

Article

Evaluating Post-Quantum Cryptography in IoT Networks: Communication, Fragmentation, and Reliability

Eric Sakk , Guobin Xu , Jianzhou Mao and Shuangbao Wang

Department of Computer Science, Morgan State University, Baltimore, MD 21251, USA;
guobin.xu@morgan.edu (G.X.); jianzhou.mao@morgan.edu (J.M.); shuangbao.wang@morgan.edu (S.W.)

* Correspondence: eric.sakk@morgan.edu

Abstract

Post-quantum cryptographic (PQC) algorithms are being developed to guard against quantum-computing attacks, but their behavior in constrained Internet of Things (IoT) environments remains an important topic of discussion. In this work, we study the impact of deploying PQC protocols in IoT networks using the Open Quantum Safe (liboqs) framework. In particular, key encapsulation and digital signature schemes are evaluated in terms of their computational performance, communication costs, and energy consumption. Our results indicate that although PQC operations can be completed in microseconds using general-purpose processors, substantially larger key and ciphertext sizes introduce significant communication overhead. When mapped to common IoT protocols such as Bluetooth Low Energy (BLE), IEEE 802.15.4 (Zigbee), and LoRa, these larger payloads must be divided into multiple packets. In low-payload LoRa networks, for example, ML-KEM handshakes can require up to 62 packets. This level of fragmentation increases latency and energy consumption, thus potentially affecting reliability. Furthermore, when packet delivery probabilities approaching 99% are achieved, handshake success rates can drop to values approaching 50%. These results suggest that communication metrics, rather than computational performance, pose key challenges to PQC deployment in constrained IoT settings.

Keywords: post-quantum cryptography; Internet of Things; ML-KEM; packet fragmentation; communication overhead

1. Introduction

Progress in quantum computing has raised concerns regarding the security of classical public-key cryptographic systems such as RSA and elliptic curve cryptography (ECC), particularly in light of Shor's algorithm [1]. As a result, post-quantum cryptographic (PQC) algorithms have been developed to address these threats. Recent efforts led by the National Institute of Standards and Technology (NIST) have resulted in the selection of schemes such as ML-KEM (Kyber) and ML-DSA (Dilithium) [2]. At the same time, the rapid expansion of the Internet of Things (IoT) has led to billions of interconnected devices operating under tight resource constraints. These devices often have limited computational power, memory, and energy availability. In addition, widely used IoT communication protocols, including Bluetooth Low Energy (BLE), IEEE 802.15.4 (Zigbee), and LoRa, impose strict limits on packet size and data rate. Therefore, communication efficiency is a central concern in system design [3].



Academic Editor: Francesco
Buccafurri

Received: 4 May 2026

Revised: 4 June 2026

Accepted: 8 June 2026

Published: 10 June 2026

Copyright: © 2026 by the authors.

Licensee MDPI, Basel, Switzerland.

This article is an open access article distributed under the terms and

conditions of the [Creative Commons](https://creativecommons.org/licenses/by/4.0/)

[Attribution \(CC BY\)](https://creativecommons.org/licenses/by/4.0/) license.

The feasibility of deploying PQC on constrained platforms has been considered in a number of studies. For example, it has been shown that several PQC schemes can be implemented on embedded devices with reasonable computational overhead [4]. Other work has focused on integrating PQC into existing protocols such as Transport Layer Security (TLS), demonstrating that post-quantum key exchange is feasible, albeit with increased communication cost [5]. More recent efforts have extended these evaluations to IoT-class devices and networked environments, confirming that PQC algorithms are computationally practical but introduce larger keys and ciphertexts that increase communication overhead [5].

Despite these advances, much of the existing literature emphasizes computational performance and protocol integration, implicitly assuming unhindered network communications. However, comparatively less attention has been paid to how PQC interacts with communication constraints inherent in IoT networks. More specifically, the effects of limited MTU sizes, packet fragmentation, and reliability have not been systematically quantified. This gap in the literature is important because PQC schemes produce significantly larger messages than traditional approaches. Therefore, while prior work has established feasibility at the algorithm and protocol levels, packet-level system effects have not been fully captured.

In this work, we examine the deployment of PQC protocols in IoT networks from a system-level perspective. Using the Open Quantum Safe (liboqs) framework, we evaluate key encapsulation and digital signature schemes. Additionally, we analyze their impact on communication overhead, latency, energy consumption, and packet fragmentation across several IoT protocols. By explicitly incorporating network constraints into the analysis, we offer a clearer picture of the challenges that can arise when transporting PQC from theory to practice in real-world IoT systems.

The remainder of this paper is organized as follows. Section 2 reviews related work on PQC and IoT security. Section 3 describes the methodology, including benchmarking, network modeling, fragmentation, and reliability analysis. Section 4 presents the experimental results. Section 5 provides deployment guidelines for PQC algorithm selection in IoT networks. Finally, Section 6 concludes the paper and discusses future work.

2. Related Work

Interest in post-quantum cryptography (PQC) has increased significantly due to potential threats posed by quantum computers [6]. Early efforts to address security issues focused on identifying and analyzing quantum-resistant cryptographic primitives. Beyond individual algorithms, quantum-safe approaches, including both quantum key distribution (QKD) and PQC, have been explored in practical settings [7].

Along the same timeline, the expansion of the Internet of Things (IoT) has led to large-scale distributed systems operating under resource and communication constraints. As a result, the architectural foundations of IoT and key challenges such as scalability, heterogeneity, and limited device capabilities have been studied [8]. Security remains a concern in these environments, with previous work addressing vulnerabilities arising from communication constraints and deployment conditions [9].

More recent studies have considered how these challenges evolve in the presence of quantum-capable adversaries, particularly for constrained IoT systems [10]. A considerable amount of work has examined the performance of PQC algorithms on constrained and embedded platforms. For example, it has been demonstrated that several PQC schemes can be implemented on resource-limited devices with manageable computational overhead [4]. These results suggest that modern lattice-based algorithms such as ML-KEM and ML-DSA are viable from a computational standpoint, even in embedded environments. Recent work has continued to improve the efficiency of PQC implementations on constrained

platforms. In particular, optimized implementations of lattice-based schemes such as Kyber and Dilithium on Cortex-M microcontrollers demonstrate that these algorithms can achieve practical performance even on resource-limited devices [11]. Additionally, improvements in the microarchitectural performance of PQC implementations on embedded platforms have been offered. For example, ML-KEM and ML-DSA implementations for the Arm Cortex-M7 have been optimized, achieving substantial speedups through improved instruction scheduling and pipeline utilization [12]. Such results continue to confirm that PQC algorithms can achieve high computational efficiency on constrained hardware. However, these studies primarily focus on implementation-level efficiency, including execution time and memory usage, and do not address the communication-level effects that arise when these algorithms are deployed in constrained IoT networks. In particular, the impact of large key and ciphertext sizes on packet fragmentation, transmission overhead, and end-to-end reliability remains largely unexplored.

In addition to standardized cryptographic approaches, lightweight security mechanisms based on chaotic maps and nonlinear dynamical systems have also been explored for constrained environments, particularly in multimedia and image-encryption applications [13,14]. Such approaches have demonstrated potential advantages in computational simplicity and lightweight implementation, although currently they address a different problem domain than standardized post-quantum cryptography.

There has also been interest in integrating PQC into existing communication protocols. Explorations into post-quantum key exchange within the Transport Layer Security (TLS) protocol have shown that such integration is feasible but introduces additional communication overhead [5]. More recent work has extended these evaluations to networked and IoT-like environments. Trade-offs identified between security, performance, and communication cost when deploying PQC protocols remain a vital area of research. However, much of the existing literature emphasizes computational performance and protocol integration. Less attention has been given to how PQC interacts with the communication constraints inherent to IoT networks. In particular, the combined effects of limited MTU sizes, packet fragmentation, and transmission reliability have not been systematically quantified. Some prior work has examined reliability in IoT systems more generally [15], but without explicitly considering the impact of PQC-induced message sizes. This work seeks to provide a realistic evaluation of PQC deployment within constrained IoT networks. In order to do this, we focus specifically on communication overhead, packet fragmentation, and handshake reliability. By combining cryptographic benchmarking with network analysis, we offer new insights into the challenges of deploying PQC in practical IoT environments.

3. Methodology

Our goal is to present a system-level evaluation of PQC algorithms in IoT environments. In this section, we define our methodology that combines cryptographic benchmarking with network-aware modeling to analyze the impact of PQC on communication overhead, latency, energy consumption, packet fragmentation, and reliability. By combining measurements of computational performance with communication and reliability analysis, our approach enables a comprehensive assessment of the feasibility of applying PQC protocols in constrained systems.

3.1. Experimental Framework

All experiments were conducted using the Open Quantum Safe (liboqs) library, which provides implementations of standardized and candidate PQC algorithms. The evaluation was performed in a Python-based environment. This framework supports both key encap-

sulation mechanisms (KEMs) and digital signature algorithms. The primary algorithms evaluated in this study include:

- ML-KEM (Kyber) with parameter sets 512, 768, and 1024;
- ML-DSA (Dilithium);
- Falcon;
- SPHINCS+.

These algorithms were selected to represent the range of NIST-standardized PQC schemes and to capture trade-offs between performance, key size, and security level.

3.2. Performance Benchmarking

The computational performance of each algorithm was evaluated by measuring the execution time of key cryptographic operations. For KEM algorithms, the following operations were benchmarked:

- Key generation;
- Encapsulation;
- Decapsulation.

For digital signature algorithms, the following operations were evaluated:

- Key generation;
- Signing;
- Verification.

Each operation was executed multiple times. For timing-variability experiments, 500 independent measurements were collected for each algorithm. Mean execution times were recorded to reduce the impact of system-level variability. These measurements provide a baseline for comparing computational cost across algorithms.

3.3. Communication Overhead Analysis

To evaluate communication overhead, the sizes of public keys, ciphertexts, and signatures were recorded for each algorithm. For KEM-based handshakes, the total number of bytes exchanged between communicating parties was calculated as the sum of the public key and ciphertext sizes. This analysis enables comparison between PQC and classical cryptographic approaches, highlighting the increase in message size introduced by post-quantum schemes.

3.4. IoT Network Modeling

To assess the impact of PQC on constrained networks, representative IoT communication protocols were modeled, including:

- Bluetooth Low Energy (BLE) with a 247-byte MTU;
- IEEE 802.15.4 (Zigbee) with a 127-byte frame size;
- LoRa with payload sizes of 51 bytes and 222 bytes.

These protocols were selected to represent a range of low-power, low-bandwidth communication environments commonly used in IoT deployments.

3.5. Packet Fragmentation Analysis

Packet fragmentation was modeled by dividing the total handshake message size by the maximum payload size supported by each protocol. The number of packets required to transmit a complete handshake was computed as:

$$N_{\text{packets}} = \left\lceil \frac{S_{\text{handshake}}}{S_{\text{MTU}}} \right\rceil \quad (1)$$

where $S_{\text{handshake}}$ is the total number of bytes exchanged and S_{MTU} is the maximum payload size of the network. Equation (1) enables a direct comparison of fragmentation across different PQC algorithms and communication protocols. Additionally, this packet-count model serves as the basis for characterizing transmission delay growth and handshake reliability attenuation due to larger message sizes when PQC protocols are applied.

3.6. Transmission Latency and Energy Model

Transmission latency was estimated based on the total number of bytes exchanged and the effective data rate of each communication protocol. The corresponding transmission latency model is derived as follows. Let $S_{\text{handshake}}$ denote the total number of bytes exchanged during a cryptographic handshake, and let R denote the effective communication data rate in bits per second. The ideal payload transmission time is given by

$$T_{\text{payload}} = \frac{8S_{\text{handshake}}}{R} \quad (2)$$

where the factor of 8 converts bytes into bits. However, packet fragmentation introduces additional overhead because each packet incurs protocol-specific processing costs such as packet headers, acknowledgements, and channel access delays. Let T_{oh} denote the average overhead associated with transmitting a single packet. The total transmission latency can therefore be approximated as

$$T_{\text{total}} = N_{\text{packets}} T_{\text{oh}} + \frac{8S_{\text{handshake}}}{R} \quad (3)$$

Substituting Equation (1) into Equation (3) yields

$$T_{\text{total}} = \left\lceil \frac{S_{\text{handshake}}}{S_{\text{MTU}}} \right\rceil T_{\text{oh}} + \frac{8S_{\text{handshake}}}{R}. \quad (4)$$

Equation (4) demonstrates that transmission latency increases due to both the total number of transmitted bytes and the packet fragmentation overhead. For a fixed protocol payload size S_{MTU} , larger PQC handshake messages increase N_{packets} , which, in turn, increases both transmission overhead and overall latency (particularly in constrained networks with small MTU values).

In this work, the value of p is treated as an abstract per-packet success probability rather than a protocol-specific constant. The purpose of this model is to illustrate the cumulative impact of packet fragmentation on handshake reliability. In practical deployments, packet success probability may depend on factors such as payload size, channel conditions, interference, retransmission mechanisms, spreading factor (LoRa), and connection parameters (BLE). Consequently, the reliability model presented here should be interpreted as a first-order analytical approximation rather than a detailed protocol-specific channel model.

Finally, energy consumption was approximated by modeling the energy required to transmit each packet and assuming a fixed energy cost per transmitted byte. This simplified model is intended to provide a relative comparison rather than an exact hardware-specific energy estimate. In this way, we can provide an estimate of the relative energy impact of PQC handshakes as compared to classical cryptographic protocols in constrained environments.

3.7. Handshake Reliability Analysis

To evaluate communication reliability, a probabilistic model was used to estimate the likelihood of successful handshake completion. Assuming an independent packet success

probability p , the probability of successfully transmitting all packets in a handshake is given by:

$$P_{\text{success}} = p^{N_{\text{packets}}} \quad (5)$$

This model captures the cumulative effect of packet loss across fragmented transmissions and provides insight into how increased message size impacts overall communication reliability. Equation (5) follows directly from the assumption of independent packet delivery. If the probability of successfully transmitting a single packet is p , then the probability of successfully transmitting all N_{packets} packets required for a handshake is given by the product of the individual packet success probabilities:

$$P_{\text{success}} = \underbrace{p \times p \times \dots \times p}_{N_{\text{packets}} \text{ terms}} = p^{N_{\text{packets}}}. \quad (6)$$

Substituting Equation (1) into Equation (5) yields

$$P_{\text{success}} = p^{\left\lceil \frac{S_{\text{handshake}}}{S_{\text{MTU}}} \right\rceil}. \quad (7)$$

Equation (7) demonstrates that handshake reliability decreases as the number of fragmented packets increases. For a fixed packet success probability p , larger PQC handshake messages increase N_{packets} , thereby reducing the probability that all packets required for a successful handshake are delivered. Consequently, larger PQC message sizes can lead to lower overall handshake success probabilities, even when the probability of successfully delivering an individual packet remains high.

4. Results

4.1. KEM Computational Performance

We first evaluate the computational performance of key encapsulation mechanisms (KEMs), including ML-KEM-512, ML-KEM-768, and ML-KEM-1024. The execution time of key generation, encapsulation, and decapsulation operations was measured using the experimental framework described in Section 3. Our results are summarized in Table 1.

Table 1. Computational performance of ML-KEM algorithms.

Algorithm	KeyGen (s)	Encap (s)	Decap (s)
ML-KEM-512	0.000049	0.000020	0.000017
ML-KEM-768	0.000024	0.000025	0.000026
ML-KEM-1024	0.000032	0.000031	0.000034

These results indicate that all ML-KEM variants evaluated in this test achieve low execution times with operations completing on the order of tens of microseconds. Although the computational cost increases slightly as a function of the security level, the variation across parameter sets remains tolerable. ML-KEM-512 exhibits the fastest encapsulation and decapsulation times, while the execution time of ML-KEM-1024 increases due to larger parameter sizes. Even at the highest security level, overall latency remains acceptable.

To further illustrate this behavior, the decapsulation times are visualized in Figure 1.

As shown in Figure 1, decapsulation time increases with the security level of the ML-KEM parameter set. Despite this trend, all operations remain within the microsecond range, confirming that PQC computational overhead is minimal. These findings indicate that computation is unlikely to be the primary limiting factor in PQC deployment. Given this observation, the impact of communication overhead and network constraints must

also be examined to understand the performance of PQC protocols when deployed in IoT environments.

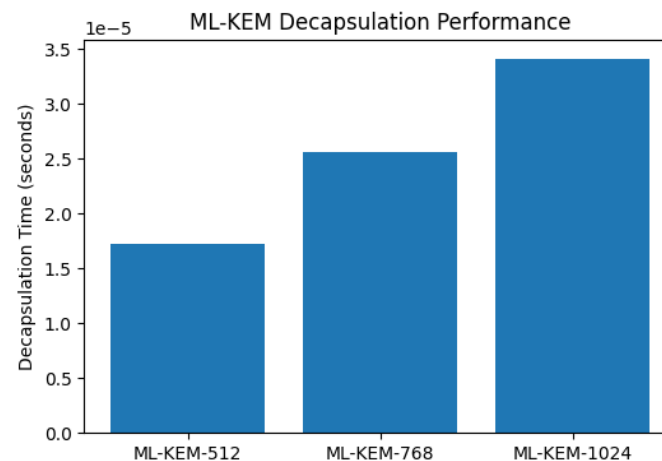


Figure 1. Decapsulation time for ML-KEM parameter sets.

Timing Variability Analysis

To assess the consistency of PQC operations and provide a proxy for potential side-channel leakage, we analyzed the variability of encapsulation execution time across multiple runs. While this study does not perform a full side-channel attack analysis, timing variability can serve as an indicator of implementation stability. The results are summarized in Table 2.

For each ML-KEM parameter set, timing statistics were computed from 500 independent encapsulation measurements. The reported mean, standard deviation, minimum, and maximum values were calculated directly from the collected samples without outlier removal or filtering. These statistics are intended to provide a descriptive characterization of runtime variability rather than a formal statistical significance analysis.

Table 2. Timing variability of ML-KEM encapsulation operations.

Algorithm	Mean (s)	Std. Dev. (s)	Min (s)	Max (s)
ML-KEM-512	0.000021	0.000020	0.000017	0.000375
ML-KEM-768	0.000025	0.000002	0.000024	0.000048
ML-KEM-1024	0.000038	0.000009	0.000033	0.000136

These results show that ML-KEM operations are completed with relatively low variability across repeated executions. ML-KEM-768 exhibits the most consistent timing, while ML-KEM-512 shows greater spread due to occasional high-latency measurements (as indicated by the maximum execution time). While these results do not indicate significant variability, they do highlight the need for caution when considering the potential for side-channel leakage. It should be noted that this analysis only provides an approximate indication of timing behavior. A more complete treatment of side-channel resistance would require a set of hardware-level measurements and carefully controlled experimental conditions.

4.2. KEM Communication Overhead

To evaluate the communication implications of deploying PQC, the sizes of public keys, ciphertexts, and shared secrets for each ML-KEM parameter set were measured. These measurements identify the total amount of data exchanged during a key encapsulation handshake and, thus, directly influence the transmission throughput in constrained networks. These results are summarized in Table 3.

Table 3. Communication overhead of ML-KEM algorithms.

Algorithm	Public Key (Bytes)	Ciphertext (Bytes)	Shared Secret (Bytes)
ML-KEM-512	800	768	32
ML-KEM-768	1184	1088	32
ML-KEM-1024	1568	1568	32

The results indicate that ML-KEM algorithms introduce substantial communication overhead when compared with classical cryptographic systems. Public key and ciphertext sizes range from about 800 bytes to over 1500 bytes, while the shared secret remains constant at 32 bytes. Hence, there is a noticeable increase in message size over elliptic curve-based key exchange implementations (which typically require only tens of bytes). As a result, PQC-based handshakes involve the transmission of kilobyte-scale payloads. Such message sizes have the potential to introduce considerable pressure on constrained communication networks. Observe that this increase in size is a function of the security level where, for example, ML-KEM-1024 requires nearly double the overhead of ML-KEM-512. This result highlights an important trade-off between security and communication efficiency in bandwidth-limited IoT environments. Therefore, although we have confirmed that PQC algorithms are computationally efficient, communication requirements could significantly impact system performance. In the following section, we extend this analysis by evaluating total handshake cost and its impact on transmission latency in constrained networks.

4.3. Handshake Cost and Transmission Analysis

Building on the communication overhead analysis in Section 4.2, we now evaluate the total cost of a complete ML-KEM handshake, including both data exchange and transmission latency. For each parameter set, the total number of bytes exchanged is computed as the sum of the public key and ciphertext sizes. Transmission times are estimated based on the modeled network data rate, thereby providing a system-level view of PQC deployment cost. These results are summarized in Table 4.

Table 4. Handshake performance and transmission cost of ML-KEM algorithms.

Algorithm	Handshake Time (s)	Bytes Exchanged	Secrets Match	Transmit Time (ms)
ML-KEM-512	0.000618	1568	True	50.176
ML-KEM-768	0.000152	2272	True	72.704
ML-KEM-1024	0.000167	3136	True	100.352

Because the handshake timing values were obtained from single end-to-end executions in the Python/liboqs v0.15.0 environment, they should be interpreted as illustrative measurements rather than statistically averaged cryptographic benchmarks. The main purpose of Table 4 is to compare the communication cost, bytes exchanged, and resulting transmission time across ML-KEM parameter sets. Notice that while the measured cryptographic handshake times remain in the microsecond-to-submillisecond range, the total data exchanged during the process is significant. Additionally, ML-KEM-1024 requires over 3 KB of data exchange, resulting in substantially greater transmission overhead than ML-KEM-512. Although computation requirements remain negligible, communication overhead increases directly with message size and quickly dominates total handshake latency. As a consequence, higher-security parameter sets introduce a tangible performance penalty in bandwidth-limited environments. Evaluating PQC performance based solely on computational benchmarks is insufficient. Network metrics such as transmission la-

tency, packet fragmentation, and communication reliability must also be considered in IoT deployments where communication resources are constrained.

4.4. Classical vs. PQC Communication Comparison

To better understand the communication overhead introduced by post-quantum cryptography, we compare ML-KEM handshake sizes with representative classical cryptographic schemes, including RSA-2048 and ECDH-P256. The total number of bytes exchanged during each handshake is summarized in Table 5.

Table 5. Comparison of handshake communication cost between classical and PQC algorithms.

Algorithm	Bytes Exchanged
RSA-2048	256
ECDH-P256	64
ML-KEM-512	1568
ML-KEM-768	2272
ML-KEM-1024	3136

To visualize the magnitude of this difference, the handshake sizes are shown in Figure 2.

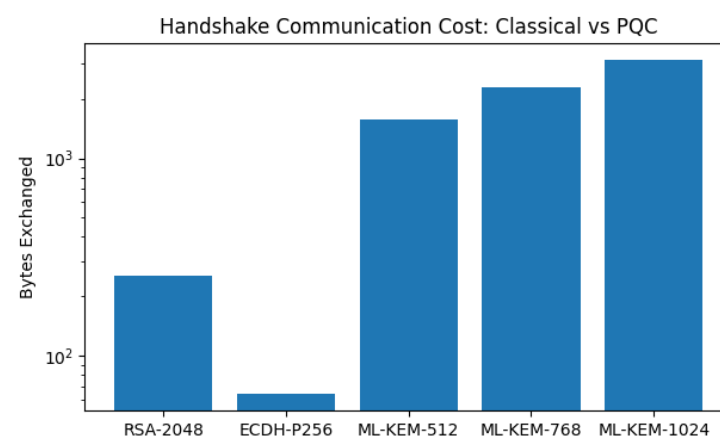


Figure 2. Handshake communication cost for classical cryptographic schemes and ML-KEM parameter sets (logarithmic scale).

These results reveal that ECDH-P256 requires only 64 bytes and RSA-2048 approximately 256 bytes. Meanwhile, ML-KEM-based handshakes range from over 1.5 KB to more than 3 KB. The logarithmic scale used in Figure 2 highlights the magnitude increase in communication overhead due to PQC implementations. Even the lowest-security PQC configuration (ML-KEM-512) requires over six times the data of RSA-2048 and nearly twenty-five times that of ECDH-P256. While these byte-level comparisons highlight the increase in communication size, the impact at the system level is more clearly observable when considering packetization in constrained IoT networks. Table 6 compares the number of packets required for PQC and classical ECDH handshakes across various representative protocols.

This table further illustrates the increase in the communication burden beyond what is apparent from byte-level comparisons alone. Depending upon the protocol and algorithm applied, PQC handshakes can require from 7× to over 30× more packets than classical ECDH. More packets will result in higher transmission latency, increased energy consumption, and a greater probability of packet loss. Furthermore, such effects compound as the number of required packets increases, particularly in low-MTU networks such as LoRa. These findings highlight a fundamental challenge in PQC deployment. Post-quantum meth-

ods provide strong security guarantees; however, they also impose significantly higher communication demands at both the byte and packet levels.

Table 6. Packet overhead of PQC handshakes relative to classical ECDH.

Protocol	Algorithm	PQC Packets	ECDH Packets	Multiplier
BLE (247 B)	ML-KEM-512	7	1	7.0
BLE (247 B)	ML-KEM-768	10	1	10.0
BLE (247 B)	ML-KEM-1024	13	1	13.0
Zigbee (127 B)	ML-KEM-512	13	1	13.0
Zigbee (127 B)	ML-KEM-768	18	1	18.0
Zigbee (127 B)	ML-KEM-1024	25	1	25.0
LoRa (51 B)	ML-KEM-512	31	2	15.5
LoRa (51 B)	ML-KEM-768	45	2	22.5
LoRa (51 B)	ML-KEM-1024	62	2	31.0
LoRa (222 B)	ML-KEM-512	8	1	8.0
LoRa (222 B)	ML-KEM-768	11	1	11.0
LoRa (222 B)	ML-KEM-1024	15	1	15.0

4.5. Energy Consumption Analysis

In addition to transmission latency, communication overhead directly impacts energy consumption in constrained IoT devices. To quantify this effect, we estimate the energy cost of ML-KEM handshakes based on the total transmission time and a simplified radio energy model. Specifically, energy consumption is computed as $E = P \times t$, where P is the radio power (assumed to be 100 mW) and t is the transmission time. These results are summarized in Table 7.

Table 7. Estimated energy cost of ML-KEM handshakes.

Algorithm	Handshake Time (s)	Bytes Exchanged	Transmit Time (ms)	Energy (mJ)	Secrets Match
ML-KEM-512	0.000618	1568	50.176	5.0176	True
ML-KEM-768	0.000152	2272	72.704	7.2704	True
ML-KEM-1024	0.000167	3136	100.352	10.0352	True

These results confirm our expectation that energy consumption should increase proportionally with the amount of data transmitted. As the security level increases from ML-KEM-512 to ML-KEM-1024, the total energy cost approximately doubles as a result of the corresponding increase in communication overhead. Clearly, larger message sizes will impose an energy penalty on resource-constrained devices. In battery-powered environments, increased energy consumption could reduce device lifetime or necessitate more frequent recharging. Once again, we see that it is communication, rather than computation, that influences PQC performance in constrained settings.

4.6. Digital Signature Performance

In addition to key encapsulation mechanisms, we evaluate the performance of representative post-quantum digital signature schemes, including ML-DSA-44, Falcon-512, and SPHINCS+-SHA2-128f-simple. For each algorithm, we measure key generation, signing, and verification times, as well as the resulting signature size. These results are summarized in Table 8.

Table 8. Performance of post-quantum digital signature algorithms.

Algorithm	KeyGen (s)	Sign (s)	Verify (s)	Signature Size (Bytes)
ML-DSA-44	0.000089	0.000108	0.000040	2420
Falcon-512	0.010253	0.000376	0.000069	654
SPHINCS+-SHA2-128f-simple	0.000445	0.010285	0.001203	17,088

Observe that ML-DSA-44 achieves fast key generation, signing, and verification but also produces large signatures of about 2.4 KB. Falcon-512 yields smaller signatures (approximately 654 bytes) but incurs a higher key generation cost. SPHINCS+-SHA2-128f-simple exhibits a relatively small key generation cost but induces higher signing and verification times. In addition, its signature size exceeds 17 KB, making it considerably larger than both ML-DSA and Falcon.

These differences reveal the associated trade-offs between computational performance and communication overhead. In constrained IoT environments, signature size could become a critical factor, as large signatures increase transmission latency, energy consumption, and the likelihood of packet fragmentation. Obviously, large signature sizes further exacerbate communication overhead, reinforcing the importance of communication-aware PQC deployment. These results continue to reinforce our broader conclusion that communication overhead plays a central role in the practical deployment of PQC-based systems.

4.7. Cryptographic Agility

To evaluate the relative operational cost of different PQC signature schemes, we consider the total execution time required for a complete cryptographic operation, including key generation, signing, and verification. This metric provides a simple measure of cryptographic agility, and it indicates how efficiently a system can perform end-to-end cryptographic tasks. These results are summarized in Table 9.

Table 9. Total execution time for PQC signature algorithms (cryptographic agility).

Algorithm	Full Operation Time (s)
ML-DSA-44	0.001371
Falcon-512	0.020837
SPHINCS+-SHA2-128f-simple	0.011896

In this experiment, ML-DSA-44 achieves the lowest total execution time, making it the most computationally efficient among the evaluated schemes. Falcon-512 exhibits the highest overall cost. This is mainly due to its expensive key generation phase. Lastly, SPHINCS+-SHA2-128f-simple falls between the two. These results highlight the importance of algorithm selection in systems requiring cryptographic agility. However, computational differences still remain secondary to communication costs. Trade-offs between computational cost and communication overhead must be carefully considered when selecting PQC schemes.

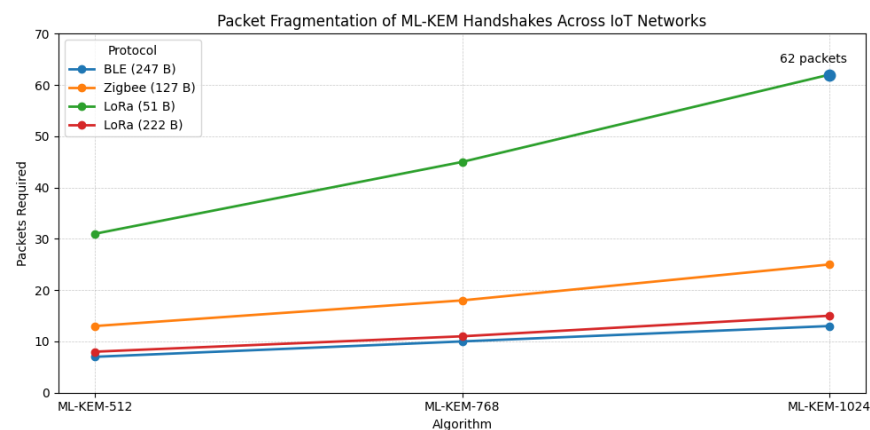
4.8. Packet Fragmentation in IoT Networks

To evaluate the impact of PQC communication overhead on constrained networks, we analyze the number of packets required to transmit ML-KEM handshake data across representative IoT protocols. Specifically, we consider Bluetooth Low Energy (BLE), IEEE 802.15.4 (Zigbee), and LoRa networks with varying maximum transmission unit (MTU) sizes. The number of packets required for each configuration is summarized in Table 10.

Table 10. Packet fragmentation of ML-KEM handshakes across IoT protocols.

Protocol	MTU (Bytes)	Algorithm	Handshake Size (Bytes)	Packets Required
BLE (247 B MTU)	247	ML-KEM-512	1568	7
BLE (247 B MTU)	247	ML-KEM-768	2272	10
BLE (247 B MTU)	247	ML-KEM-1024	3136	13
IEEE 802.15.4 (127 B)	127	ML-KEM-512	1568	13
IEEE 802.15.4 (127 B)	127	ML-KEM-768	2272	18
IEEE 802.15.4 (127 B)	127	ML-KEM-1024	3136	25
LoRa (51 B payload)	51	ML-KEM-512	1568	31
LoRa (51 B payload)	51	ML-KEM-768	2272	45
LoRa (51 B payload)	51	ML-KEM-1024	3136	62
LoRa (222 B payload)	222	ML-KEM-512	1568	8
LoRa (222 B payload)	222	ML-KEM-768	2272	11
LoRa (222 B payload)	222	ML-KEM-1024	3136	15

To illustrate the fragmentation behavior across IoT protocols, the number of packets required for each configuration is shown in Figure 3.

**Figure 3.** Packet fragmentation of ML-KEM handshakes across IoT networks with varying MTU sizes.

Observe that PQC communication overhead can lead to significant packet fragmentation in constrained IoT networks. Even under favorable conditions, such as BLE with a 247-byte MTU, ML-KEM handshakes require between 7 and 13 packets. For more constrained protocols, the impact is even more severe. For example, in IEEE 802.15.4 networks, depending on the security level, the number of packets increases to between 13 and 25. The most extreme case occurs in low-payload LoRa networks, where ML-KEM-1024 requires up to 62 packets to complete a single handshake. Figure 3 highlights this behavior.

Analysis of the average payload utilization shows that packets are transmitted near their maximum capacity for all of the protocols tested. For example, BLE transmissions utilize approximately 224–241 bytes per packet, while LoRa transmissions with a 51-byte payload consistently approach the maximum payload size. This means that the observed fragmentation is not due to inefficient packet utilization. Instead, it is a direct consequence of the disparity between PQC message sizes and the limited payload capacities of constrained IoT protocols. This degree of fragmentation has several implications. It increases transmission latency and energy consumption due to the need for sequentially transmitting multiple packets. Most importantly, the likelihood of transmission failure is increased since the success of any handshake depends upon the successful delivery of all packets. Therefore, while PQC algorithms are computationally efficient, their large message sizes result in fragmentation overhead that can impact system performance.

4.9. Handshake Reliability Analysis

To quantify the impact of packet fragmentation on system reliability, we analyze the probability of successful handshake completion under packet loss. A value $p = 0.99$ is used as a representative high-reliability link assumption to illustrate the cumulative effect of packet fragmentation and is not intended to represent a specific BLE, Zigbee, or LoRa deployment. Assuming an independent packet success probability of $p = 0.99$, the probability of a successful handshake is given by p^N , where N is the number of packets required for the handshake. These results are summarized in Table 11.

To illustrate the relationship between packet count and reliability, the handshake success probabilities are shown in Figure 4.

Table 11. Handshake success probability across IoT protocols ($p = 0.99$ per packet).

Protocol	Algorithm	Packets	Success Probability
BLE (247 B)	ML-KEM-512	7	0.932065
BLE (247 B)	ML-KEM-768	10	0.904382
BLE (247 B)	ML-KEM-1024	13	0.877521
Zigbee (127 B)	ML-KEM-512	13	0.877521
Zigbee (127 B)	ML-KEM-768	18	0.834514
Zigbee (127 B)	ML-KEM-1024	25	0.777821
LoRa (51 B)	ML-KEM-512	31	0.732303
LoRa (51 B)	ML-KEM-768	45	0.636185
LoRa (51 B)	ML-KEM-1024	62	0.536268
LoRa (222 B)	ML-KEM-512	8	0.922745
LoRa (222 B)	ML-KEM-768	11	0.895338
LoRa (222 B)	ML-KEM-1024	15	0.860058

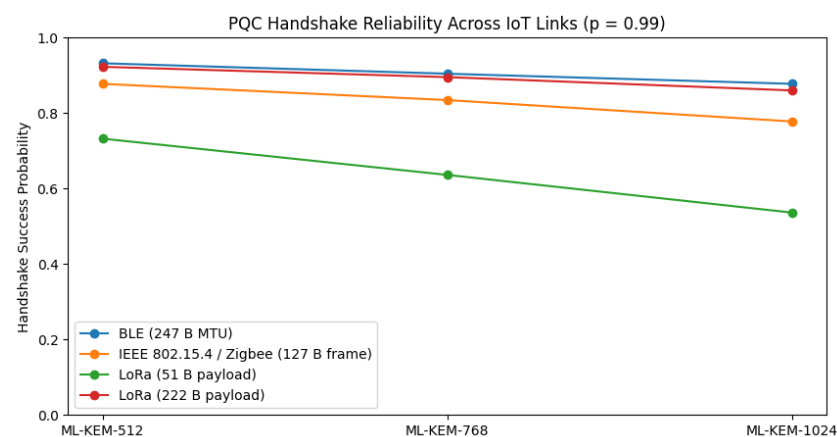


Figure 4. Handshake success probability across IoT protocols assuming a per-packet success rate of $p = 0.99$.

Observe that for BLE networks, success probabilities remain above 87% across all parameter sets, reflecting the relatively small number of packets required. In contrast, IEEE 802.15.4 networks experience a noticeable decline, with success probability dropping below 78% for ML-KEM-1024. As would be expected, handshake reliability degrades rapidly as the number of required packets increases. Even with a 99% packet success rate, multi-packet PQC handshakes exhibit significant reductions in overall success probability.

The most severe degradation occurs in low-payload LoRa networks. In the worst case, ML-KEM-1024 requires 62 packets, resulting in a handshake success probability of approximately 53.6%. This represents nearly a 50% chance of handshake failure, even when individual packet delivery is highly reliable.

These results highlight a critical system-level limitation: fragmentation amplifies the impact of packet loss. As a result, PQC deployment in constrained IoT networks may require retransmission mechanisms, protocol redesign, or cross-layer optimization to maintain acceptable reliability.

5. Deployment Guidelines for PQC Algorithm Selection

The results presented in this study provide practical guidance for selecting PQC algorithms in constrained IoT environments. Since communication overhead, packet fragmentation, and handshake reliability vary significantly across protocols, algorithm selection should be matched to the characteristics of the underlying network. Based on the simulation results presented in this study, Table 12 summarizes recommended PQC configurations for representative IoT environments.

Table 12. Recommended PQC configurations for representative IoT protocols.

Protocol	Recommended Configuration
BLE	ML-KEM-512 or ML-KEM-768
IEEE 802.15.4 (Zigbee)	ML-KEM-512 preferred
LoRa (51-byte payload)	ML-KEM-512
LoRa (222-byte payload)	ML-KEM-512 or ML-KEM-768
Digital Signatures	Falcon-512 when communication overhead is critical

For Bluetooth Low Energy (BLE), all evaluated ML-KEM parameter sets are feasible due to the relatively large 247-byte MTU. However, ML-KEM-512 and ML-KEM-768 provide a favorable balance between security and communication overhead. ML-KEM-1024 may be employed when higher security requirements justify the additional transmission cost.

For IEEE 802.15.4 (Zigbee) networks, the smaller 127-byte frame size results in increased packet fragmentation. Consequently, ML-KEM-512 is generally the most practical choice, while ML-KEM-768 may be appropriate in applications requiring additional security. The use of ML-KEM-1024 should be carefully evaluated due to the increased packet count and corresponding reduction in handshake reliability.

For LoRa networks, communication constraints become substantially more significant. In particular, low-payload LoRa configurations experience severe packet fragmentation, resulting in a large number of transmitted packets and reduced handshake success probabilities. Under such conditions, ML-KEM-512 is generally preferred, while ML-KEM-768 and ML-KEM-1024 should be considered only when the increased communication cost can be accommodated through retransmission mechanisms or protocol-level optimization.

The digital signature results indicate that signature size is also an important consideration in constrained environments. Falcon-512 provides the smallest signature size among the evaluated schemes and may therefore be attractive when communication overhead is a primary concern. In contrast, SPHINCS+ produces substantially larger signatures and may be less suitable for highly constrained networks despite its strong security properties.

Application requirements also play an important role in PQC parameter selection. For example, environmental monitoring and low-risk sensing applications may prioritize communication efficiency and battery life, making ML-KEM-512 an attractive choice. Industrial control systems, smart-grid infrastructure, and other applications with elevated security requirements may justify the use of ML-KEM-768 despite the additional communication overhead. In contrast, applications involving sensitive medical information, critical infrastructure, or long-term confidentiality requirements may warrant the increased

security margin provided by ML-KEM-1024, even when accompanied by higher packet fragmentation, latency, and energy costs. Consequently, PQC deployment decisions should balance both communication constraints and application-specific security requirements.

Overall, the results suggest that PQC algorithm selection in IoT deployments should consider not only cryptographic security level but also communication overhead, packet fragmentation, and reliability constraints imposed by the underlying network protocol. These recommendations are intended as deployment guidelines rather than strict design rules, since application-specific security requirements, network conditions, and retransmission policies may influence the final algorithm selection.

6. Conclusions

In this work, we have presented a system-level evaluation of post-quantum cryptographic (PQC) deployment in constrained Internet of Things (IoT) environments. By combining cryptographic benchmarking with network modeling, we have highlighted challenges to deploying PQC protocols in real-world IoT systems. Even highly reliable links ($p = 0.99$) can yield handshake success probabilities near 50% under the fragmentation model considered in this study, thereby underscoring the need for communication-aware PQC design.

The results highlight a fundamental shift in system behavior when transitioning from classical to post-quantum systems. With computational performance no longer being the primary bottleneck, increased message sizes associated with PQC introduce significant stress on constrained communication channels. This shift has important implications for the design of secure IoT systems, where communication efficiency, rather than computation, becomes the dominant factor influencing performance and reliability.

The present study has focused on the impact of PQC message sizes, packet fragmentation, and communication reliability under representative protocol constraints. Dynamic network effects such as channel fading, node mobility, retransmission behavior, and multi-node contention were not explicitly modeled. Extending the analysis to network-level simulation environments will enable evaluation of PQC performance under more realistic and time-varying IoT operating conditions. Furthermore, protocol-level mitigation techniques should be studied as a means of reducing communication overhead in constrained networks, including header compression, MTU negotiation, packet aggregation, and application-layer chunking. Therefore, additional investigation is needed to determine the effectiveness of such mitigation mechanisms in reducing packet fragmentation, latency, energy consumption, and handshake failure rates associated with PQC deployment.

This work has identified communication-level effects using liboqs benchmarking and network modeling. However, an important next step is the validation of these results on hardware platforms commonly used in IoT deployments such as Cortex-M4 and Cortex-M7 microcontrollers. Additionally, FPGA-based implementations and other hardware acceleration strategies will also be explored in order to support efficient PQC processing and mitigate the impact of large message sizes in constrained networks. Such experiments will enable direct comparison between modeled and measured execution time, energy consumption, and communication overhead, thereby providing additional validation of the conclusions presented in this work.

These findings suggest that traditional approaches to protocol design and security evaluation may be insufficient in the post-quantum era. Instead, for the effective deployment of PQC in IoT environments, future work will require a holistic, system-level perspective that accounts for interactions between cryptographic primitives, network protocols, and device constraints. Our study also highlights the need for cross-layer optimization. Such an

approach could therefore lead to protocol redesign, network-aware algorithm selection, and hardware-assisted acceleration, ultimately enabling the adoption of PQC in IoT systems.

Author Contributions: Conceptualization, E.S. and G.X.; methodology, E.S., G.X. and J.M.; software, E.S. and J.M.; validation, E.S.; formal analysis, E.S.; investigation, E.S.; resources, E.S.; data curation, E.S.; writing—original draft preparation, E.S.; writing—review and editing, E.S., G.X., J.M. and S.W.; visualization, E.S.; supervision, G.X.; project administration, G.X. and E.S.; funding acquisition, G.X. and S.W. All authors have read and agreed to the published version of the manuscript.

Funding: This research was funded by NSF grant number 2329053 and NSF grant number 2502912.

Data Availability Statement: Data available on request due to privacy.

Conflicts of Interest: The authors declare no conflicts of interest.

Abbreviations

The following abbreviations are used in this manuscript:

PQC	Post-Quantum Cryptography
IoT	Internet of Things
KEM	Key Encapsulation Mechanism
ML-KEM	Module-Lattice Key Encapsulation Mechanism (Kyber)
ML-DSA	Module-Lattice Digital Signature Algorithm (Dilithium)
QKD	Quantum Key Distribution
ECC	Elliptic Curve Cryptography
TLS	Transport Layer Security
MTU	Maximum Transmission Unit
BLE	Bluetooth Low Energy
LoRa	Long Range (Low-Power Wide-Area Network)
liboqs	Open Quantum Safe library

References

- Shor, P.W. Algorithms for quantum computation: Discrete logarithms and factoring. In *Proceedings of the 35th Annual Symposium on Foundations of Computer Science*; IEEE: Santa Fe, NM, USA, 1994; pp. 124–134.
- National Institute of Standards and Technology (NIST). *Post-Quantum Cryptography Standardization*; NIST: Gaithersburg, MD, USA, 2024. Available online: <https://csrc.nist.gov/projects/post-quantum-cryptography> (accessed on 5 March 2026).
- Raza, S.; Wallgren, L.; Voigt, T. SVELTE: Real-time intrusion detection in the Internet of Things. *Ad Hoc Netw.* **2013**, *11*, 2661–2674. [CrossRef]
- Kannwischer, M.J.; Rijneveld, J.; Schwabe, P.; Stoffelen, K. pqm4: Testing and Benchmarking NIST PQC on ARM Cortex-M4; Cryptology ePrint Archive; Report 2019/844. Available online: <https://eprint.iacr.org/2019/844> (accessed on 5 March 2026).
- Bos, J.W.; Costello, C.; Naehrig, M.; Stebila, D. Post-quantum key exchange for the TLS protocol from the ring learning with errors problem. In *Proceedings of the IEEE Symposium on Security and Privacy*; IEEE: San Jose, CA, USA, 2015; pp. 553–570.
- Wallden, P.; Kashefi, E. Cyber Security in the Quantum Era. *Commun. ACM* **2019**, *62*, 120. [CrossRef]
- Xu, G.; Mao, J.; Sakk, E.; Wang, S.P. An overview of quantum-safe approaches: Quantum key distribution and post-quantum cryptography. In *Proceedings of the 57th Annual Conference on Information Sciences and Systems (CISS)*; IEEE: Baltimore, MD, USA, 2023; pp. 1–6.
- Gubbi, J.; Buyya, R.; Marusic, S.; Palaniswami, M. Internet of Things (IoT): A Vision, Architectural Elements, and Future Directions. *Future Gener. Comput. Syst.* **2013**, *29*, 1645–1660. [CrossRef]
- Assiri, A.; Almagwashi, H. IoT Security and Privacy Issues. In *Proceedings of the 1st International Conference on Computer Applications & Information Security (ICCAIS)*; IEEE: Riyadh, Saudi Arabia, 2018; pp. 1–5.
- Cheng, C.; Lu, R.; Petzoldt, A.; Takagi, T. Securing the Internet of Things in a Quantum World. *IEEE Commun. Mag.* **2017**, *55*, 116–120. [CrossRef]
- Abdulrahman, A.; Hwang, V.; Kannwischer, M.J.; Sprenkels, D. Faster Kyber and Dilithium on the Cortex-M4. In *Applied Cryptography and Network Security (ACNS)*; Springer: Cham, Switzerland, 2022; pp. 853–887.

12. Abdulrahman, A.; Kannwischer, M.J.; Lim, T.-H. Enabling Microarchitectural Agility: Taking ML-KEM & ML-DSA from Cortex-M4 to M7 with SLOTHY. In *Proceedings of the ACM Asia Conference on Computer and Communications Security (ASIA CCS)*; ACM: New York, NY, USA, 2025.
13. Ding, S.; Shi, F.; Erkan, U.; Toktas, A.; Li, Q.; Wang, C.; Gao, S.; Mou, J. Design of a three-dimensional logistic map and its application to seafood image encryption. *J. Supercomput.* **2026**, *82*, 225. [[CrossRef](#)]
14. Ding, S.; Wang, X.; Zhu, P.; Erkan, U.; Toktas, A.; Li, Q.; Mou, J.; Gao, S. SRA: A sine-based reconstruction approach to improve 2D chaotic map performance. *Eur. Phys. J. Plus* **2026**, *141*, 182. [[CrossRef](#)]
15. Papakotoulas, A.; Terzidis, A.; Hadjiefthymiades, S. ERITA: Ensuring the Reliability of Internet of Things-Based Applications. In *Proceedings of the International Symposium on Networks, Computers and Communications (ISNCC)*; IEEE: Doha, Qatar, 2023; pp. 1–6.

Disclaimer/Publisher’s Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.