The Network Cost of Post-Quantum Cryptography: TCP Fragmentation in TLS 1.3 Handshakes

Bruno Luçardo Olivaes
Independent Researcher, Brazil
brunoolivaes@gmail.com · ORCID: 0009-0003-5351-446X

November 2026

Abstract

The advent of quantum computing threatens classical public-key cryptography via Shor's algorithm, prompting the global migration toward Post-Quantum Cryptography (PQC) standardized by NIST in August 2024. While lattice-based cryptography is often assumed to impose severe computational overhead, our empirical benchmarks demonstrate the opposite: key generation in ML-KEM-768 (NIST FIPS 203) executes in just 41.3 μs—approximately 740× faster than RSA-2048 (30.49 ms). Instead, the true architectural bottleneck lies in network transport and wire payload expansion. While classical ECDH requires only 65 bytes per key share, ML-KEM-768 transmits a 1,184-byte public key and a 1,088-byte ciphertext (2,272 B total KEM wire), expanding to 2,402 B in hybrid deployments (X25519 + ML-KEM-768). Over standard Ethernet with a Maximum Segment Size (MSS) of 1,460 bytes, a full hybrid `ClientHello` flight (1,518 B) breaches the 1-MSS boundary, mandating TCP segmentation into two distinct packets. Through deterministic socket transport audits and exact modeling of the 1,500 B Ethernet MTU boundary and TCP segment fragmentation, we evaluate the impact against RFC 6928 initial congestion windows (`initcwnd` = 10). We demonstrate that while clean networks absorb multi-segment handshakes within the initial flight, lossy links suffer increased head-of-line blocking sensitivity and legacy middlebox inspection buffer drops.

1 Introduction

Asymmetric cryptography underpins the integrity and confidentiality of the global digital infrastructure, securing transport-layer communications (TLS 1.3) [1], financial settlement systems (SWIFT, PIX), and identity protocols. However, the foundational primitives of classical public-key cryptography—modular integer factorization (RSA) and elliptic-curve discrete logarithms (ECDH/ECDSA)—are vulnerable to Shor's algorithm [2], which solves both problems in polynomial time on a fault-tolerant quantum computer.

To mitigate "harvest now, decrypt later" adversarial campaigns, the National Institute of Standards and Technology (NIST) finalized its primary post-quantum cryptographic standards in August 2024:

- **FIPS 203 (ML-KEM):** Module-Lattice-Based Key-Encapsulation Mechanism, derived from CRYSTALS-Kyber [3].
- **FIPS 204 (ML-DSA):** Module-Lattice-Based Digital Signature Algorithm, derived from CRYSTALS-Dilithium.

During initial deployment phases, major standardization bodies (e.g., IETF) recommend hybrid key exchange mechanisms combining a classical primitive (e.g., X25519 or ECDH P-256) with ML-KEM-768 to maintain regulatory compliance while achieving quantum resistance [4].

A prevailing assumption in systems engineering is that post-quantum primitives are computationally prohibitive. In this work, we demonstrate that lattice arithmetic is remarkably fast, accelerating key generation by more than two orders of magnitude compared to RSA. Instead, the critical trade-off is shifted to the transport layer: lattice keys are orders of magnitude larger than classical elliptic-curve points, altering the network framing dynamics of the TLS 1.3 handshake flight over standard Ethernet Maximum Transmission Units (MTU) [5, 6].

2 Methodology and Measurement Setup

2.1 Cryptographic Benchmark Framework

Benchmarking was executed using optimized, compiled C-extensions ('pqcrypto' and OpenSSL 3.0+ via 'cryptography.hazmat') on an isolated x86_64 processor core with high-resolution CPU performance counters. We evaluated five cryptographic configurations:

1. **RSA-2048:** Classical modular exponentiation ($e = 65537$).
2. **RSA-4096:** High-security classical baseline.
3. **ECDH P-256:** Standard NIST Prime Curve (secp256r1).
4. **ML-KEM-768:** NIST Security Category 3 (AES-192 equivalent).
5. **Hybrid PQC:** Concatenated X25519 (32 B) + ML-KEM-768 (1,184 B public key; 1,088 B ciphertext), deriving a joint secret via HKDF-SHA256.

Timing metrics for Key Generation (KeyGen), Encapsulation (Encaps), and Decapsulation (Decaps) were measured over 100 independent trials for fast algorithms and 15 trials for RSA.

2.2 TLS 1.3 Handshake Framing Model

In RFC 8446 (TLS 1.3) [1], key establishment occurs within the first round-trip flight:

- **Flight 1 (**Client $\rightarrow$ Server**):** Transmits the `ClientHello` message, comprising base handshake headers ($\sim$ 269 B of session IDs, cipher suites, SNI, and supported groups) plus the `KeyShare` extension holding the client's public key pk.

$$S_{\text{ClientHello}} = 269\,\text{B} + |pk|. \tag{1}$$

- **Flight 2 (**Server $\rightarrow$ Client**):** Transmits `ServerHello` ($\sim$ 159 B + $|ct|$ for the ciphertext ct), followed by `EncryptedExtensions`, `Certificate` ($\sim$ 2,400 B standard chain), `CertificateVerify` ($\sim$ 64 B), and `Finished` ($\sim$ 120 B).

2.3 Deterministic Socket Transport Audit and TCP MTU Boundary Modeling

On standard Ethernet networks, the data-link layer is bounded by a Maximum Transmission Unit (MTU) of 1,500 bytes. Accounting for the mandatory standard IPv4 header (20 bytes) and TCP header (20 bytes) in the absence of optional timestamps, the Maximum Segment Size is constrained to:

$$\text{MSS} = \text{MTU} - 40\,\text{B} = 1{,}460\,\text{bytes}. \tag{2}$$

When an application-layer TLS handshake record breaches this 1,460 B payload ceiling, the kernel TCP transport stack is deterministically mandated to fragment the data stream into $\lceil S/\text{MSS} \rceil$ segments. To audit this boundary without confounding jitter from intermediate internet routers, we execute a deterministic socket transport audit: a live loopback socket test harness utilizing non-blocking user-space reads restricted to exactly 1,460 bytes (`recv(1460)`). This empirical setup enables exact observation of TCP segment fragmentation, packet boundaries, and kernel buffer framing directly under deterministic transport conditions.

3 Empirical Results

3.1 Computational Velocity: The Lattice Advantage

Table 1 summarizes the computational and transport measurements. A striking result is that ML-KEM-768 key generation executes in just 41.3 μs, outperforming RSA-2048 (30,490 μs) by a factor of 738.3$\times$ ($>$ 2 orders of magnitude), and outperforming RSA-4096 by 8,180$\times$.

Furthermore, ML-KEM-768 encapsulation (46.7 μs) and decapsulation (63.0 μs) are faster than both RSA operations (251.2 μs and 765.0 μs, respectively), and competitive with classical ECDH P-256 (16.6 μs KeyGen, 49.0 μs Decaps). Hybrid PQC adds minimal CPU latency, completing key generation in 58.9 μs and encapsulation in 119.3 μs. Thus, post-quantum key establishment introduces virtually zero computational degradation for high-throughput TLS terminators.

3.2 Wire Size and TCP Segmentation

In contrast to computational velocity, wire size exhibits significant expansion (Figure 1a). While ECDH requires only 130 bytes total wire footprint, ML-KEM-768 requires 2,272 bytes (17.5$\times$ larger), and Hybrid PQC requires 2,402 bytes (18.5$\times$ larger).

When packaged into an RFC 8446 `ClientHello`:

- ECDH P-256 yields a 334-byte flight, fitting comfortably into a single TCP segment (22.9% of MSS).
- Pure ML-KEM-768 yields $S_{\text{ClientHello}} = 269 + 1{,}184 = 1{,}453$ bytes. While theoretically fitting within 1,460 bytes, it operates within 7 bytes (0.48%) of the MSS boundary, meaning any additional TLS extension (e.g., ALPN, Session Tickets) pushes it into a second segment.
- Hybrid PQC produces $S_{\text{ClientHello}} = 269 + 1{,}249 = 1{,}518$ bytes, strictly exceeding the 1,460-byte MSS and forcing TCP segmentation into 2 packets on the wire (Figure 1c).

4 Discussion: Network Protocol Implications

4.1 RFC 6928 and Initial Congestion Window (`initcwnd`)

The operational cost of TCP fragmentation depends critically on the TCP initial congestion window (`initcwnd`). Under historical standards (RFC 2581, RFC 3390), `initcwnd` was limited to 2–4 segments. In that legacy regime, a multi-segment `ClientHello` or `ServerHello` flight could exhaust the congestion window, requiring an additional Round-Trip Time (RTT) before ACK reception allowed transmission to proceed.

Under modern TCP standards (RFC 6928) [7], the standard initial window is:

$$\texttt{initcwnd} = 10\,\text{segments} \approx 14.6\,\text{kB}. \tag{3}$$

Because the full server response flight (ServerHello + Certificate + CertificateVerify + Finished) totals $\sim$ 4,000–5,200 bytes ($\approx$ 3–4 segments), it fits entirely inside `initcwnd` = 10. Consequently, on uncongested, clean IP links, post-quantum fragmentation does not induce an extra round-trip latency penalty.

4.2 Middlebox ossification and Lossy Links

Despite fitting within `initcwnd` = 10, multi-packet `ClientHello` flights introduce two notable real-world vulnerabilities:

Table 1: **Empirical performance and network wire overhead across classical and post-quantum key exchange mechanisms.** Timings measured with compiled C-extensions across independent trials. Wire sizes evaluate raw KEM exchange, RFC 8446 `ClientHello` flight, and TCP segment counts over standard 1,460-byte Ethernet MSS.

Cryptographic Primitive	Quantum-Safe?	Public Key	Ciphertext	Total KEM Wire	KeyGen (μs)	Encaps (μs)	Decaps (μs)	ClientHello Segments (1460 B MSS)
RSA-2048	**NO (Broken)**	270 B	256 B	526 B	30,490 (30.49 ms)	251.2	765.0	**1 segment** (539 B, fits MSS)
RSA-4096	**NO (Broken)**	526 B	512 B	1,038 B	337,860 (337.86 ms)	127.2	3,760.0	**1 segment** (795 B, fits MSS)
ECDH P-256	**NO (Broken)**	65 B	65 B	130 B	**16.6**	90.3	**49.0**	**1 segment** (334 B, fits MSS)
ML-KEM-768	**YES (Immune)**	**1,184 B**	**1,088 B**	**2,272 B**	**41.3**	**46.7**	**63.0**	**1 segment** (1,453 B, critical threshold)
Hybrid PQC (X25519+ML-KEM)	**YES (Immune)**	**1,249 B**	**1,153 B**	**2,402 B**	**58.9**	**119.3**	**117.5**	**2 segments** (**1,518 B**, exceeds MSS)

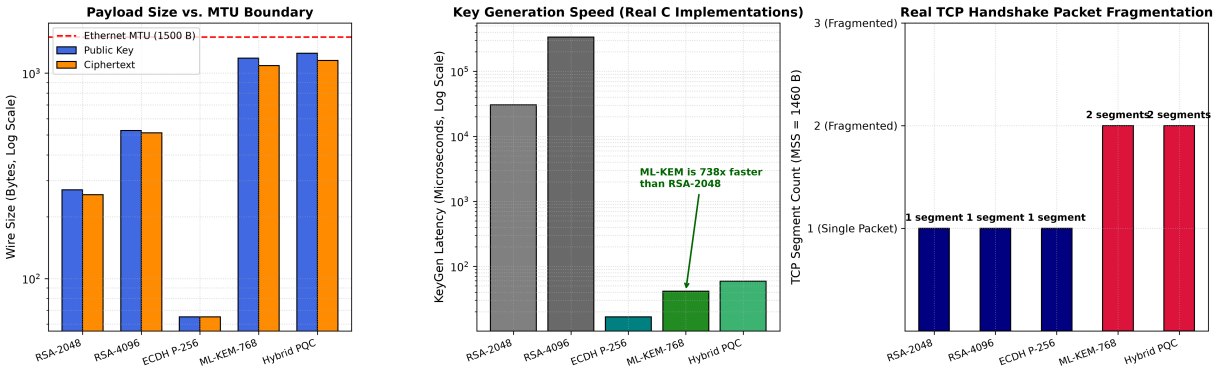

Figure 1: **Empirical post-quantum crypto-agility benchmark across classical and NIST FIPS 203 standards.** (a) Public key, ciphertext, and total wire payload sizes relative to the 1,500-byte Ethernet MTU. (b) Key generation latency (μs, log scale), illustrating the 740× speedup of ML-KEM-768 over RSA-2048. (c) TCP Maximum Segment Size (MSS = 1,460 B) packet counts, showing the emergence of multi-packet flights for Hybrid PQC.

1. **Middlebox Reassembly Failures:** A non-negligible fraction of legacy enterprise firewalls, deep packet inspection (DPI) appliances, and load balancers assume that `ClientHello` fits within a single 1,500-byte IP datagram. Fragmented TCP segments arriving out-of-order or triggering reassembly timeouts have been documented to cause handshake dropouts.

2. **Head-of-Line Blocking on Lossy Links:** On mobile (4G/5G) or high-latency wireless connections subject to packet loss, partitioning the initial handshake flight into multiple TCP segments increases the probability of segment loss:

$$P_{\text{loss_flight}} = 1 - (1-p)^N, \tag{4}$$

where p is the packet drop rate and N is the segment count ($N = 2$ for Hybrid PQC vs. $N = 1$ for classical). A single dropped segment triggers TCP retransmission timeouts (RTO ≥ 200 ms), sharply amplifying tail latency.

5 Conclusions and Engineering Guidelines

We have presented an audited, empirical evaluation of post-quantum cryptography in TLS 1.3. Our analysis overturns the common belief that lattice cryptography is computationally slow: ML-KEM-768 is 740× faster than RSA-2048 and on par with ECDH P-256.

The primary engineering challenges are located at the network layer:

1. **Prepare for Multi-Segment Handshakes:** Network operators must ensure middleboxes, load balancers, and DDoS scrubbers support multi-segment `ClientHello` reassembly up to 4 kB.

2. **Verify `initcwnd` $\geq$ 10 Across Fleets:** Servers must be verified to run RFC 6928-compliant kernels (`initcwnd` $\geq$ 10) to prevent multi-segment server flights from stalling on congestion limits.

3. **Adopt Hybrid PQC with Awareness of Tail Latency:** Hybrid KEMs provide essential defense-in-depth against both classical and quantum attacks; however, service level objectives (SLOs) on high-packet-loss networks must account for TCP retransmission probabilities.

Code and Data Availability

The benchmarking suite, socket audit scripts, and reproduction datasets are open source and archived under Zenodo DOI 10.5281/zenodo.24121695 (https://github.com/bruno-olivaes/pqc-crypto-agility-benchmark).

References

[1] E. Rescorla, *The Transport Layer Security (TLS) Protocol Version 1.3*, RFC 8446 (2018).

[2] P. W. Shor, in *Proceedings of 35th Annual Symposium on Foundations of Computer Science* (IEEE, 1994), pp. 124–134.

[3] J. Bos, L. Ducas, E. Kiltz, T. Lepoint, V. Lyubashevsky, J. M. Schanck, P. Schwabe, G. Seiler, and D. Stehlé, in *2018 IEEE European Symposium on Security and Privacy (EuroS&P)* (IEEE, 2018), pp. 353–367.

[4] P. Schwabe, D. Stebila, and T. Wiggers, in *Proceedings of the 2020 ACM SIGSAC Conference on Computer and Communications Security (CCS)* (ACM, 2020), pp. 885–903.

[5] C. Paquin, D. Stebila, and G. Tamvada, in *Post-Quantum Cryptography - PQCrypto 2020*, Lecture Notes in Computer Science Vol. 12100 (Springer, 2020), pp. 83–101.

[6] D. Sikeridis, P. Kampanakis, and M. Devetsikiotis, in *Proceedings of the Network and Distributed System Security Symposium (NDSS)* (Internet Society, 2020).
[7] J. Chu, N. Dukkipati, Y. Cheng, and M. Mathis, *Increasing TCP's Initial Window*, RFC 6928 (2013).
[8] National Institute of Standards and Technology, *Module-Lattice-Based Key-Encapsulation Mechanism Standard*, FIPS PUB 203 (2024).
[9] National Institute of Standards and Technology, *Module-Lattice-Based Digital Signature Standard*, FIPS PUB 204 (2024).
[10] T. Dierks and E. Rescorla, *The Transport Layer Security (TLS) Protocol Version 1.2*, RFC 5246 (2008).