under packet loss.
- Ideal for modern web, video streaming, and gaming workloads.

TCP/TLS vs QUIC Connection Flow

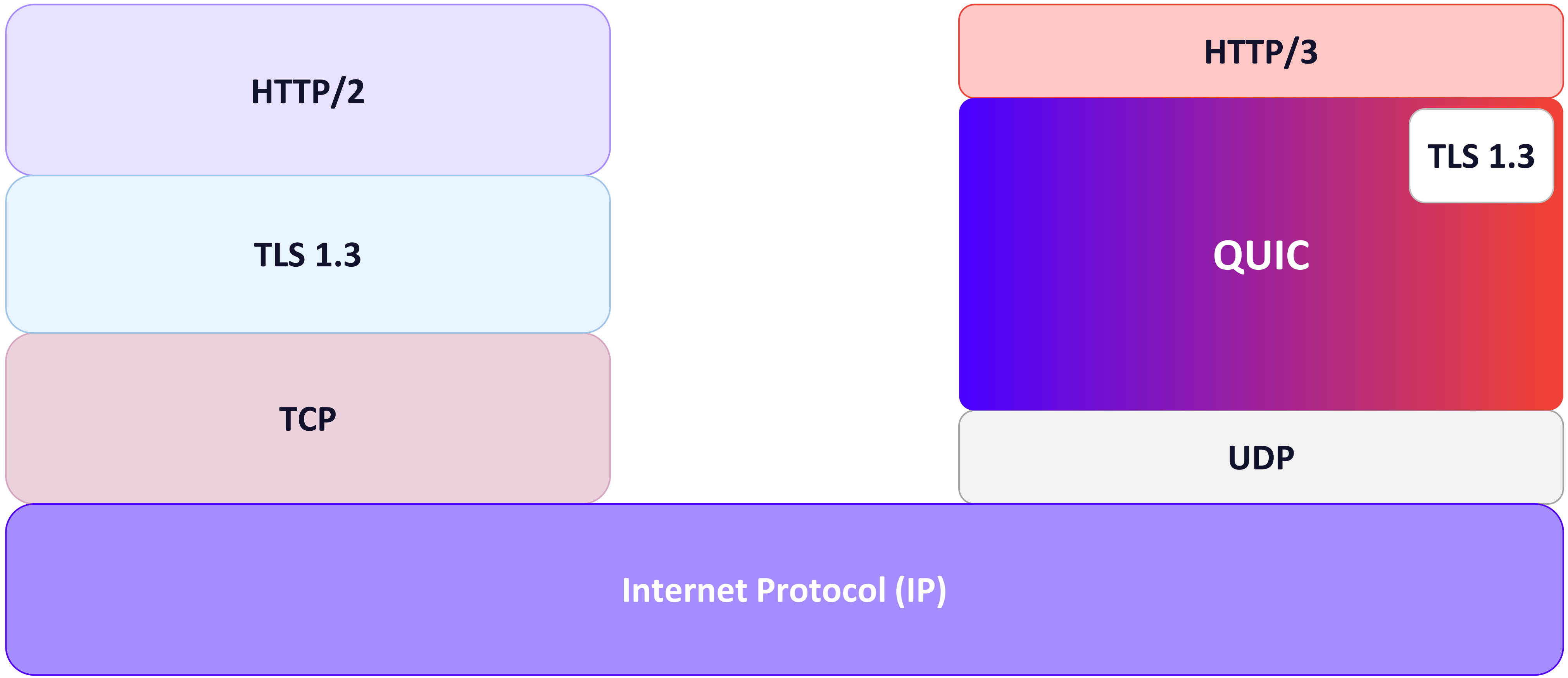


TCP + TLS 1.3

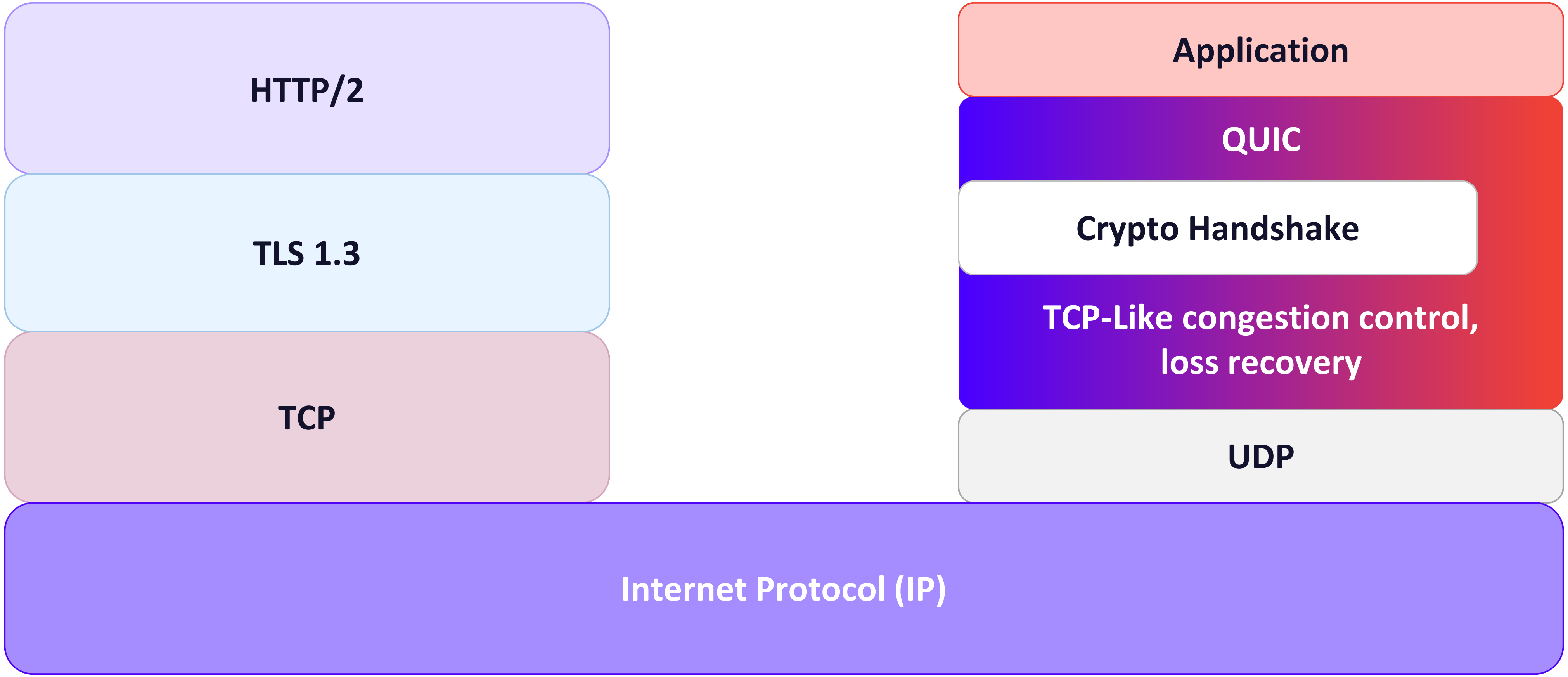


QUIC

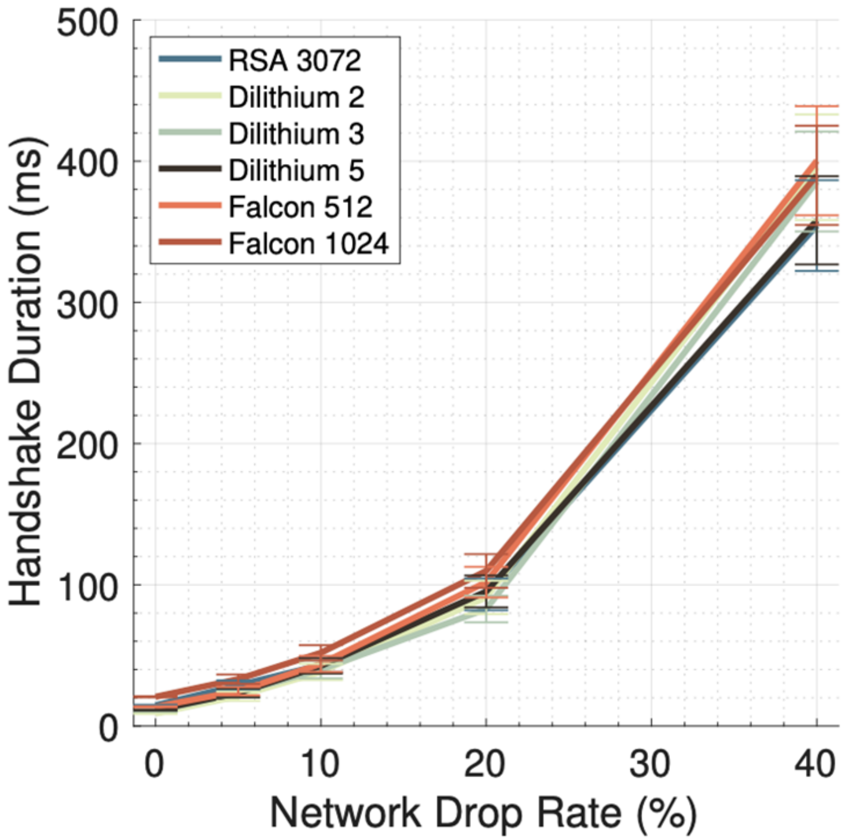
TCP/TLS vs QUIC OSI Stack



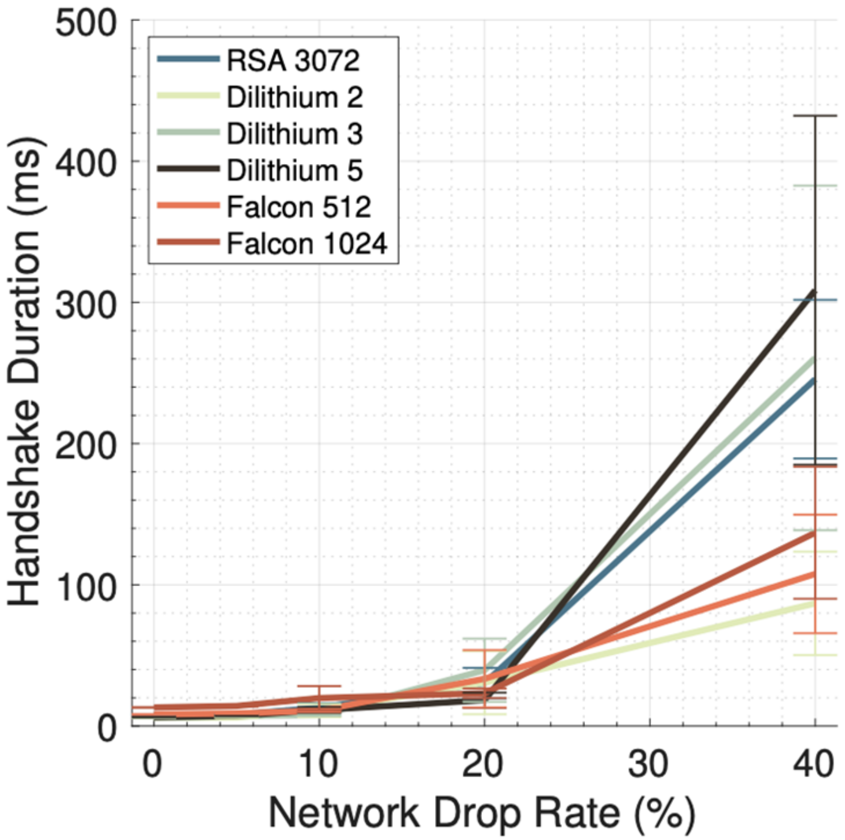
TCP/TLS vs QUIC OSI Stack



Web Performance over TCP/TLS vs QUIC



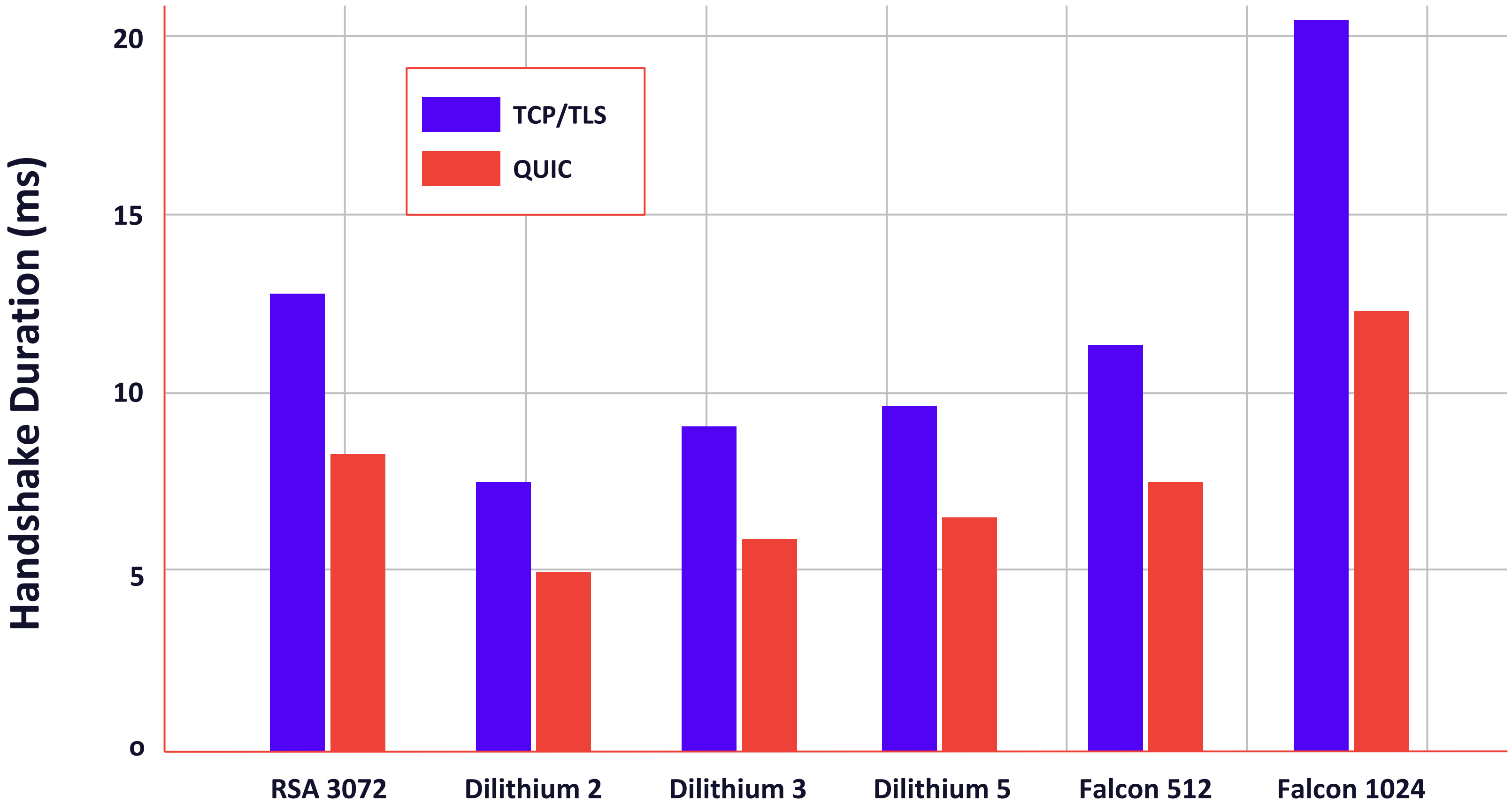
(a) TCP/TLS packet dropping



(b) QUIC packet dropping

Source: University of Colorado, Dept of Computer Science QUIC Protocol with PQC authentication

Handshake Duration over TCP/TLS vs QUIC



Source: University of Colorado, Dept of Computer Science QUIC Protocol with PQC authentication

Conclusion: PQC Migration must be holistic

PQC introduces overhead

- Larger key sizes and hybrid handshakes in **TLS/TCP** increase latency and CPU load.
- Classical web stacks may see slower connection setup and higher handshake cost.

QUIC + HTTP/3 offset these impacts

- Integrated **TLS 1.3**, **0-RTT/1-RTT** handshakes, and **stream multiplexing** minimize round trips.
- Connection migration and improved congestion control deliver real-world latency gains.
- Together, they reclaim or surpass the performance lost to PQC overhead.

Enterprise implication

- Treat **PQC adoption as part of full web-PKI modernization**, not an isolated crypto swap.
- Upgrading to **HTTP/3 / QUIC + PQC-ready TLS** builds a future-proof, quantum-resistant web stack.
- Aligns security and performance goals—moving toward the same architecture already used by **tech giants (Google, Cloudflare, Meta, etc.)**.

The Path Forward

- Plan end-to-end upgrade of **TLS, web servers, CDNs, and PKI**.
- Test PQC + QUIC interoperability early.
- Position your organization for the **next decade of secure, high-performance Internet standards**.

Thank You

appviewx.com

info@appviewx.com