Frontiers in Computing and Intelligent Systems
ISSN: 2832-6024 | Vol. 18, No. 2, 2026

Transport Thresholds and Network-Impairment Amplification Mechanisms in TLS 1.3 Post-Quantum Key Exchange

-- A Cross-Layer Experimental Study of ML-KEM, HQC, and Hybrid Schemes

Jiayi Zhao *

School of Information and Control Engineering, Qingdao University of Technology, Qingdao, Shandong, 266520, China
*** Corresponding author Email:** 3198827677@qq.com

Abstract: Post-quantum key-encapsulation mechanisms substantially enlarge TLS 1.3 handshake material, but their end-to-end impact depends on the combined effects of TLS encapsulation, TCP segmentation, and network conditions. Using OpenSSL 3.0.21, liboqs 0.16.0, and oqs-provider 0.12.0-dev, this study conducts cross-layer experiments on X25519, ML-KEM-768, two implementation-specific HQC groups, hqc1 and hqc3 (not NIST security levels), and the corresponding hybrid schemes. It measures ClientHello, ServerHello, TLS record, and TCP carriage structures; locates segmentation-transition boundaries through adjacent-MTU scans; and completes 180,000 formal connection-establishment measurements plus 6,400 targeted review measurements in separate RTT, per-direction bandwidth, and independent per-direction loss experiments. The results show that ML-KEM is generally close to X25519 when there is no packet loss or material bandwidth bottleneck. At 1 Mbps, the latency increase of each scheme is broadly consistent with serialization estimates derived from bidirectional TCP payload. Packet loss first amplifies P95 and P99; under 10% independent loss in each direction, 30 of 18,000 measurements did not complete and verify the connection within 10 s. The higher median latency and long tail of HQC-L3 recur in both the formal batches and targeted review, whereas the single-batch anomaly of X25519-HQC-L1 does not. The study establishes a traceable evidence chain from cryptographic material, TLS encapsulation, and TCP segmentation to network impairment and connection-establishment performance, providing experimental support for MTU/MSS headroom assessment and post-quantum TLS deployment.

Keywords: TLS 1.3; Post-quantum Key Exchange; ML-KEM; HQC; TCP Segmentation and Network Impairment.

1. Introduction

Post-quantum migration is time-critical for communications that require long-term confidentiality. In an ideal quantum-computing model, Shor's algorithm can efficiently solve integer factorization and discrete logarithm problems, fundamentally threatening the long-term security of RSA, finite-field Diffie-Hellman, and elliptic-curve cryptography [1]. An adversary may also collect ciphertext today and decrypt it retrospectively once sufficiently capable quantum computers become available.

Over an established TCP connection, TLS 1.3 normally completes a full handshake in 1 RTT [2]. NIST published FIPS 203 in 2024, formally standardizing ML-KEM [3], and selected the code-based HQC for standardization in 2025 as a complement to lattice-based ML-KEM [4]. The IETF subsequently published the hybrid key-exchange framework in RFC 9954 and continued developing concrete TLS mappings for ML-KEM and X25519MLKEM768 [5]-[7].

The engineering difficulty of post-quantum migration is that KEM public keys and ciphertexts are typically much larger than X25519 shares. The added material enlarges ClientHello and ServerHello, bringing the handshake close to or across TCP segmentation boundaries. On high-RTT, low-bandwidth, or lossy links, the additional bytes may be further amplified by serialization delay, transport recovery, and connection-failure probability. Reporting cryptographic operation speed alone is therefore insufficient to predict real TLS connection-establishment performance.

This study focuses on the cross-layer mechanisms that produce performance differences rather than on a single latency ranking. Its main contributions are as follows: (1) a measurement decomposition that links KEM shares to handshake messages, TLS records, and TCP carriage, together with adjacent-integer MTU scans that locate ClientHello and ServerHello segment-count transitions; (2) a controlled one-factor-at-a-time evaluation of RTT, per-direction bandwidth, and independent per-direction loss across six conventional, pure-PQC, and hybrid groups; (3) joint treatment of the median, nearest-rank P95/P99, and incomplete connections so that distribution center, tail behavior, and completion risk are not conflated; and (4) an evidence-consistency procedure based on three formal batches, within-batch effect comparisons, drift diagnostics, and targeted reviews, preventing isolated batch anomalies from being promoted to algorithmic conclusions.

2. Background and Related Work

2.1. KEM and Hybrid Key Exchange in TLS 1.3

TLS 1.3 carries ephemeral key-exchange material in the key_share extension [2]. For a KEM, the client sends an encapsulation public key and the server returns a ciphertext; the resulting shared secret then enters the existing TLS 1.3 key schedule. RFC 9954 represents a hybrid group by direct concatenation: component shares and shared secrets are joined in a fixed order without adding another round trip [5].

As of July 2026, the pure ML-KEM mapping is specified in draft-ietf-tls-mlkem-09, while hybrid groups such as X25519MLKEM768 are specified in draft-ietf-tls-ecdhe-mlkem-05; both are cited as Internet-Drafts marked "Work in Progress" [6], [7].

2.2. Transport Characteristics of ML-KEM and HQC

ML-KEM-768 is based on module-lattice hardness assumptions. FIPS 203 specifies an encapsulation public key of 1184 B and a ciphertext of 1088 B [3]. In this experiment, the client and server shares of X25519MLKEM768 add 32 B to the corresponding ML-KEM values, matching the 1216 B and 1120 B specified in the draft [7]. HQC is based on a syndrome-decoding problem related to quasi-cyclic codes and uses larger material. **Naming convention—***Throughout this paper, HQC-L1 and HQC-L3 are presentation aliases for the implementation groups hqc1 and hqc3 exposed by the evaluated OQS toolchain. The suffixes L1/L3 do not denote NIST security categories and do not represent the final standardized HQC parameter sets.*

2.3. Performance Studies and Research Gap

The Open Quantum Safe project provides liboqs and TLS integration tools and is a common foundation for post-quantum protocol prototyping [8]. Paquin et al. established a PQC benchmarking framework for TLS and showed that cryptographic computation and network transmission must be evaluated jointly [9]. Henrich et al. further examined bandwidth, latency, and packet loss, finding that large-material schemes such as HQC, BIKE, and FrodoKEM are more sensitive to network conditions [10]. Zheng et al. analyzed ML-KEM performance in TLS 1.3 from the perspective of implementation optimization [11].

Later work broadened the evaluation scope. Sosnowski et al. compared the system overhead of post-quantum key exchange and signatures through black-box and white-box measurements [12]. Kampanakis and Childs-Klein extended the metric to connection completion including application data and studied jitter, bandwidth, and packet loss [13]. Henrich et al. added post-quantum signatures, hybrid schemes, certificate chains, and physical-server configurations [14]. Souvatzidaki and Limniotis used a newer OQS toolchain to compare the handshake performance of ML-KEM, HQC, and BIKE under different loss rates [15]. Rios and Montenegro performed a multi-layer bottleneck analysis from cryptographic primitives and TLS handshakes to VPN applications, showing that isolated primitive benchmarks cannot directly represent protocol- and application-layer behavior [16].

Accordingly, the contribution is not the mere inclusion of network impairment or layered measurements. It lies in comparing standardized ML-KEM with current HQC implementations in one toolchain; locating ClientHello and ServerHello MTU/MSS segment-count transitions through adjacent-integer scans; reporting distribution center, tail latency, and incomplete connections as distinct outcomes; and constraining conclusions through three formal batches, within-batch differences, drift diagnostics, and targeted review measurements.

3. Research Questions and Experimental Method

3.1. Research Questions

RQ1: How do the six schemes differ in key-exchange material, TLS handshake messages, and TCP carriage state?

RQ2: How do RTT, bandwidth, and independent random loss in each direction affect the distribution center, tail latency, and connection-completion rate?

RQ3: Is the additional cost of each hybrid scheme relative to its corresponding pure-PQC scheme stable and practically meaningful?

RQ4: Are the observed differences reproducible across batches, and do anomalies persist in targeted review measurements?

3.2. Experimental Platform and Scheme Groups

The experimental platform uses OpenSSL 3.0.21, liboqs 0.16.0 (commit 5a1a854b), and oqs-provider 0.12.0-dev (commit 1670a8a) [17]-[19]. The client runs in the host network namespace and the server in pqc-v3-ns. They communicate through the v3c0/v3s0 veth pair, with the server at 10.203.0.2:44333. tc netem at both ends independently controls RTT, per-direction bandwidth, and independent random loss in each direction [20]. tcpdump captures PCAPNG on v3c0, and SSLKEYLOGFILE is used to decode TLS records. Figure 1 shows the overall experimental topology and the chain for data acquisition, statistical analysis, review, and archival.

The timer records its start immediately before creating the client process and records completion when CONNECTION ESTABLISHED is first detected. A successful sample must also contain both the TLSv1.3 protocol marker and Verification: OK. This metric is defined as client-observed connection-establishment latency, hereafter connection-establishment latency. It includes client-process startup, TCP connection establishment, and the TLS handshake, but excludes TLS shutdown. The per-attempt timeout is 30 s for the RTT experiment and 10 s for the bandwidth and loss experiments. A sample exceeding the threshold or lacking complete connection and verification markers is counted as an incomplete connection.

The experiments run in a QEMU virtual machine with Ubuntu 24.04.4 LTS as the guest system (Linux 7.0.0-28-generic, aarch64), 4 vCPUs, and 5.8 GiB of visible memory. All three formal experiment classes were completed in this virtual machine without changes to processor or memory allocation. No other high-load task was run, and the VM was not paused, restored from a snapshot, suspended with the host, or migrated. Linux TCP used CUBIC congestion control and the default fq_codel queue discipline; TCP timestamps were enabled and MTU probing was disabled.

The baseline captures, MTU scans, three formal experiment classes, and targeted reviews all used the same server certificate and private key, with the client using that certificate as the trust anchor. The certificate was a self-signed RSA-2048 certificate with a single-certificate chain, CN=localhost, and sha256WithRSAEncryption as the signature algorithm; its validity period covered all formal experiments. Authentication material and its fixed overhead therefore remained constant across the six schemes and all network conditions.

Table 1. Experimental Groups and Handshake Material

Group	Type	Client share (B)	Server share (B)
X25519	Conventional	32	32
ML-KEM-768	Pure PQC	1184	1088
X25519MLKEM768	Hybrid	1216	1120
HQC-L1 (hqc1)	Pure PQC	2241	4433
HQC-L3 (hqc3)	Pure PQC	4514	8978
X25519-HQC-L1	Hybrid	2273	4465

Note—*HQC-L1/HQC-L3 are aliases for the OQS groups hqc1/hqc3; L1/L3 are implementation labels, not NIST security categories. X25519-HQC-L1 is the x25519_hqc1 hybrid group.*

3.3. Cross-Layer Message Measurement and Threshold Scanning

Complete handshakes for all six groups were first captured at MTU 1500. The extracted fields included ClientHello handshake length, ServerHello handshake length, TLS record length, TCP payload, and per-direction TCP data-segment count. Application-layer content was then held constant while the MTU was searched downward from a higher value, with adjacent integers rechecked around each state change. The upper boundary is defined as the minimum MTU that still preserves the original segment count; the lower boundary is the MTU 1 B below it, where one additional TCP segment appears. This terminology describes TCP segmentation and the effective-MSS threshold rather than broadly labeling the phenomenon as IP fragmentation.

3.4. Network-Impairment Experiments and Statistical Definitions

RTT, bandwidth, and packet loss were evaluated with a controlled one-factor-at-a-time design. The RTT experiment applied neither bandwidth restriction nor loss; the bandwidth experiment fixed RTT at 70 ms and injected no loss; and the loss experiment fixed RTT at 70 ms without bandwidth restriction. Bandwidth limits were applied separately in the client and server directions, and loss was injected independently in both directions. Thus, the 1%, 3%, 5%, and 10% values in this paper mean independent random loss in each direction, not one aggregate loss probability for the round-trip path. Each formal case combines three batches, each containing 100 warm-up attempts and 1000 formal measurements. The scripts used a fixed deterministic condition and scheme order within each batch; this choice is treated explicitly as a threat to causal attribution. Anomalous bandwidth and loss cases were reviewed separately, and review data did not overwrite the formal batches.

Table 2. Formal Experimental Design

Factor	Conditions	Cases	Formal measurements	Warm-up	Targeted review
RTT	35/70/200 ms	18	54,000	5,400	-
Bandwidth	1000/100/1 Mbps (each direction)	18	54,000	5,400	2,400
Packet loss	1/3/5/10% (independent in each direction)	24	72,000	7,200	4,000

The primary latency metrics are the median and nearest-rank P95 and P99. For n successful latency samples sorted in ascending order, the p quantile is the observation at rank k, where k is the smallest integer greater than or equal to pn. Incomplete connections are excluded from latency quantiles for successful samples but included in the denominator of the completion rate and in the incomplete count. Tail latency and connection completion are therefore reported separately. The pooled distribution across three batches is used for the main results, batch medians are used for drift diagnosis, and targeted review is used only to determine whether an anomaly recurs.

For key comparisons, median differences are first calculated within the same batch and network condition and are then cross-checked against the pooled three-batch distribution and targeted review results. Because attempts are nested within execution batches and may share short-timescale environmental conditions, the analysis does not treat all 3000 pooled attempts as exchangeable independent replicates for null-hypothesis testing. Inference instead relies on effect magnitude, within-batch direction, pooled quantiles, and recurrence in targeted review. A result is described as a stable effect only when these sources of evidence are consistent; differences whose direction changes across batches or does not recur in review are reported only as drift or single-batch anomalies.

3.5. Data Archival and Reproducibility

For every attempt, the formal performance experiments save connection-establishment latency, client exit status, and success markers, while also archiving per-attempt TSV files, case summaries, batch summaries, run information, control scripts, and log indexes. Anomaly reviews are stored separately and do not overwrite the original batches. All figures and tables in the paper are regenerated from archived data rather than manually entered values. Complete checkpoint names, SHA-256 values, and figure-input mappings are provided in Appendix A and the accompanying provenance manifest.

4. Handshake Expansion and Transport Thresholds

4.1. Message Sizes and Baseline Segmentation Structure

For X25519, the ClientHello handshake message and the TLS record containing ServerHello are 216 B and 122 B, respectively. The corresponding values are 1360 B and 1178 B for ML-KEM-768, 1392 B and 1210 B for X25519MLKEM768, 2417 B and 4523 B for HQC-L1, and 4690 B and 9068 B for HQC-L3. The HQC-L3 ClientHello is approximately 1.94 times the size of HQC-L1, while its ServerHello TLS record is approximately 2.00 times as large.

At the baseline MTU, the target ClientHello and ServerHello records of X25519 and the ML-KEM family are each carried in one segment. HQC-L1 and its hybrid group use two segments for ClientHello and four for ServerHello, while HQC-L3 uses four and seven, respectively. No retransmission, reordering, or loss was observed in the baseline captures, and the maximum TCP payload was 1448 B.

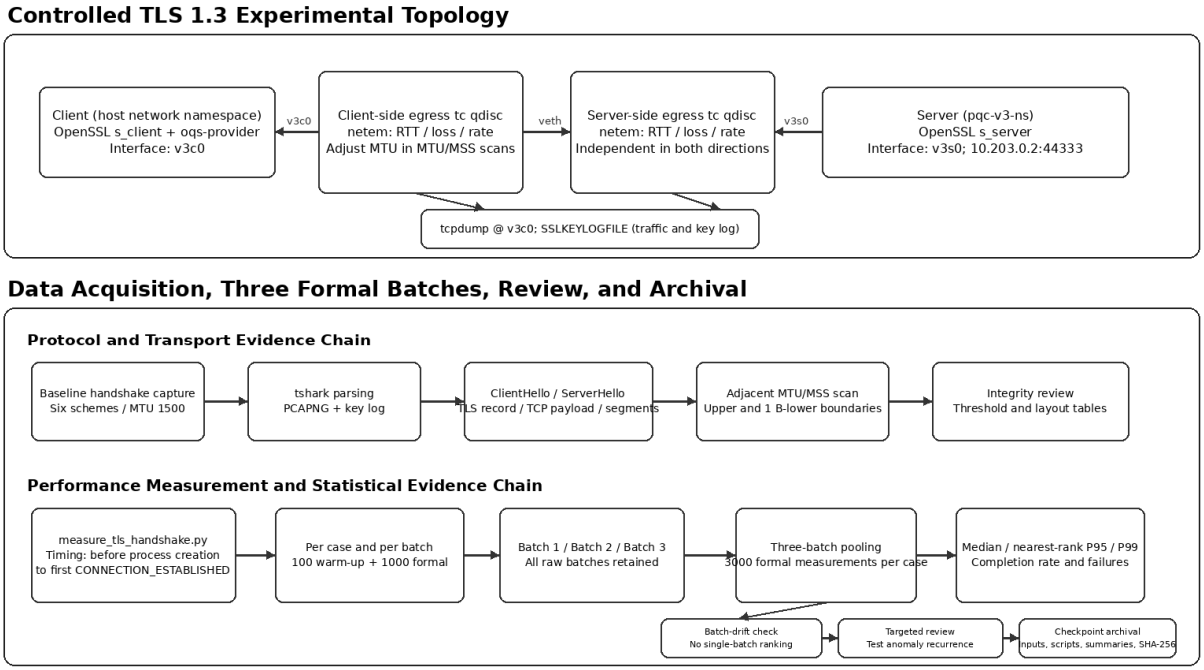


Fig 1. Controlled TLS 1.3 topology and the data chain from protocol- and transport-layer evidence through three formal batches, targeted reviews, and checkpoint archival.

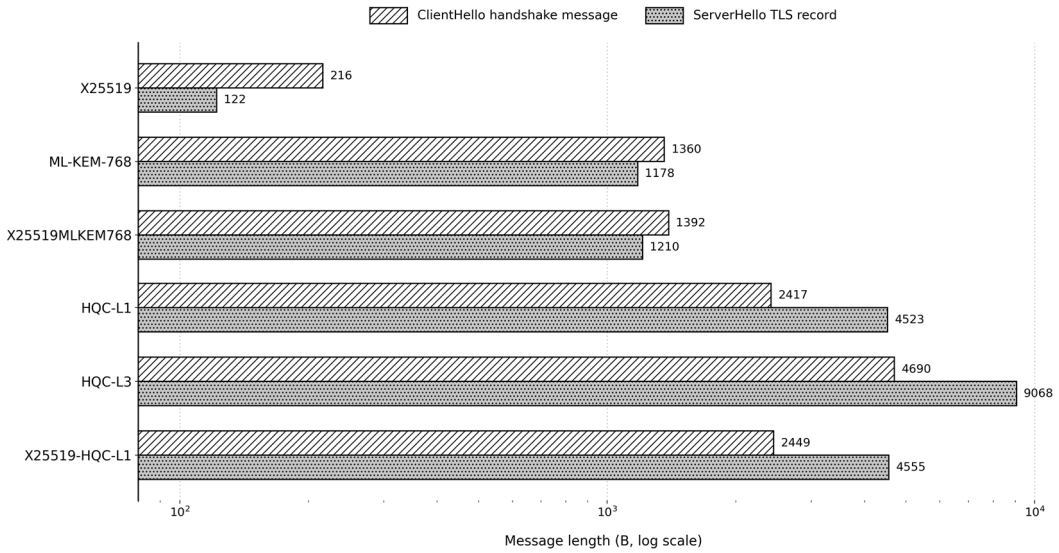


Fig 2. Handshake-message sizes for the six schemes (logarithmic horizontal axis).

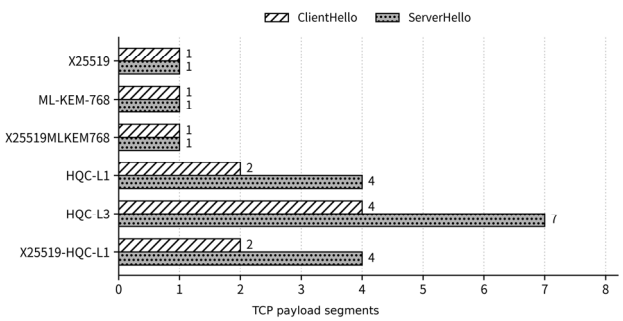


Fig 3. Number of TCP segments carrying the target ClientHello and ServerHello at MTU 1500.

4.2. B. Hybrid-Material Increment and Threshold Shift

For both hybrid schemes, the client and server shares are each 32 B larger than in the corresponding pure-PQC scheme. For ML-KEM, which is originally carried in one segment, these 32 B move the ClientHello upper-bound MTU from 1421 to 1453 and the ServerHello upper bound from 1235 to 1267, a 32 B shift in both cases. For HQC-L1, which already spans multiple segments, the ClientHello upper bound moves from 1265 to 1281, a shift of only 16 B, while the ServerHello upper bound moves from 1184 to 1192, a shift of only 8 B. Because the added bytes are distributed across several existing TCP segments, the threshold shift need not always equal the message-size increment.

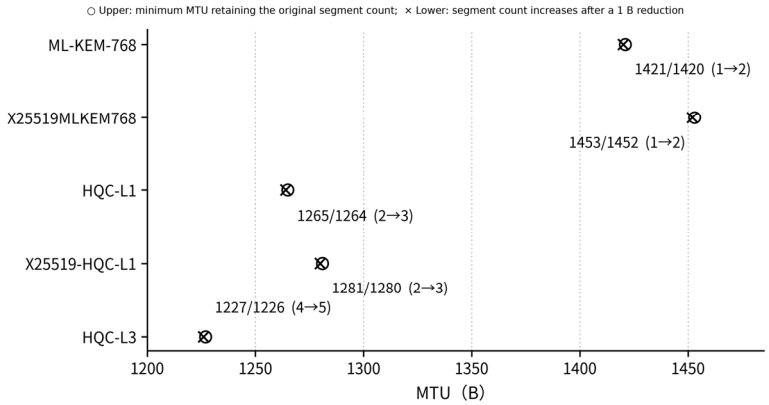


Fig 4. Adjacent-MTU segmentation thresholds for ClientHello. Circles denote the upper boundary; crosses denote the lower boundary, where reducing the MTU by 1 B increases the segment count

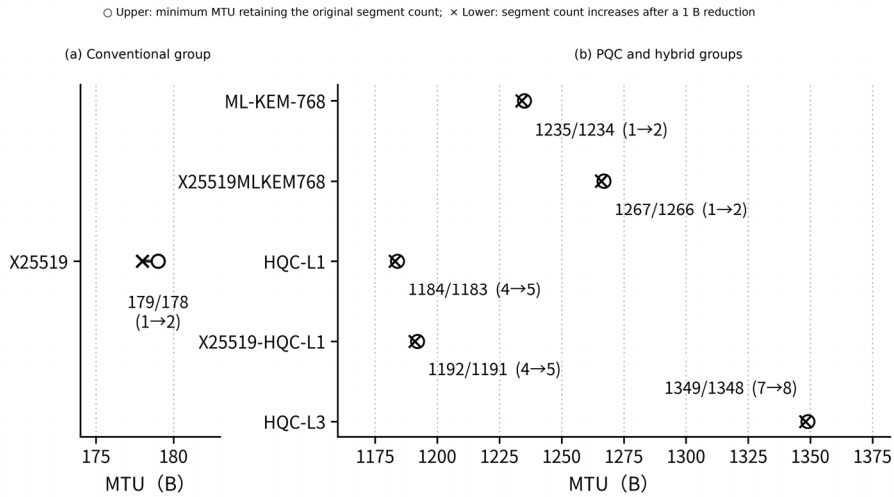


Fig 5. Adjacent-MTU segmentation thresholds for ServerHello. The conventional group and the PQC/hybrid groups use separate axis scales.

Table 3. Main Segmentation Thresholds (Upper/Lower)

Group	**ClientHello MTU**	**ServerHello MTU**
X25519	Not scanned	179/178 (1→2)
ML-KEM-768	1421/1420 (1→2)	1235/1234 (1→2)
X25519MLKEM768	1453/1452 (1→2)	1267/1266 (1→2)
HQC-L1	1265/1264 (2→3)	1184/1183 (4→5)
HQC-L3	1227/1226 (4→5)	1349/1348 (7→8)
X25519-HQC-L1	1281/1280 (2→3)	1192/1191 (4→5)

The absolute threshold values are not monotonic in message length because the TCP segment count is jointly determined by message length and effective MSS [21]. The effective MSS can be expressed as MSS_eff = MTU - H_IP - H_TCP,opt, where H_TCP,opt includes the basic TCP header and options. At MTU 1500, the maximum TCP payload observed in this experiment was 1448 B, corresponding to 52 B of IP/TCP header and option overhead. For example, the ML-KEM-768 ClientHello is still carried in one TCP data segment at MTU 1421, but requires two segments at MTU 1420. This is a TCP segmentation-boundary transition rather than IP fragmentation. From an engineering perspective, the more important quantity is the headroom between the current message and the next segment-count transition. When a tunnel, VPN, or other encapsulation reduces the effective MTU, even a small total byte increment may change the TCP carriage state.

5. Connection-Establishment Performance under RTT

All 54,000 formal connections in the RTT experiment completed and passed verification. As RTT increased from 35 ms to 200 ms, the latency distributions of all groups shifted to the right, showing that round-trip delay forms the principal common baseline of connection-establishment latency. For “ML-KEM-768 minus X25519,” the pooled three-batch P99 differences at 35, 70, and 200 ms were -0.210, 0.721, and 0.826 ms, respectively. However, the directions of within-batch median differences were inconsistent, and the third batch at 200 ms showed a positive difference of 10.576 ms. The data therefore support the conclusion that the two are generally close, but not a stable ranking of superiority.

HQC-L3 had the highest median, P95, and P99 under all three RTT conditions. For “HQC-L3 minus HQC-L1,” the pooled three-batch median differences were 5.542, 11.344, and 22.747 ms. All nine within-batch comparisons across batch × RTT were positive. The batch-difference ranges at 35, 70, and 200 ms were 4.210-6.031, 9.308-14.727, and 23.103-24.718 ms, respectively. This same-direction effect across conditions and batches is consistent with the mechanism by which large, multi-segment messages incur additional waiting on high-RTT links, but it does not justify attributing an individual slow sample to a specific TCP event.

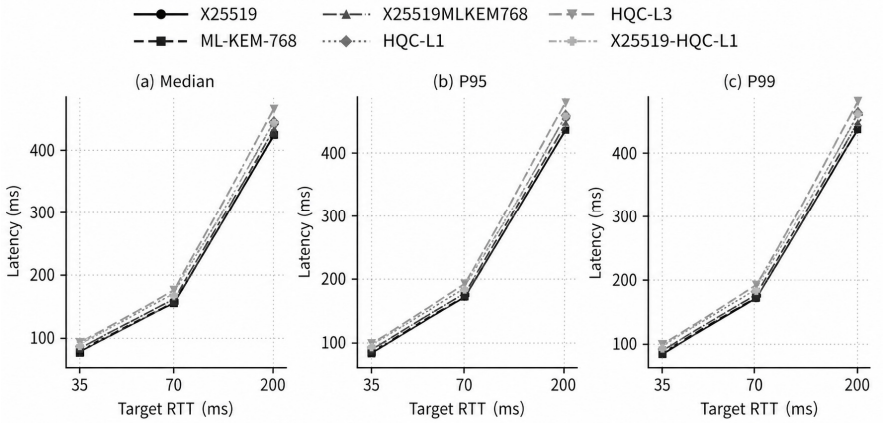


Fig 6. Pooled three-batch median, P95, and P99 connection-establishment latency at RTTs of 35, 70, and 200 ms. Each point contains 3000 formal measurements; P95/P99 use the nearest-rank method

Neither hybrid group showed a stable, monotonic additional latency. The within-batch differences of X25519MLKEM768 relative to ML-KEM-768 and of X25519-HQC-L1 relative to HQC-L1 changed sign across conditions and batches. The former was -8.143 ms in the second 70 ms batch, and the latter was -11.373 ms in the third 200 ms batch. Because the direction was unstable, the effect of the additional 32 B was smaller than environmental drift and the transport cost of the primary PQC material.

Batch drift. The third RTT batch increased across several groups and conditions, but not by a fixed offset. At 35 ms, for example, the HQC-L1 batch medians were 82.857, 82.912, and 90.405 ms, whereas the X25519 medians were 81.359, 80.630, and 80.874 ms. The main results therefore use pooled three-batch distributions while retaining the batch ranges, avoiding scheme rankings based on a single batch.

6. Systematic Amplification Under Bandwidth Limits

All 54,000 formal connections and 2,400 targeted review connections in the bandwidth experiment completed and passed verification. Across the 18 within-batch comparisons of "100 Mbps minus 1000 Mbps," the median at 100 Mbps was higher in nine cases and lower in nine cases, an exact split. Targeted review further showed that the third-batch increase did not persist. The data therefore do not support a practically meaningful stable ranking between 100 and 1000 Mbps.

By contrast, 1 Mbps created a clear bottleneck. Relative to 100 Mbps, all 18 within-batch median differences across six schemes and three batches were positive. The batch increase ranged from 7.576 to 13.570 ms for X25519 and from 136.274 to 139.985 ms for HQC-L3. The pooled median increase rose from 10.499 ms for X25519 to 138.184 ms for HQC-L3; complete results are given in Table 4. HQC-L3 P95 and P99 increased by 140.475 and 133.983 ms, respectively. The similar increases in median, P95, and P99 indicate a broad distribution shift rather than an effect caused only by a few anomalous tail samples.

To test whether this effect primarily arose from link serialization, the theoretical increase when bandwidth falls from 100 Mbps to 1 Mbps was estimated from the bidirectional TCP payload B in the baseline PCAP using $\Delta t_ser = 8B(1/R_1 - 1/R_2)$. The theoretical values for ML-KEM-768, X25519MLKEM768, HQC-L1, and X25519-HQC-L1 were 34.848, 35.355, 69.712, and 70.219 ms, respectively; the corresponding measured median increases were 33.662, 34.620, 70.591, and 71.177 ms. The theoretical scale is broadly consistent with the measurements, supporting the interpretation that low bandwidth mainly converts additional bytes into serialization delay. The deviations for X25519 and HQC-L3 show that fixed overhead, multi-segment scheduling, and implementation behavior still affect the absolute values.

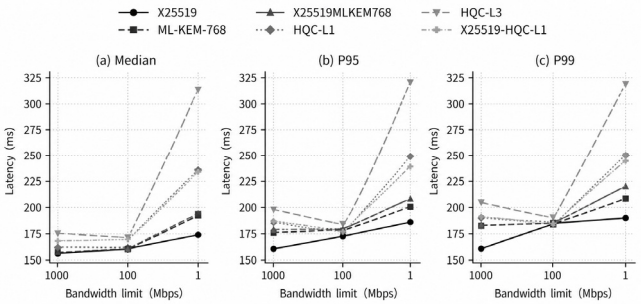


Fig 7. Median, P95, and P99 connection-establishment latency under different per-direction bandwidth limits at fixed RTT 70 ms with no packet loss. The horizontal axis decreases from 1000 Mbps to 1 Mbps.

Table 4. Increase In Connection-Establishment Latency At 1 Mbps Relative To 100 Mbps (Ms)

Group	Median	**P95**	**P99**
X25519	10.499	4.823	5.745
ML-KEM-768	33.662	22.318	22.650
X25519MLKEM768	34.620	27.944	29.238
HQC-L1	70.591	72.116	66.180
HQC-L3	138.184	140.475	133.983
X25519-HQC-L1	71.177	69.434	63.742

At 1 Mbps, X25519MLKEM768 exceeded ML-KEM-768 by only 2.069, 6.534, and 7.594 ms in median, P95, and P99, respectively. The difference between X25519-HQC-L1 and HQC-L1 remained within measurement fluctuation. Thus, the principal transport burden of a hybrid scheme comes from its post-quantum component rather than from the additional X25519 share.

Drift check and targeted review. Figure 8 compares the formal batches, pooled result, and targeted review for three representative 100 Mbps cases.

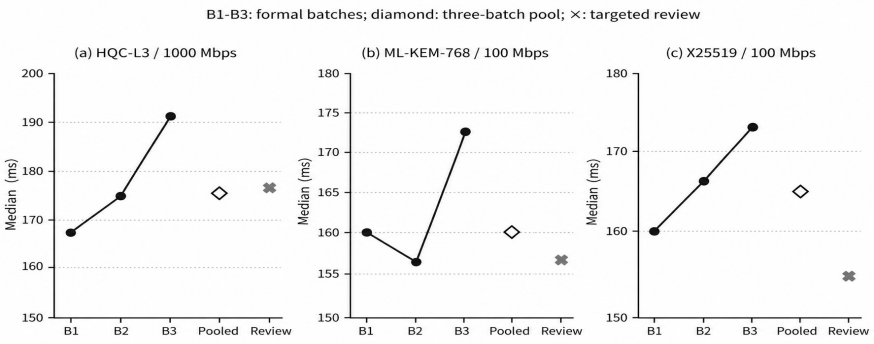


Fig 8. Batch medians, pooled three-batch results, and targeted reviews for three representative 100 Mbps cases. The reviews did not support persistence of the third-batch increase

Figure 8 uses the 100 Mbps cases of HQC-L3, ML-KEM-768, and X25519 to illustrate the elevated third batch. Review values returned near those of the first two batches, indicating that the small differences between 100 and 1000 Mbps are better explained as batch drift than as a stable bandwidth effect. The original third batch remains in the pooled data to avoid post hoc point removal.

7. Long Tails and Connection Completion Under Packet Loss

The independent random-loss experiment in each direction produced a distribution shape different from those under RTT and bandwidth variation. All 54,000 formal connections at 1%, 3%, and 5% completed and passed verification, but P95 and P99 rose before the median changed substantially. The median of HQC-L3 exceeded that of HQC-L1 in all 12 batch × loss-rate comparisons, with differences ranging from 2.688 to 144.405 ms; under the 10% condition, the three batch differences ranged from 110.322 to 144.405 ms. At 1%, X25519 had a P95 of 164.703 ms, whereas HQC-L1, HQC-L3, and X25519-HQC-L1 reached approximately 329.523, 361.769, and 341.360 ms. These same-direction results are consistent with the explanation that multi-segment handshake messages have a larger exposure surface to loss.

At 1% independent random loss in each direction, the P99 values of all six groups fell between 1.178 and 1.227 s. At 3%, HQC-L3 and X25519-HQC-L1 reached approximately 1.533 s and 1.502 s, respectively; at 5%, several post-quantum groups exceeded 2.2 s. The median was not strictly monotonic between 3% and 5%, because it can remain in the fast-sample region as long as fewer than 50% of samples enter the slow path.

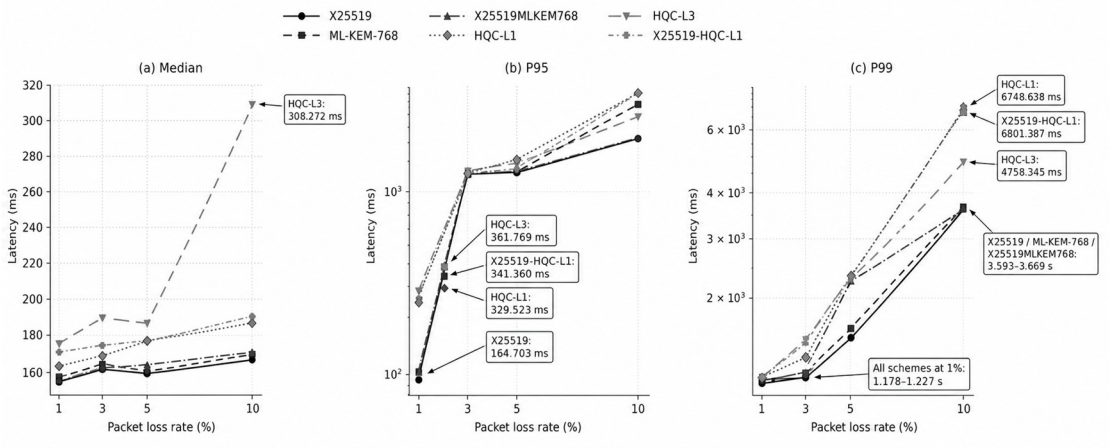


Fig 9. Median, P95, and P99 connection-establishment latency under different independent random loss rates in each direction at fixed RTT 70 ms without bandwidth limitation. P95/P99 use logarithmic vertical axes.

The 10% independent random-loss condition in each direction was the only formal condition with incomplete connections: 17,970 of 18,000 completed, 30 were incomplete, and the completion rate was 99.8333%. HQC-L1 and X25519-HQC-L1 had 9 and 8 incomplete connections, respectively, whereas X25519 had only 1. In targeted review, however, both HQC-L1 and hybrid HQC completed all attempts, while X25519 and HQC-L3 each had 1 incomplete attempt. Incomplete attempts remain in the completion-rate denominator and are not imputed into successful-sample latency quantiles. The small number of incomplete events should therefore be treated as a risk signal rather than as sufficient evidence for a strict reliability ranking.

Table 5. Connection Completion and Successful-Sample Latency At 10% Independent Random Loss In Each Direction (10 S Timeout)

Group	Comp.	Incomp.	Completion rate	Median (ms)	P99 (ms)
X25519	2999	1	99.9667%	168.110	3593.055
ML-KEM-768	2995	5	99.8333%	171.322	3668.699
X25519MLKEM768	2998	2	99.9333%	172.592	3596.535
HQC-L1	2991	9	99.7000%	187.685	6748.638
HQC-L3	2995	5	99.8333%	308.272	4758.345
X25519-HQC-L1	2992	8	99.7333%	191.020	6801.387

Review results. Figure 10 compares the formal batches, pooled results, and targeted reviews for HQC-L3 and X25519-HQC-L1 under 10% independent random loss in each direction.

The HQC-L3 medians in the three formal batches at 10% were 287.183, 332.505, and 307.114 ms, with a pooled median of 308.272 ms. The review median increased further to 349.746 ms, and the review P99 was 6447.759 ms. Both the elevated distribution center and the severe long tail appeared in all three formal batches and in review, giving this degradation strong cross-batch and review consistency.

Among the 12 within-batch median differences of X25519-HQC-L1 relative to HQC-L1, 7 were positive and 5 negative, so no stable direction emerged. In the third batch at 10%, the difference reached 123.651 ms and raised the batch median to 311.824 ms, while the first two batches were only 188.056 and 187.343 ms. The review median fell to 169.133 ms, and P99 fell from 6801.387 ms in the pooled formal result to 3515.843 ms; the overall third-batch increase did not recur. That batch remains in the pooled data, but it cannot be interpreted as a stable algorithmic property of hybrid HQC.

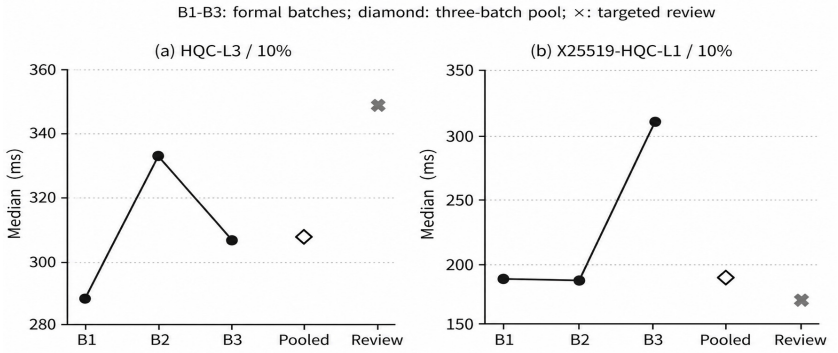


Fig 10. Batch medians, pooled three-batch values, and targeted reviews for HQC-L3 and X25519-HQC-L1 under 10% independent random loss in each direction.

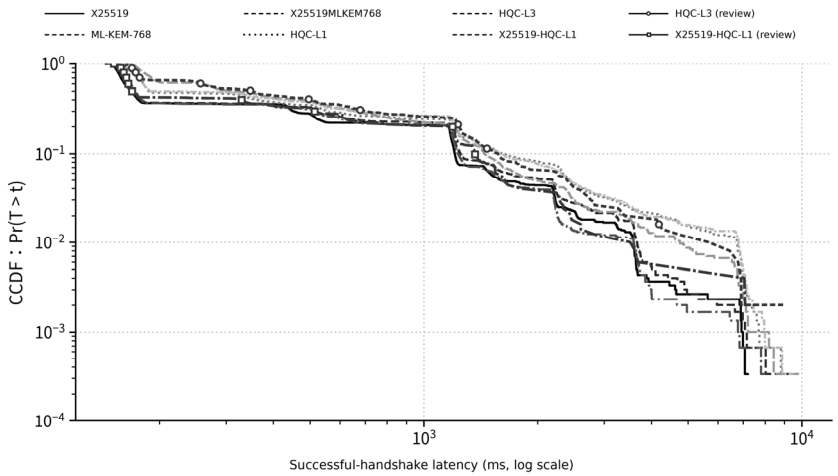


Fig 11. CCDF of successful-sample connection-establishment latency under 10% independent random loss in each direction. The three formal batches and two targeted review cases are shown together

In the 10% formal experiment, the highest P99 values were observed for X25519-HQC-L1 and HQC-L1, at 6801.387 and 6748.638 ms, respectively, while HQC-L3 reached 4758.345 ms. This shows that an elevated median, extreme P99, and number of incomplete connections are distinct risk dimensions. The defining feature of HQC-L3 is that its degradation recurred in review, not that it was absolutely worst in every statistic.

8. Cross-Layer Discussion and Deployment Implications

8.1. A. From Material Size to Network-Impairment Amplification

The experiments form a continuous evidence chain from

cryptographic material to TLS messages, TCP segmentation and thresholds, network impairment, and client-observed connection-establishment latency. Message sizes and thresholds are direct protocol measurements, while the RTT, bandwidth, and loss experiments observe how these structural differences manifest under different network mechanisms. The results show that a larger message does not imply substantially higher latency under every condition. Although ML-KEM is much larger than X25519, it remains close when there is no packet loss or bandwidth bottleneck. Material differences become clearly amplified only when bandwidth is constrained, segmentation increases, or packet loss triggers recovery paths.

8.2. Differences Among the Three Network-Impairment Classes

RTT mainly raises the common time baseline while widening the gap between HQC-L3 and HQC-L1. Low bandwidth converts byte volume into a predictable serialization cost. Random loss increases the proportion of samples entering slow paths, primarily amplifying P95 and P99 and producing a small number of incomplete connections at high loss. A single mean or a generic "handshake overhead" therefore cannot simultaneously describe the distribution center, tail behavior, and connection-completion risk.

8.3. MTU/MSS Headroom and Hybrid Deployment

Engineering assessment should not merely determine whether a message exceeds one TCP segment; it should also examine the remaining headroom to the next segment-count transition. The ML-KEM family remains single-segment at ordinary MTUs, but additional encapsulation may push ClientHello across a boundary. HQC is already multi-segment, and a lower MTU may change 2 segments to 3, 4 to 5, or 7 to 8. Conversely, the additional 32 B in the hybrid groups does not create "double overhead"; their performance remains primarily determined by the PQC component.

8.4. Evidence Consistency of the Main Findings

Table 6 summarizes consistency among the formal batches, within-batch comparisons, and targeted reviews. Only results that are directionally consistent across batches and not contradicted by review are described as stable effects. Results with inconsistent direction or without recurrence in review are not used to rank schemes.

Table 6. Evidence-Consistency Matrix for The Main Findings

Main finding	Formal data	Within-batch consistency	Targeted review	Assessment
1 Mbps bandwidth bottleneck	18/18 median differences for 1 vs 100 Mbps were positive; all three quantiles moved in the same overall direction	Consistent	Formal effect far exceeds drift	Stable main effect
HQC-L3 degradation under high loss	Relative to HQC-L1: 12/12 batch × loss median differences were positive	Consistent	Elevated center and long tail recurred	Consistent across batches and review
ML-KEM and X25519 are close under RTT	Pooled P99 difference <1 ms; batch-median directions were mixed	Partial	Not performed	Generally close; no stable superiority ranking
Ranking at 100 vs 1000 Mbps	Within-batch comparisons: 9 positive and 9 negative	Inconsistent	Third-batch increase did not persist	Stable ranking not supported
Third-batch increase of X25519-HQC-L1	Relative to HQC-L1: 7 positive and 5 negative; abnormally large in the third 10% batch	Inconsistent	Did not recur	Retained as a single-batch anomaly
Ranking by incomplete count at 10%	30/18,000 incomplete; low event count	Distribution changed across batches	Event distribution changed in review	Strict reliability ranking not supported

8.5. Threats to Validity and Study Boundaries

This study was conducted in a single-host veth and Network Namespace topology, which isolates the principal network variables. However, all formal samples came from one virtual machine and one software build and cannot represent cross-hardware platforms, the public Internet, wireless links, or environments with complex middleboxes. RTT, bandwidth, and packet loss were evaluated one factor at a time rather than in a full three-dimensional factorial matrix. MTU/MSS thresholds also depend on the current TLS implementation, Linux TCP stack, handshake content, and additional encapsulation and should not be treated as universal constants.

Network conditions and schemes were executed in a fixed order within each batch, so temporal drift may be partially confounded with execution order. Pooling three batches, using within-batch differences, diagnosing batch drift, and conducting targeted reviews reduce this risk but cannot fully replace randomization or counterbalanced execution. The Network Namespace and veth topology were removed after the experiment, so the offload-switch state during the formal runs cannot be reconstructed retrospectively. TCP-segmentation judgments therefore rely on the payload sizes and segment counts actually observed in the archived PCAP files.

The formal performance experiments did not preserve a complete retransmission capture matched one-to-one with every slow sample. Accordingly, this paper uses phrases such as "consistent with transport recovery" or "supports this mechanism-based explanation" rather than attributing every P99 event directly to a specific retransmission. The successful-sample distribution in the 10% loss experiment is right-censored by the 10 s timeout, and no survival model is fitted; its P95/P99 values should therefore be interpreted conditionally on completion within 10 s and together with the reported completion rate. The study focuses on key exchange and connection establishment and does not simultaneously evaluate post-quantum certificates, signatures, session resumption, QUIC, application-data transfer, energy consumption, or multi-client concurrency.

9. Conclusion

RQ1. The six schemes differ first in handshake material and TCP carriage structure. At MTU 1500, the target ClientHello and ServerHello of the ML-KEM family remain

single-segment, whereas HQC-L1, HQC-L3, and hybrid HQC already form multi-segment messages and exhibit reproducible discrete segment-count transitions when the MTU decreases by 1 B.

RQ2. RTT mainly raises the common baseline; 1 Mbps bandwidth converts message size into a stable serialization cost; and random packet loss preferentially amplifies P95/P99 while producing a small number of incomplete connections at 10%. The formal loss experiment contained 72,000 measurements, and 30 of the 18,000 measurements at 10% did not complete and verify the connection within 10 s.

RQ3. The differences between the two hybrid groups and their corresponding pure-PQC groups changed across conditions and batches and did not form a stable, monotonic additional penalty. The current data indicate that the primary transport burden of a hybrid scheme comes from its PQC component rather than from the additional 32 B X25519 share.

RQ4. The elevated distribution center of HQC-L3 under RTT and loss conditions is consistent across batches, and its high-loss long tail recurs in targeted review. The overall increase of X25519-HQC-L1 in the third batch did not persist in review and should therefore be treated as a single-batch anomaly.

In summary, post-quantum TLS deployment performance cannot be judged solely from KEM operation speed or material byte count. It requires joint examination of TLS encapsulation, current TCP segment count, MTU/MSS headroom to the next threshold, and the RTT, bandwidth, and packet loss of the target network. Under the software versions, single-host TCP emulation topology, and client-observed connection-establishment metric used in this study, ML-KEM and its X25519 hybrid remain close to X25519 when there is no loss or material bandwidth bottleneck, whereas the large-material HQC implementation groups show more pronounced center and tail degradation under low bandwidth and random loss. These conclusions concern only transport performance within the experimental scope and do not constitute an overall ranking of algorithm security, standards maturity, or complete deployment solutions.

Data and Code Traceability

Research assets are preserved in layers: raw evidence, processed results, formal batches, targeted reviews, and final checkpoints. Protocol-layer assets include baseline PCAPNG files, key logs, TLS-record and TCP-field TSV files, and MTU-scan results. Performance-layer assets include per-attempt RTT, bandwidth, and loss TSV files, three-batch summaries, incomplete-connection records, and review data. Raw formal batches were not modified in response to anomaly reviews, and every figure and table is mapped to explicit inputs and generation scripts. The assets are currently archived as companion material and can be organized as anonymized supplementary files, with a README and provenance manifest, for the requirements of the selected venue.

The final archived checkpoints are RTT_ FORMAL_ 3000_SUCCESS (20260721_050033), BANDWIDTH_ COMPLETE_SUCCESS (20260723_124823), and LOSS_ COMPLETE_SUCCESS (20260723_194637), with SHA-256 prefixes 2376304f, a803d95e, and e2514944, respectively. Complete names, full hashes, file-by-file inventories, and mappings from figure inputs to scripts are provided in the accompanying Figure and Data Provenance Manifest.

References

[1] Shor, P. W. (1994). Algorithms for quantum computation: Discrete logarithms and factoring. In Proceedings of the 35th Annual Symposium on Foundations of Computer Science (pp. 124–134). https://doi.org/10.1109/SFCS.1994.365700.

[2] Rescorla, E. (2026). The transport layer security (TLS) protocol version 1.3. RFC 9846. https://doi.org/10.17487/RFC9846.

[3] National Institute of Standards and Technology. (2024). Module-lattice-based key-encapsulation mechanism standard. FIPS 203. https://doi.org/10.6028/NIST.FIPS.203.

[4] Alagic, G., et al. (2025). Status report on the fourth round of the NIST post-quantum cryptography standardization process. NIST IR 8545. https://doi.org/10.6028/NIST.IR.8545.

[5] Stebila, D., Fluhrer, S., & Gueron, S. (2026). Hybrid key exchange in TLS 1.3. RFC 9954. https://doi.org/10.17487/RFC9954.

[6] Connolly, D. (2026, July 20). ML-KEM post-quantum key agreement for TLS 1.3. IETF Internet-Draft draft-ietf-tls-mlkem-09. Work in progress.

[7] Kwiatkowski, K., Kampanakis, P., Westerbaan, B. E., & Stebila, D. (2026, May). Post-quantum hybrid ECDHE-MLKEM key agreement for TLSv1.3. IETF Internet-Draft draft-ietf-tls-ecdhe-mlkem-05. Work in progress.

[8] Mosca, M., & Stebila, D. (2016). Post-quantum key exchange for the Internet and the Open Quantum Safe project. Cryptology ePrint Archive, Paper 2016/1017.

[9] Paquin, C., Stebila, D., & Tamvada, G. (2020). Benchmarking post-quantum cryptography in TLS. In Post-Quantum Cryptography (LNCS 12100, pp. 72–91). https://doi.org/10.1007/978-3-030-44223-1_5.

[10] Henrich, J., Heinemann, A., Wiesmaier, A., & Schmitt, N. (2023). Performance impact of PQC KEMs on TLS 1.3 under varying network characteristics. In Information Security (LNCS 14411, pp. 267–287). https://doi.org/10.1007/978-3-031-49187-0_14.

[11] Zheng, J., Zhu, H., Dong, Y., Song, Z., Zhang, Z., Yang, Y., & Zhao, Y. (2024). Faster post-quantum TLS 1.3 based on ML-KEM: Implementation and assessment. In Computer security - ESORICS 2024 (LNCS 14983, pp. 123–143). https://doi.org/10.1007/978-3-031-70890-9_7

[12] Sosnowski, M., et al. (2023). The performance of post-quantum TLS 1.3. In Companion of the 19th International Conference on Emerging Networking Experiments and Technologies (CoNEXT Companion '23) (pp. 19–27). https://doi.org/10.1145/3624354.3630585

[13] Kampanakis, P., & Childs-Klein, W. (2024). The impact of data-heavy, post-quantum TLS 1.3 on the time-to-last-byte of real-world connections. In Workshop on Measurements, Attacks, and Defenses for the Web (MADWeb 2024). https://doi.org/10.14722/madweb.2024.23010

[14] Henrich, J., Schmitt, N., Alnahawi, N., & Heinemann, A. (2026). A lot of data and added complexity. How does PQC affect the performance of my TLS connection? In Information security (LNCS 16186, pp. 107–128). https://doi.org/10.1007/978-3-032-08124-7_6

[15] Souvatzidaki, K., & Limniotis, K. (2025). Post-quantum key exchange in TLS 1.3: Further analysis on performance of new cryptographic standards. Cryptography, 9(4), 73. https://doi.org/10.3390/cryptography9040073

[16] Rios, R., & Montenegro, J. A. (2026). Post-quantum cryptography: A multi-layer bottleneck analysis. International Journal of Information Security, 25, 89. https://doi.org/10.1007/s10207-026-01242-0

[17] OpenSSL Project. (2026, June 9). OpenSSL 3.0.21. OpenSSL 3.0 LTS release.

[18] Open Quantum Safe Project. (2026). liboqs (Version 0.16.0, commit 5a1a854b).

[19] Open Quantum Safe Project. (2026). oqs-provider (Version 0.12.0-dev, commit 1670a8a).

[20] Hemminger, S. (2005). Network emulation with NetEm. OSDL.

[21] Eddy, W. (2022). Transmission control protocol (TCP). RFC 9293.