# The 1-RTT Penalty: Quantifying the Recurring Cost of PQC Fragmentation in LEO Handovers and ISLs

Young Eun Kwon
Korea University
equation@korea.ac.kr

Ji Won Yoon
Korea University
jiwon_yoon@korea.ac.kr

***Abstract*—Low Earth Orbit (LEO) satellite networks operate under strict latency and reliability constraints, yet require Post-Quantum Cryptography (PQC) to secure them from future threats. The large signature sizes of most PQC algorithms, however, conflict with these network-level constraints. Through comprehensive ns-3 simulations (modeling fragmentation, packet loss, and handovers), this paper demonstrates that network performance, not raw CPU computation, is the dominant factor for PQC in LEO.**

**We find that PQC certificates exceeding the 1500-byte MTU, like Dilithium (2,588 B), incur IP fragmentation. While aggressive TCP congestion windows might mask the initial latency in ideal conditions, we demonstrate that this multi-packet nature induces a severe reliability penalty regardless of window size. Specifically, fragmentation doubles the exposure to packet loss, increasing the probability of a catastrophic TCP RTO (1,000 ms+) during 'Rain Fade' events to 51%, compared to just 30% for the single-packet Falcon (858 B). This results in a massive 100-500% latency penalty in lossy conditions, rendering Dilithium's 18 µs CPU advantage negligible. Finally, we prove a Full-PQC data verification model is infeasible, creating a 345 ms CPU bottleneck and confirming the necessity of a Hybrid-PQC approach.**

**We conclude that the Falcon-based hybrid protocol is the only solution that simultaneously avoids both network-level (fragmentation, RTO) and CPU-level (bottleneck) penalties, establishing it as the most practical and robust quantum-resistant solution for future LEO satellite networks.**

## I. Introduction

Low Earth Orbit (LEO) satellite constellations are emerging not merely as a communication technology, but as a transformative critical infrastructure poised to deliver low-latency, high-bandwidth internet access to the entire globe. By interconnecting thousands of satellites, LEO networks will eliminate communication dead zones, becoming the backbone for critical sectors including IoT, global mobility, and defense.

However, the long-term security and viability of this infrastructure face a profound threat. Modern communication relies entirely on public-key cryptosystems like RSA and ECC, which are vulnerable to Shor's algorithm [1]. The advent of a large-scale quantum computer could render these systems insecure. As satellite infrastructure is deployed for operational lifespans exceeding a decade, failing to build in quantum resistance now exposes the entire network to catastrophic future risk. While 'Harvest Now, Decrypt Later' attacks threaten data confidentiality today, the compromise of signature schemes poses an existential threat to long-term infrastructure control. Without PQC signatures, satellites launched today remain vulnerable to impersonation and hijacking by quantum adversaries throughout their operational life, allowing attackers to inject malicious commands or spoof ground stations in the future.

In response, NIST has finalized Post-Quantum Cryptography (PQC) standards, including ML-DSA (Dilithium), SLH-DSA (SPHINCS+), and Falcon. The trade-off for their quantum-resistant security is a massive increase in key and signature sizes compared to ECC.

This PQC overhead directly conflicts with the intrinsic constraints of LEO satellite communication, which are defined by a complex interplay of network challenges [2]. These links are constrained by a 1,500-byte Maximum Transmission Unit (MTU), just like most IP networks. They suffer from unreliable link quality, as Ka and Ku-band frequencies are highly susceptible to 'Rain Fade,' a phenomenon causing sudden and severe bursts of packet loss. Finally, the high velocity of satellites mandates frequent handovers, a process where any authentication delay translates directly into service disruption. PQC algorithms like Dilithium (≈2,588 B) or SPHINCS+ (8 KB+) exceed this 1,500-byte MTU, forcing their certificates to be fragmented across multiple IP packets [3]. This paper identifies this fragmentation as the central, fundamental problem hindering the deployment of PQC in LEO networks. We hypothesize that this fragmentation, when combined with TCP's conservative congestion control and the unreliable nature of LEO links, creates a catastrophic network penalty in the form of cascading RTT and fatal ReTransmission Timeout (RTO) delays. We posit that these network-layer penalties, not raw CPU computation, will dominate PQC performance in this environment.

While the detrimental effects of IP fragmentation on reliability are well-established principles in general networking, their specific implications for the NIST PQC migration in LEO constellations remain unquantified. The interplay between specific PQC artifact sizes and LEO-specific dynamics, such as frequent handovers and 'Rain Fade,' creates unique failure modes that general theory does not predict. This paper moves

Workshop on Security of Space and Satellite Systems (SpaceSec) 2026
23 February 2026, San Diego, CA, USA
ISBN 979-8-9919276-1-1
https://dx.doi.org/10.14722/yyy.2026.[23|24]xxx
www.ndss-symposium.org

beyond qualitative theoretical risks to quantitatively measure the operational cost of these algorithms in space.

To validate this hypothesis, this paper presents the first high-fidelity PQC performance evaluation framework for LEO networks, integrating the ns-3 network simulator with the Open Quantum Safe (OQS) library. We designed and executed eight comprehensive experiments modeling fragmentation, persistent and dynamic packet loss, ISL multi-hop, ground station handovers, and CPU load.

Our findings are clear. Network performance is an overwhelmingly more critical metric than CPU computation speed. The MTU-exceeding Dilithium certificate incurs a minimum 1-RTT (10 ms) additional network latency penalty compared to the single-packet Falcon (858 B), a direct result of its interaction with conservative TCP baselines. While Dilithium's CPU verification (25 µs) was 18 µs faster than Falcon's (43 µs), this advantage was rendered entirely negligible by the 10,000 µs network penalty.

This 1-RTT penalty is not a one-time cost. We demonstrate that it amplifies proportionally (10 ms $\rightarrow$ 30 ms) as ISL hops increase and, critically, occurs recursively at every handover, hindering seamless mobility. Furthermore, during a 'Rain Fade' event (30% PER), the multi-packet Dilithium faced a 51% probability of a catastrophic TCP RTO (1,000 ms+), proving it fatally vulnerable to link instability.

Finally, we confirm the necessity of a hybrid model. A 'Full-PQC' approach, verifying all data packets, was proven infeasible, creating a 345 ms CPU bottleneck. This validates that a hybrid protocol, using Falcon for certificates and ECDSA for data, is essential. In conclusion, the Falcon-based hybrid protocol is the only approach that simultaneously avoids the network penalty (fragmentation, RTO) and the CPU bottleneck. We strongly suggest it as the most practical, robust, and efficient quantum-resistant security solution for future LEO satellite networks.

Our contributions are thus threefold. First, we present the first high-fidelity simulation framework (ns-3 + OQS) that models the dynamic network characteristics of LEO environments for PQC evaluation. Second, we provide the first quantitative analysis linking NIST PQC artifact sizes to LEO-specific failure modes. By empirically validating the catastrophic impact of fragmentation during Rain Fade events, we reframe MTU-compliance not merely as a discretionary network optimization, but as a primary reliability constraint for future satellite security architectures. Third, we validate that the Falcon-based hybrid protocol is the only robust and efficient quantum-resistant solution for LEO networks, offering a concrete architectural blueprint for the post-quantum transition.

## II. Related Works

### A. The NIST PQC Standardization

The impending threat of quantum computing, specifically Shor's algorithm, has rendered the long-term security of classical public-key cryptography like RSA and ECC become useless. In response, the U.S. National Institute of Standards and Technology (NIST) initiated a multi-year competition to standardize quantum-resistant algorithms [1]. This process come to an end with the selection of CRYSTALS-Kyber for key encapsulation and three signature schemes, CRYSTALS-Dilithium (ML-DSA), Falcon, and SPHINCS+ (SLH-DSA) [4].

This standardization provides a clear path forward, but also introduces a critical trade-off. A massive increase in public key and signature sizes in exchange for quantum resistance. As our own analysis confirms, these sizes vary dramatically, from Falcon (858 B) and Dilithium (2,588 B) to SPHINCS+ (8 KB+). This size increase, particularly for signatures used in certificates, creates a new, challenging performance landscape that simple computational benchmarks fail to capture.

### B. PQC Performance Benchmarking

A significant body of literature has centered on benchmarking the raw computational performance of NIST PQC candidates [5]. These studies provide in-depth analysis of CPU cycles or wall-clock time for key generation, signing, and verification across diverse platforms, from high-performance servers to constrained IoT devices [6]. Several papers specifically compare Dilithium and Falcon, noting Falcon's compact signature size and fast verification, which contrast with its more complex signing operation [7]. Other research has evaluated the protocol-level impact of PQC on terrestrial network protocols, such as measuring handshake latency in TLS 1.3 [3] or in general wireless environments like 5G [8].

However, this prior work overwhelmingly focuses on computational cost within stable, high-bandwidth terrestrial contexts. Consequently, this body of work, while valuable, fails to analyze the severe network-level penalties which are originated from packet fragmentation and cascaded into RTT and RTO delays, that arise when large cryptographic payloads are transmitted over uniquely constrained, high-latency, and lossy links.

### C. Classical Security for LEO Satellite Networks

There is a rich history of research into secure authentication and key management for satellite networks. These studies correctly identify the core challenges of LEO environments, including high latency, limited contact windows, and the need for seamless handovers [9]. Numerous protocols have been proposed to address these issues, but these solutions are almost universally based on classical cryptography, such as Elliptic Curve Cryptography (ECC) [10] or novel identity-based schemes.

While this body of work is fundamental to understanding LEO network dynamics, its reliance on classical primitives renders it insufficient for providing long-term security against quantum adversaries. Our research provides a critical bridge, presenting the first rigorous performance evaluation of NIST-standardized PQC algorithms when deployed within this specific, challenging LEO network context.

### D. LEO Network Simulation and PQC Integration

Researchers frequently use network simulators like ns-3 and STK to model LEO constellations [2]. These simulations are robust but have traditionally focused on non-security aspects, such as Layer 3 routing protocol performance like OSPF and IS-IS [11], flow control, topology management, or resilience to network-layer attacks like DDoS [12].

To date, a notable scarcity of research exists on integrating PQC with satellite communications. The few existing papers are often theoretical, explore different technologies such as Quantum Key Distribution (QKD) [13], or focus on bit-level PHY/MAC layer simulations. To our knowledge, no prior research has integrated a PQC library (OQS) with a high-fidelity packet-level simulator (ns-3) to conduct a practical, protocol-level performance analysis of PQC authentication in a LEO-specific environment. Our work is the first to provide this concrete, quantitative analysis of the network-layer consequences.

## III. Methodology

To quantitatively dissect the performance of PQC authentication in LEO networks, we developed a high-fidelity simulation framework. This framework integrates a detailed network-layer model using ns-3 (Network Simulator 3) with a real-world cryptographic library, liboqs (Open Quantum Safe). This approach allows us to systematically capture the critical interplay between network-layer phenomena (latency, fragmentation, loss) and application-layer computational load.

### A. Environmental Setup

All simulations, including network latency measurements and CPU-bound cryptographic operations, were conducted on a single physical server to ensure consistent and reproducible results. The server was running Ubuntu 22.04.5 LTS (Kernel 6.8.0-generic), equipped with an Intel(R) Xeon(R) E-2226G CPU @ 3.40GHz (6 cores) and 31GiB of RAM.

The simulation framework was built using ns-3 version 3.43 and the Open Quantum Safe (liboqs) library version 0.15.0-rc1. All C++ code was compiled using g++ (Ubuntu 11.4.0).

Our testbed simulates a foundational LEO satellite-to-ground (sat2gs) link. The baseline topology consists of two nodes: a GroundStationApp (server) and a SatelliteApp (client). We use ns3::PointToPointHelper to establish the communication link, configuring it with parameters representative of a LEO connection: 1 Gbps data rate and a 5 ms one-way propagation delay, which results in a baseline 10 ms Round-Trip Time (RTT).

The authentication exchange is modeled over ns3::TcpSocket. We made two critical configurations to the TCP stack to isolate the impact of fragmentation. First, we set the global default InitialCwnd (Initial Congestion Window) to 1 packet. While modern implementations often use larger windows, this strictly enforces a cold start behavior, enabling us to isolate the mechanical latency penalty of multi-RTT exchanges. Second, we disable TCP delayed ACKs (DelAckCount = 0) to prevent kernel-level timers from introducing a confounding variable, thus ensuring a more deterministic acknowledgment flow. Finally, a standard 1,500-byte Maximum Transmission Unit (MTU) is enforced on the link, which serves as the primary constraint for our fragmentation analysis.

Regarding the Initial Congestion Window, we acknowledge that RFC 6928 proposes a value of 10 segments. However, we deliberately configured InitialCwnd=1 to model a conservative baseline typical in high-BER satellite links, where aggressive transmission can exacerbate congestion collapse [14]. Crucially, this baseline allows us to mechanically isolate the impact of fragmentation from windowing optimizations. As we demonstrate in Section IV, the primary penalty of fragmentation in LEO is not merely latency, but a severe degradation in reliability (RTO) which persists regardless of the window size.

### B. Cryptographic Scenarios and Payloads

We model the authentication process as the client establishing a TCP connection and sending a 1-byte request, to which the server responds by sending a payload equivalent in size to a PQC certificate. We analyze two primary quantum-resistant scenarios. Falcon-512 (Falcon), using its public key size of 858 bytes, and Dilithium (ML-DSA-44), using its public key size of 2,588 bytes.

This size differential is the core independent variable of our investigation. The Falcon-512 public key (858 B) fits comfortably within a single 1,500-byte IP packet. In contrast, the Dilithium public key (2,588 B) fundamentally exceeds the 1,500-byte MTU. This forces the TCP stack to fragment the payload into two packets: a 1,460-byte full segment and a 1128-byte subsequent segment. We explicitly excluded stateful hash-based signatures (e.g., SPHINCS+ at 8 KB+) as their multi-kilobyte sizes would require 6+ packets, making them prohibitively latent and unreliable under this baseline model. We also model a computationally-trivial Hybrid-PQC baseline to represent the data-plane verification load in a hybrid protocol.

### C. Experimental Design

Our methodology comprises the eight experiments summarized in Table I. These are logically grouped to progressively dissect PQC performance, starting from a baseline case and building to complex, dynamic LEO-specific scenarios.

*1) Baseline Network and CPU Cost Analysis:* Our first experiments establish the foundational costs. In Experiment 1 (Baseline Fragmentation), we measure the total time from the client's request to the reception of the full certificate payload. This isolates the pure network-level penalty caused by Dilithium's fragmentation interacting with the InitialCwnd=1 setting. We immediately decouple this from computational cost in Experiment 3 (Network vs. CPU Latency). By integrating liboqs and std::chrono, we execute a real OQS_SIG_verify operation after the final network packet is received. This allows us to precisely measure and compare the T_Network and T_CPU components of the total authentication latency.

TABLE I: Summary of the Eight Simulation Experiments (Exp. 1-8)

| Exp. ID | Title | Objective | Topology & Key Model / Variable | Payloads | Core Metric | Related Finding |
|---|---|---|---|---|---|---|
| **Exp. 1** | Baseline | Measure the baseline network penalty of MTU-exceeding certificates. | P2P (10 ms RTT) `InitialCwnd=1`, MTU=1500 B | 858 B vs. 2,588 B | Total Latency (ms) | Finding 1: Baseline Penalty |
| **Exp. 2** | Persistent Loss | Measure the RTO probability under persistent, low-level packet loss. | P2P (10 ms RTT) `RateErrorModel` (PER = 1%) | 858 B vs. 2,588 B | RTO Probability (via RngRun) | Finding 2: Link Unreliability |
| **Exp. 3** | CPU vs. Network | Determine if network latency or CPU computation is the dominant factor. | P2P (10ms RTT) `OQS_SIG_verify()` call | 858 B vs. 2,588 B | $T_{Network}$ (ms) vs. $T_{CPU}$ ($\mu$s) | Finding 1: Baseline Penalty |
| **Exp. 4** | CA Load | Analyze the CA load trade-off for a 'Hybrid-PQC' periodic renewal model. | 100-to-1 Star (UDP) `keyValidity` (60 s vs. 3,600 s) | N/A (1B UDP) | Avg. Load (req/sec) | Finding 4: Deployment Feasibility |
| **Exp. 5** | ISL Multi-Hop | Measure the *amplification* of the 1-RTT penalty over multi-hop ISL routes. | 3-Hop Chain (30 ms RTT) E2E RTT = 30 ms | 858 B vs. 2,588 B | Total Latency (ms) | Finding 3: Dynamic Topology |
| **Exp. 6** | Rain Fade | Evaluate protocol performance under a 'Rain Fade' (burst-loss) event. | P2P (10 ms RTT) `RateErrorModel` (PER = 30%, 50 ms) | 858 B vs. 2,588 B | Total Latency (ms) / RTO | Finding 2: Link Unreliability |
| **Exp. 7** | CPU Bottleneck | Determine the feasibility of a 'Full-PQC' (verify all data) model. | P2P (10 ms RTT) `for` loop (10,000 verifies) | N/A (CPU only) | CPU Bottleneck (ms) | Finding 4: Deployment Feasibility |
| **Exp. 8** | Handover | Determine if the 1-RTT penalty is a recurring cost during handovers. | P2P Handover (A $\rightarrow$ B) New TCP Connection | 858 B vs. 2,588 B | Total Latency (ms) | Finding 3: Dynamic Topology |

*2) Link Unreliability and Robustness:* We then analyze the impact of LEO link unreliability. In Experiment 2 (Persistent Packet Loss), we attach an ns3::RateErrorModel to the satellite's receiver, configured to drop packets with a uniform 1% probability (PER). This measures the increased ReTransmission Timeout (RTO) probability for multi-packet transmissions versus single-packet ones. In Experiment 6 (Dynamic Loss 'Rain Fade'), we model a more realistic, transient burst-loss event. Using Simulator::Schedule, we programmatically change the RateErrorModel's error rate to 30% only during the exact 50ms window of the TCP certificate transmission, testing the protocol's robustness to sudden, severe atmospheric interference.

*3) Dynamic Topology and Mobility:* To evaluate performance at a constellation scale, we model network dynamics. In Experiment 5 (ISL Multi-Hop Latency), we replace the single link with a 3-hop linear chain (CA $\rightarrow$ SatA $\rightarrow$ SatB $\rightarrow$ SatC), each with a 5 ms delay, resulting in a 30ms E2E RTT. This experiment measures how the 1-RTT fragmentation penalty amplifies as the end-to-end RTT increases. In Experiment 8 (Ground Station Handover), we model mobility. A HandoverSatelliteApp connects to Station A, and upon completion, immediately initiates a new connection to Station B. This quantifies the recurring authentication cost, demonstrating that the fragmentation penalty is incurred at every handover.

*4) Deployment Model Feasibility:* Finally, we investigate the practical feasibility of different PQC deployment models. Experiment 7 (Real-Time Stream CPU Bottleneck) tests the Full-PQC model by running the OQS_SIG_verify function in a for loop 10,000 times after the initial authentication. This quantifies the CPU bottleneck of verifying every single data packet. In Experiment 4 (CA Load), we analyze the trade-offs of the hybrid model this justifies. We simulate 100 satellites (numSatellites) periodically contacting a central CA via UDP to renew short-lived keys (e.g., 60s vs. 3,600s), measuring the resulting requests-per-second load on the Certificate Authority.

## IV. Evaluation

This section presents the quantitative results from the eight experiments detailed in our methodology. Our findings validate the central hypothesis of this paper: network-level penalties, driven by certificate size and link dynamics, are the dominant factor in PQC performance, rendering raw CPU computation speed a secondary concern. We group our findings into four key areas.

### A. The 1-RTT Network Penalty

Our baseline experiments (Exp. 1 and 3) were designed to isolate the fundamental costs of network transmission versus CPU computation. As shown in Experiment 1 (Table II), the Falcon protocol, with an MTU-compliant 858 B certificate, established the irreducible minimum latency for this link: 20 ms. This represents 1-RTT (10 ms) for the TCP three-way handshake, followed by 1-RTT (10 ms) for the client's 1-byte request and the server's single-packet data response.

In contrast, the Dilithium protocol (2,588 B) incurred a total latency of 30 ms. Analysis of the ns-3 trace files unequivocally confirms this is a direct, mechanical result of IP fragmentation interacting with TCP's InitialCwnd=1 setting. The server sent the first 1,460-byte fragment, then was forced by the TCP congestion control protocol to wait for the client's ACK—a full 10 ms RTT—before its congestion window opened to allow the transmission of the second 1,128-byte fragment. This fragmentation, therefore, directly imposes a non-negotiable 10 ms, or 1-RTT, network latency penalty.

Experiment 3 (Table III) confirms this network-first conclusion. While Dilithium (25 µs) was 18 µs faster than Falcon (43 µs) in raw CPU verification, this 18 µs computational advantage was rendered statistically negligible, overshadowed by the 10,000 µs (10 ms) network penalty. The final latency of $\approx$20.043 ms for Falcon versus $\approx$30.025 ms for Dilithium proves that any performance analysis focused solely on CPU speed, while ignoring network-layer interactions, is fundamentally flawed for this environment.

TABLE II: Analysis of Fragmentation Latency (Base RTT = 10 ms)

| Metric | Falcon | Dilithium |
|---|---|---|
| Certificate Size | 858 bytes | 2,588 bytes |
| Packet Count | 1 | 2 |
| Network Latency (RTTs) | 2 RTTs (1x Handshake + 1x Data) | 3 RTTs (1x Handshake + 2x Data) |
| **Total Auth. Latency** | **20 ms** | **30 ms** |

TABLE III: Network Latency vs. CPU Verification Latency

| Metric | Falcon (858 B) | Dilithium (2,588 B) |
|---|---|---|
| Network Latency | 20 ms | 30 ms |
| CPU Verify Latency | 43 $\mu$s | 25 $\mu$s |
| Total Latency | $\approx$ 20.043 ms | $\approx$ 30.025 ms |
| **Network Penalty** | **N/A** | **+10,000 $\mu$s** |
| **CPU Advantage** | **N/A** | **-18 $\mu$s** |

### *B. Impact of LEO Link Unreliability*

While the previous section analyzed latency in ideal conditions, this section addresses the critical question of reliability. It might be argued that increasing the TCP Initial Congestion Window would eliminate the 1-RTT latency penalty. However, we demonstrate that fragmentation imposes a fundamental reliability penalty that no windowing optimization can mask. By requiring two packets to be successfully delivered instead of one, Dilithium doubles the probabilistic exposure to packet loss.

Experiment 2 (Table IV) demonstrates this statistically. We introduced a persistent 1% PER. Running the simulation with different random seeds, the single-packet Falcon protocol proved robust, avoiding a catastrophic TCP ReTransmission Timeout (RTO) until RngRun=34. The two-packet Dilithium protocol, however, failed much sooner, triggering a 1,000 ms+ RTO penalty at RngRun=8. This indicates that even with low background noise, the mean time to failure decreases significantly with fragmentation.

TABLE IV: RTO Vulnerability under 1% Persistent Packet Loss

| Metric | Falcon | Dilithium |
|---|---|---|
| Packet Count | 1 | 2 |
| Normal Latency | 20 ms | 30 ms |
| RTO Latency | 1,020 ms | 1,040 ms |
| **First RTO Triggered at** | **RngRun = 34** | **RngRun = 8** |

Experiment 6 (Table V) provides the definitive rebuttal to the 'InitialCwnd' argument. We modeled a severe Rain Fade event (30% PER burst-loss). The results prove that window optimizations are futile in this context. Even if a larger window allows Dilithium's two packets to be transmitted in a single burst, the physical reality remains: two distinct packets must traverse the link. The single-packet Falcon transmission, despite the high loss probability, successfully completed in 20 ms (70% success rate). In contrast, Dilithium faced a compounded failure mode. Although the first packet was transmitted, the second packet was lost. Because the flight size was small, this loss triggered a catastrophic TCP RTO, causing latency to skyrocket to 1,040 ms.

These results confirm that the well-known theoretical fragility of fragmented packets translates into severe operational risks in the LEO environment. Our simulation data transforms the abstract networking principle of fragmentation avoidance into a concrete PQC selection criterion: specifically, selecting Dilithium over Falcon increases the risk of critical handover failures by over 20 percentage points, from 30% to 51%, during common atmospheric disturbances.

This confirms that the penalty of Dilithium is not merely a 10 ms wait for an ACK, but a drastic increase in the probability of service interruption. With two packets each facing a 30% loss chance, the probability of a successful transmission drops to only $0.7 \times 0.7 = 49\%$. Thus, in common LEO weather conditions, Dilithium faces a 51% probability of a 1,000 ms+ delay, a reliability flaw that persists regardless of the TCP Initial Congestion Window size.

TABLE V: Performance under 'Rain Fade' (30% PER Burst Loss)

| Metric | Falcon (1 Packet) | Dilithium (2 Packets) |
|---|---|---|
| Network Latency | 20 ms | 1,040 ms |
| CPU Verify Latency | 41 $\mu$s | 26 $\mu$s |
| **Total Auth. Latency** | **20 ms** | **1,040 ms** |
| **Result** | **Success** | **RTO Failure** |

### *C. Penalty Amplification in Dynamic Topologies*

We then tested the impact of this 1-RTT penalty on the dynamic scale and mobility of a LEO constellation. In Experiment 5 (Table VI), we simulated a 3-hop ISL chain, increasing the E2E RTT from 10 ms to 30 ms. As hypothesized, the penalty amplified in lockstep with the E2E RTT. Falcon (2 RTTs total) completed in 60 ms (30 ms handshake + 30 ms data), while Dilithium (3 RTTs total) completed in 90 ms (30 ms handshake + 30 ms data + 30 ms penalty). The 1-RTT network penalty, now 30 ms, scaled directly with the link's end-to-end latency.

TABLE VI: 1-RTT Penalty Amplification over 3-Hop ISL (E2E RTT = 30 ms)

| Metric | Falcon (858 B) | Dilithium (2,588 B) |
|---|---|---|
| Network Latency (Network RTTs) | 60 ms (2 RTTs) | 90 ms (3 RTTs) |
| CPU Verify Latency | 44 $\mu$s | 32 $\mu$s |
| Total Latency | $\approx$ 60.044 ms | $\approx$ 90.032 ms |
| **Amplified Penalty** | **N/A** | **+30 ms** |

Experiment 8 (Table VII) confirmed this penalty is not a one-time cost but is a recurring tax on mobility. We simulated a satellite handover from Ground Station A to Ground Station B. The Falcon-based protocol authenticated with Station A in 20

ms and, after the handover, authenticated with Station B in an identical 20 ms. The Dilithium-based protocol, however, paid the 10 ms penalty on both connections, requiring 30 ms for Station A and another 30 ms for Station B. This demonstrates that the 10 ms fragmentation penalty is a recurring cost paid at every single handover, severely compromising seamless mobility.

TABLE VII: Recurring 1-RTT Penalty During Handover

| Metric | Falcon | Dilithium |
|---|---|---|
| Initial Auth Latency (to GS-A) | 20 ms | 30 ms |
| Handover Auth Latency (to GS-B) | 20 ms | 30 ms |
| **Recurring 1-RTT Penalty** | **No** | **Yes (+10 ms each time)** |

### D. Justification for the Hybrid Protocol

Finally, our last experiments (Exp. 7 and 4) investigated the only remaining alternative, a Full-PQC model versus our proposed hybrid solution. Experiment 7 tested the viability of using an MTU-compliant algorithm Falcon to verify every data packet in a real-time stream. As shown in Experiment 7 (Table VIII), this Full-PQC model was proven non-viable, creating a debilitating 345ms CPU bottleneck just to verify 10,000 packets. In contrast, the Hybrid PQC model, representing computationally trivial verification, introduced 0 ms of CPU delay. This finding is critical. It proves that even MTU-compliant algorithms like Falcon are only feasible for the initial authentication. They cannot be used for per-packet data verification in a real-time stream. This leaves a hybrid protocol as the only logical path forward.

TABLE VIII: CPU Bottleneck of Full PQC vs. Hybrid Model (10,000 Packet Verifications)

| Metric | Full PQC (Falcon) | Hybrid Model (ECDSA) |
|---|---|---|
| Initial Network Latency | 20 ms | 20 ms |
| CPU Verify Latency | 345 ms | 0 ms |
| **Total Latency** | **365 ms** | **20 ms** |
| **Result** | **CPU Bottleneck** | **No Bottleneck** |

Experiment 4 (Table IX) directly addressed the practicality of this hybrid model, analyzing the load on the Certificate Authority (CA) from frequent key renewals. The results showed a clear, inverse relationship between key validity (v) and CA load. A 1-hour validity (v=3,600 s) generated a trivial load of 0.03 req/sec, while a 1-minute validity (v=60 s) generated a manageable 1.67 req/sec. This confirms that the CA load in a hybrid model is not a fundamental design flaw, but a minor, tunable engineering parameter, validating the protocol's practicality.

TABLE IX: CA Load vs. Key Validity Duration (100 Satellites over 3 Hours)

| Key Validity (v) | Total Requests | Average Load (req/sec) |
|---|---|---|
| 60 s (1 minute) | 18,000 | 1.67 |
| 600 s (10 minutes) | 1,800 | 0.17 |
| 3,600 s (1 hour) | 300 | 0.03 |

## V. Discussion

### A. The Network-Aware PQC Design

The central finding of this paper is that the long-standing focus on CPU-based PQC benchmarks is dangerously misleading when applied to high-latency, constrained networks like LEO. Our results from Experiment 1 and 3 are unequivocal. An 18 µs CPU advantage between Dilithium and Falcon is rendered statistically irrelevant when overshadowed by a 10,000µs (10 ms) network penalty.

This finding necessitates a paradigm shift in how we evaluate PQC for constrained environments. The primary metric for protocol selection should not be raw computational speed, but MTU-compliance. The Falcon certificate (858 B) succeeds not because it is computationally superior, but because its design is network-aware, fitting comfortably within a single packet. Conversely, Dilithium (2,588 B) fails because it is network-agnostic, blindly exceeding the MTU and triggering a cascade of predictable, negative network-layer interactions. This fragmentation, combined with TCP's InitialCwnd=1, is the unmitigated root cause of the 1-RTT penalty. This also explains why SPHINCS+ (8 KB+), requiring 6+ packets, was justifiably excluded from this analysis, as it would be fundamentally non-functional.

### B. Fragmentation as a Critical Reliability Vulnerability

Our findings in Experiments 2 and 6 show that fragmentation is not merely a latency issue; it is a critical reliability and availability vulnerability that persists regardless of TCP window optimizations. In LEO networks, a 1,000 ms+ TCP RTO is not just a lag spike. It can be a catastrophic service failure. It can cause a handover to fail, a real-time data link to drop, or a critical control command to be lost.

The Rain Fade simulation was particularly illuminating in refuting the efficacy of larger Initial Congestion Windows. Even if a larger window allows packets to be sent in a single burst, the physical necessity of transmitting multiple fragments remains. The 51% probability of an RTO for Dilithium, versus 30% for Falcon, is not a small statistical difference. It means the protocol is unreliable more than half the time under adverse, but common, atmospheric conditions. This makes single-packet algorithms like Falcon fundamentally more robust and resilient, a non-negotiable feature for mission-critical satellite infrastructure.

### C. Implications for LEO Architecture and Hybrid Protocols

The dynamic topology results reveal that the 1-RTT penalty is not a one-time cost, but a recurring and amplifying tax on network mobility and scale.

The handover simulation is particularly damning for LEO mobility. The recurring 10 ms penalty incurred at every single handover makes Dilithium-like schemes fundamentally unsuitable for the seamless mobility LEO networks promise.

This latency floor directly compromises real-time service continuity. This problem is further compounded when considering network scale. Our multi-hop ISL simulation demonstrated that the penalty amplifies proportionally with end-to-end latency, scaling from 10 ms to 30 ms as the RTT increased. This suggests that MTU-non-compliant schemes will not scale efficiently for long-haul, cross-constellation data routes, creating a systemic performance bottleneck.

These network-level findings, combined with Experiment 7's discovery of a 345 ms CPU bottleneck in any Full-PQC model, provide the definitive justification for a hybrid approach. Crucially, this bottleneck would be orders of magnitude worse on radiation-hardened space-grade processors, which typically run much slower than our simulation server. This confirms that the Falcon-based hybrid protocol is the only solution that simultaneously avoids both the network-level penalty and the application-level CPU bottleneck.

Furthermore, this hybrid approach presents no contradiction to post-quantum security goals. While 'Harvest Now, Decrypt Later' threatens confidentiality, this is mitigated by the PQC KEM during the handshake. Addressing the concern regarding initial authentication sans per-packet verification, we emphasize that our model adheres to standard secure channel principles. Once the PQC handshake authenticates the endpoints, data integrity is maintained not by heavy asymmetric signatures, but by efficient, quantum-resistant symmetric primitives. Thus, the absence of per-packet PQC signatures represents a necessary engineering optimization rather than a security degradation, ensuring continuous protection with negligible overhead. The manageable CA load demonstrated in Experiment 4 validates this trade-off as practical, efficient, and secure.

### *D. Limitations and Future Work*

We acknowledge the limitations of our paper. First, our network topology was simplified to P2P and linear chain to isolate variables. A real-world LEO constellation is a complex mesh with dynamic routing and higher jitter. We argue this simplification only strengthens our findings, as a more complex topology would likely introduce more jitter and packet re-ordering, further exacerbating the problems of multi-packet transmissions.

A related concern is the exclusive focus on TCP. It might be argued that UDP-based protocols like QUIC or IKEv2 with application-layer fragmentation could mitigate these issues. However, we contend that the reliability penalty described in this paper is a function of physics, not just transport logic. Whether a large certificate is fragmented by IP for TCP or by the application layer for QUIC, it physically requires multiple packets to traverse the link. In a lossy LEO environment, the probability of successful message delivery decreases geometrically with each additional packet, regardless of whether the retransmission is handled by the kernel (TCP) or the user space (UDP). Thus, adopting UDP does not eliminate the fundamental reliability risk posed by MTU-exceeding payloads.

Second, our simulation focused on the PQC certificate as the payload, as it is the largest single object in an authentication handshake and the primary driver of fragmentation. We deliberately excluded Key Encapsulation Mechanisms from the simulation scope. This decision is grounded in the fact that NIST-standardized KEMs, such as ML-KEM, are relatively compact. For instance, Kyber-512 has a ciphertext size of 768 bytes and a public key size of 800 bytes, both of which fit comfortably within the 1,500-byte MTU. Thus, unlike the large Dilithium signatures ($\approx$2.5 KB) analyzed in this work, KEMs do not inherently trigger the fragmentation penalty. We are therefore confident that the signature artifact remains the dominant network bottleneck.

Future work should involve implementing a full PQC KEM-Signature handshake within ns-3 to validate these findings in a full-stack TLS 1.3 model. Furthermore, extending this simulation to a more complex LEO network simulation framework, such as ns3-leo or integration with STK, would be a valuable next step.

## VI. Conclusion

LEO satellite networks are essential for future global communications, but they face an urgent need to transition to PQC to defend against quantum threats. This transition is uniquely challenging in the LEO environment, which is defined by strict latency budgets, a 1,500-byte MTU, unreliable links, and frequent handovers. This paper addressed this challenge by designing and executing a high-fidelity simulation framework, integrating ns-3 and liboqs, to analyze the practical network performance of PQC authentication.

We have quantitatively demonstrated that network performance, not raw CPU computation, is the dominant factor for PQC authentication in LEO networks. We identified that PQC certificates exceeding the 1,500-byte MTU, such as Dilithium (ML-DSA-44) at 2,588 bytes, incur IP fragmentation. While conservative TCP settings expose this as a baseline 1-RTT latency penalty, our analysis reveals that fragmentation introduces a fundamental structural vulnerability that extends far beyond simple delays.

This vulnerability amplifies proportionally over multi-hop ISL routes and recurs at every ground station handover, compromising seamless mobility. Crucially, we demonstrate that this is a reliability penalty that persists regardless of TCP window optimizations. In a simulated Rain Fade event, the fragmented Dilithium exchange faced a 51% probability of a catastrophic 1,000 ms+ TCP RTO, compared to 30% for the single-packet Falcon. This confirms that selecting non-MTU-compliant algorithms increases the risk of critical service failure by over 20 percentage points during atmospheric disturbances, rendering CPU-side advantages negligible. This leads to our topline performance takeaway. The use of certificates exceeding the MTU imposes a latency penalty ranging from 50% in baseline scenarios to over 5,000% in common lossy conditions, rendering them unsuitable for critical LEO infrastructure. Finally, we proved that a Full-PQC model, which verifies every data packet, is infeasible, creating a 345

ms CPU bottleneck and confirming the necessity of a hybrid protocol for real-time data streams.

Based on this evidence, we conclude that the Falcon-based hybrid protocol is the only solution that is simultaneously practical, robust, and efficient. Falcon's compact 858-byte certificate is MTU-compliant, which natively avoids the entire class of network-level penalties, the 1-RTT latency, the RTO vulnerability, and the recurring handover tax. Simultaneously, the hybrid model avoids the application-level CPU bottleneck. Consequently, our findings serve as an urgent design guideline, future LEO security standards must prioritize MTU-compliant algorithms like Falcon to ensure the resilience of next-generation critical infrastructure.

## ACKNOWLEDGMENT

This work was supported by Korea Research Institute for defense Technology planning and advancement (KRIT) grant funded by the Korea government (Defense Acquisition Program Administration), (KRIT-CT-24- 001, Defense Space Security Research Lab, 2025)

## REFERENCES

[1] L. Chen, D. Moody, and Y. Liu, "Nist post-quantum cryptography standardization," *Transition*, vol. 800, no. 131A, p. 164, 2017.

[2] T. Schubert, L. Wolf, and U. Kulau, "ns-3-leo: Evaluation tool for satellite swarm communication protocols," *IEEE access*, vol. 10, pp. 11 527–11 537, 2022.

[3] D. Sikeridis, P. Kampanakis, and M. Devetsikiotis, "Post-quantum authentication in tls 1.3: A performance study," *Cryptology ePrint Archive*, 2020.

[4] S. and, "Post-quantum cryptography — csrc," 2022. [Online]. Available: https://csrc.nist.gov/Projects/post-quantum-cryptography/selected-algorithms

[5] K. Basu, D. Soni, M. Nabeel, and R. Karri, "Nist post-quantum cryptography-a hardware evaluation study," *Cryptology ePrint Archive*, 2019.

[6] T. Liu, G. Ramachandran, and R. Jurdak, "Post-quantum cryptography for internet of things: a survey on performance and optimization," *arXiv preprint arXiv:2401.17538*, 2024.

[7] C. Paquin, D. Stebila, and G. Tamvada, "Benchmarking post-quantum cryptography in tls," in *International Conference on Post-Quantum Cryptography*. Springer, 2020, pp. 72–91.

[8] S. Hoque, A. Aydeger, E. Zeydan, and M. Liyanage, "Analysis of post-quantum cryptography in user equipment in 5g and beyond," in *2025 IEEE 50th Conference on Local Computer Networks (LCN)*, 2025, pp. 1–9.

[9] Y. Liu, L. Ni, and M. Peng, "A secure and efficient authentication protocol for satellite-terrestrial networks," *IEEE Internet of Things Journal*, vol. 10, no. 7, pp. 5810–5822, 2022.

[10] C. Poomagal and G. Sathish Kumar, "Ecc based lightweight secure message conveyance protocol for satellite communication in internet of vehicles (iov)," *Wireless Personal Communications*, vol. 113, no. 2, pp. 1359–1377, 2020.

[11] N. Pandey, D. Kumar, and H. Palwal, "Simulation based comparative study on eigrp/is–is and ospf/is–is," *International Journal of Engineering Research and General Science*, vol. 3, no. 2, pp. 204–214, 2015.

[12] W. Guo, J. Xu, Y. Pei, L. Yin, C. Jiang, and N. Ge, "A distributed collaborative entrance defense framework against ddos attacks on satellite internet," *IEEE Internet of Things Journal*, vol. 9, no. 17, pp. 15 497–15 510, 2022.

[13] S.-J. Chen and Y.-H. Tsai, "Quantum-safe networks for 6g an integrated survey on pqc, qkd, and satellite qkd with future perspectives," *Computing&AI Connect*, vol. 2, no. 1, pp. 1–10, 2025.

[14] P. Papadimitriou and V. Tsaoussidis, "On tcp performance over asymmetric satellite links with real-time constraints," *Computer Communications*, vol. 30, no. 7, pp. 1451–1465, 2007, wired/Wireless Internet Communications. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S014036640700028X