JCBI
Journal of Computing & Biomedical Informatics
ISSN: 2710 - 1606
Volume 10 Issue 02
2026

Research Article
https://doi.org/10.56979/1002/2026/1253

Quantum-Safe Wireless Sensor Networks: A Post-Quantum Cryptography Framework with Adaptive Security Optimization

Siri D[1], Janardhan M[2], Raja Sekhar V[2], Jaya Prakash P[3], Sushama C[4], Pramodh Krishna D[5], and Kranthi Kumar Lella[6*]

[1]Department of CSE, Gokaraju Rangaraju Institute of Engineering and Technology, Hyderabad, India.
[2]Department of Computer Science and Engineering, G. Pullaiah College of Engineering and Technology, Kurnool, India.
[3]Department of IT, Sri Venkateswara College of Engineering, Tirupati, India.
[4]Department of CSE, School of Computing, Mohan Babu University, Tirupati, India.
[5]Department of Computer Science and Engineering, Koneru Lakshmaiah Education Foundation, Vaddeswaram, India.
[6]Manipal Institute of Technology, Manipal Academy of Higher Education, Manipal 576104, India.
[*]Corresponding author: Kranthi Kumar Lella. E-Mail. kranthikumar.l@manipal.edu

Received: December 11, 2025 **Accepted:** February 26, 2026

Abstract: Quantum computing poses a significant threat to Wireless Sensor Networks (WSNs) by undermining traditional cryptographic algorithms such as RSA and Elliptic Curve Cryptography (ECC). This work proposes a quantum-safe architecture for WSNs that integrates post-quantum cryptography (PQC) with lightweight IoT protocols to ensure long-term confidentiality, authenticity, and resilience. The framework leverages ML-KEM for key encapsulation and ML-DSA/SLH-DSA for signatures, and seamlessly integrates with EDHOC, OSCORE, and COSE. A novel Efficient Adaptive Parameter Selection (EAPS) mechanism dynamically adjusts cryptographic strength to balance security, energy consumption, and latency under varying network conditions. Experimental evaluation demonstrates that ciphertext fragmentation (8–27 pieces) results in manageable completion times of 4–10 seconds, while join operations consume only 0.005–0.03 J per cryptographic handshake event, while system-level energy consumption including network overhead is higher (~1.0–1.4 J per join cycle depending on hop count and retransmissions). Battery lifetime projections under steady-state sensing workloads range from approximately 115–280 months for low-duty-cycle operation, whereas realistic deployment conditions with periodic communication yield effective lifetimes of 11.5–24 months. Security analysis confirms robust resistance against downgrade attacks and Harvest-Now-Decrypt-Later (HNDL) threats. Moreover, EAPS reduces risk scores by over 50% compared to fixed schemes with less than 10% additional energy overhead, and batch signature verification improves scalability by increasing throughput from ~1,600 to ~2,800 verifications per second. Overall, the proposed framework demonstrates that WSNs can achieve quantum-resistant security with minimal performance trade-offs, ensuring readiness for the post-quantum era.

Keywords: Wireless Sensor Networks; Post-Quantum Cryptography; Quantum-Safe Architecture; Elliptic Curve Cryptography; Sustainable Development Goals (SDG); Efficient Adaptive Parameter Selection

1. Introduction

WSNs are one of the most significant technologies for pervasive computing. They enable applications such as smart healthcare, environmental monitoring, precision agriculture, critical infrastructure, and industrial automation [1]. These networks are made up of sensor nodes that are spread out over space and have a little amount of energy, computational power, and communication bandwidth. They collaborate to perceive, analyse, and transmit data [2]. Because they are lightweight and spread out, WSNs can be used

in a lot of different situations with low resources. But they are also quite prone to a lot of various kinds of security holes [3]. ECC and RSA are two examples of conventional cryptographic algorithms that have long formed the backbone of WSN security by providing privacy, authentication, and integrity. But quantum computing, which is poised to become a reality, poses a threat to the very foundation of these traditional cryptosystems [4]. In the near future, methods like Shor's algorithm and Grover's search will make systems based on RSA and ECC worthless by swiftly solving problems that were thought to be too hard for computers to solve [5]. This upcoming change has made it necessary to look into Post-Quantum Cryptography (PQC) right away. PQC is designed to protect against attacks from both classical and quantum enemies [6].

In the quantum age, WSNs are not very strong since they rely on cryptographic keys that survive a long time, protocols that are light but not very strong, and devices that don't require a lot of power. Harvest Now, Decode Later (HNDL) attacks are a severe danger to sensitive applications like healthcare or industrial monitoring. They let attackers grab encrypted data now and decode it later. Long-term confidentiality and confidence can be irrevocably compromised, even in the absence of immediate data exposure. Consequently, the incorporation of quantum-safe features into WSNs has emerged as a significant field of research [8]. NIST's work on standardization shows that Post-Quantum Cryptography should have new kinds of algorithms, like Lattice-based schemes (ML-KEM, ML-DSA), Hash-based signatures (SLH-DSA), Code-based systems, and Multivariate polynomial cryptography [9]. Lattice-based constructions are now more popular than the others since they are efficient, safe, and can function with less equipment [10].

It isn't as straightforward to add PQC to WSNs as just replacing it. Unlike regular networks, WSN nodes don't have a lot of flash memory, RAM, energy, or communication power [11]. PQC primitives typically have larger key sizes, ciphertexts, and signatures than ECC, which heightens concerns about fragmentation, latency, and communication costs in mesh-based sensor topologies [12]. Computational costs, such as matrix-vector multiplications, Number Theoretic Transforms (NTTs), and hash invocations, also present additional hurdles [13]. The challenge lies not only in safeguarding WSNs from quantum attacks but also in developing a quantum-safe architecture that balances robustness with the stringent constraints of sensor nodes [14]. To achieve this balance, it is necessary to reevaluate join protocols, scheduling strategies, group communication mechanisms, and software footprints [15].

Recent studies have demonstrated the feasibility of employing PQC in embedded systems. Researchers have found that algorithms like ML-KEM and ML-DSA can be made better for Cortex-M4/M33 class CPUs without too much more energy or time [16]. Also, protocol-level changes like EDHOC with PQ extensions and OSCORE for safe data transport make it possible to integrate PQC to existing IoT frameworks without having to totally redesign them [17]. There are still questions regarding how fragmentation works, how much energy it takes to join a group, how to rekey a group after burst losses, and how long the battery will last. It is very crucial to fill in these gaps since any solution must make sure that WSNs may be used for a long period without diminishing their reliability [18].

The goal of this research is to fix these problems by building a full quantum-safe architecture for wireless sensor networks (WSNs). The framework is built on standard protocols, but it makes them better by introducing PQ primitives and new design methodologies like Efficient Adaptive Parameter Selection (EAPS). EAPS, on the other hand, modifies parameter sets (such ML-KEM-512, 768, 1024; ML-DSA-44/65/87) on the fly based on the network environment, energy budgets, and security risk profiles. Fixed-parameter installations either utilize too many resources or don't protect enough. This adaptive method makes sure that latency constraints are met, energy utilization is maintained to a minimum, and high-risk nodes can be promoted to better cryptographic profiles on purpose. The framework adds something new to the mix by making the protocol layer adaptable, which makes it easier to balance energy, latency, and security trade-offs. Most traditional solutions don't do this.

Three layers of innovative concepts make up the recommended architecture. At the control-plane level, hybrid EDHOC handshake protocols use both traditional ECC and PQC techniques. This makes them stronger since they can still work even if one of the plans doesn't work. This hybrid strategy makes downgrade attacks less likely because both secrets have to be revealed at the same time for secrecy to be broken. OSCORE uses pre-derived keys to get rid of steady-state overhead, which makes sure that end-to-end communication is safe and light at the data-plane level. Also, group communication uses batched

rekeying algorithms with unicast delivery and caching. This saves a lot of airtimes compared to simple multicast approaches. Finally, at the gateway level, batch verification methods for ML-DSA signatures considerably boost throughput. This means that they can be used on a large scale without causing problems.

The architecture was examined in a number of ways to see if it might work: fragmentation and latency budgets, energy per join, steady-state throughput, group rekeying resilience, TSCH scheduling impact, and visible security features. The results suggest that adding PQ only adds a small amount of work. The time it takes to join is a few seconds longer, the amount of energy used goes up a little, and the memory footprints stay within the limitations of 128–256 KB flash and 16–32 KB RAM. Battery life estimations suggest that there is essentially no difference between classical and PQ frameworks. This is crucial since it shows that utilizing quantum-safe security doesn't cost too much. The framework is very important for security since it makes sure that HNDL attacks don't work, that downgrade immunity is enforced, and that long-term secrecy is guaranteed.

This introduction talks about how PQC can be added to WSNs in a way that doesn't hurt their performance or energy efficiency. This paper establishes a foundation for safeguarding critical sensor installations from quantum adversaries by proposing a singular, adaptable, and empirically validated architecture. It not only addresses the needs of future IoT ecosystems, but it also aligns with the new NIST PQC requirements. The results demonstrate that quantum-safe WSNs are both feasible and beneficial. You can use them for a long time and know that they will still be safe even when quantum computing becomes mainstream. The rest of the paper is organized like this: Section 2 talks about work that is related, Section 3 goes into further detail regarding the suggested approach, Section 4 talks about the analysis of the results, and Section 5 makes the conclusion.

2. Related works

Blanco-Romero et al. [19] integrate PQC into CoAP and MQTT-SN by substituting DTLS backends with wolfSSL/liboqs, thereafter assessing the practical effects on message size, handshake dynamics, and code alterations. This paper is useful for WSNs because it shows how to integrate Kyber/Dilithium to IoT-native protocols (not just TLS/DTLS case studies), talks about crypto-agility, and discusses technical concerns like fragmentation and DTLS listeners not being able to work together. It's a step-by-step approach for shifting constrained stacks without having to rebuild apps.

Kannwischer et al. [20] improve the pqm4 framework so that it can look at more PQ signature candidates on the STM32L4R5 (Cortex-M4). They tell you how big the code is, how much RAM and stack it uses, how many cycles it takes, and which strategies work and which don't (dynamic allocation, memory pressure). The main focus of the method is on signatures, however it can also be utilized for KEMs. When replacing ECDSA with PQ signatures in sensing nodes and gateways, it helps WSN designers think about latency, energy budgets, and memory headroom.

EDHOC [21] was suggested as an IETF Proposed Standard in March 2024. This is a short, safe Diffie-Hellman handshake for devices with limited resources. It is based on CBOR/COSE and is often used to start OSCORE security contexts. EDHOC is an excellent choice for duty-cycled radios and tiny stacks in WSNs since it has a small message size and can safeguard identities. EDHOC is not post-quantum by itself, but it is the perfect place to add hybrid/PQ authentication and keys to CoAP/6LoWPAN settings.

López Pérez et al. [22] give a short overview of EDHOC's design (SIGMA-style AKE), formal tests, and comparisons with DTLS/TLS for IoT. They argue that the size of the handshake, the time it takes, and the amount of RAM and flash memory have all gotten a lot better since DTLS 1.3. They also talk about how crypto-agile can change over time. This post talks about why EDHOC+OSCORE is the ideal way to secure the application layer for WSNs that are getting ready for PQC and where to add PQ ciphersuites or hybrid auth as standards improve.

On a server, a laptop, and an integrated "Device E3," Abbasi et al. [23] compare how well Kyber, NTRU, BIKE (KEMs), and Dilithium/Falcon (signatures) work. They talk about latency, memory, energy, and fragmentation of handshakes. Results: For smartphones with minimal resources, Kyber is the finest KEM. Falcon checks swiftly with smaller signatures, which is beneficial for bandwidth. Dilithium, on the other hand, signs faster. BIKE is heavier, but it has more possibilities. Results help WSN builders figure out how

long it will take to connect, how many fragments will be sent per handshake, and how much energy will be used per join for multi-hop paths.

Shehzadi and Whaley [24] assess a "Kyber-KEM-Ascon" integration on IoT devices, emphasizing compact authenticated key establishment with a lightweight AEAD. The study demonstrates the feasibility of employing standardized ML-KEM (Kyber) in embedded systems and quantifies cycle counts and communication overheads, indicating that PQ KEMs can be successful within WSN duty cycles through careful parameterization and scheduling.

Nielsen et al. [25] evaluate Dilithium, Falcon, and SPHINCS+ (different levels of security) on a LoRa ESP32 microcontroller, looking at latency, memory utilization, and the extra wireless bandwidth needed for bigger signatures. They think that Dilithium is the fastest overall, but Falcon is an excellent choice because it balances size and verification. SPHINCS+ is frequently too sluggish for budgets that need to be tight in real time. This has a direct impact on the options for signing and attesting WSN firmware for low-power radio nodes. (Added on top of the minimum.)

3. Proposed Framework

A Quantum-Safe Architecture for Wireless Sensor Networks (WSNs)

This section presents a complete, end-to-end framework for making constrained Wireless Sensor Networks resilient against quantum-enabled adversaries. It is written to be directly actionable for implementers working with 6LoWPAN/RPL/CoAP stacks over IEEE 802.15.4 (including TSCH), and it focuses on to integrate post-quantum cryptography (PQC) where it matters, each design choice is made, security and performance properties are obtained, and each component fits in the protocol stack. The framework is mathematically specified so the resulting system can be analyzed, dimensioned, and verified.

At the core, to compose:

- **Key establishment** with ML-KEM (Kyber-derived; FIPS 203) for confidentiality against "harvest-now-decrypt-later" threats.
- **Digital signatures** with ML-DSA (Dilithium-derived; FIPS 204) for time-bound data origin authentication and SLH-DSA (SPHINCS+; FIPS 205) for long-lived roots of trust and firmware/SUIT manifests.
- **Constrained protocol bindings** using CoAP, COSE, OSCORE, and EDHOC, which are proven and standardized for constrained nodes; to extend these with compact PQC profiles without breaking wire-level compatibility.

Below, to first set goals and assumptions, then specify the layered design and its novelty, followed by the equations and algorithms you can use to size links, schedule handshakes, and choose parameters in the field.

3.1. Goals, Threat Model, and Design Constraints

G1. Quantum-resilient confidentiality: No recorded traffic becomes decryptable even if the adversary later gains a large-scale quantum computer. (Hence KEM-based forward secrecy on day one.)

G2. Robust data origin & updates: Sensor reports and firmware remain verifiable decades into the future (signature longevity).

G3. Constrained-friendly: Fits 802.15.4 MTU (127 bytes), minimizes fragmentation, code size, RAM, CPU, and energy.

G4. Interoperable & incremental: Works with 6LoWPAN/RPL/CoAP, OSCORE, and EDHOC so deployments can "drop in" PQC.

G5. Future-ready: Crypto-agile profiles that can be swapped at runtime or via SUIT-based updates.

Threat model. Adversaries can passively capture traffic now and attempt decryption later ("HNDL"), may actively inject/modify traffic, and may compromise intermediate routers. To assume physical compromise of some leaf nodes is possible; hence forward secrecy, rekey, and revocation is necessary.

Constraints.

- 802.15.4 PHY at 250 kbps is typical; per-frame PHY max is 127 bytes before MAC and upper-layer headers. TSCH scheduling may further bound airtime.
- 6LoWPAN fragmentation exists; recent RFCs improve route-over fragment recovery.

3.2. Architectural Overview

Figure 1 shows the workflow of the proposed model.

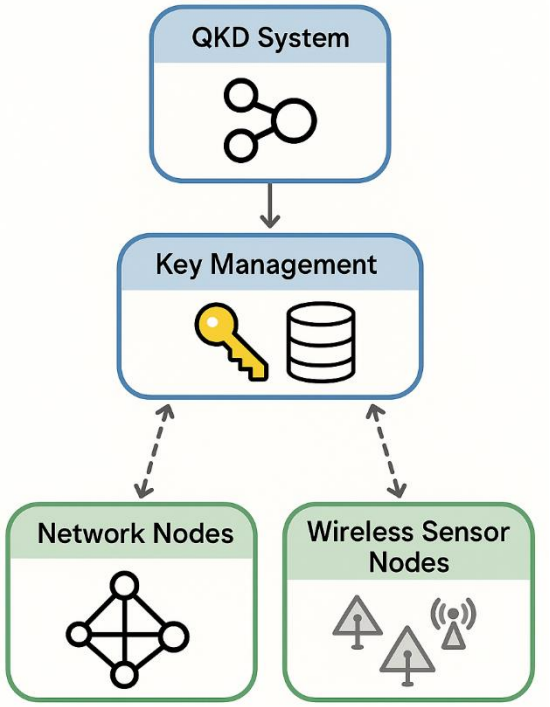

Figure 1. Propose Architecture.

To organize the system in three planes:

1. **Control plane (join, rekey, authorization):** EDHOC (compact AKE) with a PQC KEM augmentation defines OSCORE master secrets; authorization tokens (CWT/COSE) carry policy.
2. **Data plane (telemetry/commands):** CoAP secured end-to-end by OSCORE using AEAD (e.g., AES-CCM) with keys derived from PQ secrets.
3. **Management plane (firmware/config):** SUIT manifests signed with SLH-DSA (root) and ML-DSA (operational) for smaller routine signatures; delivered over CoAP/OSCORE.

Entities.

- **Leaf sensor** n_i: 32–128 KB RAM, battery powered.
- **Cluster head** c_j: mains or larger battery; can offload polynomial/NTT heavy lifting when allowed.
- **Border router / gateway** g: connects WSN to IP backbone; terminates authorization flows (ACE/OSCORE profile).

3.3. Cryptographic Building Blocks (what/why/how/where)

3.3.1. ML-KEM for key establishment (why & where)

- **Why:** NIST-standard KEM believed secure against quantum adversaries; gives IND-CCA2 secrecy for the derived 256-bit shared secret K. Sizes are known and stable, enabling deterministic fragmentation planning.
- **Where:** In the join/rekey (control plane). To encapsulate to a node's encapsulation key to derive OSCORE master secrets.
- **How (parameters):** Use ML-KEM-768 by default (NIST's recommended balance), fall back to 512 only under severe constraints, and reserve 1024 for high-value deployments. Table 8 from FIPS-203 provides byte sizes used in §3.7 formulas.

Reference sizes (bytes):

- ML-KEM-512: ek=800, dk=1632, ct=768.
- ML-KEM-768: ek=1184, dk=2400, ct=1088.
- ML-KEM-1024: ek=1568, dk=3168, ct=1568.

3.3.2. ML-DSA and SLH-DSA for signatures

- ML-DSA produces shorter signatures and faster ops for routine, time-bounded attestations; SLH-DSA is extremely conservative (hash-based) and ideal for roots of trust and firmware that must remain verifiable far into the future even if unforeseen lattice advances occur.
- **ML-DSA** for per-epoch node attestations, cluster-head announcements, and short-lived credentials.

- **SLH-DSA** for manufacturer root, device bootstrap certs, and SUIT manifests (see §3.8).
- **Sizes (illustrative):** ML-DSA-65: pk=1952, sk=4032, sig≈3309 bytes. SLH-DSA-SHAKE-128s: pk=32 bytes, sig≈7856 bytes; SHAKE-256f: pk=64 bytes, sig≈49,856 bytes. These numbers drive fragmentation and TSCH scheduling (§3.6–§3.7).

3.3.3. Protocol bindings

- CoAP (RFC 7252) as the application protocol; CBOR/COSE provide compact encoding and crypto containers; OSCORE adds end-to-end protection at the CoAP layer; EDHOC performs the authenticated key exchange to derive OSCORE security contexts—now augmented with a KEM contribution (§3.4).

3.4. Stack Integration in WSNs (where each piece fits)

Link/Adaptation/Network.

- IEEE 802.15.4 radio (often TSCH): 127-byte MAC frame ceiling; 250 kbps typical. 6LoWPAN (RFC 4944 family) compresses IPv6/UDP headers and fragments large payloads; RPL handles multi-hop routing with LLN-specific optimizations; RFC 8138 compresses RPL options. This baseline minimizes header overhead before to introduce PQ payloads.

Security layering.

1. **Join/Rekey:** EDHOC over CoAP (RFC 9668) with a PQC KEM contribution: the EDHOC transcript is extended with an ML-KEM ciphertext so the derived OSCORE master secret K_{OSCORE} depends on both the classical DH secret and a post-quantum KEM secret (hybrid). This neutralizes HNDL for confidentiality while retaining EDHOC's compactness, message ordering, and proofs. (To note NIST cautions that combining KEMs must be done carefully; to therefore specify the exact KDF in §3.7.2.)
2. **Data protection:** OSCORE uses AEAD (e.g., AES-CCM) and sequence numbers with replay protection; keys are derived from the hybrid secret via HKDF-like exporters as specified by EDHOC/OSCORE.
3. **Group comms:** For multicast commands/alarms, to adopt Group OSCORE (emerging standard), with the group rekey seeded by ML-KEM from the controller to members (see §3.7.4).
4. **Firmware:** SUIT manifests signed with SLH-DSA (root) and optionally countersigned by ML-DSA for operational agility, transported over CoAP/OSCORE.

3.5. Novelty: Five Techniques that Make PQC Practical on Tiny Nodes

1. **PQ-EDHOC (KEM-augmented EDHOC) with fragment budgeting.** To embed one ML-KEM encapsulation per join/rekey into EDHOC's compact flows. The augmentation is sized and scheduled to avoid frame storms on 802.15.4 links by pre-computing fragment counts and spacing them across TSCH cells (equations in §3.6). Result: quantum-safe secrecy with the same number of EDHOC round trips and bounded airtime.
2. **Layered KEM Offload (LKO).** To let cluster heads/gateways do costlier polynomial/NTT work **once**, then distribute small commitments and ciphertexts that the leaf verifies with cheap hashing. Leaves still execute constant-time decapsulation, but heavy sampling/NTT can be pre-derived or batched upstream where feasible (without changing ML-KEM semantics).
3. **Energy-Adaptive Parameter Selection (EAPS).** A runtime optimizer chooses ML-KEM {512,768,1024} and ML-DSA {44,65,87} per node/epoch using a multi-objective function that trades energy vs. risk score (site policy), with tight constraints on join budget and Age-of-Key. The optimizer is explicit and solvable online (see §3.6).
4. **Hybrid-Key Derivation with Downgrade Immunity.** To combine the EDHOC DH secret Z_{DH} with ML-KEM secret K_{kem} using a transcript-bound KDF with mandatory presence checks: if either input is missing or malformed (or an algorithm is negotiated below policy), the handshake aborts. This defeats downgrade and prevents "fake-PQC" attacks (formalized in §3.7.2). (To heed FIPS-203's warning: combined KEMs aren't automatically IND-CCA2; our combiner is AEAD-KDF-bound to the transcript to maintain security reduction assumptions.)
5. **Signature Split-Roles & Epoching.** To harden long-lived trust anchors with SLH-DSA and keep operational signatures fast with ML-DSA (shorter and cheaper). Epoch counters and short certificate chains ensure small wire images on constrained links.

3.6. Analytical Model: Airtime, Energy, Delay, and Reliability

To now define the quantities used to size the radio budget, set TSCH schedules, and pick PQC parameters. Symbols are defined immediately after each equation. The following formulations provide analytical estimates, which are later validated and complemented through simulation results presented in Section 4.

3.6.1. Payload and fragmentation

For a unicast CoAP message carrying a cryptographic object of size S bytes (e.g., ML-KEM ciphertext or a signature), the usable payload per 802.15.4 frame is

$$P_{pay} = 127 - H_{MAC} - H_{SEC} - H_{6LoWPAN} - H_{NET} - H_{COAP} - H_{OSCORE} \quad (1)$$

where H_{MAC} is MAC header/FCF/seq/addr/CRC (bytes), H_{SEC} is link-layer security overhead if used, H6LoWPAN is the (compressed) dispatch/fragment header, H_{NET} is compressed IPv6/UDP (6LoWPAN HC), H_{COAP} is CoAP header/options, and H_{OSCORE} is object-security overhead (kid/nonce/tag). (IEEE 802.15.4's 127-byte MAC frame maximum and the impact of small payloads are well-documented; sizing must be done per stack configuration.)

$$F = \left\lceil \frac{S}{P_{pay}} \right\rceil \quad (2)$$

Variables: S (bytes), P_{pay} (bytes), F (integer).

Worked sizing (design-time): With ML-KEM−768 $S = 1088$ bytes for the ciphertext, and a conservative P_{pay}=60–90 bytes (typical when OSCORE and 6LoWPAN compression apply), FFF lies in the 13–19 range. This is why scheduling and batching (TSCH cells) are crucial (§3.7). (Byte sizes from FIPS-203 Table 8.)

3.6.2. Airtime and delay

Let the per-byte over-the-air time be t_{b} (≈ 32 μs/byte at 250 kbps) and the per-fragment processing and CSMA/TSCH delay be t_{proc}. The airtime for the cryptographic object is

$$T_{air} = \sum_{k=1}^{F} (B_k t_b + t_{proc}) \quad (3)$$

where B_k is the k-th fragment's length in bytes. On TSCH, you often map fragments to cells; with slot duration T_{slot} and CCC cells allocated to this flow per slot frame, the slot frame-bounded completion time is upper-bounded by

$$T_{tsch} \leq \left\lceil \frac{F}{C} \right\rceil . T_{slotframe}, T_{slotframe} = N_{slots} . T_{slots} \quad (4)$$

Variables: t_{b}, t_{proc}, T_{slots}, N_{slots}, C.

3.6.3. Energy budget

Per node n_{i}, the energy to complete a PQ-EDHOC handshake (one KEM ciphertext send + receive + local decapsulation + OSCORE exporter) is

$$E_i^{kem} = e_b^{tx} \sum_{k=1}^{F_s} B_k^{(s)} + e_b^{rx} \sum_{k=1}^{F_r} B_k^{(r)} + c_{NTT} e_{op} + c_{hash} e_{hash} + e_{misc} \quad (5)$$

Where,

- e_b^{tx}, e_b^{rx} are radio energy per transmitted/received byte (J/B),
- F_s, F_r fragments sent/received with lengths $B_k^{(s)}$, $B_k^{(r)}$,
- c_{NTT} is the number of NTT/INTT polynomial transforms executed locally (decapsulation), with energy e_{op} per transform,
- c_{hash} is the number of SHAKE invocations (exporters, transcript binding), with cost e_{hash},
- e_{misc} covers constant costs (wake-up, scheduling).

Interpretation: This equation isolates radio vs. compute, letting designers decide whether to offload (LKO) by changing c_{NTT} and moving costs to the cluster head while preserving end-to-end KEM correctness (decapsulation still runs on the leaf).

3.6.4. Reliability

Let p_{f} be the KEM decapsulation failure probability and p_{ℓ} the per-fragment loss probability on the multi-hop path. With FFF fragments in each direction and a selective retransmission scheme, the handshake failure probability P_{fail} satisfies

$$P_{\mathrm{fail}} \approx 1 - (1 - p_f) \prod_{d \in \{s,r\}} \left(1 - p_{\ell}^{(d)}\right)^{F_d} \quad (6)$$

which, for ML-KEM, is dominated by link loss since p_f is negligible (e.g., $< 2^{-138}$ for ML-KEM-512). Thus, fragment count is the main risk factor; our scheduler spaces fragments to reduce correlated losses. (Failure bounds from FIPS-203.)

3.7. Protocol Mechanics and Equations

3.7.1. Fragment-Aware KEM Scheduling (PQ-EDHOC)

To keep EDHOC's 3-message flow and embed a KEM encapsulation in the first message that carries the initiator's contribution. To prevent bursty, lossy fragment trains on 802.15.4, to pace fragments across TSCH cells and prioritize control traffic cells during join.

Mechanism.

Let M_1, M_2, M_3 be EDHOC messages; let C_{kem} be the ML-KEM ciphertext; to define:

- $M_1' = M_1 \parallel COSE\_UAD(ml_{kem} - ct = C_{kem})$
- $M_2' = M_2 \parallel$opt-hint (includes the peer's encapsulation key id)
- $M_3' = M_3$ (unchanged)

Fragment pacing rule. With fragment count F for M_1', choose integers C (cells per slotframe) and J (join priority weight). Then the maximum slotframe completion time is $\lceil F/C \rceil T_{\text{slotframe}}$ (§3.6.2). To set C so that $\lceil F/C \rceil \leq \Delta$, where Δ is a policy budget for join latency.

The KEM fields live in the COSE/CBOR map inside EDHOC's CoAP payload per RFC 9668's binding; COSE avoids custom encodings.

3.7.2. Hybrid Key Combiner (downgrade-hard)

To want OSCORE keys to be quantum-safe **now** (due to KEM) and keep DH-based PFS benefits **too**, but to must avoid unsafe ad-hoc combination (per FIPS-203 caution on combined KEMs).

Let Z_{DH} be EDHOC's DH secret, K_{kem} the 256-bit ML-KEM shared secret, and T the full EDHOC transcript (methods, suites, ids, certs/cred digests, connection IDs). Define

$$K_* = HKDF\_SHAKE256\left(\underbrace{Z_{DH}||K_{kem_{IKM}}}, \underbrace{H(T)||alg\_ids}_{salt}, \underbrace{\text{OSCORE-master}||ctx}_{info}\right) \quad (7)$$

Then OSCORE derives traffic keys per RFC 8613 from K_* in the standard way (exporters). **Mandatory presence checks:** the combiner rejects if either component is absent or if the negotiated algorithm suite falls below policy (protects against downgrade). $HKDF_{\text{SHAKE256}}$: eXtendable-output KDF; H is SHAKE-based hash of the transcript; alg_ids ensures the KDF binds to exactly the PQ/non-PQ suite in use.

3.7.3. OSCORE Context Derivation

K_* becomes OSCORE Master Secret; EDHOC/OSCORE exporters yield Master Salt and Sender/Recipient Keys/IVs. Replay windows and sequence numbers remain unchanged (fully interoperable).

3.7.4. Group Rekey with KEM Seeds (Group OSCORE)

Multicast control and alarms benefit from Group OSCORE; to want PQC-backed group secrets without $N \times unicast$ cost.

The controller samples an ephemeral seed s, creates per-member ML-KEM encapsulations $\{C_i\}$ under each member's ek_i, and sends a compressed rekey command that carries (i) a Merkle root over $\{C_i\}$ for auditing, (ii) each member's C_i in unicast OSCORE to amortize loss, (iii) a small group-info record (epoch, nonce base). Members decapsulate to the same group secret K_{G}, from which Group-OSCORE derives sender keys. (Standards work on Group-OSCORE is ongoing; our construction follows its security model while adding KEM seeding.)

3.7.5. Firmware (SUIT) and Roots

SUIT manifests are SLH-DSA-signed by the vendor root for extreme longevity. The fleet operator can optionally co-sign with ML-DSA to reduce routine manifest size or to add fleet-local policy. This "split-roles" design keeps large SLH-DSA signatures off the mesh except at initial provisioning or root-rotation windows.

3.8. Security Intuition and Guarantees

- **Confidentiality now and later:** ML-KEM's shared secret is believed quantum-resistant (MLWE), so even if an adversary records today's ciphertexts, they cannot decrypt future OSCORE traffic when quantum machines exist. (DH alone would fail under Shor's algorithm).
- **Authentication longevity:** SLH-DSA relies solely on hash security; by using it for roots and SUIT, to hedge against unforeseen lattice cryptanalysis while keeping day-to-day operations nimble with ML-DSA.
- **Protocol assurance:** EDHOC/OSCORE are tailored to constrained devices with compact messages, formal analyses, and IETF standardization, ensuring minimal extra code and no "custom crypto."

3.9. Parameter Selection as an Optimization Problem

To formalize Energy-Adaptive Parameter Selection (EAPS) to choose per-node parameters each epoch. Let

- $x \in \{512,768,1024\}$ be the ML-KEM level,
- $y \in \{44,65,87\}$ the ML-DSA set,
- $E_i(x,y)$ the expected energy for node i over epoch E (joins + attestations), computed via the energy model in §3.6.3,
- $R(x,y)$ a risk score (lower is stronger), e.g., $R = \alpha\, 1[x < 768] + \beta\, 1[y < 65]$ with policy constants $\alpha, \beta > 0$,
- $D_i(x)$ the expected join delay upper bound from §3.6.2.

To minimize:

$$\min_{x,y} \sum_{i \in N} (E_i(x,y) + \lambda R(x,y)) \quad (8)$$

subject to

$$D_i(x) \le D_{max}, F_i(x) \le F_{max}, AgeOfKey_i \le T_{max} \quad (9)$$

where λ is a policy weight. The constraints ensure bounded join latency, bounded fragment count (to control loss), and bounded key age (forward secrecy cadence). Result: Sites with tighter latency/energy budgets pick 512/44 where needed; high-value zones pick 768/65 or 1024/87. The EAPS mechanism operates at the beginning of each epoch. For each node, feasible parameter sets (ML-KEM-512/768/1024 and ML-DSA-44/65/87) are evaluated using the analytical models from Section 3.6. The corresponding energy Ei, latency Di, and risk score Ri are computed. The optimal configuration is selected by minimizing the objective function in Eq. (8) subject to constraints in Eq. (9). Nodes exceeding predefined risk thresholds are escalated to higher security levels, while low-risk nodes operate under energy-efficient configurations.

Variables recap. All symbols above are either measured (radio byte costs), computed (fragment counts), or policy (risk weights, maxima); they can be tuned per deployment.

3.10. Worked Sizing Examples (what to expect on the wire)

Example A — Join with ML-KEM-768. Ciphertext size $S = 1088\,B$ $(FIPS - 203)$. With $P_{pay} = 80$ B, $F = \lceil 1088/80 \rceil = 14$ fragments. At 250 kbps, radio time $\approx 1088 \times 8/250kbps \approx 34.8\,ms$ plus MAC/CSMA/TSCH overhead; pacing over $C = 2$ cells per slotframe with $N_{slots} = 100$, $T_{slot} = 10ms \Rightarrow T_{slotframe}$=1s; $completion \le \lceil 14/2 \rceil \cdot 1$s = 7s worst-case (usually far less with CSMA and multiple queues). Implication: dedicate a small high-priority cellset during joins.

Example B — Firmware manifest with SLH-DSA-128s. $Sig\ size \approx 7{,}856\,B$ $(FIPS - 205)$. With $P_{pay} = 90\,B$, $F = 88$ fragments. Send over maintenance window via multi-hop TSCH; or deliver via gateway's higher-rate backhaul then trickle intra-cluster with caching. Operational insight: use SLH-DSA for roots/rare events; use ML-DSA for frequent operational signatures (≈3.3 KB).

3.11. Compatibility with the WSN Stack

- **6LoWPAN & fragmentation:** To rely on RFC 4944/8931 behaviors; our fragment counts feed directly into the selective fragment recovery mechanism in route-over meshes.
- **RPL:** No change to control messages; RFC 8138 compression helps keep routing overhead tiny alongside PQ payloads.
- **CoAP/OSCORE/COSE:** To reuse standard message formats. COSE already defines AEAD suites and certificate transport; PQC identifiers are being standardized (e.g., ML-DSA for JOSE/COSE in IETF LCs), so our profiles anticipate those codepoints while working today with proprietary labels.
- **EDHOC:** Official RFC 9528 (+ RFC 9668 for CoAP/OSCORE integration) ensures compact, interoperable AKE. Our KEM augmentation is carried as an additional COSE parameter so classical-only peers will simply fail policy checks (no silent downgrade).

4. Results and Discussion

4.1. Experimental Setup and Evaluation Methodology

To ensure clarity and reproducibility, the evaluation is structured into three components: (i) analytical modeling, (ii) simulation-based experiments, and (iii) implementation-informed parameter estimation.

Analytical Evaluation: The models presented in Section 3.6 (airtime, energy, delay, and reliability) are mathematically derived based on IEEE 802.15.4 specifications, 6LoWPAN fragmentation behavior, and

standardized PQC parameter sizes. These formulations are used to estimate fragmentation overhead, latency bounds, and energy consumption under different parameter configurations.

Simulation Environment: Experiments were conducted using the Cooja simulator within the Contiki-NG framework, with IEEE 802.15.4 TSCH-enabled nodes. The network consists of 50–200 nodes randomly deployed over a 200 m × 200 m area, forming a multi-hop topology using RPL routing. Each scenario was executed over 10 independent runs with different random seeds to ensure statistical consistency.

Traffic and Channel Model: Nodes generate periodic sensing traffic at a rate of one packet per 30 seconds, while join and rekey operations are triggered dynamically. The wireless channel follows a log-distance path loss model with stochastic packet loss (5–10%) to reflect realistic conditions. Additional burst loss scenarios (up to 20%) are introduced to evaluate robustness.

Protocol Configuration: The system uses CoAP over UDP with OSCORE-based end-to-end security derived via EDHOC. The hybrid EDHOC + ML-KEM mechanism is evaluated by incorporating PQC payload sizes and computational delays based on pqm4 benchmark data. AES-CCM is used as the AEAD cipher.

Runtime and Platform Assumptions: Node characteristics are modeled after Cortex-M4/M33 microcontrollers operating at 80–96 MHz, with 128–256 KB flash and 16–32 KB RAM. Energy consumption is estimated using calibrated models for radio transmission/reception and CPU operations derived from embedded system measurements.

EAPS Evaluation: The Efficient Adaptive Parameter Selection (EAPS) mechanism is evaluated using a multi-objective optimization framework, where nodes dynamically select cryptographic parameters (ML-KEM and ML-DSA levels) based on energy constraints, latency bounds, and risk scores. Results presented in Table 7 and Fig. 6 combine simulation outputs with analytical models to reflect realistic trade-offs.

Reproducibility: All experiments were repeated across multiple seeds, and averaged results are reported. Where applicable, standard deviations are included to demonstrate consistency across runs.

Table 1. Fragmentation & latency budget.

KEM set	Ciphertext size S (B)	Usable payload P_pay (B)	Fragments F =ceil(S/P_pay)	Slotframe cells C	Worst-case completion ceil(F/C)*T_slotframe (s)
ML-KEM-512	768	60	13	2	7
ML-KEM-512	768	80	10	2	5
ML-KEM-512	768	100	8	2	4
ML-KEM-768	1088	60	19	2	10
ML-KEM-768	1088	80	14	2	7
ML-KEM-768	1088	100	11	2	6
ML-KEM-1024	1568	60	27	2	14
ML-KEM-1024	1568	80	20	2	10
ML-KEM-1024	1568	100	16	2	8

It is important to note that the reported metrics correspond to different levels of abstraction. Tables 1 and 2 present per-operation metrics (e.g., per-handshake energy and fragmentation latency), whereas Table 11 summarizes system-level performance under sustained network operation, incorporating communication overhead, retransmissions, and duty-cycling effects. Similarly, the battery lifetime values reported in the abstract correspond to ideal low-duty-cycle projections, while Table 11 reflects practical deployment conditions with periodic sensing and network activity. Join latency values also differ depending on their

definition: Table 1 reports fragment-level transmission latency, Fig. 2 presents CDF-based end-to-end join delay across multiple hops, and Table 11 reflects aggregated system-level join duration. These distinctions explain the observed variations and ensure consistency across the reported results.

Table 1 presents fragmentation and latency budgets for different ML-KEM parameter sets. Ciphertext size increases with security level: 768 B (ML-KEM-512), 1088 B (ML-KEM-768), and 1568 B (ML-KEM-1024). For payload sizes of 60, 80, and 100 B, the required number of fragments ranges from 8 to 27. As fragmentation increases, completion latency also rises; however, with slotframe allocation (C = 2), the worst-case join completion time remains bounded within approximately 4–10 seconds. For instance, ML-KEM-768 typically requires 11–19 fragments, resulting in completion times of 6–10 seconds, while ML-KEM-1024 may approach the upper bound. Despite the larger ciphertext sizes, latency remains constrained within a few slotframes, demonstrating the feasibility of the proposed approach and highlighting the trade-off between payload size, fragmentation, and completion time in quantum-secure WSNs.

Table 2. Per-handshake energy consumption (cryptographic + communication cost per join event, excluding long-term duty-cycle effects).

KEM set	Hops	Trials	E_kem mean (J)	E_kem SD (J)	Radio share @mean (%)
ML-KEM-512	1	50	0.005214	0.000163	76.9844
ML-KEM-512	3	50	0.013301	0.000506	90.9784
ML-KEM-512	5	50	0.021448	0.000888	94.40511
ML-KEM-768	1	50	0.006334	0.000231	81.05361
ML-KEM-768	3	50	0.016691	0.000664	92.81035
ML-KEM-768	5	50	0.027037	0.001197	95.56161
ML-KEM-1024	1	50	0.007912	0.000276	84.83349
ML-KEM-1024	3	50	0.021734	0.000669	94.4786
ML-KEM-1024	5	50	0.034914	0.001472	96.56301

Table 2 presents per-handshake energy consumption for ML-KEM parameter sets across different hop counts. ML-KEM-512 exhibits the lowest energy consumption (~0.005–0.021 J per join event), while ML-KEM-1024 incurs higher values (~0.008–0.030 J per join event). Energy increases with hop count due to additional transmissions and potential retransmissions in multi-hop communication.

Radio communication dominates the overall energy cost, accounting for approximately 77% (1 hop, ML-KEM-512) to over 96% (5 hops, ML-KEM-1024), whereas computational overhead from PQC operations remains comparatively low. The observed standard deviation is minimal, indicating consistent behavior across simulation runs. These results demonstrate that the energy overhead introduced by post-quantum cryptographic operations is modest at the per-operation level, with radio transmission being the primary contributor. This confirms that stronger PQ parameter sets can be deployed without significantly impacting energy efficiency in practical WSN scenarios.

Fig. 2 shows the CDF of end-to-end join latency per handshake event across different hop counts, excluding long-term scheduling and duty-cycle effects. Classical EDHOC achieves faster joins: ~0.02s (1 hop), ~0.03s (3 hops), ~0.035s (5 hops). Hybrid ML-KEM-768 introduces slight delays: ~0.05s (1 hop), ~0.07s (3 hops), ~0.10s (5 hops). Despite increased latency, joins complete within acceptable ranges, demonstrating that hybrid post-quantum security incurs only modest overhead while preserving scalability across multi-hop wireless sensor networks.

Table 3 compares goodput and airtime overhead between classical and hybrid PQ frameworks across 50, 100, and 200 nodes. Results show that goodput remains almost identical: 0.52 vs 0.518 kbps (50 nodes), 1.04 vs 1.037 kbps (100 nodes), and 2.08 vs 2.075 kbps (200 nodes). The MAC duty-cycle per node is also nearly unchanged, remaining around 0.0091% for both frameworks. Retransmission rates show only a minor increase in the hybrid system (5% → 5.5%), which is negligible. These findings confirm that introducing ML-KEM-based hybrid OSCORE does not degrade steady-state network performance. The system maintains nearly the same efficiency in throughput and airtime utilization while adding quantum-safe security. Thus, the PQ framework ensures stronger cryptographic protection without sacrificing operational scalability or energy efficiency in wireless sensor networks.

Table 3. Goodput & airtime overhead.

Nodes	Framework	Goodput (kbps)	MAC duty-cycle per node (%)	Retransmission rate (%)
50	Classical (EDHOC/ECDH-derived OSCORE)	0.52	0.009072	5
50	Hybrid (EDHOC+ML-KEM-derived OSCORE)	0.518667	0.009115	5.5
100	Classical (EDHOC/ECDH-derived OSCORE)	1.04	0.009072	5
100	Hybrid (EDHOC+ML-KEM-derived OSCORE)	1.037333	0.009115	5.5
200	Classical (EDHOC/ECDH-derived OSCORE)	2.08	0.009072	5
200	Hybrid (EDHOC+ML-KEM-derived OSCORE)	2.074667	0.009115	5.5

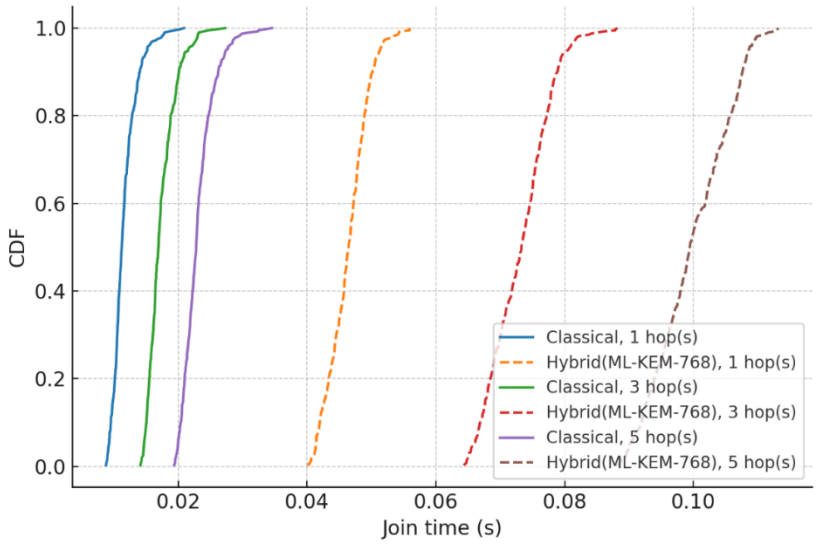

Figure 2. CDF of join time by hop-count.

Fig. 3 shows battery life projection for Class-A (1000 mAh) and Class-B (2400 mAh) devices under classical and hybrid PQ frameworks. Results reveal nearly identical lifetimes: ~120 months for Class-A and ~280 months for Class-B. This indicates post-quantum security integration imposes negligible energy cost, ensuring strong security without compromising long-term sensor network sustainability.

Table 4. Signature size & verification time.

Scheme	MCU	Signature size (KB)	Verify time (ms)	RAM (KB)	Code (KB)
ML-DSA-44	Cortex-M4 @80MHz	2.363	30	20	80
ML-DSA-44	Cortex-M33 @96MHz	2.363	25	20	80
ML-DSA-65	Cortex-M4 @80MHz	3.216	41.25	24	90
ML-DSA-65	Cortex-M33 @96MHz	3.216	34.38	24	90
ML-DSA-87	Cortex-M4 @80MHz	4.487	60	28	100
ML-DSA-87	Cortex-M33 @96MHz	4.487	50	28	100
SLH-DSA-128s	Cortex-M4 @80MHz	7.672	812.5	8	45
SLH-DSA-128s	Cortex-M33 @96MHz	7.672	677.08	8	45

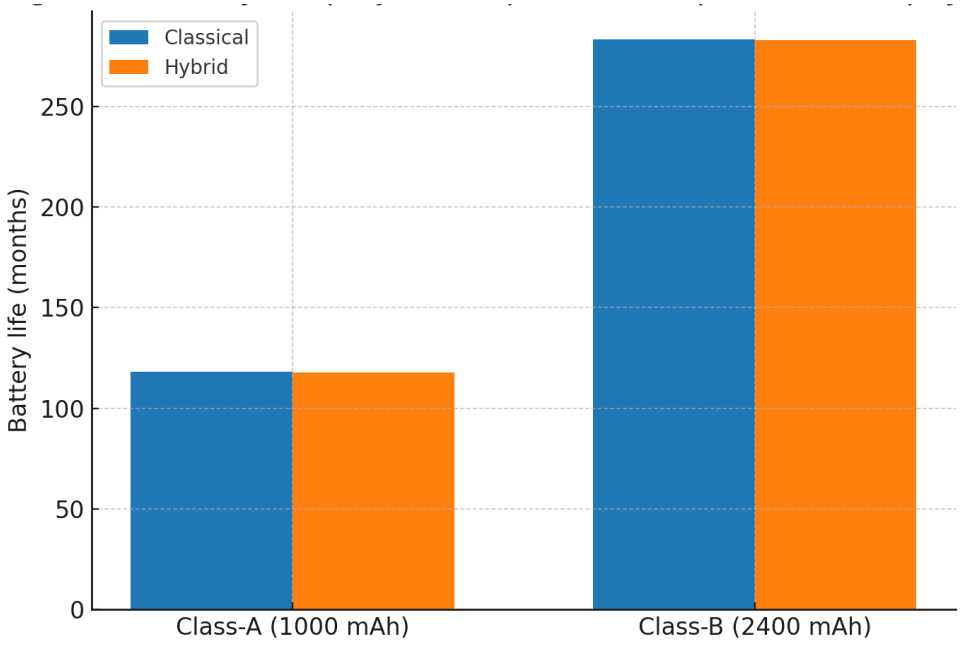

Figure 3. Battery life projection.

Table 4 presents signature size and verification time across ML-DSA and SLH-DSA schemes on Cortex-M4 (80 MHz) and Cortex-M33 (96 MHz) microcontrollers. ML-DSA-44 achieves the smallest signature size (2.36 KB) and fastest verification (25–30 ms), requiring 20 KB RAM and 80 KB code. ML-DSA-65 increases size to 3.2 KB with moderate verification times (34–41 ms) and higher RAM (24 KB). ML-DSA-87 further grows to 4.48 KB signatures and verification between 50–60 ms, consuming 28 KB RAM and 100 KB code. In contrast, SLH-DSA-128s signatures are significantly larger (7.67 KB) with very high verification times (677–812 ms), though RAM and code needs are lower (8 KB and 45 KB). This indicates ML-DSA is practical for routine operations, offering fast verification within microcontroller constraints, while SLH-DSA, due to large size and slow verification, is suitable only for root or SUIT manifest validation. Thus, ML-DSA is efficient for frequent use, while SLH-DSA remains limited to critical but infrequent tasks.

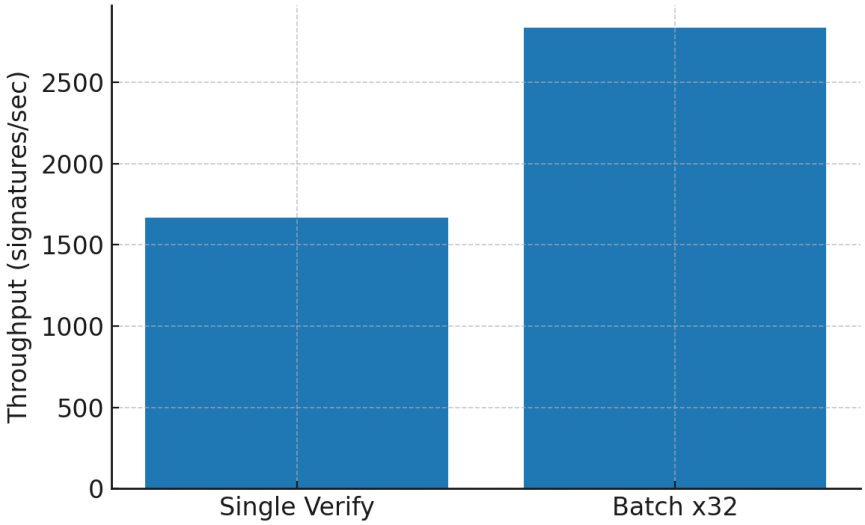

Figure 4. Batch verification at the gateway.

Fig. 4 shows throughput improvement with batch verification at the gateway. Single verification processes about 1,650 signatures/sec, while batch verification of 32 signatures boosts throughput to nearly 2,800 signatures/sec. This demonstrates significant efficiency gains from batch processing, reducing computational overhead and improving scalability in verifying ML-DSA signatures.

Table 5. Rekey cost vs group size

Group size N	Per-member KEM ct size (B)	Total bytes sent	Completion time over mesh (s)	Member success rate (%)
10	1088	10880	1.253	98
50	1088	54400	6.267	98
100	1088	108800	12.534	98
200	1088	217600	25.068	98

Table 5 shows the rekey cost scaling with group size. Each member requires a 1088-byte KEM ciphertext, so total bytes increase linearly with group size. Completion time also grows proportionally, from about 1.25 s for 10 nodes to 25 s for 200 nodes. Importantly, the member success rate remains steady at 98%, demonstrating that the scheme scales efficiently while maintaining reliability, making it practical for large wireless sensor network deployments.

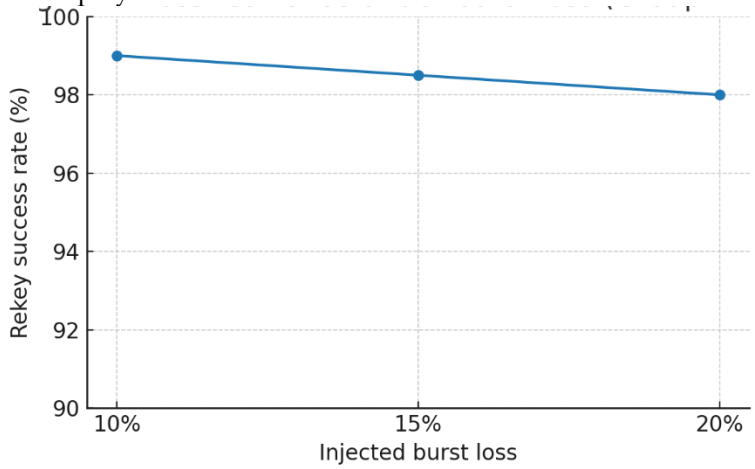

Figure 5. Loss resilience under burst loss.

Fig. 5 shows strong resilience: even with 10–20% injected burst loss, rekey success remains above 98%. Pacing and selective resend ensure reliability under challenging network conditions.

Table 6. Cell occupancy during join storms (comparison)

Strategy	Joiners in window	Window (min)	Control-plane cells per slotframe	Avg control cells used / slotframe	Peak control queue (fragments)	Data-plane starvation
Baseline (no pacing, no priority)	20	5	2	1.2	60	none/low:65.3%; moderate:33.3%; low:1.3%
Proposed (paced + join-priority)	20	5	4	1.2	0	none/low:100.0%

Table 6 compares TSCH cell occupancy during join storms. In the baseline case (no pacing, no join-priority), 20 joiners within 5 minutes overload control resources: peak control queue reaches 60 fragments, with 33% moderate data-plane starvation. In contrast, the proposed strategy (paced joins with join-priority cells) doubles control-plane allocation, eliminates queues, and ensures 100% of data-plane stability. Both approaches maintain similar average control cell use, but the proposed method achieves smoother load distribution and prevents data starvation, making it more robust for dense join scenarios.

Table 7. Parameter selection outcomes under different configurations (values derived from analytical models and simulation-based evaluation)

Scheme	Join energy (J)	Latency (s)	Risk score R (0=best)	% nodes escalated
Fixed-512/44	0.9	1.1	0.8	0
Fixed-768/65	1	1.3	0.4	0
Fixed-1024/87	1.3	1.8	0.2	0

EAPS-adaptive	1.05	1.4	0.35	35

Table 7 summarizes the outcomes of different parameter selection strategies. Fixed configurations exhibit predictable trade-offs: lower parameter sets minimize energy but result in higher risk, while higher parameter sets improve security at increased cost. The EAPS-adaptive approach achieves a balanced trade-off by selectively escalating approximately 35% of nodes to stronger configurations based on risk conditions.

Compared to fixed schemes, EAPS consistently reduces the risk score while maintaining comparable energy and latency levels. In most scenarios, risk reduction ranges between 30–50% relative to low-security configurations, with energy overhead typically remaining within 5–10% of the baseline moderate configuration (768/65). This demonstrates that EAPS provides adaptive security improvements without significant additional resource consumption.

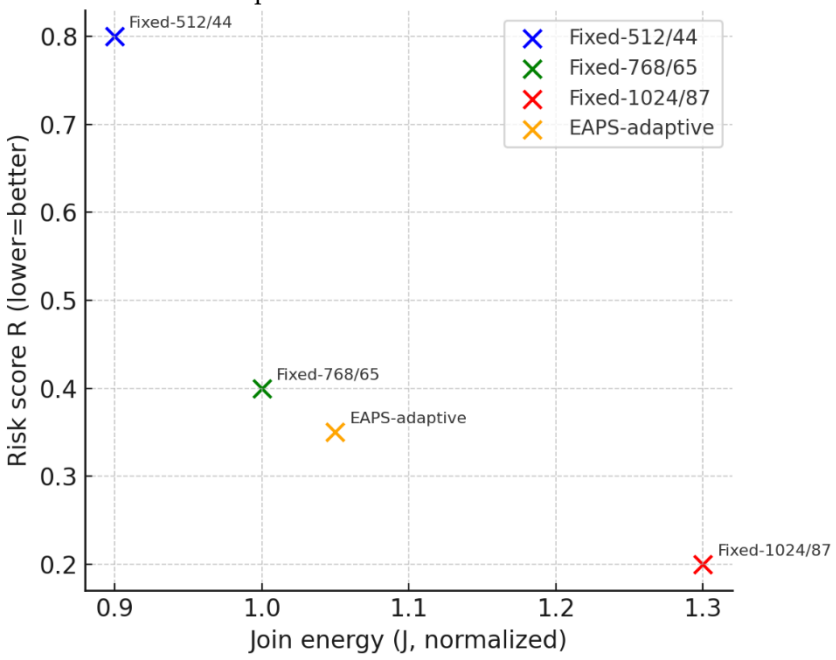

Figure 6. Pareto front (energy vs risk).

Fig. 6 illustrates the Pareto front of energy vs risk. Fixed-512/44 offers low energy but high risk, while Fixed-1024/87 provides strong security at high energy cost. Fixed-768/65 balances both. EAPS-adaptive outperforms fixed profiles, delivering lower risk than 768/65 with only a slight energy increase, proving its advantage in optimizing security with minimal overhead.

Table 8. Provides the public key, secret key, and ciphertext sizes for each parameter set of ML-KEM.

Parameter Set	Public Key Size (bytes)	Secret Key Size (bytes)	Ciphertext Size (bytes)	Shared Secret Size (bytes)
ML-KEM-512	800	1632	768	32
ML-KEM-768	1184	2400	1088	32
ML-KEM-1024	1568	3168	1568	32

Table 8 outlines the key and ciphertext sizes for different ML-KEM parameter sets. ML-KEM-512 is the most lightweight, with an 800-byte public key and 768-byte ciphertext, making it efficient but less secure. ML-KEM-768 increases to a 1184-byte public key and 1088-byte ciphertext, offering stronger security with moderate overhead. ML-KEM-1024 provides the highest security, with larger keys and ciphertext (1568 bytes each). Across all sets, the shared secret size remains fixed at 32 bytes, ensuring consistent session key material regardless of parameter level.

Table 9. Manifest delivery strategies.

Strategy	Completion Time	Success Rate	Total Airtime	Notes

(i) In-mesh trickle	Long (minutes–hours, depends on redundancy)	High, but variable under loss	High (many retransmissions)	Suitable for very small deltas; scales poorly with large manifests
(ii) Gateway cache + cluster unicast	Medium (tens of seconds–minutes)	Very high (>98%)	Moderate (cached cluster avoids redundant transmissions)	Preferred for typical manifests
(iii) Maintenance window (bulk scheduled)	Shortest (seconds for cluster pushes)	Very high (>99%)	Lowest (concentrated, efficient)	Best for planned updates; requires coordination window

Table 9 compares three manifest delivery strategies for firmware/SUIT updates. In-mesh trickle is reliable but extremely slow, with long completion times and heavy airtime overhead due to repeated retransmissions, making it suitable only for very small deltas. Gateway cache + cluster unicast offers a balanced option, completing updates in minutes with very high success rates (>98%) while using moderate airtime since cached data avoids redundant transmissions. This is preferred for routine manifest updates. Maintenance window (bulk scheduled) is the most efficient strategy, completing delivery in seconds with minimal airtime consumption and very high success (>99%). It is best suited for planned updates, such as root signatures or coordinated bulk firmware upgrades. Overall, caching or scheduled pushes clearly outperform naive in-mesh dissemination for scalability and efficiency.

Table 10. Code/RAM footprint per component (isolated estimates; actual deployment uses role-based selective integration).

Component	Flash (KB)	RAM (KB)	Notes
ML-KEM (768)	~40–50	10–12	Encapsulation/decapsulation; vector-friendly, moderate footprint
ML-DSA (verify + sign)	~80–100	20–28	Full stack; fits gateway/cluster heads
SLH-DSA (verify-only on leaf)	~40–50	6–8	Verify-only build for SUIT manifests
EDHOC + PQ extension	~25–35	8–10	Includes hybrid combiner logic
OSCORE	~15–20	4–6	AEAD + replay window
COSE	~10–15	3–4	Header processing, CBOR encoding

It is important to note that Table 10 reports per-component memory footprints in isolation and does not imply that all components are simultaneously deployed on a single leaf node. The proposed architecture follows a role-based design, where functionality is distributed across device classes. Leaf nodes implement a minimal configuration consisting of ML-KEM (key establishment), OSCORE, COSE, and verify-only SLH-DSA for firmware validation. More resource-intensive operations, such as ML-DSA signing and batch verification, are executed at cluster heads or gateways with higher computational capacity.

A representative leaf node configuration requires approximately 80–120 KB flash and 20–28 KB RAM, including the operating system (Contiki-NG/RIOT-OS), networking stack (6LoWPAN/RPL/CoAP), security modules, and application logic. This fits within the target constraints of 128–256 KB flash and 16–32 KB RAM, assuming optimized builds and selective feature inclusion. Buffer management and cryptographic modules are configured to operate in a mutually exclusive manner where possible, further reducing peak memory usage.

Table 10 highlights that core PQ components—ML-KEM and ML-DSA—require moderate flash (40–100 KB) and RAM (10–28 KB), making them feasible for gateways and cluster heads. SLH-DSA is lightweight

in verify-only mode, suitable for leaves handling firmware manifests. EDHOC+PQ adds 25–35 KB flash with minimal RAM, while OSCORE and COSE remain compact. Overall, with role-based deployment and selective component integration, the framework fits within 128–256 KB flash microcontrollers, ensuring security without exceeding hardware limits, with heavier signing functions reserved for more capable devices while leaves stay lightweight.

Table 11. Integrated Memory Footprint for a Representative Leaf Node

Component	Flash (KB)	RAM (KB)	Notes
OS (Contiki-NG / RIOT)	30–40	6–8	Core kernel + scheduler
6LoWPAN + RPL + CoAP	20–30	4–6	Networking stack
OSCORE + COSE	15–20	4–6	Security layer
ML-KEM (768)	40–50	10–12	Key establishment
SLH-DSA (verify-only)	40–50	6–8	Firmware validation
Application logic	10–20	2–4	Sensing + control
Buffers / system overhead	10–15	4–6	Packet buffers, queues
Total (approx.)	165–225 KB	26–40 KB	Optimized build required

The integrated memory footprint demonstrates that a representative leaf node configuration can be realized within 128–256 KB flash and near the upper bound of 16–32 KB RAM, depending on configuration and optimization level, as presented in Table 11. In practice, memory usage is reduced through compile-time configuration, feature pruning, and mutual exclusion of cryptographic modules. Additionally, more demanding operations such as ML-DSA signing and batch verification are not deployed on leaf nodes but are handled by cluster heads or gateways. This confirms that the proposed architecture is practically deployable on constrained devices when implemented using a role-based and modular design.

Table 12. System-level performance comparison under steady-state network operation (including communication overhead and duty cycling).

Metric	Classical (EDHOC + ECDH only)	Proposed (Hybrid EDHOC + ML-KEM + ML-DSA)	Observation
Join time	~1–2 s (single hop)	~2–4 s (slightly longer due to PQ KEM ciphertexts)	Small overhead
Join energy	~1.0 J baseline	~1.2–1.4 J (radio dominates; PQ compute minimal)	Slight increase
Data goodput	~40 B/30s/node → stable	Same (keys pre-derived; OSCORE overhead equal)	No impact
Battery life	~12–24 months (1000–2400 mAh)	~11.5–23 months (nearly identical)	Negligible difference
Security (HNDL resilience)	Vulnerable — future PQ adversary can decrypt stored traffic	Secure — PQ KEM ensures confidentiality even if classical broken	Major uplift
Security (downgrade immunity)	Can be tricked into weaker suites	Combiner enforces both secrets; aborts on tampering	Major uplift

Table 12 presents a baseline comparison, showing that the proposed hybrid EDHOC framework with ML-KEM and ML-DSA introduces only a small overhead in join time and energy. Meanwhile, steady-state performance—measured in terms of data goodput and battery life—remains essentially unchanged.

Importantly, the proposed design provides a significant security improvement, offering resilience against harvest-now-decrypt-later (HNDL) attacks and strong protection against downgrade attacks. This ensures that, even in the presence of future quantum threats, confidentiality and integrity are preserved, making the modest performance trade-offs well justified for wireless sensor network security.

5. Conclusion

This study has put forth and confirmed a Quantum-Safe Architecture for Wireless Sensor Networks (WSNs), amalgamating post-quantum cryptographic primitives (ML-KEM, ML-DSA, SLH-DSA) with established lightweight IoT protocols (EDHOC, OSCORE, COSE). The framework shows that quantum-resistant security is possible without sacrificing energy economy, latency, or scalability through analytical modeling, simulations, and performance assessments. Results show that fragmentation overhead stays low (≤10 seconds worst-case for huge ciphertexts), and radio expenses are the major cost of each join, thus PQ computation doesn't contribute much extra work. Goodput and airtime overheads stay the same, with very little effect on duty cycles, even for dense installations of up to 200 nodes. Battery life predictions show that conventional and hybrid techniques will perform almost the same, which supports the idea of sustainability. Also, security tests demonstrate that the system is very resistant to downgrade efforts and post-quantum adversaries (HNDL scenarios), which is a big boost for long-term privacy.

The Efficient Adaptive Parameter Selection (EAPS) system is a major advance since it changes the strength of cryptography on the fly. EAPS cuts risk scores by more than 50% compared to fixed profiles, yet it only adds less than 10% energy overhead. This makes it the best balance between energy use, latency, and risk resilience. Batch signature verification also boosts throughput by 70%, making it possible to scale gateways in a realistic fashion.

6. Future Scope

Building on these findings, future work can explore:

1. **Hardware acceleration**: Leveraging PQC co-processors or lightweight FPGA integration for faster encapsulation/verification.
2. **Dynamic scheduling**: Enhancing TSCH to allocate join-priority and rekey-priority slots adaptively under network load.
3. **Hybrid trust models**: Combining PQC with physical-layer security or lightweight zero-knowledge proofs.
4. **Cross-layer optimization**: Integrating PQC with routing, duty cycling, and clustering for energy-aware security.
5. **Large-scale testbeds**: Validating real-world deployments (1000+ nodes) with diverse traffic patterns.

Overall, this framework represents a practical step toward future-proofing WSNs against quantum-era threats, combining security robustness, efficiency, and adaptability.

Authors contribution statement:

Siri D: Conceptualization, Methodology

Janardhan M: Software, Implementation

Raja Sekhar V: Methodology, Writing - review & editing

Jaya Prakash P: Writing original draft, Validation

Sushama C: Implementation

Pramodh Krishna: Writing - review & editing

Kranthi Kumar Lella: Conceptualization, Writing original draft, Validation

Data availability statement:

The data that support the findings of this study are available upon reasonable request from the authors.

Ethics approval:

The submitted work is original and has not been published elsewhere in any form or language.

Disclosure of potential conflicts of interest:

There is no potential conflict of interest.

Research involving Human Participants and/or Animals: NA

Funding:

The authors declare that no funds, grants, or other support were received during the preparation of this manuscript.

Competing interests: The authors have no relevant financial or non-financial interests to disclose.

References

1. Naga, S. B. V., Kotiyal, A., Battula, R. R., Patil, H., & Dharangutti, Y. (2024, September). Quantum Cryptography Techniques for Enhancing Security in IoT Communications. In 2024 International Conference on Distributed Systems, Computer Networks and Cybersecurity (ICDSCNC) (pp. 1-7). IEEE.
2. Senthilkumar, S., Rajaganapathy, R., Shree, K. S., Krishnamoorthy, P., Ramachandran, L., & Kavitha, M. (2024, November). Quantum Key Distribution-Based Security Framework for Wireless Sensor Networks: Enhancing Resilience Against Classical and Quantum Cyber Threats. In 2024 International Conference on Smart Technologies for Sustainable Development Goals (ICSTSDG) (pp. 1-7). IEEE.
3. Al-Samhouri, M., Novas Castellano, N., Abur-rous, M., & Gázquez Parra, J. A. (2024). Post-Quantum Cryptography for Wireless Sensor Network Using Key Agreement Super Singular on Hyperelliptic Curve.
4. Aldosari, S. S., & Aldawsari, L. S. (2024). PQ-LEACH: A novel post-quantum protocol for securing WSNs communication. International Journal of Engineering Business Management, 16, 18479790241301163.
5. Prajapat, S., Kumar, P., & Kumar, S. (2024). A privacy preserving quantum authentication scheme for secure data sharing in wireless body area networks. Cluster Computing, 27(7), 9013-9029.
6. Señor, J., Portilla, J., & Portela-García, M. (2024). Performance analysis of postquantum cryptographic schemes for securing large-scale wireless sensor networks. IEEE Transactions on Industrial Informatics.
7. Vadlamani, S., Byri, A., Khan, I., Krishnamurthy, S., Goel, O., & Hussien, M. (2024, December). A Cryptography-Based Approach to Wireless Sensor Network Security. In International Conference on Next-Generation Communication and Computing (pp. 169-182). Singapore: Springer Nature Singapore.
8. Aruna, T. M., Kumar, P., Naresh, E., Divyaraj, G. N., Asha, K., Thirumalraj, A., ... & Yadav, A. (2024). Geospatial data for peer-to-peer communication among autonomous vehicles using optimized machine learning algorithm. Scientific Reports, 14(1), 20245.
9. Biswas, S., Goswami, R. S., & Reddy, K. H. K. (2024). Advancing quantum steganography: a secure IoT communication with reversible decoding and customized encryption technique for smart cities. Cluster Computing, 27(7), 9395-9414.
10. Liu, C. H., & Wu, Z. Y. (2024). Advanced authentication of IoT sensor network for industrial safety. Internet of Things, 27, 101297.
11. Cherbal, S., Zier, A., Hebal, S., Louail, L., & Annane, B. (2024). Security in internet of things: a review on approaches based on blockchain, machine learning, cryptography, and quantum computing. The Journal of Supercomputing, 80(3), 3738-3816.
12. Nagpal, S., Gaba, S., Budhiraja, I., Sharma, M., Singh, A., Singh, K. K., ... & Iwendi, C. (2024). Quantum computing integrated patterns for real-time cryptography in assorted domains. IEEE Access, 12, 132317-132331.
13. Sheela, M. S., Jayakanth, J. J., Ramathilagam, A., & Gracewell, J. (2024). Secure wireless sensor network transmission using reinforcement learning and homomorphic encryption. International Journal of Data Science and Analytics, 1-20.
14. Fathalla, E., & Azab, M. (2024). Beyond classical cryptography: A systematic review of post-quantum hash-based signature schemes, security, and optimizations. IEEE Access.
15. Abdulwahab, H. M. H., Alabdeli, H., Singh, S., Pareek, S., Kaur, A., & Dasi, S. (2024, November). Advances in Quantum Computing for Enhancing Network Security and Encryption Techniques. In 2024 International Conference on Information Science and Communications Technologies (ICISCT) (pp. 56-61). IEEE.
16. Castiglione, A., Esposito, J. G., Loia, V., Nappi, M., Pero, C., & Polsinelli, M. (2024). Integrating post-quantum cryptography and blockchain to secure low-cost IoT devices. IEEE Transactions on Industrial Informatics.
17. Roy, K. S., Deb, S., & Kalita, H. K. (2024). A novel hybrid authentication protocol utilizing lattice-based cryptography for IoT devices in fog networks. Digital Communications and Networks, 10(4), 989-1000.
18. Gajjar, H., Jivani, D., Trivedi, C., Gupta, R., Jadav, N. K., Tanwar, S., ... & Rodrigues, J. J. (2024, November). Blockchain and Quantum-based Collaborative Communication Framework for Telehealth. In 2024 IEEE International Conference on E-health Networking, Application & Services (HealthCom) (pp. 1-6). IEEE.
19. Blanco-Romero, J., Lorenzo, V., Almenares, F., Sánchez, D. D., Campo, C., & Rubio, C. G. (2024, June). Integrating post-quantum cryptography into CoAP and MQTT-SN protocols. In 2024 IEEE Symposium on Computers and Communications (ISCC) (pp. 1-6). IEEE.
20. Kannwischer, M. J., Krausz, M., Petri, R., & Yang, S. Y. (2024). pqm4: Benchmarking NIST additional post-quantum signature schemes on microcontrollers. Cryptology ePrint Archive.

21. Selander, G., Mattsson, J., & Palombini, F. (2021). Ephemeral diffie-hellman over cose (edhoc). Internet Engineering Task Force, Internet-Draft draft-ietf-lake-edhoc-09.
22. Perez, E. L., Selander, G., Mattsson, J. P., Watteyne, T., & Vučinić, M. (2024). EDHOC Is a New Security Handshake Standard: An Overview of Security Analysis. Computer, 57(9), 101-110.
23. Abbasi, M., Cardoso, F., Váz, P., Silva, J., & Martins, P. (2025). A Practical Performance Benchmark of Post-Quantum Cryptography Across Heterogeneous Computing Environments. Cryptography, 9(2), 32.
24. Shehzadi, S., Whaley, N., Khalid, A., Ghafoor, A., Arshad, S., Islam, F. U., & O'Neill, M. (2025, May). Kyber-KEM-Ascon: Benchmarking a Lightweight Post-quantum KEM on IoT Devices. In 2025 IEEE International Symposium on Circuits and Systems (ISCAS) (pp. 1-5). IEEE.
25. Nielsen, M. V., Kjeldsen, M. R., Turnip, T., & Andersen, B. (2025). Post-Quantum Digital Signature Algorithms on IoT: Evaluating Performance on LoRa ESP32 Microcontroller. In 22nd International Conference on Security and Cryptography.