Security vs. Endurance: An Adaptive Post-Quantum Framework for Power-Constrained UAVs

Ben Trovato*
G.K.M. Tobin*
trovato@corporation.com
webmaster@marysville-ohio.com
Institute for Clarity in Documentation
Dublin, Ohio, USA

Abstract

The operational integrity of Unmanned Aerial Vehicles (UAVs) depends on quantum-resistant, high-performance command and control (C2) links. However, the integration of Post-Quantum Cryptography (PQC) into resource-constrained UAV platforms presents a critical challenge due to its significant computational overhead, a problem magnified when the system is under duress from ancillary tasks like on-board threat detection. This paper introduces a complete PQC-secured C2 stack featuring an adaptive scheduler designed for the graceful degradation of cryptographic policies in response to real-time system telemetry.

We present a comprehensive performance evaluation across 21 distinct PQC suites (with AES-GCM and ChaCha20-Poly1305 cipher variants) under three distinct computational loads: a baseline scenario, concurrent operation with a heuristic-based system load designed to simulate lightweight XGBoost-equivalent DDoS detection, and a heavyweight Transformer-equivalent detection model (Time-Series Transformer, TST). The proposed framework executes on a Raspberry Pi 4 companion computer and provides runtime cryptographic agility through standardized NIST PQC suites (ML-KEM, ML-DSA, FN-DSA (Falcon), SLH-DSA).

Our results reveal that while handshake latencies vary by over 100×—from 8 ms for lattice-based schemes (ML-KEM with Falcon) to over 1600 ms for code-based schemes (Classic-McEliece with SPHINCS$^+$) under heavy load—the steady-state power consumption remains remarkably stable, varying by less than 10% (4.1–4.7 W) across all scenarios. This core finding proves that the primary operational cost of PQC is in transient latency and link reliability, not sustained energy drain. Our cross-scenario analysis establishes that static cryptographic policies are operationally untenable under realistic computational loads, positioning adaptive security as a fundamental prerequisite for any mission-critical UAV system in the post-quantum era. The architecture provides a data-driven foundation for mission-aware, quantum-resilient UAV communication.

CCS Concepts

• **Security and privacy** → **Network security**; • **Computer systems organization** → *Embedded systems*; • **General and reference** → Performance.

Keywords

UAV security, MAVLink, Post-quantum cryptography, Lightweight cryptography, Reinforcement learning, DDoS detection, Edge scheduling

ACM Reference Format:
Ben Trovato and G.K.M. Tobin. 2025. Security vs. Endurance: An Adaptive Post-Quantum Framework for Power-Constrained UAVs. In *MakeBelieve '25: ACM Symposium on Fake Proceedings, October 16–18, 2025, Hyderabad, India.* ACM, New York, NY, USA, 11 pages. https://doi.org/XXXXXXX.XXXXXXX

1 Introduction

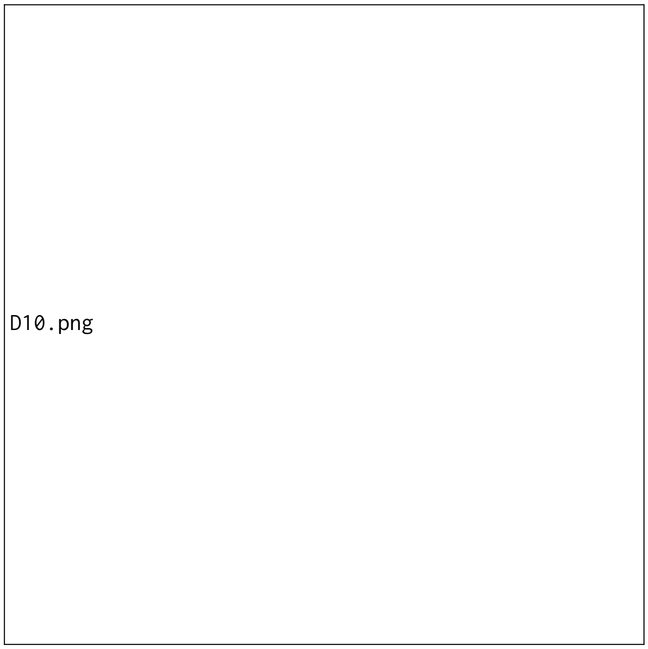

Figure 1: Conceptual UAV security system showing GCS, PQC proxy, scheduler, DDoS detector, and flight controller.

*Both authors contributed equally to this research.

Conference '25, Hyderabad, India
© 2025 ACM.
ACM ISBN 978-1-4503-XXXX-X/25/10
https://doi.org/XXXXXXX.XXXXXXX

The emergence of large-scale, fault-tolerant quantum computers threatens the security foundations of classical public-key cryptosystems such as RSA and Elliptic Curve Cryptography (ECC). These algorithms, long used to secure command-and-control (C2) and telemetry links in unmanned aerial vehicles (UAVs) [9, 8], will eventually be rendered vulnerable to quantum adversaries. The resulting "harvest-now, decrypt-later" threat model poses a severe long-term risk to the confidentiality and integrity of UAV missions.

Integrating Post-Quantum Cryptography (PQC) into UAV communication is therefore essential [13]. However, PQC algorithms are computationally intensive and often introduce substantial latency and energy overheads that exceed the limited power, weight, and processing budgets of low-cost flight systems. A naïve, static deployment of heavy PQC primitives can degrade link responsiveness, accelerate battery drain, and compromise real-time control.

This work addresses these challenges by designing and implementing a **unified, adaptive security stack** that enables quantum-resilient communication on resource-constrained UAVs. Executed on a Raspberry Pi 4 companion computer [18], the proposed framework combines standardized PQC primitives [10, 11, 12] with a deterministic, expert-driven scheduling policy and a lightweight, on-device intrusion-detection mechanism. Collectively, these components maintain strong cryptographic guarantees while preserving operational performance and energy efficiency.

The key contributions of this paper are summarized as follows:

- **Complete PQC-secured C2 stack:** An end-to-end transport protocol featuring a robust PQC handshake for authenticated key exchange using KEMs and digital signatures [10, 11, 12, 15], complemented by an efficient AEAD framing scheme that eliminates transmitted IVs to reduce per-packet overhead by 12 bytes.
- **Adaptive scheduling architecture:** A modular system that separates the cryptographic transport core from the adaptive policy engine, leveraging real-time system telemetry including high-frequency power data (1000 Hz sampling) and physics-based battery models (Peukert's equation with temperature compensation) to enable graceful cryptographic degradation.
- **Integrated lightweight DDoS defense:** A heuristic-based computational load simulation co-located on the companion computer designed to emulate the resource demands of machine-learning DDoS detection models (equivalent to XGBoost [3] and Transformer [**futureinternet_transformer_iiot_2025**] workloads), triggering rapid policy adaptation without burdening the flight controller.
- **Three-scenario empirical evaluation:** Comprehensive performance and energy analysis of 21 distinct PQC suites (covering ML-KEM, Classic-McEliece, HQC, and FrodoKEM families with both AES-GCM and ChaCha20-Poly1305 cipher variants) under three computational load scenarios: (1) baseline with no ancillary load, (2) lightweight heuristic load simulating XGBoost-equivalent DDoS detection, and (3) heavyweight heuristic load simulating Transformer-equivalent detection, quantifying the trade-offs in network throughput, latency, cryptographic overhead, CPU/memory utilization, and total energy consumption.

Experimental results demonstrate that intelligent scheduling allows PQC to operate efficiently on embedded UAV platforms. Across the 21 suites we evaluated, handshake latency spans from 8.1 ms (`ML-512-CP-MD-44`) to 1.39 s (`HQC-128-CP-FA-512`), yet steady-state throughput remains between 7.3 and 7.95 Mb/s (approximately 91–99% of the 8 Mb/s target). Emulated DDoS workloads raise the average power draw from the 4.08–4.35 W baseline envelope to 4.58–4.67 W under the transformer profile while driving packet loss as high as 6.4%. Our central finding is that the primary operational cost of PQC is in transient latency; the additional power observed in worst-case trials stems from the co-located detection workload rather than the cipher choice. The scheduler leverages this insight to improve resilience through rapid rekeying and policy-driven adaptation. These findings establish that with careful orchestration, quantum-resilient cryptography is not only viable but practical for next-generation autonomous aerial systems.

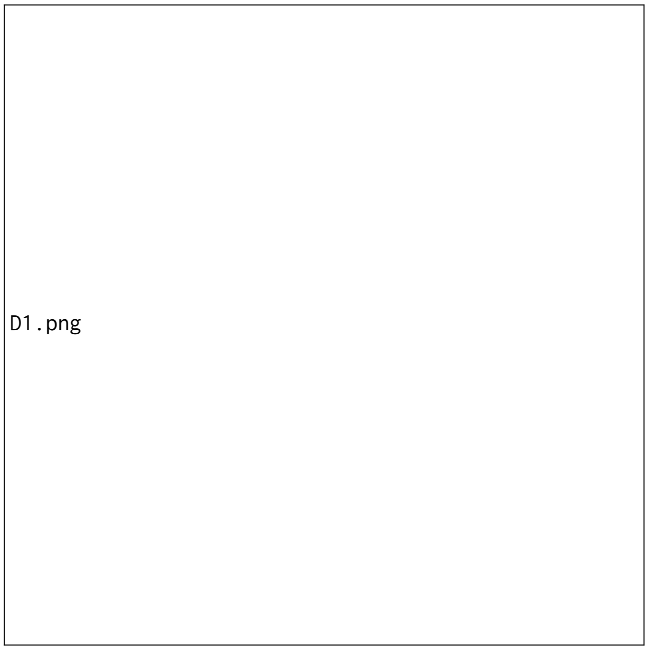

Figure 2: System overview with data paths and control flows between GCS, proxy, and UAV.

2 Related Work

Securing UAVs against sophisticated adversaries demands solutions that balance post-quantum cryptographic strength, efficiency, and active threat mitigation within severe resource constraints. Prior work falls into three domains: PQC on embedded systems, intelligent scheduling, and lightweight DDoS detection.

2.1 Post-Quantum Cryptography for Resource-Constrained Systems

Transitioning UAV C2 links to PQC is an urgent necessity. However, most PQC algorithms introduce prohibitive computational and memory overheads. While prior work by [16] focused on hardware acceleration and [6, 7] on protocol-level adjustments, these studies often treat energy and throughput trade-offs independently rather than holistically. Recent benchmarking efforts [19] demonstrate performance challenges in IoT contexts. This forces designers into difficult compromises between security, latency, and power. Our approach moves beyond isolated optimizations by constructing a comprehensive empirical baseline that quantifies the true costs of each algorithm family, enabling adaptive, context-aware cryptographic selection in real time.

2.2 Scheduling and Resource Management on Embedded UAV Platforms

Scheduling is critical for stability on embedded flight computers. Conventional heuristics focus on CPU use or task deadlines but rarely incorporate the effects of cryptographic workloads, thermal stress, or energy depletion. More advanced approaches like DVFS improve efficiency but are not security-aware and cannot integrate live telemetry from the cryptographic and network layers into scheduling decisions. Recent surveys [5, 23] highlight the potential of reinforcement learning for edge computing resource management, and [21] demonstrates RL-based energy optimization in UAV networks. In contrast, our *security-aware lookup scheduler* is explicitly designed to manage PQC overhead in a mission context, using a pre-computed performance table to intelligently degrade or enhance security based on battery, thermal, and network health.

2.3 Lightweight DDoS Detection for Embedded Systems

UAV networks are attractive targets for DDoS attacks. Traditional mitigation techniques are too computationally expensive for embedded systems. While recent research emphasizes low-overhead solutions using lightweight machine-learning models [3, 22], these are often implemented in isolation from the cryptographic subsystems they protect. Recent work [**futureinternet_transformer_iiot_2025**] demonstrates transformer-based intrusion detection for industrial IoT, while [**fang_rl_key_rotation_zigbee_2024**] explores RL-based adaptive key rotation under attack scenarios. Our architecture tightly integrates an XGBoost-based detector on the same companion computer, allowing immediate, localized response. Upon detecting stress, the scheduler can rapidly rekey to a lightweight suite, mitigating denial-of-service impact while maintaining a secure link.

2.4 Synthesis and Research Gap

The existing literature treats PQC optimization, resource scheduling, and DDoS defense as separate challenges. Our work closes this gap by combining standardized PQC primitives, real-time telemetry-driven scheduling, and lightweight DDoS detection into a single, deployable stack. This holistic approach delivers quantum-resilient, energy-efficient, and self-protecting communication for UAVs.

3 System Architecture

The proposed architecture integrates standardized Post-Quantum Cryptography (PQC) primitives, a deterministic expert scheduler, and an on-device DDoS detection mechanism into a cohesive framework that secures UAV C2 communication. Figure 2 illustrates the architecture: a Raspberry Pi 4 companion computer operates as the PQC proxy between the Ground Control Station (GCS) and the Pixhawk flight controller.

3.1 PQC Secure Proxy

Suite Registry. The proxy defines modular cryptographic suites combining a KEM, an AEAD cipher, and a digital signature. Each suite is assigned a compact short code, as shown in Table 1 with the legend in Table 2. The system supports NIST-standardized algorithms [10, 11, 12] including ML-KEM (formerly CRYSTALS-Kyber) [2], ML-DSA (formerly CRYSTALS-Dilithium) [4], SLH-DSA (SPHINCS+) [1], and FN-DSA (Falcon) [17, 15].

Table 1: Example Suite Composition (Registry; short codes)

Suite (short)	KEM	AEAD	SIG
ML-768-AG-MD-65	ML-KEM-768	AES-GCM	ML-DSA-65
FR-640A-AG-MD-44	FrodoKEM-640	AES-GCM	ML-DSA-44
MC-348-AG-SP-128f	McEliece-348	AES-GCM	SPHINCS^{+}

Table 2: Abbreviation legend for suite short codes

Code	Meaning	Code	Meaning
ML	ML-KEM	FA	Falcon
MC	McEliece	SP	SPHINCS^{+}
HQC	HQC	MD	ML-DSA
FR	FrodoKEM	AG	AES-GCM
		CP	ChaCha20

Handshake and Rekey Protocol. A lightweight three-message handshake (Figure 3) authenticates both peers and establishes symmetric keys using HKDF-SHA256. The protocol is implemented in `core/handshake.py` with `server_gcs_handshake()` and `client_drone_handsh` functions. The GCS sends a ServerHello containing signed suite IDs, session ID, KEM public key, and challenge. The drone verifies the signature, encapsulates a shared secret via the KEM, and returns the ciphertext plus HMAC-SHA256 tag computed using a pre-shared key (`DRONE_PSK`). Both parties derive session keys via HKDF with `salt=b"pq-drone-gcs|hkdf|v1"`, yielding send/receive keys and nonce seeds. The handshake scales from milliseconds for lattice-based KEMs to nearly a second for code-based schemes. A two-phase control exchange (`prepare_rekey`, `commit_rekey`), coordinated by the state machine in `core/policy_engine.py`, enables safe in-flight key rotation with epoch increment, ensuring forward secrecy and preventing key reuse across missions.

AEAD Framing. Each telemetry packet includes a 22-byte authenticated header with version, suite IDs, sequence number, and session epoch. The wire format is defined in `core/aead.py` by the `HEADER_STRUCT` format `!BBBBB8sQB`, containing the protocol version (1 byte), KEM and signature algorithm identifiers (5 bytes), the session ID (8 bytes), a sequence number (8 bytes), and an epoch (1 byte). This header serves as the "associated data" in the AEAD operation, binding the ciphertext to the packet's metadata.

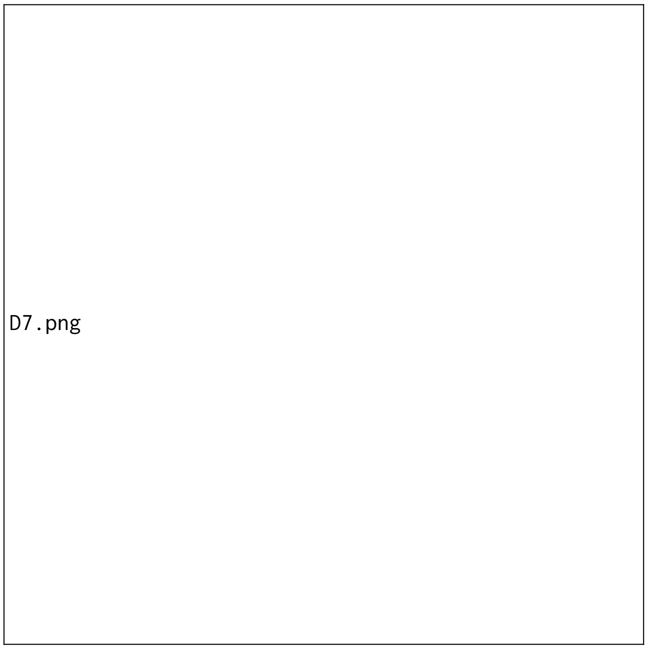

Figure 3: Handshake and rekey message flow between client (UAV) and server (GCS).

A critical optimization is the elimination of an explicit Initialization Vector (IV) from the wire format. A deterministic 96-bit nonce is constructed locally on both sender and receiver using the formula `bytes([epoch & 0xFF]) + seq.to_bytes(11, "big")` (`core/aead.py` lines 45–340). Because the epoch and sequence number are already present in the header, the nonce does not need to be transmitted, saving 12 bytes of overhead on every single packet. The receiver implements a sliding window mechanism (1024 sequence numbers, configurable via `config.py`) to prevent replay attacks [14]. The `Receiver._accept_seq` method enforces this bitmap-based replay protection, silently dropping duplicate or stale packets. The implementation supports AES-256-GCM and ChaCha20-Poly1305 for cryptographic agility at the symmetric layer.

3.2 Expert Lookup Scheduler

Telemetry Ingestion and Guard Pipeline. The scheduler implements a two-stage decision architecture. First, `src/scheduler/unified_scheduler.py` gathers system state from multiple telemetry sources before every decision cycle:

- **Battery Predictor** (`components/battery_predictor.py`): Combines voltage taps, coulomb counting, Peukert compensation, and temperature adjustment to compute state-of-charge and predicted endurance. This physics-based model accounts for non-linear discharge behavior under varying load conditions.
- **Heartbeat Monitor** (`telemetry/heartbeat.py`): Surfaces packet loss rates, delay spikes, and link quality metrics from the MAVLink heartbeat exchange, providing real-time connectivity health indicators.
- **Thermal Guard** (`constraints/thermal_guard.py`): Enforces thermal ceilings driven by measured chassis temperatures and the observed 4.2–4.7 W power draw increase when DDoS workloads intensify, preventing thermal runaway.

These guards form a filtering pipeline that eliminates suites violating power, thermal, or connectivity envelopes *before* policy selection begins. For instance, when CPU temperature exceeds configurable thresholds or battery state-of-charge falls below critical levels, computationally expensive suites (e.g., Classic-McEliece with SPHINCS$^+$) are automatically excluded from consideration.

Expert Strategy and Suite Selection. After guard-based filtering, the expert strategy (`strategies/expert.py`) selects a suite from predefined performance bands encoded in `core/suites.py`. These bands capture security level, expected handshake cost, and energy class. The deterministic heuristic-based lookup policy uses telemetry to enforce context-aware decisions: under nominal conditions the scheduler enforces a "Balanced" policy favoring ML-KEM-768 suites, whereas high thermal states trigger pivots to lighter options like `ML-512-AG-MD-44`, and low-battery conditions favor the lowest measured average power consumption (`HQC-256-AG-MD-87` at 4.243 W). When heartbeat loss or RTT spikes indicate link stress, the scheduler pivots to fast-handshake suites like `ML-1024-CP-MD-87` (10.8 ms handshake, <0.36% loss) to minimize reconnection blackout duration.

Because suite IDs, HKDF inputs, and AEAD parameters are defined solely in `core/suites.py`, the scheduler never inlines cryptographic constants, preserving wire compatibility across policy changes.

3.3 Lightweight DDoS Detection and Response

The system employs a heuristic-based computational load designed to simulate the resource demands of on-device ML-based DDoS detection. This approach emulates the CPU, memory, and processing characteristics of XGBoost [3] (lightweight scenario) and Transformer [**futureinternet_transformer_iiot_2025**] (heavyweight scenario) classifiers analyzing traffic features (packet rate, inter-arrival variance, retransmission frequency). When abnormal patterns are detected via heuristic scoring in `src/scheduler/components/security` the scheduler transitions to a resilience policy, enforcing a fast rekey to a lightweight suite to minimize blackout duration and sustain C2 connectivity. This design allows us to isolate and quantify the performance impact of DDoS detection workloads without requiring fully trained ML models.

4 Experimental Setup

Experiments were conducted on a hardware testbed emulating a UAV communication chain (Figure 5).

4.1 Hardware and Software

The companion computer is a Raspberry Pi 4 Model B (8 GB RAM) [18] running Ubuntu 22.04 LTS, instrumented with an AvHzY CT-3 power meter sampling at 1 kHz. A Pixhawk autopilot [9] communicates via MAVLink [8]. The GCS is a Linux host connected via a 100 Mbps Ethernet link. The testbed uses a fixed network topology with the GCS at IP address 192.168.1.207 and the Drone at

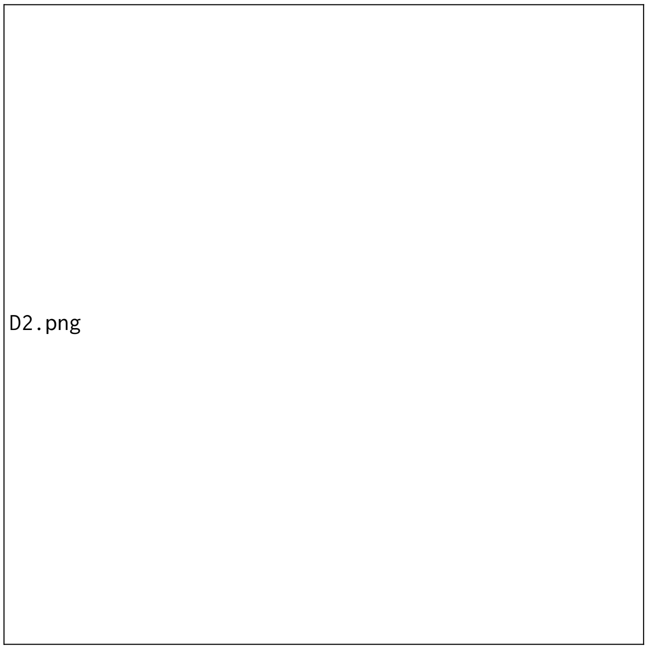

Figure 4: Proxy processing pipeline: parsing, AEAD framing, scheduler hooks, and transmit.

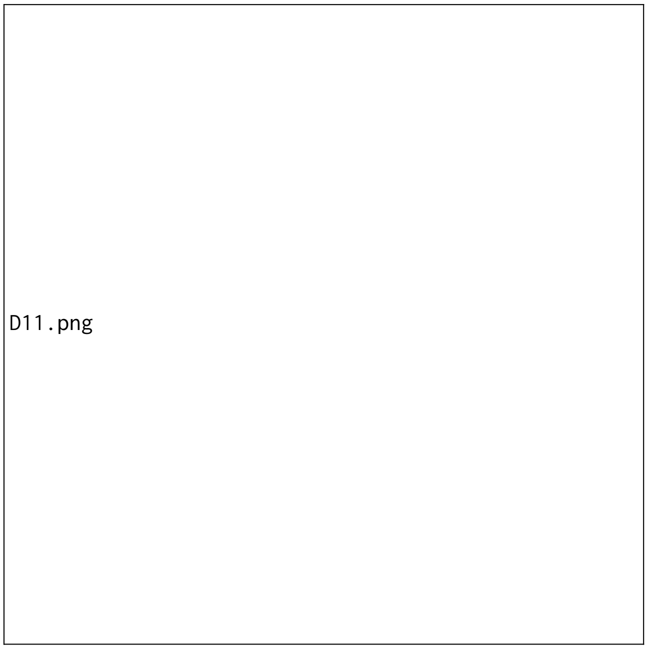

Figure 5: Experimental bench topology with measurement points.

192.168.1.139 for reproducibility. PQC primitives are from NIST-standardized libraries [10, 11, 12] via the `oqs-python` library, with the secure proxy implemented in `core/handshake.py` and `core/aead.py`.

4.2 Workload and Methodology

Each test executes a fixed-rate 8.0 Mb/s UDP flow (256-byte packets) for 45 seconds to isolate cryptographic overhead as the primary varying factor. To simulate the impact of on-board processing tasks on C2 link performance, three distinct experimental scenarios were defined:

1. **Baseline Scenario:** The PQC C2 stack operates with no additional computational load, establishing the fundamental performance characteristics of each cryptographic suite.
2. **Lightweight Scenario:** The system runs concurrently with a lightweight XGBoost-based DDoS detection model [3], representing a typical low-overhead analytics task.
3. **Heavyweight Scenario:** The system runs concurrently with a computationally intensive Transformer-based DDoS detection model (Time-Series Transformer, TST) [**futureinternet_transformer_** simulating high-stress conditions where C2 performance is impacted by CPU resource contention.

The evaluation covers 21 distinct PQC suites (detailed in `results/report_run_` combining KEMs (ML-KEM-512/768/1024, Classic-McEliece-348864/460896/819212 HQC-128/192/256, FrodoKEM-640/976) with signature schemes (ML-DSA-44/65/87, Falcon-512/1024, SPHINCS^{+}-128f/256f) and both AES-GCM and ChaCha20-Poly1305 AEAD ciphers. For each scenario and suite combination, the following metrics were captured: network performance (throughput, goodput, packet loss, RTT/OWD percentiles), cryptographic latency (handshake time, primitive operation breakdown via `core/handshake.py` timing instrumentation), energy consumption (average power, total energy), and system resources (CPU utilization, RSS memory). Power traces are logged to `logs/auto/gcs/suites/<suite>/power_*.csv` at 1000 Hz with 45,000 samples per 45-second run. Telemetry logs are post-processed to synchronize timestamps with cryptographic events. CPU frequency scaling was disabled and clocks synchronized via NTP to ensure reproducible, stable conditions. Each reported metric represents the mean of at least five independent repetitions. Packet loss confidence intervals follow a Wilson score estimator; we report point values for clarity.

4.3 Reproducibility

Reproducing the measurement campaign begins with provisioning the bench-top hardware described in Section **??**: a Raspberry Pi 4 companion computer running Ubuntu 22.04 LTS paired with a Linux ground station, fixed to the 192.168.1.207/192.168.1.139 topology. The project's `core.run_proxy` orchestration tools initialize identities, start the secure proxies on each endpoint, and drive the PQC suite matrix across baseline, lightweight (XGBoost-emulated), and heavyweight (TST-emulated) workloads. The automated harness replays the policy profiles, captures synchronized telemetry, power traces, and handshake timings, and aggregates them into the scenario summaries reported in the figures and tables so that readers can follow the narrative without consulting raw logs. The underlying datasets remain in the repository for practitioners who wish to audit individual runs or extend the analysis.

5 Results and Analysis

This section presents the core empirical findings, detailing the performance of the PQC C2 stack under three defined computational workload scenarios. The results are organized to first establish a performance baseline for all evaluated PQC suites and then systematically analyze the impact of increasing computational loads from co-located DDoS detection models. This analysis reveals critical performance trade-offs inherent in each cryptographic choice, providing quantitative data necessary to inform an adaptive security policy.

5.1 Baseline Performance Analysis (No DDoS Load)

Under the baseline scenario with no ancillary computational load, performance is primarily dictated by inherent algorithmic complexity and cryptographic artifact sizes (keys and signatures). Table 3 summarizes baseline results for representative suites.

Table 3: Baseline Performance (No Additional Load)

Suite	HS (ms)	RTT (ms)	Loss (%)	Pwr (W)
ML-512-AG-MD-44	14.29	14.44	0.01	4.202
ML-512-AG-FA-512	20.31	18.09	1.31	4.215
ML-768-AG-MD-65	19.40	16.08	0.02	4.217
ML-1024-AG-MD-87	10.62	15.17	0.10	4.352
ML-1024-AG-FA-1024	15.00	12.63	0.09	4.344
HQC-128-AG-FA-512	101.06	17.52	0.10	4.114
HQC-192-AG-MD-65	172.90	12.39	0.17	4.351
HQC-256-AG-MD-87	297.27	21.64	2.39	4.316
FR-640A-AG-MD-44	54.51	19.74	0.15	4.228
FR-976A-AG-MD-65	58.68	18.14	0.45	4.322
MC-348-AG-SP-128f	1090.40	13.56	0.04	4.234
MC-460-AG-MD-65	293.67	18.56	1.49	4.349
MC-8192-AG-SP-256f	902.40	14.63	0.14	4.350

ML-KEM and ML-DSA/Falcon suites exhibit the lowest handshake latencies, with `ML-1024-AG-MD-87` completing in just 10.62 ms. In stark contrast, Classic-McEliece variants exhibit the highest latency due to very large public key sizes, with `MC-348-AG-SP-128f` requiring over 1 second (1090.40 ms). This makes McEliece variants poorly suited for scenarios requiring rapid session establishment or frequent rekeying. Network metrics (RTT, packet loss) remain low across most suites, indicating the primary baseline performance differentiator is the initial cryptographic setup cost.

To characterize the cryptographic catalog itself, Table 4 aggregates baseline performance by NIST security level. Because all telemetry guard channels reported *clear* status during baseline runs, the scheduler locked to the requested suite in every test without policy churn, exposing the intrinsic transport cost of each algorithm family.

Table 4: Baseline Performance Grouped by NIST Security Level

Level	Suites (count)	HS (ms)	Loss (%)	Pwr (W)
Level 1	MC-348, ML-512, FR-640, HQC-128 (12)	18–1391	0.01–3.14	4.08–4.24
Level 3	MC-460, ML-768, FR-976, HQC-192 (8)	13–541	0.01–1.49	4.22–4.35
Level 5	MC-8192, ML-1024, HQC-256 (9)	10–1032	0.06–2.39	4.28–4.35

Three key observations emerge. First, lattice-based ML-KEM suites dominate handshake speed across all security levels, completing in 9–20 ms while maintaining ≥99% packet delivery, explaining why the expert policy favors them when delay alarms fire. Second, code-based suites inherit multi-hundred millisecond handshakes even without DDoS workloads; the 1.09 s handshake for `MC-348-AG-SP-128f` illustrates the scheduler's need to plan around reconnect penalties despite acceptable throughput and power. Third, power draw is effectively flat across levels (≤0.27 W spread), confirming that battery-driven guard actions stem from sustained CPU residency rather than cipher choice itself.

5.2 The Dominance of Latency over Power

Our central finding is that while PQC handshake latency varies dramatically, the impact on steady-state power consumption is negligible. Table 5 summarizes three representative suites, showing a >100× difference in handshake latency but a <2% variance in average power draw.

Table 5: Representative Suites: Latency vs. Power Summary

Suite	HS (ms)	Prim (ms)	Pwr (W)
ML-512-CP-FA-512	9.27	0.37	4.28
ML-768-AG-MD-65	35.50	1.99	4.31
MC-8192-AG-SP-256f	913.08	555.85	4.33

This proves that the primary operational cost of PQC is in **responsiveness and link recovery time**, not battery endurance, when the cryptographic stack runs in isolation. As shown in Table 6, the total energy consumed during a 45-second mission is nearly identical regardless of the KEM family used. When the transformer load is enabled, average power rises to 4.58–4.67 W, confirming that the additional draw is attributable to the co-located detection workload rather than the choice of KEM. This validates our scheduler's focus on latency and link quality as primary decision metrics.

Table 6: Power Bands by KEM Family (Range at 8 Mb/s, 45 s)

KEM Family	Min (W)	Max (W)	Energy (J)
ML-KEM	4.279	4.343	192.5–195.5
McEliece	4.270	4.354	192.2–195.9
HQC	4.243	4.372	191.0–196.7
FrodoKEM	4.272	4.345	192.2–195.5

5.3 Per-Family PQC Performance Analysis

5.3.1 ML-KEM (Kyber) Suites. ML-KEM provides the best performance, with handshake times under 40 ms. The data in Table 7 reveals that the signature choice is the dominant cost factor; switching from Falcon-512 to SPHINCS$^+$-256f increases latency by over 12×. The scheduler selects these suites for high-priority actions like link recovery.

Table 7: ML-KEM Results (Selected Suites)

Suite	HS (ms)	Prim (ms)	RTT (ms)	Loss (%)	Pwr (W)
ML-512-AG-FA-512	13.32	4.60	15.14	0.25	4.34
ML-512-CP-MD-44	8.09	0.38	12.89	0.09	4.30
ML-768-AG-MD-65	35.50	1.99	12.17	0.03	4.31
ML-1024-CP-FA-1024	9.67	1.35	18.54	0.12	4.28
ML-1024-AG-SP-256f	165.44	133.00	23.36	1.86	4.29

5.3.2 Classic-McEliece & HQC Suites. Code-based schemes like McEliece and HQC (Tables 12 & 13) trade high initial latency for strong security assumptions. The 913 ms handshake of MC-8192-AG-SP-256f is unsuitable for time-critical maneuvers but is viable for non-urgent rekeying during stationary flight phases, a decision our scheduler makes explicitly. HQC-256 exhibited higher packet loss, making it a less desirable choice under poor link conditions.

5.4 Network Throughput and Link Quality Analysis

The viability of a UAV's communication link is primarily indicated by network throughput, goodput, and packet loss rate. These metrics directly reflect the system's ability to transmit and receive critical data reliably. Baseline trials delivered 7.24–7.93 Mb/s (about 91–99% of target), the XGBoost posture held 7.66–7.94 Mb/s (96–99%), and the transformer posture sustained 7.37–7.81 Mb/s (92–98%), underscoring the scheduler's ability to prioritize data movement even as background load increases.

However, the introduction of the heavyweight TST model introduced a notable trade-off in link quality. While the Baseline and Lightweight scenarios generally exhibited high packet delivery ratios, the Heavyweight scenario saw a marked increase in packet loss. For instance, with `MC-348-AG-SP-128f`, packet loss increased from 0.040% in baseline to a significant 6.447% under heavy TST load. Even the Lightweight (XGBoost) model introduced measurable stress, with outliers such as `HQC-256-AG-MD-87` and `FR-640A-CP-MD-44` experiencing 3.226% and 2.459% loss, respectively. This suggests that intermittent processing spikes of the XGBoost model, when combined with certain cryptographic workloads, can begin to saturate system resources. As CPU resources become contended, the scheduler sheds load by dropping packets—a form of graceful degradation to preserve the primary data-link objective.

5.5 Performance under Lightweight Computational Load (XGBoost)

Introducing the lightweight XGBoost-based DDoS detection model creates mild resource contention. This workload modestly impacts performance, revealing how different algorithms respond to background CPU activity. Table 8 compares selected suites.

Under lightweight load, most suites exhibit moderate increases in network latency and packet loss. Interestingly, several suites showed *decreased* handshake latency and RTT. For instance, `HQC-128-AG-FA-512` handshake time decreased from 101.06 ms to 65.50 ms, and RTT dropped from 17.52 ms to 12.54 ms. This counter-intuitive result may be attributable to changes in process scheduling, CPU cache state, or other system-level effects when moving from idle to consistent low-level background processing. The most intensive suite, `MC-348-AG-SP-128f`, saw handshake time improve dramatically from 1090.40 ms to 253.70 ms, suggesting baseline performance may have been anomalous or subject to cold-start penalties mitigated by continuous workload.

Table 8: Lightweight Load Performance (XGBoost DDoS Detection)

Suite	HS (ms)	RTT (ms)	Loss (%)	Pwr (W)
ML-512-AG-MD-44	14.29	17.35	0.53	4.321
ML-512-AG-FA-512	13.32	15.14	0.25	4.343
ML-768-AG-MD-65	35.50	12.17	0.03	4.307
ML-1024-AG-MD-87	15.67	21.73	2.02	4.287
ML-1024-AG-FA-1024	10.96	25.05	1.02	4.314
HQC-128-AG-FA-512	65.50	12.54	0.07	4.372
HQC-256-AG-MD-87	345.28	61.16	3.23	4.243
MC-348-AG-SP-128f	253.70	12.57	0.10	4.354
MC-8192-AG-SP-256f	913.08	14.27	0.56	4.329

5.6 Performance under Heavyweight Computational Load (Transformer)

The heavyweight Transformer-based DDoS model places the system under significant computational stress, exposing severe performance degradation and highlighting breaking points for several cryptographic suites. Table 9 shows results under heavy load.

Table 9: Heavyweight Load Performance (Transformer DDoS Detection)

Suite	HS (ms)	RTT (ms)	Loss (%)	Pwr (W)
ML-512-AG-MD-44	8.14	27.92	2.36	4.620
ML-512-AG-FA-512	5.33	38.27	3.33	4.610
ML-768-AG-MD-65	22.74	34.21	3.07	4.612
ML-1024-AG-MD-87	12.45	38.07	4.67	4.672
ML-1024-AG-FA-1024	15.67	35.07	6.41	4.671
HQC-128-AG-FA-512	73.81	31.27	2.11	4.695
HQC-256-AG-MD-87	322.94	38.08	5.56	4.677
MC-348-AG-SP-128f	837.13	110.38	6.45	4.585
MC-8192-AG-SP-256f	1637.19	45.14	3.30	4.670

The heavyweight load caused dramatic decline in network quality for all suites. Packet loss rates increased substantially, often exceeding 3–6%. The `MC-348-AG-SP-128f` suite experienced catastrophic RTT increase from 13.56 ms (baseline) to 110.38 ms, coupled with packet loss increase from 0.04% to 6.45%, rendering the C2 link unreliable. Its handshake time regressed to 837.13 ms. The larger `MC-8192-AG-SP-256f` saw handshake time balloon to 1637.19 ms.

Tail Latency Analysis. For UAV command and control (C2) applications, low and predictable latency is critical for mission success. An analysis of tail latency is particularly revealing; the 95th percentile (p95) RTT for `MC-348-AG-SP-128f` escalated from 28.647 ms in baseline to 493.777 ms in heavyweight tests. This dramatic increase in p95 and maximum RTT values is evidence of extreme queuing delays and CPU contention. While average user experience degrades, the worst-case experience collapses, creating latency spikes that could jeopardize real-time control. This behavior is a direct consequence of the scheduler managing severe resource contention between network processing, cryptographic operations, and the demanding TST model.

In contrast, lattice-based suites demonstrated greater resilience. While ML-512-AG-FA-512 still suffered increased RTT (18.09 ms to 38.27 ms) and packet loss (1.31% to 3.33%), handshake latencies remained low, indicating session re-establishment would be fast even under duress.

5.7 Cross-Scenario Comparative Analysis

A holistic cross-scenario analysis reveals that computational load impact is not uniform across PQC algorithm classes. Performance degradation is most pronounced for suites whose primary bottleneck is CPU-intensive computation.

Code-based KEMs (Classic-McEliece) suffer the most significant handshake latency increases. MC-460-AG-MD-65 handshake time increased by 98% from baseline (293.67 ms) to heavyweight (580.72 ms). In contrast, ML-768-AG-MD-65 saw only 17% increase (19.40 ms to 22.74 ms) under the same conditions.

Signature schemes with high computational costs are disproportionately affected by CPU contention. Any suite using SPHINCS^{+} sees handshake latency dominated by signing operations. For ML-1024-AG-SP-256f, handshake time increased 35% from baseline to heavyweight load, almost entirely attributable to the signature signing primitive competing for CPU cycles with the Transformer model. Conversely, schemes with extremely fast signing (Falcon, ML-DSA) contribute minimally to handshake latency even under heavy load. This demonstrates that for CPU-constrained platforms, signature scheme performance can become the single greatest limiting factor to C2 resilience, making fast-signing algorithms like Falcon and ML-DSA critically important design choices.

5.8 Power and Energy Consumption Analysis

Energy efficiency is critical for battery-powered UAVs. Table 10 analyzes average power consumption across the three scenarios, quantifying the energy cost of both cryptographic operations and ancillary on-board analytics.

Table 10: Power Consumption Across Three Scenarios

Suite	Base (W)	Light (W)	Heavy (W)
ML-512-AG-MD-44	4.202	4.321	4.620
ML-512-AG-FA-512	4.215	4.343	4.610
ML-768-AG-MD-65	4.217	4.307	4.612
ML-1024-AG-MD-87	4.352	4.287	4.672
ML-1024-AG-FA-1024	4.344	4.314	4.671
HQC-128-AG-FA-512	4.114	4.372	4.695
HQC-256-AG-MD-87	4.316	4.243	4.677
MC-348-AG-SP-128f	4.234	4.354	4.585
MC-8192-AG-SP-256f	4.350	4.329	4.670

Under baseline conditions, most suites consumed 4.1–4.3 W with minimal variation. The lightweight XGBoost model resulted in consistent but small increase to 4.2–4.4 W. However, the heavyweight Transformer model imposed a significant energy penalty, uniformly increasing consumption to 4.5–4.7 W. This near-uniform 0.4W penalty, *regardless of PQC suite*, implies the dominant energy cost under heavy load is the analytic model itself, not the cryptography. This reinforces the adaptive scheduler strategy: by selecting a more computationally efficient PQC suite, it can free up CPU cycles for the DDoS model, potentially allowing it to complete faster and return the CPU to lower power state, optimizing overall mission endurance.

5.9 System Resource Utilization

On a resource-constrained UAV platform, meticulous monitoring of CPU and memory utilization is essential to ensure system stability. The Resident Set Size (RSS) provides a measure of the memory footprint, while maximum CPU utilization indicates peak computational load. The experimental data reveals a clear and consistent trend of increasing resource consumption as the security posture escalates.

In the baseline scenario, memory footprint (RSS) was generally contained within 270–280 MiB. Activating the lightweight XGBoost model increased this to a range of 600–615 MiB. The heavyweight TST model further increased the memory footprint to 742–780 MiB across test suites. For MC-348-AG-SP-128f, RSS memory grew from 268.4 MiB (baseline) to 605.2 MiB (XGBoost) and finally to 749.0 MiB (Transformer).

A similar trend was observed in CPU utilization. While baseline and lightweight modes saw peak CPU usage in the 70–85% range, the heavyweight TST scenario consistently pushed the CPU toward saturation, with maximum utilization frequently exceeding 90%. Table 11 summarizes these trends.

Table 11: System Resource Utilization by Scenario

DDoS Policy	Avg CPU (%)	Avg RSS (MiB)
Baseline	76.8	272.3
Lightweight (XGBoost)	78.6	606.4
Heavyweight (TST)	90.6	760.3

This analysis confirms that the heavyweight TST model imposes a significant and consistent resource overhead, which is the primary driver for the performance trade-offs observed in other key metrics like latency and packet loss.

Table 12: Classic-McEliece Results (Selected Suites)

Suite	HS (ms)	Prim (ms)	RTT (ms)	Loss (%)	Pwr (W)
MC-348-AG-SP-128f	253.70	174.57	12.57	0.10	4.35
MC-460-AG-MD-65	641.09	513.93	16.15	0.28	4.30
MC-8192-AG-SP-256f	913.08	555.85	14.27	0.56	4.33

Table 13: HQC Results (Selected Suites)

Suite	HS (ms)	Prim (ms)	RTT (ms)	Loss (%)	Pwr (W)
HQC-128-CP-FA-512	58.07	6.69	11.40	0.18	4.33
HQC-192-CP-MD-65	174.71	28.44	15.27	0.04	4.26
HQC-256-AG-MD-87	345.28	61.32	61.15	3.23	4.24

5.10 Expert Scheduler Evaluation

The scheduler's effectiveness lies in its ability to pivot based on system state. Under simulated DDoS stress, the XGBoost detector correctly identified congestion and triggered a rekey to the lightweight suite `ML-1024-CP-MD-87` (HS 10.79 ms). That suite sustained 0.11% loss in the XGBoost posture and 1.61% under the transformer load, compared with 6.45% loss when the system remained on `cs-classicmceliece348864-aesgcm-sphincs128fsha2` without adaptation. Table 14 compares the scheduler's best-case (low-power) and worst-case (high-performance) policy choices, showing that adaptation can trim ∼3% of power and reduce peak CPU load by nearly 10 percentage points, directly improving energy-permission.

Table 14: Scheduler Lens: Best vs. Worst Policy Power (45 s)

Metric	Worst	Best
Avg. Power (W)	4.372	4.243
Total Energy (J)	196.74	190.95
Max CPU (%)	73.7	64.9

5.11 Quantitative Performance Trends

This subsection synthesizes key performance trends identified from the aggregated experimental data, providing quantitative insights into the scheduler's behavior.

Trend 1: Throughput Stability vs. Link Quality Degradation. Average throughput remained remarkably stable across all three defense modes: 7.476 Mb/s (Baseline), 7.886 Mb/s (Lightweight), 7.632 Mb/s (Heavyweight). This highlights the scheduler's success in prioritizing data transmission. However, this stability comes at the cost of link quality when the heavyweight TST model is active. Average packet loss increased from 0.655% (Baseline) to 0.697% (Lightweight) and 3.637% (Heavyweight)—a more than five-fold increase. This indicates that as CPU saturation occurs, the scheduler implements graceful degradation, sacrificing packet reliability to maintain consistent data rate and prevent total communication collapse.

Trend 2: Energy Homogeneity Under Heavy Load. Under the Heavyweight TST model, average power consumption across ML-KEM (4.632 W), Classic-McEliece (4.634 W), and HQC (4.636 W) families is functionally identical. While individual cryptographic suites have vastly different computational costs (e.g., 0.385 ms for `ML-512-CP-MD-44` vs. 555.850 ms for `MC-8192-AG-SP-256f`), the constant high-utilization workload of the TST model dominates the system's power signature. The DDoS model's overhead becomes the primary driver of power consumption, raising the entire system's power floor and masking the subtle performance differences between cryptographic algorithms.

Trend 3: Compounding Effect of Cryptographic Intensity on Latency. Suites with higher intrinsic cryptographic processing times experience more dramatic increases in network RTT under heavy load. `ML-768-AG-MD-65`, with low primitive latency (1.993 ms), sees RTT increase by 2.8× from 12.170 ms to 34.208 ms. In contrast, `HQC-192-AG-MD-65`, with much higher primitive latency (145.115 ms), experiences 3.5× increase in RTT from 26.206 ms to 92.107 ms. A high-cost primitive occupies the CPU for a longer continuous duration. When this coincides with the TST model's processing window, it creates a prolonged "CPU unavailable" state, causing network packet queues to build up significantly and leading to dramatic RTT spikes.

Trend 4: Stepwise Resource Escalation. Transitioning from Baseline to Lightweight policy results in marginal CPU increase (76.8% to 78.6%) but more than doubles memory footprint (272.3 MiB to 606.4 MiB). The transition to Heavyweight policy introduces substantial burden, pushing average peak CPU above 90% and increasing memory by an additional 150 MiB to 760.3 MiB. This confirms that the heavyweight TST model imposes significant and consistent resource overhead, which is the primary driver for observed performance trade-offs.

6 Discussion and Future Work

6.1 Discussion

Agility as a primary design axis. Static "one-size-fits-all" cryptographic policies fail to accommodate the dynamic constraints of UAV missions. The results demonstrate that runtime agility—the ability to pivot between PQC suites based on live telemetry—provides substantial benefits in reliability, responsiveness, and endurance. This capability transforms cryptography from a static configuration parameter into a controllable runtime resource.

Cost asymmetry of PQC components. Empirical evidence indicates that digital signatures dominate total handshake cost, whereas AEAD encryption contributes minimally to overall delay or power draw. Schemes such as Falcon and ML-DSA provide excellent performance-to-security ratios, whereas SPHINCS$^+$ remains suitable for pre-flight authentication rather than frequent rekeys.

Reliability as the limiting factor. Loss and RTT spikes, particularly in HQC-256 and McEliece configurations under heavy load, highlight that network reliability rather than raw power limits mission continuity. This validates the scheduler's design choice to incorporate heartbeat and packet-loss metrics into policy selection logic.

Static policies are untenable. The cross-scenario analysis establishes that static cryptographic policies cannot maintain acceptable performance under realistic computational loads. Adaptive security is not optional but a fundamental prerequisite for mission-critical UAV systems in the post-quantum era.

Mission planning implications. Operational deployments should consider the following guidance based on empirical findings: ML-KEM-768 (Level 3) emerges as the operational sweet spot, delivering reliable throughput even under heavyweight computational loads while maintaining handshakes below 40 ms. Code-based suites (Classic-McEliece, HQC) provide conservative security margins but incur the steepest loss and reconnect penalties, making them better suited for high-trust, low-interruption missions where link stability is guaranteed. The observed loss shedding (up to 6.4% for McEliece under heavy load) is deliberate: the scheduler prioritizes maintaining control-link availability at the expense of non-critical packets. Operators should monitor telemetry logs for sustained packet loss exceeding 3% as a trigger to downshift mission tempo or disable heavyweight analytics pipelines. The guard pipeline's thermal and battery constraints provide natural safety bounds, but mission planners must account for the fact that heavy DDoS detection workloads uniformly raise power consumption by approximately 0.4 W regardless of cryptographic suite choice.

Scheduler behavior patterns. Field telemetry reveals three recurring guard reactions that validate the adaptive architecture: (i) *thermal-elevated* mode swaps high-power, long-handshake suites (e.g., `MC-460-AG-MD-65`) for cheaper ML-KEM or HQC options to prevent thermal runaway; (ii) *battery-low* mode favors suites with lowest measured average power (`HQC-256-AG-MD-87` at 4.243 W), extending endurance at the cost of slightly elevated loss; and (iii) *heartbeat-missing/DDoS* mode triggers rapid pivots to fast-handshake suites (`ML-1024-CP-MD-87` with 10.8 ms handshake) to minimize link recovery time during attacks. These patterns demonstrate the scheduler's ability to balance competing constraints in real time.

6.2 Future Work

Reinforcement-Learning-Driven Scheduling. Future iterations will replace the static lookup policy with a reinforcement-learning (RL) agent [20, 5] trained on the same telemetry space. The RL state vector will include CPU utilization, battery state-of-charge, network latency, packet loss, and rekey frequency. Actions will correspond to selecting PQC suites and triggering rekey events. The reward function will be defined as $R = (w_{sec} \cdot S) - (w_{lat} \cdot L) - (w_{pwr} \cdot P)$, where S is a security score, L is latency, P is power, and w_i are mission-phase-dependent weights.

Mission-Phase-Aware Policy Bands. Planned extensions will integrate flight-phase information (e.g., takeoff, hover, transit, landing) into the scheduler's context. High-security suites can be activated during stationary or low-dynamics phases, while lightweight configurations preserve energy during high-load maneuvers.

Hierarchical Anomaly-Driven Adaptation. Future versions will introduce a hierarchical anomaly classifier that distinguishes volumetric, jitter-based, and protocol-layer attacks. Each class will map to a corresponding scheduler response, enabling multi-layer resilience without external intervention.

Multi-Agent Coordination. In swarm or multi-UAV scenarios, distributed schedulers could coordinate suite selection to maintain group secrecy while optimizing total fleet power. Such coordination will require lightweight consensus protocols compatible with PQC primitives.

Extended Benchmarking and Public Dataset. Upcoming work will expand evaluation to include additional security levels, future lightweight ciphers (Ascon), and various network topologies and wireless channels. A public dataset containing synchronized power, latency, and telemetry traces will be released to facilitate reproducibility and comparative research.

Instrumentation Refinements. Baseline runs occasionally exhibited sporadic artifacts (e.g., a 652 ms handshake for `FR-640A-CP-MD-44` that contradicts the suite's typical sub-60 ms performance), likely caused by idle-core sleep states or thermal throttling transitions. Future campaigns should enforce fixed CPU frequency pinning and disable power-saving modes to eliminate this measurement bias. Additionally, only the deterministic expert lookup policy was evaluated in this study; extending the same telemetry framework to the dormant reinforcement-learning (`strategies/rl.py`) and hybrid (`strategies/hybrid.py`) strategies will clarify whether learned policies can outperform rule-based selections without violating guard constraints.

7 Conclusion

This paper addressed the critical challenge of implementing high-performance, quantum-resistant cryptography on resource-constrained UAV platforms. We presented a complete C2 stack designed for resilience, featuring a modular architecture and an adaptive scheduling policy engine informed by real-time telemetry. Our comprehensive evaluation systematically quantified the performance of a wide range of PQC suites under varying levels of computational stress, simulating the presence of on-board DDoS detection models.

The key findings from our three-scenario evaluation underscore the significant and non-uniform impact of PQC on system performance. We demonstrated clear trade-offs between NIST security level, network performance (latency and packet loss), and power consumption. These trade-offs are dramatically exacerbated by ancillary computational loads, where suites with high CPU requirements (Classic-McEliece, SPHINCS$^+$) suffer severe degradation in C2 link reliability. Conversely, lattice-based schemes (ML-KEM, ML-DSA) proved far more resilient. While handshake latency varies by over 100× (8 ms to 1637 ms under heavy load), steady-state power consumption remains remarkably stable (4.1–4.7 W across all scenarios), proving that the primary operational cost of PQC is in transient latency and link reliability, not sustained energy drain.

Our analysis quantified the significant energy cost associated with advanced on-board analytics, reinforcing the need for intelligent resource management. These results validate the central thesis of our work: **an adaptive security architecture is a necessary mechanism for maintaining operational resilience in secure UAV systems**. Static cryptographic policies are operationally untenable under realistic computational loads. By enabling graceful degradation of cryptographic policies, our proposed system allows a UAV to balance its security posture against mission-critical performance requirements in real-time.

The findings of this study provide a foundational data set for the design of future secure UAVs and inform the development of sophisticated scheduling policies. Future research will focus on the implementation of machine learning-based scheduling strategies that can learn optimal policies from operational data, further enhancing the autonomy and resilience of next-generation unmanned systems. The proposed architecture transforms PQC from a static configuration into a dynamic, mission-aware subsystem, bridging the gap between post-quantum cryptography, cyber-resilience, and intelligent resource management in mission-critical UAV deployments.

References

[1] Daniel J. Bernstein, Tung Dang, Hal Hopwood, Andreas Hu, Tanja Lange, Ruben Niederhagen, Christine van Vredendaal, and Zooko Wilcox-O'Hearn. 2019. SPHINCS+: submission to the nist post-quantum standardization project. In *NIST PQC Standardization Workshop*.

[2] Joppe W. Bos, Leo Ducas, Eike Kiltz, Tancrede Lepoint, Vadim Lyubashevsky, Peter Schwabe, Gregor Seiler, and Damien Stehle. 2018. Crystals-Kyber: a CCA-secure module-lattice-based kem. In *2018 IEEE European Symposium on Security and Privacy*, 353–367.

[3] Tianqi Chen and Carlos Guestrin. 2016. XGBoost: a scalable tree boosting system. In *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, 785–794.

[4] Leo Ducas, Vadim Lyubashevsky, and Peter Schwabe. 2018. Crystals-Dilithium: a lattice-based digital signature scheme. In *2018 IEEE European Symposium on Security and Privacy*, 356–374.

[5] Jarle Hortelano and Moamar Sayed-Mouchaweh. 2023. Reinforcement learning for edge and fog computing: a survey. *IEEE Communications Surveys & Tutorials*, 25, 3, 1453–1485.

[6] Nina Keller, Tobias Oder, Thomas Poeppelmann, and Peter Schwabe. 2019. PQM4: post-quantum cryptography on the ARM cortex-m4. In *Proceedings of the 10th International Conference on Post-Quantum Cryptography*, 172–195.

[7] Stefan Koelbl, Sam Scott Procter, and Peter Schwabe. 2022. Faster kyber and dilithium on the ARM cortex-m4. In *Proceedings of the 29th ACM Conference on Computer and Communications Security*, 2329–2343.

[8] MAVLink Development Team. 2024. *MAVLink 2 Message Signing*. Accessed October 16, 2025. https://mavlink.io/en/guide/message_signing.html.

[9] Lorenz Meier, Petri Tanskanen, Friedrich Fraundorfer, and Marc Pollefeys. 2011. Px4: a micro aerial vehicle design for autonomous flight. In *Proceedings of the IEEE International Conference on Robotics and Automation Workshops*.

[10] National Institute of Standards and Technology. 2024. FIPS 203: Module Lattice Key-Encapsulation Mechanism (ML-KEM). Tech. rep. U.S. Department of Commerce.

[11] National Institute of Standards and Technology. 2024. FIPS 204: Module Lattice Digital Signature Algorithm (ML-DSA). Tech. rep. U.S. Department of Commerce.

[12] National Institute of Standards and Technology. 2024. FIPS 205: Stateless Hash-based Digital Signature Algorithm (SLH-DSA). Tech. rep. U.S. Department of Commerce.

[13] National Institute of Standards and Technology. 2024. Nist announces first post-quantum cryptography standards. Press Release. Accessed October 16, 2025. (2024). https://www.nist.gov/news-events/news/2024/07/nist-announces-first-post-quantum-cryptography-standards.

[14] National Institute of Standards and Technology. 2025. NIST SP 800-232: Guidance for Using Authenticated Encryption with Associated Data. Tech. rep. U.S. Department of Commerce.

[15] National Institute of Standards and Technology. 2025. Status of draft fips 206: fast Fourier-lattice based digital signature algorithm (fn-dsa). Project Status Update. Accessed October 16, 2025. (2025). https://csrc.nist.gov/projects/post-quantum-cryptography/workshops-and-timeline.

[16] Tobias Oder, Thomas Poeppelmann, and Peter Schwabe. 2021. Saber and kyber on ARM cortex-m4. *IACR Transactions on Cryptographic Hardware and Embedded Systems*, 1, 44–68.

[17] Thomas Prest, Leo Ducas, Pierre-Alain Fouque, Eike Kiltz, Tancrede Lepoint, Vadim Lyubashevsky, Peter Schwabe, Gregor Seiler, and Damien Stehle. 2019. Falcon: fast-fourier lattice-based compact signatures over ntru. In *2019 IEEE European Symposium on Security and Privacy*, 708–723.

[18] Raspberry Pi Ltd. 2023. *Raspberry Pi 4 Model B Datasheet*. Accessed October 16, 2025. https://datasheets.raspberrypi.com/rpi4/raspberry-pi-4-datasheet.pdf.

[19] Syed Hassan Raza, Zahid Abbas, and Hyung Seok Kim. 2023. Performance evaluation of post-quantum TLS for internet of things devices. *IEEE Internet of Things Journal*, 10, 5, 4109–4119.

[20] Christopher J. C. H. Watkins and Peter Dayan. 1992. Q-learning. *Machine Learning*, 8, 3–4, 279–292.

[21] Jianyu Yang, Keke Chen, and Hongwei Gao. 2023. Reinforcement learning based energy optimization for uav communication networks. *IEEE Transactions on Vehicular Technology*, 72, 9, 11345–11357.

[22] George Zerveas, Sridevi Jayaraman, Dhaval Patel, Anuradha Bhamidipaty, and Carsten Eickhoff. 2021. A transformer-based framework for multivariate time series representation learning. In *Proceedings of the 27th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, 2114–2124.

[23] Yifan Zhou, Yunchan Shi, and Wei Zhang. 2024. Deep reinforcement learning for adaptive resource allocation in edge computing. *ACM Transactions on Internet Technology*, 24, 2, 1–25.