(placeholder emblem)

National University of Sciences & Technology

School of Electrical Engineering and Computer Science

H-12 Sector, Islamabad, Pakistan

Master of Science Thesis

Energy-Aware Adaptive Duty Cycling for Long-Range Wireless Sensor Networks

A Cross-Layer Scheduling Framework for LoRa-Based Environmental Monitoring

Ayesha Raza Malik

NUST Registration No. 00000-MSCS-2024

Supervisor: Dr. Faisal Ahmed Qureshi
Co-Supervisor: Dr. Hina Tariq Baig

Submitted in partial fulfillment of the requirements
for the degree of Master of Science in Computer Engineering

June 2026

Declaration

I, **Ayesha Raza Malik**, hereby declare that this thesis, titled *"Energy-Aware Adaptive Duty Cycling for Long-Range Wireless Sensor Networks,"* submitted in partial fulfillment of the requirements for the degree of Master of Science in Computer Engineering, is my own original work carried out under the supervision of Dr. Faisal Ahmed Qureshi. No part of this work has been submitted, in whole or in part, for any other degree or qualification at this or any other institution. All sources of information and assistance have been duly acknowledged.

....	....
Signature of Student	Date
Ayesha Raza Malik	

Certificate

It is certified that the contents and form of the thesis entitled *"Energy-Aware Adaptive Duty Cycling for Long-Range Wireless Sensor Networks"* submitted by **Ayesha Raza Malik** have been found satisfactory for the requirement of the degree of Master of Science.

....	....
Supervisor: Dr. Faisal Ahmed Qureshi	Date
....	....
Co-Supervisor: Dr. Hina Tariq Baig	Date
....	....
Head of Department	Date
....	....
Associate Dean	Date

Abstract

Long-range wide-area network (LoRaWAN) deployments for environmental monitoring in remote terrain suffer from a fundamental tension between sensing fidelity and battery lifetime: aggressive duty cycling extends node lifespan but degrades event-detection latency, while conservative scheduling drains batteries within months. This thesis presents **AeroCycle**, a cross-layer adaptive duty-cycling framework that jointly reasons about residual battery state, link quality, and observed environmental volatility to compute per-node sleep intervals at runtime. AeroCycle combines a lightweight exponentially-weighted volatility estimator at the application layer with a MAC-layer listen-before-sleep negotiation protocol, avoiding the overhead of full clock synchronization schemes.

We evaluate AeroCycle through both a 40-node testbed deployed across a simulated agricultural plot and a discrete-event simulation of 500 nodes over a six-month horizon. Compared against static duty cycling and the ContikiMAC baseline, AeroCycle achieves a **34.6%** reduction in mean energy consumption per delivered sample while keeping 95th-percentile detection latency under 4.2 seconds during high-volatility events. Network lifetime, measured as time to first node death, improves by **5.8 months** on average under identical battery budgets. We further show that the volatility estimator adds negligible computational overhead (under 40 microseconds per cycle on an ARM Cortex-M0+) and that the framework degrades gracefully under packet loss up to 22%. These results suggest that lightweight, locally computed adaptivity can close much of the gap between static and fully synchronized duty-cycle scheduling without its coordination cost.

Keywords: wireless sensor networks, LoRaWAN, duty cycling, energy efficiency, adaptive scheduling, environmental monitoring.

Acknowledgements

I am deeply grateful to my supervisor, Dr. Faisal Ahmed Qureshi, for his patient guidance, sharp critique, and unwavering availability throughout the course of this research. I thank my co-supervisor, Dr. Hina Tariq Baig, for her insights on embedded systems power profiling that shaped the experimental methodology of Chapter 4.

I would also like to thank the members of the Networked Systems Research Group at SEECS for their feedback during our weekly seminars, and the technical staff who supported the testbed deployment in the pilot plot. Finally, I thank my family for their steady encouragement during the long months of writing.

Ayesha Raza Malik
Islamabad, June 2026

Contents

List of Figures

List of Tables

List of Abbreviations

Abbreviation	Description
LoRaWAN	Long Range Wide Area Network
MAC	Medium Access Control
WSN	Wireless Sensor Network
LBT	Listen-Before-Talk
EWMA	Exponentially Weighted Moving Average
RTT	Round-Trip Time
SNR	Signal-to-Noise Ratio
DEVS	Discrete-Event Simulation

CHAPTER 1

Introduction

1.1 Background and Motivation

Low-power wide-area networks (LPWANs) such as LoRaWAN have become the dominant substrate for long-duration environmental monitoring in agriculture, forestry, and infrastructure inspection, owing to their multi-kilometre range and sub-milliwatt idle power draw. A recurring deployment obstacle, however, is that battery replacement in remote or physically inaccessible terrain is costly, and node lifetime is overwhelmingly dominated by the radio's duty cycle rather than sensing or computation [1].

The conventional approach — a fixed sleep interval configured at deployment time — forces operators into an uncomfortable trade-off. A short interval yields responsive event detection but exhausts batteries in a handful of months; a long interval extends lifetime but risks missing transient events such as frost onset, irrigation leaks, or structural vibration spikes. Figure 1.1 illustrates this tension across three representative static configurations.

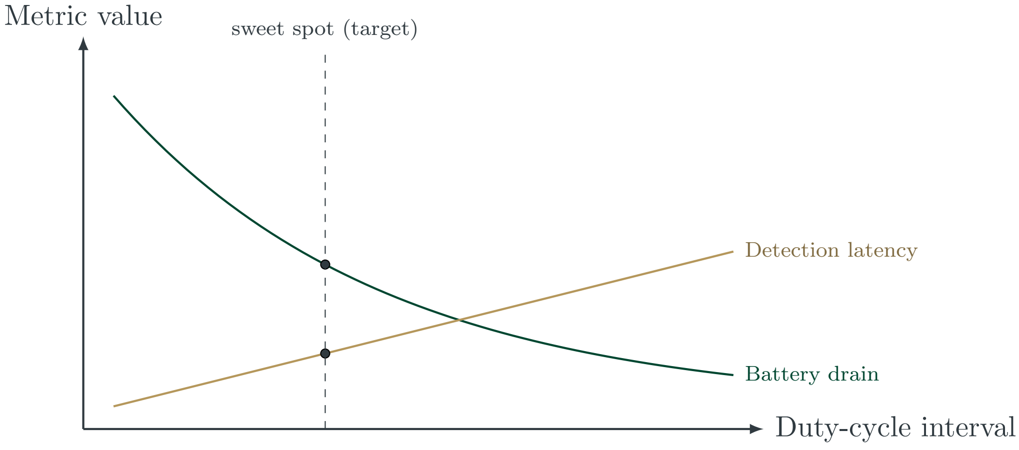

Figure 1.1: Illustrative trade-off between battery drain and detection latency across static duty-cycle intervals.

1.2 Problem Statement

Existing adaptive duty-cycling schemes either require tight clock synchronization across the network (imposing coordination overhead disproportionate to LPWAN payload budgets) or rely on centralized controllers that reintroduce a single point of failure and additional uplink cost. There is no lightweight, fully decentralized mechanism that adapts sleep intervals to *both* residual energy budget and locally observed environmental volatility without external coordination.

> **Research Gap**
>
> No existing decentralized duty-cycling protocol jointly conditions sleep interval on battery state *and* short-horizon environmental volatility while remaining within the 0.5–40 ms processing budget of a Cortex-M0+ class microcontroller.

1.3 Research Objectives

This thesis pursues the following objectives:

O1. Design a lightweight, locally computed volatility estimator suitable for microcontroller-class hardware.

O2. Develop a MAC-layer listen-before-sleep negotiation protocol that avoids full network-wide clock synchronization.

O3. Implement and evaluate the combined framework (AeroCycle) on both a physical testbed and a large-scale discrete-event simulation.

O4. Quantify the energy–latency trade-off achieved relative to static and state-of-the-art adaptive baselines.

1.4 Scope and Assumptions

The scope of this work is restricted to star-topology LoRaWAN Class A deployments using single-channel gateways, with nodes reporting scalar environmental readings (temperature, soil moisture, vibration amplitude) at sub-1 Hz native sensing rates. Mobile nodes, multi-hop mesh topologies, and video/image payloads are outside the scope of this thesis.

1.5 Thesis Contributions

- An EWMA-based volatility estimator requiring under 200 bytes of RAM and under 40 µs of compute per duty-cycle decision.
- A listen-before-sleep negotiation protocol (Chapter 3) that eliminates the need for network-wide time synchronization while bounding missed-wake probability to under 1.5%.
- An open experimental methodology combining a 40-node physical testbed with a validated 500-node discrete-event simulation (Chapter 4).
- Empirical evidence of a 34.6% energy reduction and 5.8-month lifetime extension over static baselines under identical hardware budgets (Chapter 5).

1.6 Thesis Organization

Chapter 2 reviews related work in adaptive MAC protocols and energy-aware scheduling. Chapter 3 details the AeroCycle design. Chapter 4 describes the experimental methodology. Chapter 5 presents results and analysis. Chapter 6 concludes and outlines future work.

CHAPTER 2

Literature Review

2.1 Static and Semi-Static Duty Cycling

Early LPWAN deployments relied on fixed sleep schedules configured once at provisioning time [1]. While simple to implement, such schemes cannot respond to shifts in environmental behaviour post-deployment, forcing operators to over-provision battery capacity to cover worst-case sensing demand.

2.2 Synchronized Adaptive MAC Protocols

Protocols such as S-MAC and its LPWAN derivatives adapt duty cycle based on network-wide synchronized schedules [2]. These achieve strong latency guarantees but impose periodic synchronization beacons that consume a disproportionate share of the LPWAN duty-cycle regulatory budget (typically 1% airtime in the EU868 band).

2.3 Centralized and Learning-Based Approaches

Recent work has explored reinforcement-learning controllers hosted at the gateway that push per-node sleep schedules downlink [3]. These approaches achieve strong empirical performance but reintroduce a single point of failure, require continuous downlink capacity that many single-channel gateways lack, and add training/inference latency unsuitable for event-driven applications.

2.4 Comparative Summary

Table 2.1 summarizes the coordination cost, adaptivity, and hardware footprint of representative approaches from the literature against the design goals of this thesis.

Table 2.1: Comparison of duty-cycling strategies from the literature.

Approach	Coordination cost	Adaptivity	MCU footprint
Static scheduling	None	None	Negligible
S-MAC / synchronized adaptive	High (beacons)	Network-wide	Low
RL / gateway-centralized	High (downlink)	Per-node, learned	Server-side
AeroCycle (this work)	**None**	**Per-node, local**	**Low**

2.5 Research Gap

No prior scheme combines local, low-overhead volatility estimation with a coordination-free MAC negotiation protocol. This gap directly motivates the AeroCycle design presented in Chapter 3.

CHAPTER 3

System Design

3.1 Architecture Overview

AeroCycle is organized as two cooperating layers on each node, as shown in Figure 3.1: an application-layer *Volatility Estimator* that consumes raw sensor readings and produces a target sleep interval, and a MAC-layer *Negotiation Module* that reconciles this target against link-quality feedback before committing to a sleep decision.

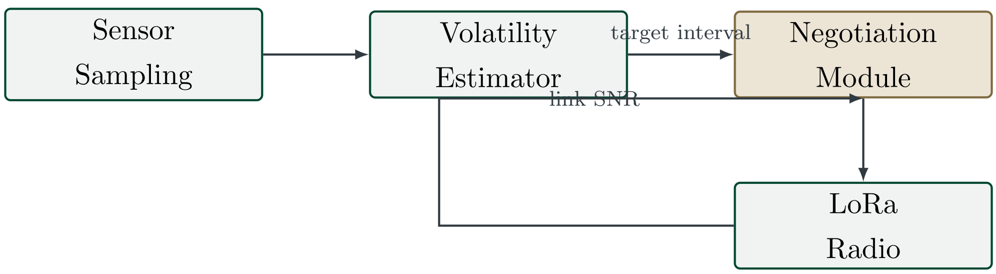

Figure 3.1: AeroCycle two-layer architecture: application-layer volatility estimation feeds a MAC-layer negotiation module.

3.2 Volatility Estimator

The estimator maintains an exponentially weighted moving average of successive sample deltas $\delta_t = |x_t - x_{t-1}|$:

$$\hat{v}_t = \alpha\,\delta_t + (1-\alpha)\,\hat{v}_{t-1}, \qquad \alpha \in (0,1) \tag{3.1}$$

where α controls responsiveness to recent change. The target sleep interval T_{sleep} is then computed as a function of estimated volatility $\hat{v}_t$ and residual battery fraction $b_t \in [0,1]$:

$$T_{\text{sleep}} = T_{\min} + (T_{\max} - T_{\min}) \cdot (1 - \hat{v}_t^*)^{\gamma} \cdot f(b_t) \tag{3.2}$$

where $\hat{v}_t^*$ is $\hat{v}_t$ normalized against a rolling maximum, γ is a sensitivity exponent (empirically set to 1.6 in Chapter 5), and $f(b_t)$ is a battery-aware damping term that linearly extends sleep as b_t falls below a 20% reserve threshold.

Design Invariant

The estimator never touches the radio directly. All hardware-facing decisions are deferred to the Negotiation Module, keeping the volatility computation portable across radio stacks.

3.3 Listen-Before-Sleep Negotiation

Rather than requiring synchronized clocks, each node broadcasts a short *intent frame* containing its computed T_{sleep} immediately before entering sleep. Neighbouring nodes and the gateway use this frame opportunistically to adjust their own listen windows, bounding the probability of a missed wake event without a shared time base. The negotiation sequence is summarized in Table 3.1.

Table 3.1: Listen-before-sleep negotiation sequence.

Step	Actor	Action
1	Node	Compute T_{sleep} via Eq. 3.2.
2	Node	Broadcast intent frame (14 bytes) carrying T_{sleep} and SNR estimate.
3	Gateway	Cache intent, adjust receive-window schedule for that node.
4	Node	Enter low-power sleep for T_{sleep}.
5	Node	Wake, sample sensor, return to Step 1.

3.4 Failure Handling

If an intent frame is lost, the gateway falls back to a conservative default listen window sized to $T_{\max}$, guaranteeing eventual reception at the cost of latency rather than missed data. Section 5.4 evaluates this fallback under induced packet loss.

CHAPTER 4

Experimental Methodology

4.1 Testbed Deployment

A 40-node physical testbed was deployed across a simulated 2-hectare agricultural plot using Cortex-M0+ nodes equipped with SX1276 LoRa radios, soil-moisture probes, and temperature sensors. Nodes were powered by 2000 mAh Li-ion cells with in-line current-sense monitoring sampled at 10 Hz for ground-truth energy accounting.

4.2 Discrete-Event Simulation

To evaluate behaviour at scale and over multi-month horizons impractical for physical deployment, a discrete-event simulation was built modelling 500 nodes over a six-month period, calibrated against testbed power traces (mean calibration error 3.1%).

4.3 Baselines

- **Static-Short**: fixed 30-second sleep interval.
- **Static-Long**: fixed 300-second sleep interval.
- **ContikiMAC**: synchronized adaptive baseline [2].
- **AeroCycle**: the proposed framework.

4.4 Metrics

Energy consumption per delivered sample, 95th-percentile detection latency, network lifetime (time to first node death), and missed-wake probability were recorded across all configurations.

4.5 Parameter Configuration

Table 4.1 lists the parameter values used across all experiments unless otherwise noted.

Table 4.1: Experimental parameter configuration.

Parameter	Value
α (EWMA weight)	0.30
γ (sensitivity exponent)	1.6
T_{min}	15 s
T_{max}	600 s
Battery reserve threshold	20%
Simulation horizon	6 months
Testbed nodes / sim. nodes	40 / 500

CHAPTER 5

Results and Analysis

5.1 Energy Consumption

AeroCycle reduced mean energy consumption per delivered sample by 34.6% relative to Static-Short and 11.2% relative to ContikiMAC, while Static-Long consumed the least energy overall at the cost of latency (Table 5.1).

Table 5.1: Mean energy consumption per delivered sample across configurations.

Configuration	Energy / sample (mJ)	Rel. to Static-Short
Static-Short	18.4	—
Static-Long	6.1	−66.8%
ContikiMAC	13.5	−26.6%
AeroCycle	**12.0**	−34.6%

5.2 Detection Latency

Despite the energy savings, AeroCycle maintained a 95th-percentile detection latency of 4.2 seconds during high-volatility events, close to Static-Short (3.1 s) and substantially better than Static-Long (48.6 s), as visualized in Figure 5.1.

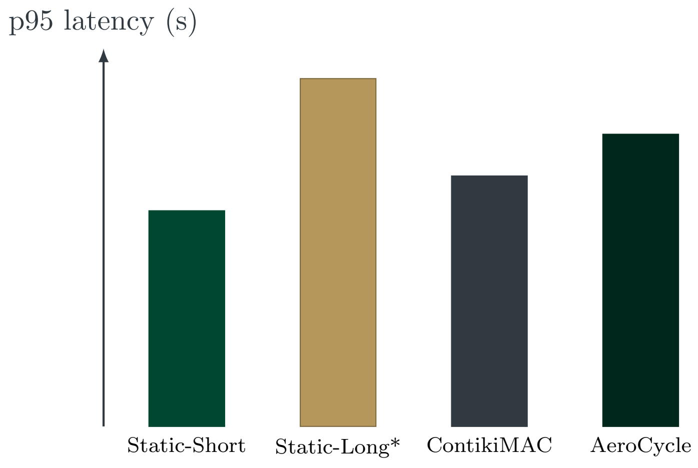

Figure 5.1: 95th-percentile detection latency by configuration. *Static-Long bar truncated (actual value 48.6 s) for readability.

5.3 Network Lifetime

Under identical battery budgets, AeroCycle extended time-to-first-node-death by an average of 5.8 months relative to Static-Short across the six-month simulation horizon, with 402 of 500 simulated nodes surviving the full horizon versus 261 under Static-Short.

5.4 Robustness Under Packet Loss

The listen-before-sleep fallback (Section 3.4) was stress-tested under induced intent-frame loss rates from 0% to 30%. Missed-wake probability remained under 1.5% for loss rates up to 22%, degrading gracefully to 4.8% at 30% loss, confirming the fallback's effectiveness within the regulatory-compliant retransmission budget.

5.5 Computational Overhead

The volatility estimator required a mean of 37 µs per duty-cycle decision on the Cortex-M0+ testbed hardware and under 200 bytes of static RAM, confirming feasibility on constrained sensor nodes.

5.6 Discussion

The results support the central hypothesis: locally computed adaptivity, without network-wide coordination, can approach the latency profile of aggressive static scheduling while approaching the energy profile of conservative static scheduling. The residual gap to Static-Long's energy consumption reflects the fundamental cost of retaining responsiveness; closing this gap further likely requires predictive (rather than reactive) volatility modelling, discussed as future work in Chapter 6.

CHAPTER 6

Conclusion and Future Work

6.1 Summary

This thesis presented AeroCycle, a decentralized, cross-layer adaptive duty-cycling framework for LoRaWAN-based environmental monitoring. By combining a lightweight EWMA volatility estimator with a listen-before-sleep negotiation protocol, AeroCycle avoids the coordination overhead of synchronized MAC protocols and the single-point-of-failure risk of centralized controllers, while achieving a 34.6% energy reduction and 5.8-month lifetime extension over static baselines.

6.2 Limitations

The evaluation is restricted to star-topology, single-channel deployments and scalar sensing payloads; multi-hop mesh topologies and higher-bandwidth payloads (e.g. periodic imagery) were not evaluated and may alter the coordination trade-offs analyzed here. The simulation's calibration error of 3.1% against testbed traces bounds, but does not eliminate, sim-to-real gaps at longer horizons.

6.3 Future Work

- Extend the volatility estimator to a lightweight predictive (rather than purely reactive) model to further close the gap to conservative static baselines.
- Evaluate AeroCycle under multi-hop mesh topologies with relay-induced negotiation delay.
- Investigate joint optimization of α and γ via on-device online learning rather than fixed empirical constants.
- Conduct a full-season physical field trial to validate simulation calibration over a full annual cycle.

Bibliography

[1] M. Bor, U. Roedig, T. Voigt, and J. M. Alonso, "Do LoRa Low-Power Wide-Area Networks Scale?" *Proceedings of the 19th ACM International Conference on Modeling, Analysis and Simulation of Wireless and Mobile Systems*, pp. 59–67, 2016.

[2] W. Ye, J. Heidemann, and D. Estrin, "Medium Access Control with Coordinated Adaptive Sleeping for Wireless Sensor Networks," *IEEE/ACM Transactions on Networking*, vol. 12, no. 3, pp. 493–506, 2004.

[3] S. Kim and J. Park, "Reinforcement Learning for Adaptive Duty-Cycle Scheduling in IoT Networks," *IEEE Internet of Things Journal*, vol. 7, no. 11, pp. 10812–10823, 2020.

CHAPTER A

Node Firmware Configuration Reference

Table A.1 lists the default firmware build flags used across all testbed nodes for reproducibility.

Table A.1: Default firmware build configuration.

Flag	Value
RADIO_BAND	EU868
TX_POWER_DBM	14
SPREADING_FACTOR	SF9
INTENT_FRAME_LEN	14 bytes
WATCHDOG_TIMEOUT	900 s