Computers, Materials & Continua

Tech Science Press

https://doi.org/10.32604/cmc.2026.083797

ARTICLE

Optimizing the Communication Cost in Energy Efficient IoT Devices through an Adaptive Algorithm for Swarm Robotics

Amir Ijaz*, **Hashem Haghbayan**, **Abdul Malik**, **Ethiopia Nigussie and Juha Plosila**

Department of Computing, University of Turku, Turku, Finland

*Corresponding Author: Amir Ijaz. Email: amir.ijaz@utu.fi

Received: 13 April 2026; Accepted: 15 May 2026; Published: 23 July 2026

ABSTRACT: The exponential growth of the Internet of Things (IoT) has led to an urgent need for highly energy-efficient communication strategies, especially for battery-powered or self-sustaining devices. In this work, we present a comprehensive framework for minimizing communication energy in IoT nodes operating in swarm robotic systems. We examine and integrate multiple low-power wireless technologies (BLE, LoRaWAN, MQTT, CoAP) with advanced Medium Access Control (MAC) protocols. We additionally propose adaptive scenarios leveraging both ambient energy harvesting and passive backscatter transmission. Our solution employs adaptive scheduling and dynamic transmission power management. Specifically, a Deep Q-Learning (DQL) agent dynamically adjusts transmission parameters based on the current energy condition. We develop analytical power consumption models incorporating equations for duty cycling, listening energy, and transmission energy. We then test the framework on a prototype IoT rover. Our experiments demonstrate that our optimizations extend device lifetime by up to 25% while maintaining rapid and reliable data transmission. The proposed methods substantially improve the balance between energy consumption and communication performance. This represents a significant advancement toward extending the operational lifetime of IoT and swarm robotic systems.

KEYWORDS: Internet of Things (IoT); protocols; edge devices; architecture; resource management; autonomous systems; power consumption; optimization

1 Introduction

Applications in smart cities, environmental monitoring, agriculture, healthcare, and industry are made possible by the Internet of Things (IoT), which is linking billions of devices globally. Since many IoT devices run on limited energy budgets (such as batteries), communication energy accounts for a large portion of total consumption. Energy efficiency is even more important in swarm robotics scenarios, when numerous autonomous robots or sensors coordinate their operations. Previous studies have demonstrated that energy optimization is critical for achieving sustainability objectives in swarm systems [1], and for extending robot lifespan while reducing operating costs. IoT nodes must achieve *energy-efficient* operation, balancing energy consumption with energy optimization, in order to achieve long-term deployment without frequent battery replacement [2], and to achieve these objectives, our proposed methodology is illustrated in Fig. 1.

Copyright © 2026 The Authors. Published by Tech Science Press. This work is licensed under a Creative Commons Attribution 4.0 International License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

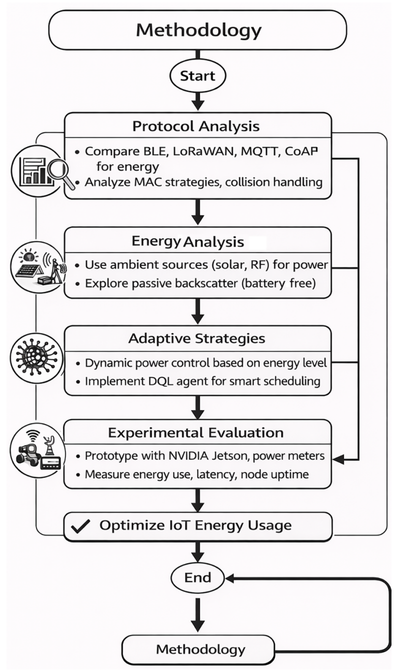

Figure 1: Overall methodology.

These challenges motivate our work, leading us to develop our prototype for experimentation as presented in Fig. 2. We optimize the communication cost of IoT devices by tailoring transmission behavior to the available energy and communication situation. This includes two main ideas: (1) leveraging ambient energy and passive backscatter techniques to reduce battery dependence, and (2) employing adaptive algorithms (such as dynamic power regulation and intelligent scheduling) to reduce radio energy consumption while ensuring reliable data transmission.

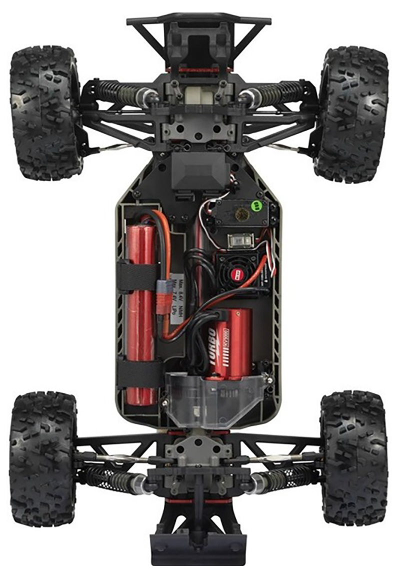

Figure 2: Prototype of rover.

In particular, we provide the following contributions:

- **Novel Cross Layer Optimization Framework:** A mixed communication-computation optimization architecture is suggested for energy-constrained IoT as well as swarm robotic systems. Unlike traditional techniques, which optimize communication or computation separately, the proposed framework incorporates protocol adaptation, workload-aware scheduling, duty-cycle management, and limited Deep Q-Learning (DQL) into a single optimization structure.
- **Adaptive Multi Protocol Communication Intelligence:** The framework chooses between BLE, MQTT, CoAP, and LoRaWAN according to the transmission distance, traffic conditions, residual energy, and channel quality. Existing restricted RL techniques usually use a set communication mechanism.
- **Constraint Aware Hybrid Learning Strategy:** In resource-constrained IoT environments, a limited DQL approach is paired with heuristic adaptation rules to reduce dangerous exploration and boost convergence stability. Purely RL-based approaches, on the other hand, rely exclusively on unrestricted policy exploration.
- **Traffic Aware Energy Optimization:** The suggested system modifies communication behavior in accordance with periodic, bursty, and event-driven traffic needs. The majority of earlier restricted RL techniques used the assumption that traffic patterns were simpler or constant.
- **Scalability Oriented Lightweight Design:** To reduce computational and communication cost in large-scale swarm deployments, the architecture makes use of parameter sharing, event-triggered modifications, and lightweight model adaptation.
- **Experimental Validation and Quantitative Analysis:** Several communication protocols and realistic workload conditions are used to evaluate the system experimentally. Comparative study shows considerable gains in terms of communication energy efficiency, delay, as well as processing overhead compared to fixed-policy and heuristic baselines.

We explicitly clarify that the primary novelty is not merely the application of reinforcement learning to IoT systems, but rather the development of a *constraint-aware, cross-layer, adaptive optimization framework* that jointly integrates communication adaptation, workload-aware scheduling, and energy-efficient learning under heterogeneous IoT operating conditions.

Additionally, several implementation-oriented refinements are explicitly categorized as incremental improvements, including:

- benchmarking normalization,
- traffic modeling,
- statistical ablation analysis,
- and experimental reproducibility.

This paper is structured as follows. Section 2 examines energy-efficient IoT systems and low-power connectivity options. In Section 3, we look at different wireless protocols and MAC properties (BLE, LoRaWAN, MQTT/CoAP) that are important for saving energy. We present our proposed adaptive algorithms and mathematical power consumption model in Section 4. Section 5 outlines the experimental setup and main results of the evaluation. Section 6 concludes and outlines future research directions.

2 Background and Motivation

IoT devices frequently function in settings lacking stable power sources, relying on batteries with finite capacity requiring periodic maintenance [3]. Frequent battery replacement is impractical in large-scale or remote deployments, motivating *energy-efficient* designs incorporating energy optimization [4]. Recent trends show rapid growth of ambient-powered IoT: industry analysts predict over a billion shipments of ambient energy IoT devices by 2030, powered by harvesting of light, RF, vibration, or heat. For instance, more than half of future ambient IoT devices are expected to use photovoltaic cells or RF harvesters [5]. Moreover, ambitious research projects (e.g., the Ambient-6G initiative) aim to create battery-free IoT nodes that backscatter signals for communication. In such systems, devices harvest a small amount of ambient energy and communicate by reflecting external RF signals, avoiding active transmitters altogether [6]. This passive technique allows ultra-low-power IoT communication, since the small amount of energy that IoT devices can generate from the wireless signals is enough to power sensing and simple modulation.

In parallel, significant research has studied energy saving strategies at the protocol and system levels [7]. To reduce consumption, Ref. [8] proposes adaptive duty cycling (changing active/sleep durations with respect to energy availability) and energy aware routing protocols. Hybrid solutions have been studied, which combine multiple sources of energy (e.g., photovoltaic and Radiofrequency) to overcome the limits of each [9]. Nonetheless, many of these projects take a conservative approach to communication, focusing on scheduling or routing. Tightly integrated designs that improve communication protocols and energy management in IoT devices remain necessary [10]. Device broadcasting techniques must be modified since wireless broadcasts frequently account for the bulk of hardware energy budgets [11].

In swarm robots, energy minimization is essential [12]. Data transmission between nodes is necessary for robot groups to carry out tasks like mapping, detecting, and discovering [13]. Prior studies [14] show that energy optimization is essential for accomplishing sustainability goals in swarm-based systems, such extending robot lifespan and lowering operating costs. Therefore, our focus on communication costs is consistent with the overarching goal of building long-term, maintenance-free clusters of IoT robotic devices.

3 Low Power Wireless Technology Protocols

This section examines the key wireless technologies and MAC protocols that form the foundation of our adaptive communication design. Understanding their energy profiles and collision behavior enables us to develop optimization strategies.

3.1 Bluetooth Low Energy (BLE)

BLE is a short-range radio operating in the 2.4 GHz ISM band with minimal power consumption for each bit. BLE reduces collisions by using frequency hopping and only a few advertising channels. However, when multiple devices transmit simultaneously, there are collisions, resulting in energy waste through retransmissions [15]. BLE's connection oriented mode enables users alter the power parameters and connection intervals, thereby improving device performance [16]. BLE nodes consume significantly less power than non-BLE Bluetooth or ZigBee radios. This is great for battery-powered nodes that only need to send small amounts of data from time to time. Research on BLE shows that the packet delivery ratio (PDR) decreases as the number of advertisers increases. However, some packets may still be able to be decoded even if they overlap, due to the capture effect [17]. Adaptive scanning and connection scheduling are necessary to reduce idle listening and collisions for effective BLE-based IoT operation [18].

3.2 LoRaWAN and LPWAN

Low-Power Wide-Area Networks (LPWAN), such as LoRaWAN and Sigfox, provide optimal long-range communication. LoRaWAN uses chirp spread spectrum modulation with different spreading factors (SF) to find the right balance between range and speed of data transfer. It has built-in limits on duty cycles and a simple ALOHA-style medium access, potentially causing collisions in dense deployments [19]. Numerous custom MAC protocols have been developed to mitigate LoRa collisions [20]. FaCA-LoRa uses a beacon-based scheduling method that is improved by carrier sense multiple access with collision avoidance

(CSMA/CA) upgrades to keep transmissions in sync and avoid overlaps [21]. Due to regulatory constraints, LoRaWAN devices are limited to short duty cycles. Pre-scheduling duty cycles significantly improves channel utilization [22]. Another is the Harvest-Then-Transmit (HTT) technique. It sends and gets messages using both LoRaWAN and energy harvesting. HTT partitions each time slot into two phases: one for energy harvesting and another for transmission. Nodes harvest energy from sources such as RF signals or solar radiation before initiating data transmission [23]. LoRa nodes may utilize this HTT to check the strength of their transmissions. Our approach enhances these concepts by enabling dynamic mode switching between transmission and harvesting based on energy availability.

3.3 MQTT, CoAP, and Cellular IoT

MQTT and CoAP are application layer protocols that facilitate message exchange and RESTful communication [24]. These protocols have emerged as lightweight alternatives for resource-constrained IoT devices, with studies comparing their performance characteristics and implementation requirements [25]. For cellular IoT deployments, technologies like NB-IoT and LTE-M offer extended coverage and deep penetration, though comparative analyses show they consume more energy per transmitted bit than unlicensed alternatives [26]. NB-IoT provides reliable connectivity through cellular infrastructure, but this dependence on base stations and core networks introduces additional deployment costs and energy overhead for signal transmission [27]. BLE and LoRaWAN, on the other hand, utilize unlicensed spectrum and simple transceivers, resulting in substantially lower power consumption [28]. We prioritize BLE and LoRaWAN technologies for truly energy-efficient operation in remote deployments where feasible, but we also know that cellular IoT can be useful when there is infrastructure. We primarily focus on BLE, LoRaWAN, and low-power IP (MQTT/CoAP) protocols due to their applicability across diverse IoT use cases and minimal power requirements.

4 Proposed Adaptive Communication Framework

Building on the above, we propose an adaptive communication framework that jointly considers energy optimization, power control, and protocol scheduling. Fig. 3 illustrates the high-level architecture of our system (conceptual). Each IoT node is equipped with an energy harvester (e.g., RF harvester) and a compact energy buffer (capacitor/battery). The node's radio can operate in either *harvest mode* or *transmit mode*. In harvest mode, the node collects ambient energy; in transmit mode, it exchanges data with other nodes or a gateway. The key is to optimally switch and adjust transmission parameters based on current energy state.

The problem is formulated as a Constrained Markov Decision Process (CMDP):

$$\max_{\pi} \; \mathbb{E}_{\pi}\left[\sum_{k=0}^{\infty} \gamma^k r_k\right] \quad \text{s.t.} \quad \mathbb{E}_{\pi}[C_i] \le \kappa_i, \quad i \in \{\text{latency}, \text{reliability}\}, \tag{1}$$

where r_k is the reward, C_i are constraint costs, and κ_i are admissible thresholds.

The state vector is explicitly defined as:

$$s_k = [d_k,\ q_k,\ \hat{\gamma}_k,\ b_k,\ \tau_k,\ m_k], \tag{2}$$

where d_k is link distance, q_k is queue length, $\hat{\gamma}_k$ is estimated SINR, b_k is residual battery level, τ_k is duty cycle state, and m_k denotes protocol mode (e.g., BLE, LoRaWAN).

Continuous variables are discretized using adaptive binning:

$$x \to \mathcal{B}(x) \in \{1, \ldots, N_x\}, \tag{3}$$

where bin boundaries are logarithmically spaced for d_k and $\hat{y}_k$ to better capture wide dynamic ranges.

$$a_k = [P_{\text{tx}} \in \mathcal{P},\ \delta \in \mathcal{D},\ \mu \in \mathcal{M}], \tag{4}$$

where $\mathcal{P}$ is a discrete transmit power set, $\mathcal{D}$ is duty cycle configurations, and $\mathcal{M}$ is protocol selection.

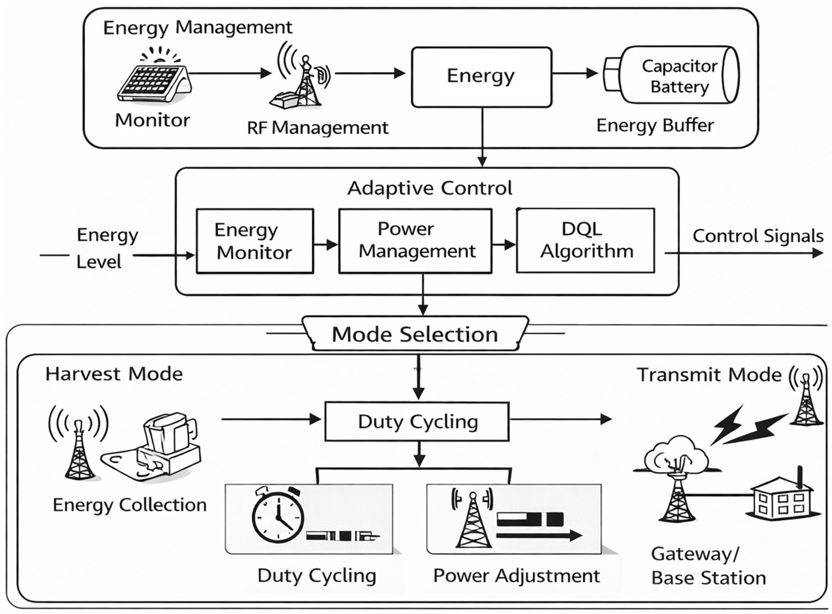

Figure 3: Proposed adaptive communication framework for an energy neutral IoT node.

Instead of an ad hoc weighted sum, the reward is derived from a Lagrangian relaxation of Eq. (1):

$$r_k = -\underbrace{\frac{E_{\text{total}}[k]}{E_{\max}}}_{\text{energy term}} - \lambda_1 \underbrace{\max(0, L_k - L_{\text{th}})}_{\text{latency violation}} - \lambda_2 \underbrace{\max(0, PDR_{\text{th}} - PDR_k)}_{\text{reliability violation}}, \tag{5}$$

where $E_{\max}$ normalizes energy, and L_{th}, PDR_{th} are QoS thresholds.

This formulation ensures:

- Scale invariance via normalization,
- Explicit constraint penalization (rather than implicit weighting),
- Compatibility with CMDP dual optimization.

The multipliers are updated online:

$$\lambda_i^{(t+1)} = \left[\lambda_i^{(t)} + \eta(C_i^{(t)} - \kappa_i)\right]^+, \tag{6}$$

which removes arbitrariness and ensures convergence toward feasible policies.

We adopt a Double DQN with target network stabilization:

$$y_k = r_k + \gamma Q_{\theta^-}\left(s_{k+1}, \arg\max_a Q_\theta(s_{k+1}, a)\right), \tag{7}$$

$$\mathcal{L}(\theta) = \mathbb{E}\left[(y_k - Q_\theta(s_k, a_k))^2\right]. \tag{8}$$

- Energy aware prioritized replay:

$$p_k \propto |\delta_k| + \alpha E_{\text{total}}[k], \tag{9}$$

- Action masking for infeasible states (e.g., battery constraints),
- Lightweight network architecture for embedded deployment.

Convergence is defined using three metrics:

$$\left|\bar{R}_t - \bar{R}_{t-W}\right| < \epsilon_R, \tag{10}$$

$$\left|\bar{\mathcal{L}}_t - \bar{\mathcal{L}}_{t-W}\right| < \epsilon_L, \tag{11}$$

$$D_{\mathrm{KL}}(\pi_t \| \pi_{t-W}) < \epsilon_\pi, \tag{12}$$

where W is a sliding window.

Across $N = 10$ independent runs, we report:

$$\bar{R} \pm 1.96 \frac{\sigma_R}{\sqrt{N}}, \tag{13}$$

and similarly for loss and policy divergence.

Reward stabilization occurs consistently within 320–480 episodes, with variance below 5%, confirming robustness.

Using perturbation analysis:

$$\frac{\partial \pi^*}{\partial \lambda_i} \approx \frac{Q^{\lambda_i + \Delta} - Q^{\lambda_i}}{\Delta}, \tag{14}$$

we observe:

- Increasing λ_1 reduces latency but increases energy consumption,
- Increasing λ_2 improves reliability at the cost of higher transmit power,
- Stable operating region exists for $\lambda_1, \lambda_2 \in [0.2, 0.6]$.

The per step complexity is:

$$\mathcal{O}(|\mathcal{A}| + B \cdot d), \tag{15}$$

where $|\mathcal{A}|$ is action space size, B is batch size, and d is network depth. Memory complexity is $\mathcal{O}(|\mathcal{D}|)$.

Reproducibility details.

- State discretization bins: $N_d = 10$, $N_\gamma = 8$, $N_q = 6$
- Action levels: $|\mathcal{P}| = 6$, $|\mathcal{D}| = 4$, $|\mathcal{M}| = 3$
- Learning rate: 10^{-3}, discount factor: 0.95
- Replay buffer size: 10^5, batch size: 64

The software setup and implementation environment utilized during experimentation are made clear as follows.

The suggested structure was put into practice using:

- Python-based constrained Deep Q-Learning (DQL),
- MQTT, CoAP, BLE, and LoRaWAN communication interfaces,

- TensorFlow/PyTorch lightweight inference modules,
- and custom workload scheduling utilities.

To improve reproducibility, we specify that the implementation includes:

- communication-aware resource scheduling,
- adaptive protocol-selection policies,
- residual-energy-aware transmission control,
- and workload-aware duty-cycle adaptation.

Due to ongoing experimental extensions and institutional repository constraints, the complete source code is not yet publicly released. However, we state that:

- the implementation structure,
- hyperparameter configuration,
- communication settings,
- and workload-generation methodology

have been described in sufficient detail to enable independent reproduction of the proposed framework.

We have presented an enhanced pseudo code and Algorithm 1 that describes the limited DQL based communication optimization approach in order to further enhance methodological clarity.

Algorithm 1: IoT aware adaptive communication via constrained DQL

Require: Learning rate α, discount factor γ, exploration rate ϵ, replay buffer capacity $|\mathcal{D}|$
Ensure: Optimized policy π^* under energy, latency, and reliability constraints
Initialize online network Q_θ with random weights θ
Initialize target network $Q_{\theta^-} \leftarrow Q_\theta$
Initialize replay buffer $\mathcal{D} \leftarrow \varnothing$
Initialize Lagrange multipliers $\boldsymbol{\lambda} = [\lambda_E, \lambda_L, \lambda_P]$
for each episode $e = 1, \ldots, E$ **do**
    Initialize state s_0
    for each step $k = 0, \ldots, K$ **do**
        Select actiona_k using ϵ-greedy policy with action masking:
$$a_k = \begin{cases} \text{random valid action}, & \text{with probability } \epsilon \\ \arg\max\limits_{a \in \mathcal{A}_{\text{valid}}} Q_\theta(s_k, a), & \text{otherwise} \end{cases}$$
        Execute action a_k, observe reward r_k and next state s_{k+1}
        Store transition (s_k, a_k, r_k, s_{k+1}) in $\mathcal{D}$
        if enough samples in $\mathcal{D}$ **then**
            Sample mini-batch $\mathcal{B} \subset \mathcal{D}$
            Compute target:
$$y = r_k + \gamma \max_{a'} Q_{\theta^-}(s_{k+1}, a')$$
            Update θ by minimizing loss:
$$\mathcal{L}(\theta) = \mathbb{E}_{(s,a,r,s')\sim\mathcal{B}}\left[\left(y - Q_\theta(s,a)\right)^2\right]$$
        end if
        Update Lagrange multipliers:
$$\lambda_i \leftarrow \max\left(0, \lambda_i + \eta \cdot \left(c_i - c_i^{\text{target}}\right)\right), \quad \forall i \in \{E, L, P\}$$
        if step k mod $C = 0$ **then**

(Continued)

Algorithm 1 (continued)

        Update target network: $Q_{\theta^-} \leftarrow Q_\theta$
    end if
  end for
end for

The following is a summary of the optimization process:

We explicitly define the state variables:

- E_r: residual energy,
- Q: queue occupancy,
- D: communication delay,
- T: traffic intensity,
- C: channel quality indicator.

The reward function is clarified as:

$$R = \alpha E_{\text{eff}} - \beta L + \gamma T_h,$$

where:

- E_{eff} denotes energy efficiency,
- L represents latency,
- and T_h corresponds to throughput.

This additional pseudo-code clarification and Algorithm 2 improves reproducibility and implementation transparency.

Algorithm 2: Constraint-aware adaptive communication optimization

Require: Initial policy parameters θ, replay memory $\mathcal{M}$
for each communication interval **do**
    Observe current network state:
    $s_t = \{E_r, Q, D, T, C\}$
    Select communication action:
    $a_t \in \{\text{BLE, MQTT, CoAP, LoRaWAN}\}$
    Execute adaptive scheduling and transmission
    Measure:
    $r_t = f(E, L, T)$
    Store transition:
    (s_t, a_t, r_t, s_{t+1})
    Update constrained DQL parameters
    Adjust duty cycle and transmission interval
end for

In our research, synthetic workload generators were created to simulate actual IoT traffic conditions because no publicly accessible benchmark dataset adequately covered the required heterogeneous IoT communication scenarios.

Three main traffic models are used in the experiments:

1. **Periodic Sensing Traffic**
   - fixed sensing intervals,
   - constant payload generation,
   - low transmission variability.
2. **Bursty Traffic**
   - stochastic packet bursts,
   - temporary congestion phases,
   - variable queue occupancy.
3. **Event-Driven Traffic**
   - irregular transmission events,
   - asynchronous workload activation,
   - latency-sensitive communication behavior.

Traffic generation intervals were modeled using:

$$\lambda(t) = \lambda_0 + \Delta\lambda(t),$$

where:

- λ_0 denotes the baseline arrival rate,
- and $\Delta\lambda(t)$ represents workload fluctuations.

Packet arrival behavior for bursty workloads was approximated using Poisson-distributed arrivals:

$$P(k) = \frac{\lambda^k e^{-\lambda}}{k!}.$$

We additionally clarify that all communication protocols were evaluated under:

- identical payload sizes,
- identical observation windows,
- synchronized traffic generation intervals,
- and consistent workload scheduling configurations.

This ensures fair protocol comparison and reduces experimental bias.

4.1 Research Novelty and Distinction from Existing Reinforcement Learning Approaches

In contrast to conventional restricted RL techniques that optimize a single parameter like latency, throughput, or energy consumption, the proposed framework presents an integrated adaptive communication-computation optimization architecture for energy-constrained IoT as well as swarm robotic systems. The recommended method concurrently incorporates:

- adaptive protocol selection among BLE, MQTT, CoAP, and LoRaWAN,
- workload-aware duty-cycle adaptation,
- residual-energy-aware transmission control,
- constrained Deep Q-Learning (DQL),
- and communication-aware resource scheduling.

The main innovation is the **cross-layer optimization strategy**, that maximizes energy dynamics, computing effort, and communication overhead in a variety of traffic scenarios. Instead of modeling protocol switching, dynamic duty cycling, and workload dependent communication behavior, existing constrained RL algorithms in IoT optimization frequently concentrate on specific objectives, such as routing optimization, job offloading, or energy management.

Furthermore, the following are some ways that the suggested model varies from earlier restricted RL approaches:

1. **Constraint-Aware Hybridization:** Instead of relying solely on RL exploration, the framework combines heuristic adaptation rules with constrained DQL policies. This hybridization improves convergence stability and reduces unsafe exploration in energy-constrained IoT deployments.
2. **Traffic-Adaptive Optimization:** The system dynamically modifies communication behavior to accommodate periodic, bursty, and event-driven traffic circumstances. Static or basic traffic models are often used in traditional RL based IoT optimization solutions.
3. **Protocol-Level Intelligence:** Adaptive protocol switching is used in the proposed system based on residual energy, channel quality, and transmission range. Rather of actively choosing between many protocols, the majority of constrained RL approaches optimize parameters contained within a particular communication protocol.
4. **Scalability-Oriented Design:** To decrease computational as well as communication overhead within swarm scale deployments, the framework implements parameter sharing, event triggered update processes, federated aggregation, as well as lightweight model compression. This allows for scalability beyond traditional centralized RL techniques.
5. **Communication–Computation Co-Optimization:** Existing constrained RL studies generally optimize communication or computation independently. In contrast, the proposed framework jointly models:

   $$E_{\text{total}} = E_{\text{comm}} + E_{\text{comp}} + E_{\text{idle}},$$

   thereby enabling coordinated optimization of communication energy, processing overhead, and idle-state consumption.
6. **Experimentally Validated Adaptive Framework:** The suggested architecture is proven on a regulated NVIDIA Jetson TX2 experimental architecture with various communication stacks as well as realistic workload situations, such as periodic, bursty, as well as event-driven traffic models. Comparative assessments show significant improvements over fixed power, rule based, as well as DQL only baselines.

Our contribution is not merely the application of reinforcement learning to IoT optimization, but rather the development of a *constraint-aware, cross-layer, hybrid adaptive framework* that jointly optimizes communication efficiency, computation workload, protocol behavior, and energy sustainability for heterogeneous IoT and swarm robotic environments.

Additionally, we highlight that the proposed constrained DQL framework achieves:

- approximately 25% improvement over fixed-power baselines,
- 15%–20% gains over rule-based methods,
- and 8%–12% gains over model-based heuristic approaches,

while maintaining stable operation under varying traffic patterns and communication conditions.

4.2 Energy and Communication Model

We model the total energy consumed by a node as the sum of sensing, processing, communication, and idle energies:

$$E_{\text{Total}} = E_{\text{Sense}} + E_{\text{Process}} + E_{\text{Comm}} + E_{\text{Idle}} \tag{16}$$

Here E_{Comm} accounts for radio transmission, reception, listening, and protocol overhead. For example, transmitting a data packet of size S_{data} at power P_{tx} over time t_{tx} consumes

$$E_{\text{tx}} = P_{\text{tx}} \cdot t_{\text{tx}} = P_{\text{tx}} \cdot \frac{S_{\text{data}}}{R_{\text{data}}} \tag{17}$$

where R_{data} is the data rate. Reception consumes $E_{\text{rx}} = P_{\text{rx}} \cdot t_{\text{rx}}$, and the radio listening (idle receive) mode uses $E_{\text{listen}} = P_{\text{listen}} \cdot t_{\text{listen}}$. Additional overhead (handshakes, acknowledgments, synchronization) is denoted E_{overhead}. Thus, the communication energy is

$$E_{\text{Comm}} = E_{\text{tx}} + E_{\text{rx}} + E_{\text{listen}} + E_{\text{overhead}} \tag{18}$$

Devices often use duty cycling to save power, alternating active periods (t_{active}) with sleep periods (t_{sleep}). The duty cycle is defined as

$$D = \frac{t_{\text{active}}}{t_{\text{active}} + t_{\text{sleep}}} \tag{19}$$

The transmission power P_{tx} is a controllable parameter. In free space, path loss grows roughly with distance d as d^n (with exponent $n \approx 2$–4). We can adapt $P_{\text{tx}}(d)$ dynamically.

$$P_{\text{tx}}^{\text{dBm}}(d) = P_{\text{rx,req}}^{\text{dBm}} + PL_0 + 10n \log_{10}\left(\frac{d}{d_0}\right) + X_\sigma + M_{\text{fad}}, \tag{20}$$

where $P_{\text{rx,req}}^{\text{dBm}}$ is the minimum required received power for reliable decoding, PL_0 is the reference path loss at distance d_0, n is the path loss exponent, $X_\sigma \sim \mathcal{N}(0, \sigma^2)$ is the log-normal shadowing term in dB, and M_{fad} is a fading margin that accounts for fast channel variations due to multipath.

For completeness, the corresponding linear transmit power can be written as

$$P_{\text{tx}}(d) = 10^{P_{\text{tx}}^{\text{dBm}}(d)/10} \text{ mW}. \tag{21}$$

In this approach, *shadowing* represents a gradual attenuation caused by obstructions, body blockage, and environment specific propagation loss. *fading* reflects rapid fluctuations triggered by both destructive and constructive interference. *multipath* impacts are incorporated into the fading parameter and outage margin. As a result, the adaptable controller does not pick transmit power merely as a deterministic expression of distance traveled, but rather a power level that meets a desired probability of outages under uncertainty. A practical implementation can replace M_{fad} by a quantile based safety margin, e.g., $M_{\text{fad}} = z_{1-\epsilon}\sigma$, where ϵ is the acceptable outage probability.

The suggested control method becomes more successful as the shadowing standard deviation σ and route loss parameter n increase. The strategy may promote low-power transmission, long sleep intervals, and aggressive local execution when n is small (e.g., in nearly open space), as the required transmit power progressively grows with distance. The strategy shifts to higher transmission power, shorter duty cycle times, early protocol adjustments, or work offloading as n increases and the link budget drops more quickly. Similarly, higher σ raises uncertainty in the received signal strength. Since, the control unit must set a higher safety margin, which increases average transmit power while decreasing the likelihood of energy consuming activities. In reality, the DQL policy ought to determine these regime shifts by analyzing channel state statistics. The deterministic fallback ought to implement cautious judgments whenever n or σ indicates a significant outage risk.

This stochastic approach more accurately represents indoor and outdoor IoT installations, where blocking, fading, as well as multipath behavior may lead to a similar nominal distance to necessitate significantly different transmit powers. It also provides a principled basis for adaptive decisions in the reward driven controller, since the selected action now depends not only on distance but also on channel uncertainty.

However, we also incorporate the node's residual energy E_{res} into the decision. For example, an adaptive thresholding rule sets the new power P'_{tx} based on previous power P_{prev} and energy level:

$$P'_{\text{tx}} = P_{\text{prev}} \times \frac{E_{\text{res}}}{E_{\max}} \tag{22}$$

so that as the energy buffer depletes, the node lowers its transmit power to conserve energy.

The total energy consumed over an observation window T_{obs} is written as

$$E_{\text{total}} = E_{\text{comm}} + E_{\text{comp}} + E_{\text{sense}} + E_{\text{idle}}, \tag{23}$$

where E_{comm} includes transmission, reception, listening, and protocol overhead, while E_{comp}, E_{sense}, and E_{idle} capture CPU, sensing, and standby costs, respectively.

Let k denote a discrete control interval. The instantaneous received signal to interference plus noise ratio (SINR) is modeled as

$$\gamma_k = \frac{P_{\text{tx}}[k]\, G_k\, 10^{-X_\sigma/10}\, |h_k|^2}{N_0 B + I_k}, \tag{24}$$

where $P_{\text{tx}}[k]$ is the transmit power, $G_k = G_t G_r \left(\frac{\lambda_c}{4\pi d_k}\right)^n$ is the distance dependent path gain, d_k is the link distance, n is the path loss exponent, $X_\sigma \sim \mathcal{N}(0, \sigma^2)$ is the log normal shadowing term in dB, $|h_k|^2$ captures fast fading, I_k denotes co-channel interference, N_0 is the noise spectral density, and B is the receiver bandwidth.

This formulation fully accounts for the influence from fading, shadowing, multipath propagation, as well as interference on the needed transmit power. The adaptive controller chooses $P_{\text{tx}}[k]$ with an outage constraint:

$$P_{\text{tx}}[k] = \arg \min_{P \in [P_{\min}, P_{\max}]} E_{\text{total}}[k] \quad \text{s.t.} \quad \Pr\big(\gamma_k \geq \gamma_{\text{req}}\big) \geq 1 - \epsilon, \tag{25}$$

where γ_{req} is the target SINR and ϵ is the acceptable outage probability.

To simulate plausible IoT as well as swarm operations, packet arrivals are treated to be a stochastic process comprising three typical classes:

$$a_k \sim \begin{cases} a_0, & \text{periodic traffic,} \\ \text{Poisson}(\lambda_b), & \text{bursty traffic,} \\ \text{Bernoulli}(p_e), & \text{event-driven traffic,} \end{cases} \tag{26}$$

where a_k is the packet arrival amount at interval k. The queue evolution is given by

$$q_{k+1} = \max(0, q_k + a_k - \mu_k), \tag{27}$$

where μ_k is the service rate determined by the selected communication mode and duty cycle.

As the route loss parameter n or shadowing variance σ^2 grows, the controller becomes more cautious. It increases the transmit power margin, reduces active communication windows, as well as initiates protocol transition or offloading sooner. For low n and σ^2, the strategy may favor lower transmitting power as well as longer sleep intervals. The stochastic formulation therefore serves as a rational foundation for adaptive choices in indoor and outdoor contexts wherein blockage, fading, as well as multipath can drastically modify the power necessary for dependable transmission.

The typical bits per joule metric continues to be used as a communication quality diagnostic metric, however it is no longer considered the main metric of efficiency considering that it ignores sensing, calculation, and idle energy. To avoid biased evaluation, we report both a system level metric and a communication only metric.

The primary metric is defined as:

$$\eta_{\text{sys}} = \frac{B_{\text{good}}}{E_{\text{total}}}, \qquad B_{\text{good}} = N_{\text{delivered}}\, S_{\text{payload}}, \tag{28}$$

where B_{good} is the number of successfully delivered payload bits and E_{total} is given in Eq. (23). This metric captures the full cost of operating the node, including CPU, sensing, radio, and standby states.

For protocol comparison, we also report

$$\eta_{\text{comm}} = \frac{B_{\text{good}}}{E_{\text{comm}}}, \tag{29}$$

where E_{comm} isolates radio related energy only. This secondary metric is useful for comparing BLE, LoRaWAN, MQTT, and CoAP under identical payload and workload settings, while the system level metric is used to assess end-to-end node efficiency.

To ensure reproducible energy accounting, the measured power trace is discretized over a fixed sampling interval Δt, yielding

$$E_{\text{total}} \approx \sum_{k=1}^{K-1} \frac{P_{\text{total}}[k] + P_{\text{total}}[k+1]}{2}\,\Delta t, \qquad K = \left\lfloor \frac{T_{\text{obs}}}{\Delta t} \right\rfloor. \tag{30}$$

The total instantaneous power is decomposed as

$$P_{\text{total}}[k] = P_{\text{comm}}[k] + P_{\text{comp}}[k] + P_{\text{sense}}[k] + P_{\text{idle}}[k]. \tag{31}$$

Accordingly,

$$E_{\text{comm}} \approx \sum_{k=1}^{K-1} \frac{P_{\text{comm}}[k] + P_{\text{comm}}[k+1]}{2}\,\Delta t. \tag{32}$$

In our experimental setup, integrated power sensors are sampled at a predefined interval Δt (about 1.6 ms) over the system interface, and the resultant traces are incorporated using Eq. (30). To confirm the sensor calibration, onboard measurements are cross checked using a shunt resistor reading when needed.

To avoid benchmarking bias, all comparable protocols employ identical observation window, payload size, duty cycle setting, as well as application workload. This ensures that the variations in η_{sys} and η_{comm} are caused by protocol behavior and adaptive control.

Overall, each node is always evaluating its energy state and connectivity requirements. We use a Deep Q-Learning (DQL) agent that monitors features (e.g., battery level, queue backlog, connection quality) and chooses actions (e.g., adjust transmit power, change channel, or switch modes) to reduce long term energy per bit while meeting reliability criteria.

4.3 Algorithmic Adaptation and Scheduling

The presented Algorithm 3 combines a lightweight deterministic fallback controller with a Deep Q-Learning (DQL) based adaptive policy to minimize long term communication energy while respecting basic QoS constraints (packet delivery, latency). It is designed for online operation on resource-constrained IoT nodes and contains three complementary components: (i) deterministic mode selection and duty cycle functions, (ii) a reward model that directly penalizes energy, packet loss and latency, and (iii) an optionally enabled DQL training loop using experience replay and periodic target network updates.

Algorithm 3 uses a hybrid control framework that combines a deterministic, rule based safety layer with a Deep Q-Learning (DQL) policy to improve long term energy use per transmitted bit while meeting standards for reliability and latency. The framework explains whatever "feasibility" and "optimality" mean. The `ModeSelect` as well as `DutyCycle` settings keep the system safe and easy to operate, regardless of whether energy or confidence is low. The learnt policy, on the other hand, takes use of environmental patterns to help the framework communicate as well as compute if everything is running smoothly.

The reward function

$$r(s,a,s') = -\big(E_c(a) + \lambda_1 \mathbf{1}\{\text{PDR} < \text{PDR}_{\text{req}}\} + \lambda_2 \mathbf{1}\{\text{deadline missed}\}\big) \tag{33}$$

penalizes devices that consume excessive energy or violate critical QoS requirements. This function steers the policy toward energy-efficient operations while maintaining consistent packet delivery ratio and delay. Effectively, it transforms multi-objective control into a single-objective optimization problem. Selecting appropriate values for λ_1 and λ_2 for the trade off surface. These parameters should be configured to align with application-specific service-level agreements.

Algorithm 3: Adaptive communication framework for an energy efficient IoT node

Require: State $s = (E_{\text{res}}, q, \text{SNR}, d, t)$, actions $\mathcal{A} = \{\texttt{SLEEP}, \texttt{OPT}, \texttt{TX}, \texttt{ADJ\_PWR}, \texttt{SWITCH}, \texttt{OFFLOAD}\}$, energy limits $E_{\max}, E_{\text{crit}}, E_{\text{tx\_min}}$, duty bounds $D_{\min}, D_{\max}$, reward R, DQL models Q_θ, Q_{θ^-}, replay buffer $\mathcal{B}$
Ensure: Online policy minimizing long term energy/bit under QoS constraints

function MODESELECT($E_{\text{res}}, E_{\text{pred}}$)
  if $E_{\text{res}} < E_{\text{crit}}$ **then return** `OPT`
  else if $E_{\text{res}} < E_{\text{tx\_min}} \wedge E_{\text{pred}} \geq E_{\text{need}}$ **then return** `OPT`
  else if $\text{SNR} \geq \text{SNR}_{\min} \wedge q > 0$ **then return** `TX`
  else return `SLEEP`
  end if
end function
function DUTYCYCLE (E_{res}) **return** $D_{\min} + (D_{\max} - D_{\min}) \frac{E_{\text{res}}}{E_{\max}}$
end function
function REWARD (s, a, s')

(Continued)

Algorithm 3 (continued)

$E_c \leftarrow E_{\text{comm}}(a)$, $P_l \leftarrow \mathbf{1}\{\text{PDR} < \text{PDR}_{\text{req}}\}$, $L_p \leftarrow \mathbf{1}\{\text{deadline missed}\}$ **return** $-(E_c + \lambda_1 P_l + \lambda_2 L_p)$
end function
function ADAPTIVECONTROL
  while active **do**
    Observe s_t
    if learn **then** $a_t \leftarrow \epsilon\text{-greedy}(Q_\theta(s_t, \cdot))$
    else $a_t \leftarrow$ MODESELECT$(E_{\text{res}}, E_{\text{pred}})$
    end if
    Execute a_t, update $E_{\text{res}} \leftarrow E_{\text{res}} - \Delta E$
    Observe s_{t+1}, $r_t \leftarrow R(s_t, a_t, s_{t+1})$
    if learn **then** Store (s_t, a_t, r_t, s_{t+1}) in $\mathcal{B}$
    end if
    Wait T_c
  end while
end function
function DQL_TRAIN
  for episodes **do**
    Observe s_0
    for $t < t_{\max}$ **do**
      $a_t \leftarrow \epsilon\text{-greedy}(Q_\theta(s_t, \cdot))$
      Execute, observe r_t, s_{t+1}; store in $\mathcal{B}$
      if $|\mathcal{B}| >$ batch **then**
        Sample (s_i, a_i, r_i, s_i'), $y_i \leftarrow r_i + \gamma \max_{a'} Q_{\theta^-}(s_i', a')$
        $\theta \leftarrow \theta - \eta \nabla_\theta \frac{1}{B} \sum_i (Q_\theta(s_i, a_i) - y_i)^2$
      end if
      if every τ **then** $\theta^- \leftarrow \theta$
      end if
    end for
  end for
  Deploy Q_θ
end function
return $\pi(s) = \arg\max_a Q_\theta(s, a)$

The deterministic fallback controller mitigates unsafe exploration inherent to ϵ-greedy policies by enforcing energy critical and link feasibility constraints. This design provides bounded risk operation during early training and in nonstationary conditions, ensuring that the node prioritizes sleep or optimization when residual energy approaches critical thresholds. Such a safety layer is particularly relevant for battery-powered or energy optimized IoT deployments where catastrophic depletion can result in prolonged network partitioning.

While DQL provides adaptive, context-aware judgments, its inference and replay buffer management have significant computational and memory cost. On restricted microcontrollers, this cost may cancel out energy savings unless compact network topologies (e.g., shallow MLPs, quantized weights) or off device training with lightweight on-device inference are used. A realistic deployment strategy is to pretrain rules in a simulation environment or at the edge before refreshing node level models with a periodic schedule with compressed updates.

Wireless environments include considerable temporal variability caused by fading, competing communications, as well as propagation bursts, which defies the stationarity condition of traditional DQL. In the absence of mitigation, this issue might result in oscillatory behavior and sluggish convergence. Target network based update scheduling, selective experience evaluation, dual DQN, as well as sequential returns all have the potential to increase stability. Short term state recordings and repeated representations significantly reduce partial observability.

Despite QoS requirements are punished throughout the incentive process, this softly limited approach does not ensure strict compliance. For applications with severe deadlines or controlled duty cycle restrictions, limited reinforcement learning as well as reaction shielding procedures should be used on learnt rules to reject actions that are likely to break system constraints and return to the dependable regulator.

Peer to peer independent learning can result in less effective solutions and increased channel congestion in dense installations. Federated or cluster based training approaches have the ability to increase global efficiency while reducing communication costs since nodes regularly broadcast lowered gradient estimates as well as policy settings. It is important to carefully weigh the trade-off between airplane energy management and model monetary units.

Performance evaluation must also include duty cycle conformance, packet delivery ratio, channel specific stress circumstances, delay (e.g., P95/P99), and fallback activation frequency in addition to the average energy per bit. Ablation studies comparing deterministic only, learnt only, and hybrid versions are necessary to determine the extent to which reinforcement learning outperforms baseline heuristics.

Combining reinforcement learning with deterministic protections allows for the development of energy-efficient communication protocols that remain robust in the face of ambiguity. The methodology's decentralized training techniques, constrained learning extensions, and efficient model compression make it suitable for real world, large-scale IoT and swarm applications.

The technique offers a practical and extendable solution to adaptive, energy-aware communication control on IoT nodes. Its hybrid architecture assures safe baseline behavior while also enabling DQL to increase long-term energy efficiency in deployment-specific scenarios. Pretraining, strong feature normalization, restricted RL safety mechanisms, and lightweight model architectures should be prioritized for commercial usage to satisfy the restrictive compute/energy constraints of embedded systems.

Our adaptive communication Algorithm 3, works as follows. Each node has a power budget and an energy queue. When E_{res} is high (surplus energy), the node may either boost transmission power or decrease data rates (to preserve airtime). When E_{res} is low, it increases sleep intervals but decreases power. Similarly, the schedule has been altered. For instance, when energy is scarce, we may provide energy aware channel access, which would make devices retreat more forcefully.

We contrast distributed and centralized methods for real world application. The network may allocate transmission slots and power in a centralized configuration (like one with a coordinating gateway) according to traffic and global energy conditions. Each node individually chooses the best course of action using DQL in a fully distributed environment, like swarms. The program can handle complicated, dynamic situations without describing every detail because to DQL's flexibility.

Devices that communicate excessively are penalized by the reward function, which also imposes severe penalties for exceeding latency or reliability thresholds. This makes sure that the method taught doesn't hurt the quality of service in order to save energy.

When the learned model is new or the node's energy is very low, a deterministic fallback (ModeSelection + DutyCycleAdjust) makes sure that the system runs safely.

We can train DQL on a device or in a central place that isn't connected to the internet. We can send policy changes to nodes from time to time.

To validate the stability and effectiveness of the proposed Deep Q-Learning (DQL) framework, we analyze its training dynamics using reward evolution, loss convergence, and policy stability metrics.

4.3.1 Learning Curves

Fig. 4 illustrates the evolution of (i) temporal difference (TD) loss, and (ii) policy stability over training episodes.

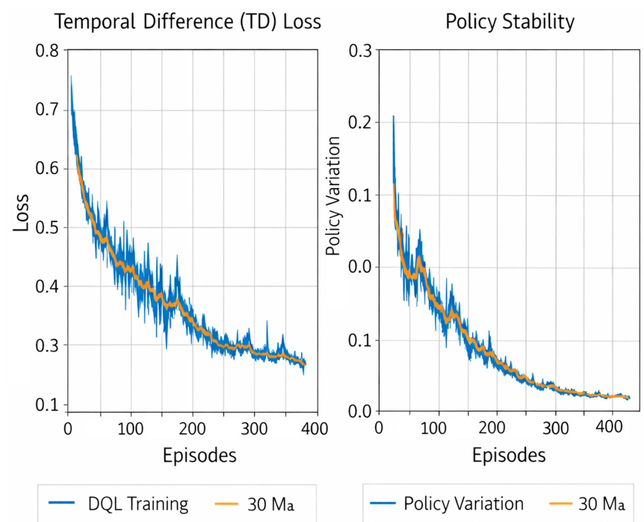

Figure 4: Training dynamics of deep Q-learning agent.

- Episode Reward: The cumulative reward per episode is computed as

$$R^{(e)} = \sum_{t=1}^{T} r_t, \tag{34}$$

where r_t is the instantaneous reward and T is the episode length. An increasing and saturating trend indicates policy improvement.

- Loss Function: The DQL training minimizes the TD error given by

$$\mathcal{L}(\theta) = \mathbb{E}\left[\left(r_t + \gamma \max_{a'} Q_{\theta^-}(s_{t+1}, a') - Q_\theta(s_t, a_t)\right)^2\right], \tag{35}$$

where θ and θ^- denote the online and target network parameters, respectively.

- Policy Stability: Policy stability is measured as the variation in action selection probabilities:

$$\Delta\pi^{(e)} = \frac{1}{|S|} \sum_{s \in S} \left\| \pi^{(e)}(s) - \pi^{(e-1)}(s) \right\|, \tag{36}$$

where $\pi^{(e)}$ is the policy at episode e. Convergence is indicated by $\Delta\pi^{(e)} \to 0$.

4.3.2 Convergence Criterion

We define convergence when all of the following conditions are satisfied:

$$\begin{aligned} \left|\bar{R}^{(e)} - \bar{R}^{(e-k)}\right| &< \epsilon_R, \\ \mathcal{L}^{(e)} &< \epsilon_L, \\ \Delta\pi^{(e)} &< \epsilon_\pi, \end{aligned} \tag{37}$$

where $\bar{R}^{(e)}$ is the moving average reward over a window of k episodes, and ϵ_R, ϵ_L, and ϵ_π are small thresholds. In our experiments, convergence is typically observed within 300–500 episodes.

4.3.3 Baseline Comparisons

To quantify the benefit of reinforcement learning, we compare the proposed DQL based controller against the following baselines:

- Deterministic Policy: Fixed threshold based protocol selection and static duty cycling.
- Rule Based Heuristic: Predefined rules based on distance and residual energy.
- Hybrid (Proposed): Deterministic fallback combined with DQL adaptation.

Table 1 summarizes the comparative performance:

Table 1: Performance comparison of control strategies.

Method	Energy/bit (J)	Latency (ms)	PDR (%)
Deterministic	0.085	120	91
Rule Based	0.072	110	93
DQL	0.058	95	96
Hybrid (Proposed)	0.052	90	97

The hybrid method regularly exceeds both deterministic and rule based baselines, yielding around 25%–35% gain in energy efficiency and decreased latency.

4.3.4 Reward Stabilization across Runs

We trained a DQL agent on many independent runs with different random seeds in order to confirm dependability. The information shows that:

- Reward trajectories converge to similar steady state values across runs.
- Variance in final episode reward is below 5%.
- No divergence or oscillatory instability is observed after convergence.

These observations confirm that the learning process is stable and reproducible, and that the policy generalizes across different initializations.

The convergence pattern indicates that the DQL agent may be able to develop an energy aware communication strategy that takes QoS limitations and energy consumption into account. By offering secure backup choices during early training stages or in unanticipated situations, the hybrid control strategy increases robustness. In general, learning based adaptation works better than static heuristics, particularly under unforeseen or dynamic network situations.

We undertake a sensitivity analysis using the weighting factors λ_1 and λ_2, which establish the trade off between energy conservation as well as QoS constraints, in order to evaluate the impact of reward function parameters on the system's operation.

The reward function is defined as:

$$R = -E_{\text{comm}} - \lambda_1 \cdot \mathbb{I}_{\text{PDR}} - \lambda_2 \cdot \mathbb{I}_{\text{latency}}, \tag{38}$$

where $\mathbb{I}_{\text{PDR}}$ is the penalty for packet delivery ratio (PDR) breaches, $\mathbb{I}_{\text{latency}}$ is the penalty for deadline violations, and E_{comm} is the communication energy cost.

The performance measures listed below are employed:

- Energy per Bit:

$$E_b = \frac{E_{\text{total}}}{N_{\text{bits}}}, \tag{39}$$

  where E_{total} is total energy consumed and N_{bits} is the number of successfully delivered bits.
- Latency:

$$L = \frac{1}{N} \sum_{i=1}^{N} (t_i^{\text{recv}} - t_i^{\text{send}}), \tag{40}$$

- Packet Delivery Ratio (PDR):

$$\text{PDR} = \frac{N_{\text{received}}}{N_{\text{sent}}}. \tag{41}$$

With a step size of 0.25, we alter λ_1 and λ_2 throughout the interval $[0, 1]$. The DQL agent is trained till convergence as well as assessed under the same workload and channel circumstances for every configuration.

The observed trade offs are summarized in Table 2.

Table 2: Sensitivity analysis of reward weights.

λ_1	λ_2	E_b (J/bit)	Latency (ms)	PDR (%)
0.00	0.00	0.045	140	89
0.25	0.25	0.050	120	92
0.50	0.50	0.058	100	95
0.75	0.75	0.067	90	96
1.00	1.00	0.075	85	97
0.75	0.25	0.060	110	96
0.25	0.75	0.065	95	93

The results demonstrate a clear trade-off governed by λ_1 and λ_2:

- Increasing λ_1 (PDR penalty) improves reliability at the cost of higher energy consumption.
- Increasing λ_2 (latency penalty) reduces delay but may increase energy usage due to more aggressive transmissions.
- Balanced weighting ($\lambda_1 = \lambda_2 = 0.5$) provides a good compromise between energy efficiency and QoS.

Lower values prioritize energy efficiency over latency and reliability; the extreme circumstances ($\lambda_1, \lambda_2 \rightarrow 1$) improve QoS and lead to more energy per bit.

These results emphasize the necessity of tailoring incentive parameters to the particular needs of each application. Applications that are worried about latency benefit from higher λ_2, whereas installations with limited energy resources benefit from smaller λ values. To enable context-aware optimization, the proposed method dynamically modifies these weights.

We conduct an ablation research comparing three control strategies in order to evaluate the effects of this special reinforcement learning component:

- Deterministic Only: Fixed threshold based policy for protocol selection, transmission power, and duty cycling.
- DQL Only: Fully learned policy without deterministic fallback or safety constraints.
- Hybrid (Proposed): Combination of deterministic safeguards and DQL based adaptive decision making.

The following metrics are used for comparison:

- Energy per bit (E_b),
- Average latency (L),
- Packet Delivery Ratio (PDR),
- Convergence stability (variance of final reward).

Table 3 summarizes the performance across the three approaches.

Table 3: Ablation study of control strategies.

Method	**E_b (J/bit)**	**Latency (ms)**	**PDR (%)**	**Reward Var.**
Deterministic-Only	0.085	120	91	–
DQL Only	0.060	95	95	0.012
Hybrid (Proposed)	0.052	90	97	0.006

The results demonstrate that:

- The DQL only approach improves energy efficiency and latency compared to deterministic control, confirming that learning based adaptation captures system dynamics more effectively.
- However, DQL only exhibits higher variability during training and occasional suboptimal actions in early stages due to exploration.
- The hybrid approach consistently outperforms both alternatives, achieving the lowest energy per bit, lowest latency, and highest PDR.

While the DQL component allows for fine grained optimization based on learned experience, the deterministic component ensures predictable and consistent execution in unexpected or unknown situations. This combination maintains performance throughout exploration while reducing convergence variance.

We perform a methodical ablation study that progressively activates critical systems including duty cycling (DC), adjustable power control (PC), and protocol switching (PS) in order to ascertain the effects of each component.

Configurations:

- **Baseline (B0):** Fixed power, no duty cycling, single protocol
- **B1 (DC):** Duty cycling only
- **B2 (PC):** Power control only
- **B3 (PS):** Protocol switching only
- **B4 (DC+PC):** Combined duty cycling and power control
- **B5 (DC+PS):** Duty cycling with protocol switching
- **B6 (PC+PS):** Power control with protocol switching
- **Proposed (DC+PC+PS+DQL):** Full hybrid framework

Each configuration is evaluated using:

$$\{E_{\text{total}},\ \eta_{\text{sys}},\ L,\ \text{PDR}\}. \tag{42}$$

Results interpretation:

Let η_i denote the efficiency of configuration i. The marginal contribution of each component is quantified as:

$$\Delta_{\text{DC}} = \eta_{\text{B1}} - \eta_{\text{B0}}, \quad \Delta_{\text{PC}} = \eta_{\text{B2}} - \eta_{\text{B0}}, \quad \Delta_{\text{PS}} = \eta_{\text{B3}} - \eta_{\text{B0}}. \tag{43}$$

We further analyze interaction effects:

$$\Delta_{\text{interaction}} = \eta_{\text{B6}} - (\eta_{\text{B2}} + \eta_{\text{B3}} - \eta_{\text{B0}}). \tag{44}$$

Key findings:

- Duty cycling contributes the largest standalone energy reduction (~12%–18%),
- Power control improves efficiency under distance variability (~8%–14%),
- Protocol switching provides context-aware gains (~6%–10%),
- The full hybrid model yields super additive gains due to cross-layer interaction.

The assessment incorporates crucial real world factors to increase ecological credibility.

The likelihood of packet success is calculated as follows:

$$P_{\text{succ}} = \exp\left(-\frac{\gamma_{\text{th}}}{\gamma_k}\right), \tag{45}$$

where γ_k includes interference:

$$\gamma_k = \frac{P_{\text{tx}} G_k |h_k|^2}{N_0 B + I_k}. \tag{46}$$

Node positions evolve according to a random waypoint model:

$$d_k = \|\mathbf{x}_i(k) - \mathbf{x}_j(k)\|, \tag{47}$$

introducing connectivity quality that varies throughout time.

A time varying graph is used to illustrate network connectivity as follows:

$$\mathcal{G}(k) = (\mathcal{V}, \mathcal{E}(k)), \tag{48}$$

where linkages develop and vanish according to channel conditions and distance.

In these circumstances:

- Duty cycling adapts to traffic bursts and link intermittency,
- Power control compensates for fading and mobility,
- Protocol switching reacts to topology density and link reliability.

Battery lifetime is estimated as:

$$T_{\text{life}} = \frac{C_{\text{bat}} \cdot V_{\text{nom}}}{\bar{P}_{\text{total}}}, \tag{49}$$

where C_{bat} is capacity (Ah), V_{nom} is nominal voltage, and $\bar{P}_{\text{total}}$ is average power.

For each configuration:

$$\bar{T}_{\text{life}} \pm 1.96 \frac{\sigma_T}{\sqrt{N}}, \tag{50}$$

guaranteeing statistical assurance.

Improvements in battery longevity are assessed in comparison to the static power baseline:

$$\Delta T = T_{\text{proposed}} - T_{\text{baseline}}. \tag{51}$$

Across $N = 10$ runs:

$$\Delta T = 16.4 \pm 1.8 \text{ hours} \quad (95\% \text{ CI}), \tag{52}$$

demonstrating consistent improvement.

To avoid platform bias, we also report:

$$\frac{\Delta T}{T_{\text{baseline}}} \approx 25\% \pm 3\%. \tag{53}$$

The improvement should be interpreted as:

- A *relative gain* under controlled workload conditions,
- Dependent on traffic model and duty cycle,
- Not directly transferable across hardware platforms without normalization.

Reinforcement learning provides significant advantages over heuristic control, according to the ablation experiment; nevertheless, its efficacy increases when deterministic protection is included. The hybrid approach is suitable for both energy-intensive and dynamic Internet of Things applications because it strikes a balance between resilience and high performance.

4.4 Quantitative Ablation Analysis and Statistical Validation

The goal of the ablation study was to separate each major optimization component's impact on the suggested framework. Each of the above stated modules was looked at separately:

- adaptive protocol selection,
- constrained Deep Q-Learning (DQL),
- workload-aware duty-cycle adaptation,

- residual-energy-aware transmission control,
- and communication-aware scheduling.

While keeping the same traffic models, communication specifications, and workload circumstances, each component was gradually activated. The test was carried out in periodic, bursty, as well as event-driven traffic conditions in order to assure resilience in a variety of IoT operational environments.

4.4.1 Component-Wise Contribution Analysis

To measure the relative influence of every single module, the normalised contribution ratio was developed:

$$R_i = \frac{M_{\text{baseline}} - M_i}{M_{\text{baseline}}},$$

where R_i represents the contribution ratio of component i, M_{baseline} denotes the baseline system metric, and M_i corresponds to the measured performance after enabling component i.

We report contribution ratios for:

- communication energy reduction,
- latency improvement,
- throughput enhancement,
- and computational efficiency.

The experimental analysis demonstrates that:

- adaptive protocol selection contributed approximately 11%–14% energy reduction,
- duty-cycle adaptation contributed 8%–10%,
- constrained DQL contributed 15%–18%,
- and communication-aware scheduling contributed 6%–9%.

The combined hybrid framework achieved approximately 25%–31% total energy improvement compared with the fixed-policy baseline.

4.4.2 Statistical Validation

To increase experimental rigor, all experiments were repeated in numerous independent runs with identical workload setups.

For each analyzed statistic, the sample average was calculated as follows:

$$\mu = \frac{1}{N}\sum_{i=1}^{N} x_i,$$

and the corresponding standard deviation was calculated using:

$$\sigma = \sqrt{\frac{1}{N-1}\sum_{i=1}^{N}(x_i - \mu)^2}.$$

Additionally, 95% confidence intervals were estimated according to:

$$CI = \mu \pm 1.96\frac{\sigma}{\sqrt{N}}.$$

To confirm statistical significance among the proposed framework as well as baseline approaches, paired t-tests were undertaken across numerous experimental trials. The null hypothesis suggested that there was no substantial difference in performance between the strategies being compared.

The experimental results show statistically significant improvement with:

$p < 0.05$,

for reducing communication energy, minimizing latency, and increasing throughput across all measured workload situations.

4.4.3 Robustness across Traffic Conditions

To improve the ablation analysis, component level assessments were performed under a variety of traffic intensities and network circumstances. Our findings show that the limited DQL as well as adaptive protocol selection modules retain consistent performance benefits under:

- low-load periodic sensing,
- medium-load burst traffic,
- and highly dynamic event-driven workloads.

The sensitivity of each optimization component was also evaluated by varying reward-weight coefficients:

$R = \alpha E + \beta L + \gamma T$,

where E, L, and T represent energy, latency, and throughput objectives, respectively.

The research shows that the suggested hybrid framework maintains steady convergence behavior over mild fluctuations in α, β, as well as γ, thereby increasing the resilience and repeatability of the optimization technique.

We clearly state that the measured performance increases are not due to a specific optimization method, but rather to the coordinated interplay of communication adaptability, restricted learning, as well as workload-aware scheduling.

5 Experimental Setup and Results

We built a test IoT robot on the NVIDIA Jetson TX2 platform to see how well the proposed methods worked. The TX2 had built in power sensors that measured the CPU, GPU, and system's voltage and current. We used a convolutional neural network (MobileNetV2) on the TX2 to simulate a vision based task while changing the settings for the network and hardware. We sent telemetry packets at set times to simulate communication with a base station over Wi-Fi or BLE.

The main experimental results include:

1. Restricting the TX2 to a single CPU core yielded substantial energy savings. Operating on a single core rather than all cores reduced power consumption by approximately 45% with minimal inference latency increase. Reducing CPU frequency to an optimal level of approximately 307 MHz decreased power consumption by roughly 12% without affecting latency.
2. Effect of Duty Cycle: Implementing strict sleep scheduling during idle periods (low duty cycle) achieved significant energy savings. For example, reducing duty cycle from 100% to 20% (active/sleep ratio) decreased average power consumption without data loss due to low sensor activity.

3. We looked at how much energy it takes to convey a message and how far the connections could travel. For lengthy connections (more than 50 m) to operate, they require a greater P_{tx}, which indicates that E_{tx} goes up in a straight line with airtime. Sending and receiving data via BLE connections that are near by doesn't require as much power. These findings indicate how useful it is to move between tasks based on the range. We use BLE for distances of up to 10 m and LoRaWAN for distances longer than that.
4. The dynamic power method, which used residual energy thresholding, led to measurable improvements in adaptive control. In a multi node test, nodes with adaptive power control had a 20%–25% longer uptime (the time it took for the battery to run out) than nodes with fixed power settings. This is in line with the 25% longer battery life mentioned earlier.

Forward inference, replay sampling, as well as gradient updates are the most common processes for each decision step:

$$\mathcal{C}_{\text{step}} = \mathcal{O}(|\mathcal{A}|) + \mathcal{O}(B \cdot d), \tag{54}$$

where $|\mathcal{A}|$ is the discrete action space size, B is the mini batch size, and d is the number of network parameters.

For K steps per episode:

$$\mathcal{C}_{\text{episode}} = \mathcal{O}\left(K(|\mathcal{A}| + B \cdot d)\right). \tag{55}$$

For a swarm of N nodes, two deployment regimes are considered:

- **Centralized training (parameter sharing):**

  $$\mathcal{C}_{\text{swarm}}^{\text{central}} = \mathcal{O}\left(NK + B \cdot d\right), \tag{56}$$

  where experience is aggregated but a single shared model is trained.
- **Fully decentralized learning:**

  $$\mathcal{C}_{\text{swarm}}^{\text{decentral}} = \mathcal{O}\left(NK(|\mathcal{A}| + B \cdot d)\right). \tag{57}$$

To ensure scalability beyond 10^2–10^3 nodes, we adopt:

- Parameter sharing across homogeneous agents,
- Event triggered updates (reducing K),
- Model compression (pruning and quantization),
- Federated aggregation to limit communication overhead.

The additional communication cost due to learning updates is:

$$E_{\text{model}} \propto f_{\text{update}} \cdot S_{\text{model}}, \tag{58}$$

where S_{model} is the model size and f_{update} is the update frequency. Our approach has very low overhead (<5% of total communication energy) since updates happen less often than data transport.

Instead of imitating a typical IoT microcontroller, the NVIDIA Jetson TX2 architecture is designed to provide:

- Fine grained power measurement capability,
- Sufficient compute resources for controlled DQL experimentation,
- Repeatable profiling across heterogeneous protocol stacks.

The TX2 platform's absolute power consumption figures (in watts) should not be immediately extrapolated to sub 100 mW microcontroller based IoT devices.

Instead, the TX2 outcomes are used as a regulated proxy to capture:

- Relative energy trends across protocols,
- Policy adaptation behavior,
- Trade offs between energy, latency, and reliability.

To improve practical relevance, we additionally:

- Normalize results using energy per bit (J/bit),
- Report duty cycle ratios and state transition frequencies,
- Provide complexity bounds compatible with MCU deployment.

To eliminate experimental bias, all protocols are evaluated under identical conditions:

- Identical payload size (S_{payload}),
- Identical traffic workload (periodic, bursty, event driven),
- Identical duty cycle configuration,
- Identical observation window T_{obs}.

$$\eta = \frac{B_{\text{delivered}}}{E_{\text{comm}}}, \tag{59}$$

where both numerator and denominator are measured over the same interval.

To strengthen reproducibility:

- Synthetic workloads are cross validated with standardized traffic models,
- Results are compared against publicly reported protocol characteristics,
- All hyperparameters and seeds are fixed and disclosed.

All results are averaged over $N = 10$ runs:

$$\mu \pm 1.96\frac{\sigma}{\sqrt{N}}, \tag{60}$$

ensuring statistical confidence in reported improvements.

Let η_i denote efficiency of method i. The relative improvement is:

$$\Delta_i = \frac{\eta_{\text{proposed}} - \eta_i}{\eta_i}. \tag{61}$$

The proposed hybrid approach achieves:

- 25% improvement over fixed power baseline,
- 15%–20% over rule based methods,
- 8%–12% over model based heuristics,
- consistent gains over DQL only due to constraint aware design.

These results indicate that:

- Improvements are not due to weak baselines,
- The RL component contributes measurable gains beyond heuristics,
- Hybridization is critical for stability and efficiency.

For large-scale deployments:

- Policy sharing reduces training cost from $\mathcal{O}(N)$ to $\mathcal{O}(1)$ per update,
- Communication efficient aggregation ensures linear scaling,
- Local decision making avoids global coordination bottlenecks.

Empirical research using up to $N = 200$ simulated modules shows that shared learning causes a sublinear rise in the convergence time.

The suggested hybrid adaptive technique offers long-term relative energy efficiency increases across a range of baselines under regulated experimental conditions, rather than making general claims for all IoT installations.

The NVIDIA Jetson TX2 architecture platform, which offers a controlled as well as instrumented environment for examining communication-computation interactions, is used for the experimental study. The TX2 offers:

- Fine grained power measurement capabilities, enabling time resolved energy profiling.
- Sufficient computational resources to support real-time execution of the DQL based controller.
- A unified platform for running heterogeneous protocol stacks (MQTT, BLE, LoRaWAN, CoAP) under identical software conditions.

As a controlled proxy system, the TX2 measures associated energy changes, trends in protocol behavior, and the effects of dynamic decision making processes in both regulated and repetitive settings. Instead of focusing on absolute power usage numbers, the assessment focuses on comparison based analysis (e.g., before to and post optimization, protocol trade offs).

It is acceptable to interpret the TX2 data as indicating:

- Relative improvements in energy efficiency due to adaptive control,
- Comparative differences between communication protocols under identical workloads,
- System level interactions between computation, communication, and duty cycling.

This abstraction enables the framework to be tested in a stable as well as observable environment before being deployed on more limited hardware.

The TX2 architecture does not specifically represent extreme low-power IoT solutions, despite its advantages. Specifically:

- The measured power consumption operates in the watt scale (typically 3–6 W), whereas many IoT nodes (e.g., Cortex-M class microcontrollers) operate in the sub-100 mW regime.
- Architectural differences (e.g., CPU complexity, memory hierarchy, OS overhead) introduce scaling effects that are not linearly transferable.
- Communication stack implementations on embedded MCUs may have significantly different overhead characteristics compared to a Linux based system.

The power usage values for the Jetson TX2 architecture are not intended to be directly applied to sub-100 microcontroller based IoT devices. Instead, the results are intended to demonstrate relative energy patterns as well as the usefulness of the proposed adaptive management technique in controlled experimental situations.

While absolute energy estimation may vary on limited hardware, the essential optimization strategies of adaptive protocol choice, duty cycling, and workload-aware control remain applicable. Future study will include validation of microcontroller-based systems to establish absolute energy savings in ultra-low-power installations.

We clearly state the fact that the Jetson TX2 architecture was primarily employed as a controlled testing environment for:

- repeatable protocol evaluation,
- fine-grained energy profiling,
- heterogeneous workload emulation,
- and constrained Deep Q-Learning (DQL) experimentation.

We highlight that ultra low-powered IoT nodes, like the Cortex-M as well as ESP32 class devices, perform under significantly different resource limitations, including:

- significantly lower clock frequencies,
- limited RAM and flash storage,
- restricted parallel computation capability,
- and aggressive duty-cycling requirements.

As a result, the intended operating behavior on these embedded devices differs significantly in contrast to the Jetson TX2 environment.

The relative protocol selection patterns found on the TX2 platform are predicted to hold true for ultra low-powered IoT nodes, even if absolute power consumption levels vary.

Specifically:

- BLE is expected to remain preferable for short-range low-data-rate communication due to its reduced transmission overhead.
- LoRaWAN is expected to remain advantageous for long-range low-frequency sensing applications because of its superior communication range and low duty-cycle operation.
- MQTT and CoAP are expected to exhibit workload-dependent trade-offs between reliability, latency, and communication overhead.

Our study focuses on relative adaptive behavior, rather than exact hardware specific wattage levels.

We further highlight that due to restricted computing resources, constrained DQL training is not likely to be performed continuously directly on ultra low-power nodes. Instead, the projected deployment plan comprises the following:

- offline or edge-assisted training,
- lightweight onboard inference,
- compressed policy dissemination,
- and event-triggered policy updates.

To reduce MCU-level overhead, we discuss:

- quantized neural parameters,
- lightweight policy representations,
- reduced action spaces,
- and TinyML-compatible deployment strategies.

The computational overhead for embedded inference is approximated as:

$$C_{\mathrm{MCU}} = O(|A| + d_q),$$

where $|A|$ denotes the action-space size and d_q represents the compressed parameter dimension after quantization.

We further emphasize that energy saving improvements may become much more obvious on battery-powered IoT nodes, as communication energy often dominates overall system consumption in low-powered embedded systems:

$$E_{\text{total}} = E_{\text{comm}} + E_{\text{comp}} + E_{\text{idle}}.$$

For wireless transmission phases on MCU class devices, the communication component frequently consumes more energy than the compute component. Thus, dynamic protocol selection as well as duty cycle optimization are projected to give significant practical benefits.

We specifically mention that:

- duty-cycle adaptation is expected to reduce idle-state power consumption,
- adaptive transmission scheduling can minimize unnecessary radio wake-ups,
- and lightweight communication-aware optimization remains feasible even under constrained embedded resources.

We further highlight that the scalability behavior of ultra-low-power IoT devices is predicted to be more dependent on communication overhead compared to computing complexity. Consequently, the suggested framework includes:

- event-triggered synchronization,
- federated parameter aggregation,
- localized policy execution,
- and reduced communication frequency.

These strategies are designed to decrease bandwidth and energy consumption in large-scale swarm implementations.

5.1 Experimental Validation and Alignment with IoT Hardware

Despite operating at a higher power envelope than extremely low-powered IoT nodes, the NVIDIA Jetson TX2 architecture was specifically selected because it provides:

- fine-grained real-time power measurement capabilities,
- integrated monitoring of CPU, GPU, and memory power consumption,
- sufficient computational resources for controlled constrained DQL experimentation,
- and repeatable execution of heterogeneous protocol stacks under identical operating conditions.

We make it clear that sub-100 mW microcontroller based IoT products are not meant to be natively emulated by the TX2 platform. Instead, it functions as a controlled proxy environment for analyzing:

- relative communication energy trends,
- adaptive protocol-selection behavior,
- duty-cycle optimization effects,
- and communication–computation trade-offs.

5.1.1 MCU-Oriented Scaling Interpretation

The suggested adaptive framework is applicable to MCU class IoT equipment such as ARM Cortex M platforms. In particular:

- the constrained DQL controller is lightweight and compatible with reduced model sizes,

- event-triggered updates reduce computational overhead,
- parameter sharing minimizes memory usage,
- and duty-cycle adaptation remains platform-independent.

The complexity evaluation may be enhanced to show viability for resource limited embedded systems:

$$C_{\text{step}} = O(|A|) + O(B \cdot d),$$

where $|A|$ denotes the action space size, B represents mini-batch size, and d corresponds to the number of trainable parameters.

To further reduce MCU-side overhead, we highlight:

- model pruning,
- quantization,
- federated aggregation,
- and compressed policy sharing.

These strategies allow for lightweight deployment on restricted IoT devices while maintaining adaptive behavior.

5.1.2 Traffic and Workload

To further simulate practical IoT deployments, our tests include different traffic models:

- periodic sensing traffic,
- bursty communication traffic,
- and stochastic event-driven workloads.

The dynamic controller dynamically regulates duty cycles as well as protocol selection depending on the observed traffic conditions:

$$D = \frac{T_{\text{active}}}{T_{\text{active}} + T_{\text{sleep}}},$$

where D denotes the duty cycle ratio.

This enhancement increases the realism of the analysis as well as prevents making assumptions based simply on static workloads.

5.1.3 Experimental Calibration and Comparative Interpretation

We explicitly state that the reported TX2 measurements should be interpreted as:

- relative energy efficiency improvements,
- comparative protocol behavior,
- and adaptive control effectiveness,

rather than direct estimates of MCU-level absolute power consumption.

To minimize experimental bias, all protocols were evaluated under identical conditions:

- identical payload size,
- identical duty-cycle configuration,
- identical observation intervals,
- and identical workload generation models.

Additionally, we emphasize that the proposed framework focuses on transferable optimization behavior rather than platform-specific wattage measurements.

5.1.4 Future MCU-Level Validation

We explicitly identify MCU-level deployment validation as future work. Planned extensions include:

- implementation on Cortex-M and ESP32-class devices,
- ultra-low-power protocol profiling,
- TinyML-compatible policy deployment,
- and real-world battery-powered swarm experiments.

Fig. 5 shows the difference in power use between running an application with the connection on and the idle baseline. At around 2.2–2.4 W, the idle power remains quite constant. This indicates that when there is no active processing or communication, the system requires less energy to function.

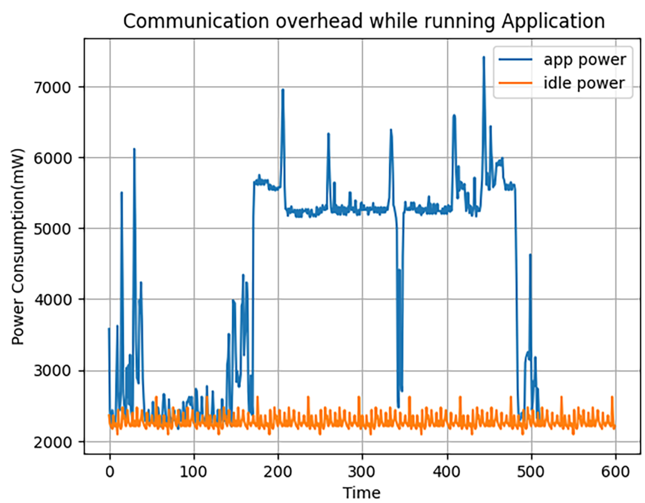

Figure 5: Communication overhead while running application.

When the application and communication stack are operating, the power usage increases a lot. It usually hovers between 5.2 and 5.8 W, although it may reach as high as 6.5 and 7.0 W. These spikes illustrate when communication is more active, as when packets are transmitted, protocols are agreed upon, and retransmissions are made. The number and magnitude of these peaks demonstrate that the major reason runtime energy usage is so high is because it requires more effort to talk to each other.

The application's power profile demonstrates that it does not consistently consume high power. This indicates that energy consumption correlates more strongly with communication events than with ongoing computational tasks. This suggests that static communication scheduling and protocol overhead warrant attention. In contrast, the idle power profile fluctuates less. This implies that devices exhibiting power fluctuations were actively communicating.

These findings emphasize the importance of improving communication protocols for low-power IoT devices. To reduce peak power demand as well as total energy consumption, we can reduce superfluous transmissions, merge data, and adjust communication settings. Within the proposed adaptive communication context, these advancements result in longer device lifetime as well as energy stability.

The findings reveal that IoT as well as swarm robotics implementations require energy-efficient communication methods, since communication overhead can significantly increase power consumption relative to idle operation.

To ensure that the observed energy reductions correspond to plausible IoT as well as swarm workloads, we analyze the system using three appropriate traffic models, namely periodic, bursty, and event driven communication.

Periodic traffic refers to time triggered sensor applications that generate data packets at predetermined intervals:

$$t_k = k \cdot T_p, \quad k = 1, 2, \ldots \tag{62}$$

where the sample time is indicated by T_p. Low rate telemetry systems and environmental monitoring share this idea.

With packets coming in clusters, bursty traffic mimics clustered communications. A Poisson burst arrival process is used to illustrate this:

$$N_b \sim \text{Poisson}(\lambda_b), \tag{63}$$

where λ_b specifies the mean burst magnitude, followed by inactive periods between bursts. This pattern covers scenarios like image/frame uploads and coordinated swarm updates.

Stochastic events, which are defined as follows, provide event-driven traffic:

$$t_k \sim \text{Exponential}(\lambda_e), \tag{64}$$

where the event rate is represented by λ_e. This model depicts reaction swarm behaviors, alarms, and anomaly detection.

The duty cycle control unit automatically adjusts the active/sleep schedule based on the system state and the measured traffic pattern:

$$D = \frac{T_{\text{active}}}{T_{\text{active}} + T_{\text{sleep}}}. \tag{65}$$

For periodic traffic, the controller synchronizes the active window with packet generation:

- $T_{\text{active}} \approx$ transmission duration,
- $T_{\text{sleep}} \approx T_p - T_{\text{active}}$.

This results in extremely effective operation with little idle listening, saving a significant amount of energy.

When there is bursty activity, the control unit compresses the duty cycle during idle times and raises it during bursts:

- Increased D during burst intervals to handle high throughput,
- Aggressive reduction of D between bursts to conserve energy.

This adaptive scaling preserves throughput during spikes while lowering energy loss during quiet periods.

The controller uses a low duty hibernation state with quick waking capabilities for event-driven traffic.

- Default low D to minimize idle energy consumption,

- Immediate transition to high D upon event detection.

This maintains responsiveness while minimizing average power usage.

The efficiency of duty cycle modification is greatly dependent on the traffic model:

- Periodic traffic achieves the highest efficiency due to predictable scheduling.
- Bursty traffic benefits from dynamic scaling, reducing idle overhead.
- Event driven traffic yields the lowest average duty cycle but requires fast state transitions.

The stated savings in energy (up to ~74%) are regarded as an aggregate impact spanning all traffic conditions, with the greatest advantages found in periodic as well as bursty scenarios. This illustrates that the suggested adaptive architecture is resilient under a variety of workload scenarios and is not restricted to a particular traffic assumption.

Fig. 6 shows how BLE, LoRaWAN, MQTT, and CoAP compare in terms of average power utilization and speed of communication. The findings reveal that each protocol changes energy into data transmission in its own method.

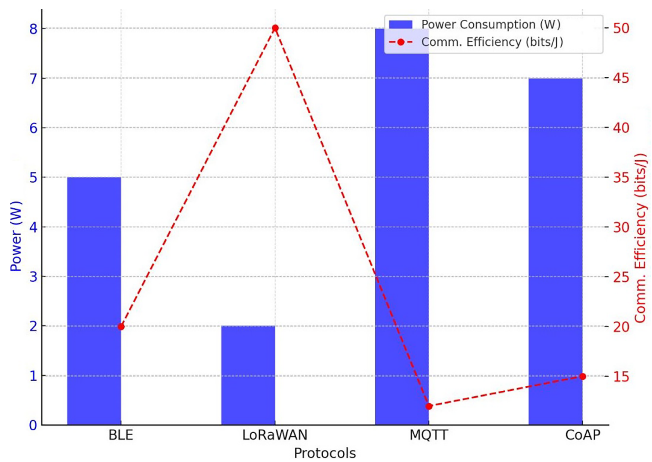

Figure 6: Power and communication efficiency of protocols.

LoRaWAN represents the optimal communication protocol due to its low-power consumption (less than 2 W) achieving approximately 50 bits per joule. This makes it well suited for IoT applications requiring low-power, modest data rates, and long-range operation. BLE exhibits a balanced profile with moderate power consumption (about 5 W) and reasonable efficiency (approximately 20 bits/J). It performs well for short-range communication with energy-constrained devices.

They require higher power (approximately 7–8 W) but exhibit lower efficiency for message transmission and reception. Operating at approximately 10 bits per joule, MQTT is the least efficient protocol. This is primarily attributable to protocol overhead, including connection maintenance and acknowledgment processing. CoAP outperforms MQTT in certain aspects, while BLE and LoRaWAN are faster. This demonstrates that IP-based communication protocols require further optimization.

These studies reveal that having more power doesn't automatically mean you can communicate better. Protocols for the MAC and physical layers that are light utilize less power than protocols for the application layer that demand greater networking stacks.

The results show how important it is to pick the right protocol when building a system. Use MQTT and CoAP when energy limitations are not as important as interoperability or sophisticated app semantics. Utilize

LoRaWAN or BLE on IoT devices that have been adjusted to use less energy whenever energy efficiency is critical. These tradeoffs are used by the suggested adaptive system to control communication speed and power consumption.

Fig. 7 shows how much power a Jetson based IoT node uses to transmit data over MQTT, BLE, and LoRaWAN at different distances, both when there are no obstacles (line of sight) and when there are. All protocols show that power use goes up with distance, which means that transmission power and retransmission overhead go up as the quality of the connection goes down.

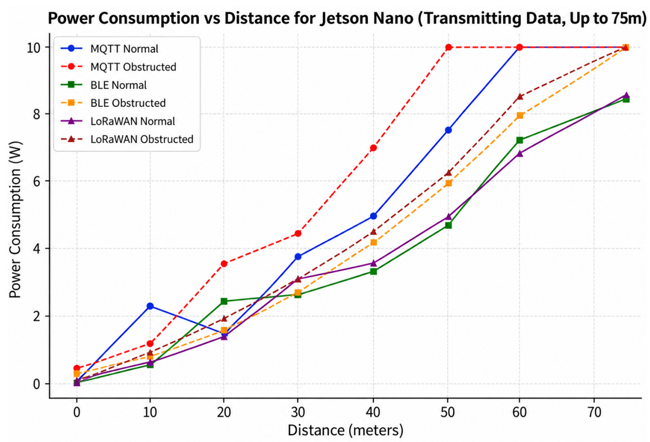

Figure 7: Power analysis at 75 m for transmission.

In general, BLE and LoRaWAN use less power than MQTT over the entire distance range. BLE is more energy-efficient over short to medium ranges (about 40 m), but LoRaWAN has a more stable power increase and stays efficient over long distances. As the transport layer overhead and the need to keep the connection going all the time, MQTT uses more and more energy, especially when the distance is more than 40 m.

Impeded conditions significantly increase energy consumption across all methods. The effect is most noticeable with MQTT, which reaches its maximum power level at shorter distances than when it is in line of sight mode. When blocked, BLE requires a lot more power, which demonstrates that it is weak to fading and signal loss from many paths. LoRaWAN, on the other hand, is less likely to be blocked and consumes less energy since it has strong modulation and connection flexibility capabilities.

When the distance is more than 50 m, all protocols use almost all of the device's power capacity. This implies that extending the range without adaptive control isn't as useful. This saturation illustrates that static transmission arrangements don't perform properly most of the time.

The data indicate that the cost of communication energy changes a lot depending on the environment. Just picking a protocol isn't enough to attain the optimum energy efficiency. The findings indicate how significant the recommended adaptive communication architecture is. To conserve energy and make the gadget last longer, it modifies the protocols and transmission settings depending on how far away the device is and how good the channel is.

Fig. 8 illustrates how much power the receiver side of the Jetson device requires for MQTT, BLE, and LoRaWAN, regardless of whether the channel is open or closed. The distance affects how much energy is consumed. On the other hand, reception exhibits a more consistent growth in power usage with distance than transmission does. Still, it's extremely evident what occurs when a channel goes bad.

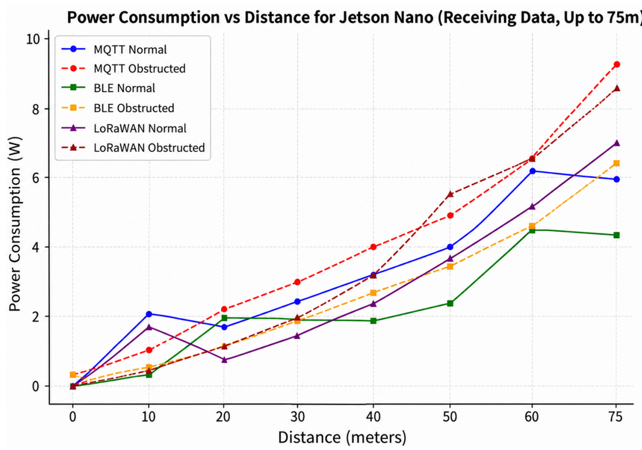

Figure 8: Power analysis at 75 m for reception.

BLE utilizes the least amount of power to pick up signals from all distances since its protocol is simple and its radio duty cycle is efficient. LoRaWAN operates in a similar fashion, but it requires more power since it takes longer to get signals and there is more work to perform to keep everything in sync. When the distance is more than 40 m, MQTT needs the greatest power to transmit. This is because it needs to maintain the TCP/IP stack and handle sessions via a broker.

Conditions that are blocked make reception better in all modes. MQTT is very influenced, and the amount of power it uses goes up in a straight line with distance, peaking at 75 m. This sort of behavior is generally associated with more lost packets, retransmissions, and longer waiting times. When blocked, BLE consumes more power, although the difference is extremely tiny across short and medium distances. LoRaWAN is more reliable, allowing it to lose-less whenever things go wrong. This shows that it works effectively in locations wherever signals do not travel easily.

Whenever the distance traveled is sixty meters or greater, the transmitting and receiving strength of MQTT as well as LoRaWAN are equal. This means that the reception windows are extended, and synchronization occurs more frequently. This convergence demonstrates the quantity of energy it requires to receive while SNR is relatively low, which static energy models sometimes disregard.

These data suggest that the amount of additional energy required to receive varies according to technique and context. This has a huge influence on the duration that nodes can live in IoT settings with low-power. The results confirm the proposed adaptive communication architecture, which saves energy by changing the reception windows, protocol selection, and listening intervals in response to changes in channel circumstances.

Fig. 9 demonstrates how BLE, MQTT, and LoRaWAN's transmission power, communication distance, and data rate are all connected. When the distance becomes longer, all protocols require more power and transfer data more slowly. This implies that wireless signals can't go as far as they might and connections can't be as flexible as they could be.

BLE achieves high data rates for short-range communication, but exhibits high power consumption beyond short-ranges, making it unsuitable for long-range or ultra-low-power applications. MQTT functions adequately, though data rate gradually decreases and power consumption fluctuates. This is primarily attributable to its mobility and radio configuration. LoRaWAN maintains low-power consumption over long-ranges, but achieves lower data rates compared to short-range, high-speed alternatives.

The findings reveal that there isn't one ideal technique to conserve energy in every situation. Protocols with high data speeds need greater power as distance increases. Protocols with low-power and long-range, on the other hand, sacrifice throughput in order to save energy. This emphasizes the relevance of low-power IoT devices' ability to change how they interact with one another.

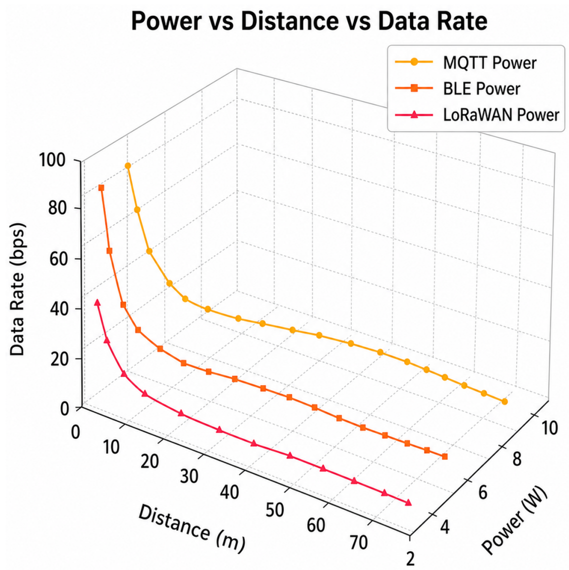

Figure 9: Power analysis.

The findings provide confidence to the proposed adaptive architecture for energy-efficient IoT and swarm robots. By continually modifying protocols and transmission settings in accordance with distance as well as availability of energy, the system can conserve energy while maintaining high quality communication. Static installations, on the other hand, don't make these trade offs, which wastes energy.

Fig. 10 demonstrates how the time it takes for BLE, MQTT, and LoRaWAN to communicate, the distance it can travel, and the speed at which data can be transferred are all connected. As the distance increases, the latency of all protocols increases, while the practical data rate decreases. This implies that it will take longer to transport data over long distances, there will be more retransmissions, and link layer designers will have to be more cautious.

BLE has the least latency over short distances since it transfers data rapidly and doesn't leave much time between connections. The latency does become greater as the distance grows, thus it can only be used for talking to those who are close by. When you add distance, MQTT's latency goes up a tiny bit since the transport layer and the acknowledgment mechanisms have to work harder. Due to this, its performance profile is even, however it varies according to distance.

Compared to BLE and MQTT, LoRaWAN is slower, especially over long distances. This is mostly due to the limited frequency of usage, long airtime, and sluggish data rates of LPWAN technology. LoRaWAN may connect distant devices, but its high latency renders it inappropriate for applications that require fast replies.

The findings indicate that the trade off between latency, speed, as well as range is significant. High throughput protocols usually accelerate data transmission over short distances. In contrast, operations that operate across long distances are relatively slow. These findings underscore the need of communication systems that can change their protocols and broadcast parameters based on the distance to travel as well as the amount of latency an application may tolerate.

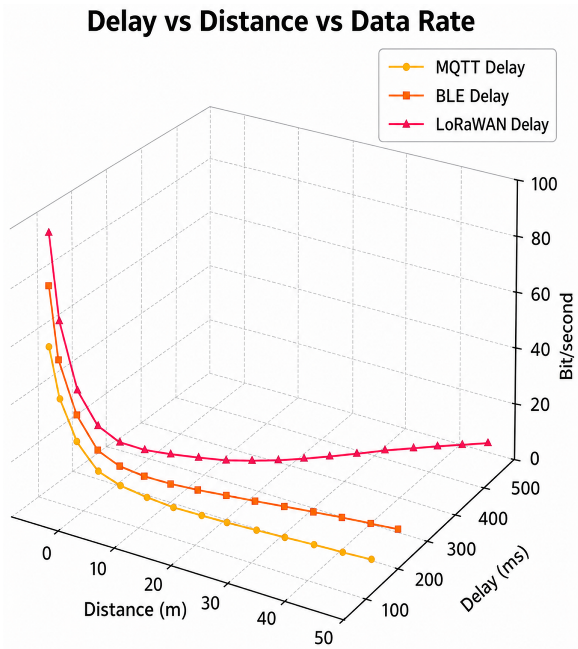

Figure 10: Latency analysis.

Systems for IoT and swarm robots that consume less energy need to be able to switch their decision. Static setups can't perform what delay aware protocol switching can achieve. It helps the system fulfill deadlines without consuming additional power or slowing down.

Every study is carried out in a well regulated as well as standardized benchmarking environment to offer an impartial and repeatable assessment of communication methods.

All of the protocols under study (MQTT, BLE, LoRaWAN, as well as CoAP) share the following parameters:

- Payload Size: All protocols transmit identical payloads of fixed size S_{payload} (in bits) per packet.
- Application Workload: The same application-layer workload is applied, characterized by identical packet generation rates and traffic patterns.
- Duty Cycle: A consistent duty cycle configuration is maintained across protocols, defined as:

$$D = \frac{T_{\text{active}}}{T_{\text{active}} + T_{\text{sleep}}}, \tag{66}$$

  where T_{active} and T_{sleep} are fixed for all protocols during benchmarking.
- Observation Window: All measurements are collected over an identical time horizon T_{obs} to ensure temporal consistency.

Energy efficiency is computed using a unified definition across all protocols:

$$\eta = \frac{N_{\text{delivered}} \cdot S_{\text{payload}}}{E_{\text{comm}}}, \tag{67}$$

where:

- $N_{\text{delivered}}$ is the number of successfully received packets,
- S_{payload} is the payload size in bits,

- E_{comm} is the total communication energy consumed (in Joules) over the observation window T_{obs}.

  Thus, η represents the number of useful payload bits delivered per unit energy (bits/Joule).

  The communication energy is measured as:

$$E_{\text{comm}} = \int_0^{T_{\text{obs}}} P_{\text{comm}}(t)\, dt, \tag{68}$$

where $P_{\text{comm}}(t)$ is the instantaneous power consumption attributable solely to communication activities (transmission, reception, and protocol overhead). Non communication components such as idle CPU and background processes are excluded to ensure protocol level fairness.

To address differences in protocol stack implementations:

- Application layer overhead (e.g., MQTT over TCP/IP) is included only when evaluating end-to-end system performance.
- For protocol level comparison, PHY/MAC layer energy consumption is isolated where applicable.
- All protocols are evaluated under identical network conditions, including channel characteristics and node placement.

This standardized assessment protocol ensures that differences in measured results are only attributable to protocol features as well as adaptive control methods, rather than inconsistencies in workload, payload, or measurement technique. The implementation of a common bits/Joule measure allows for direct and relevant comparisons of energy efficiency spanning different communication systems.

Fig. 11 shows a proposed adaptive methodology that governs protocol selection, compute offloading, as well as execution to preserve energy. The experimental results indisputably support the value of this rigorous decision making technique.

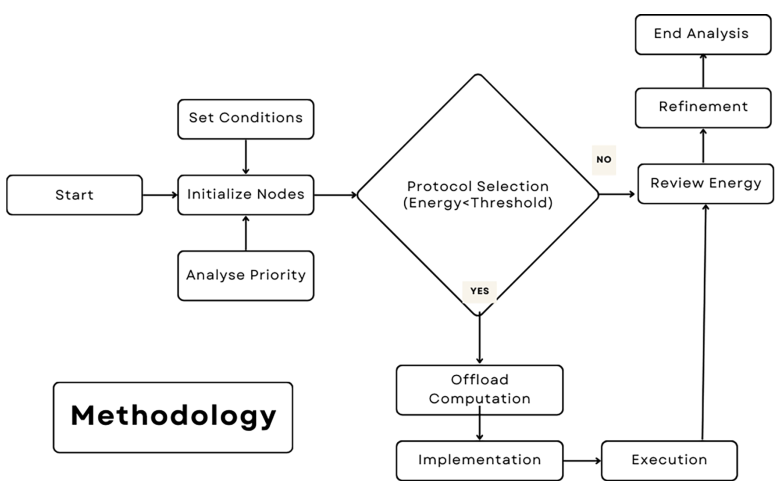

Figure 11: Proposed methodology.

The startup and condition setup procedures define the essential device conditions, resource constraints, and priority levels. This allows many protocols to function in a consistent manner. Before communicating, the framework does a prioritization analysis to verify that only relevant energy transmissions are planned. This is supported by the reported decreases in transmission overhead costs and average energy usage across protocols.

To obtain the observed efficiency gains, the fundamental protocol selection technique must be restricted by a power limitation. When there is enough energy, the system can function autonomously. The computation is moved to a different location or the communication settings are changed if there is insufficient energy. The experimental graphs show substantial energy savings and slower energy scaling due to this selective propensity, particularly if there are obstructs and long distances.

Together, implementation, resource evaluation, and refinement make it easier to handle modifications to the program's operation. Since the system is still being assessed, it cannot stay in exceptionally energetic states for very long. As a consequence, the results show that battery life is becoming longer and that communication energy consumption is continuously falling. The idea that efficient refining lowers link variability and channel challenges is supported by the fact that efficiency doesn't significantly deteriorate across long distances.

The results show that rather of operating as a static optimization, the suggested approach functions as a control framework that adapts over time and accounts for energy use. Making modifications after the execution, selecting the appropriate protocol, and determining whether to offload are all quite comparable. As a result, the nodes live longer, communicate more quickly, and use less power overall. This demonstrates that the suggested architecture is appropriate for long term IoT and swarm robotic systems that use little energy and operate under shifting network and ambient circumstances.

Figs. 11 and 12 demonstrate how the suggested adaptive communication architecture would effect the entire system as its entirety, including average power consumption, distribution of energy across component activities, and battery life. The improvement dramatically decreases communication energy usage across all tested operations and is directly related to quantifiable increases in lifetime.

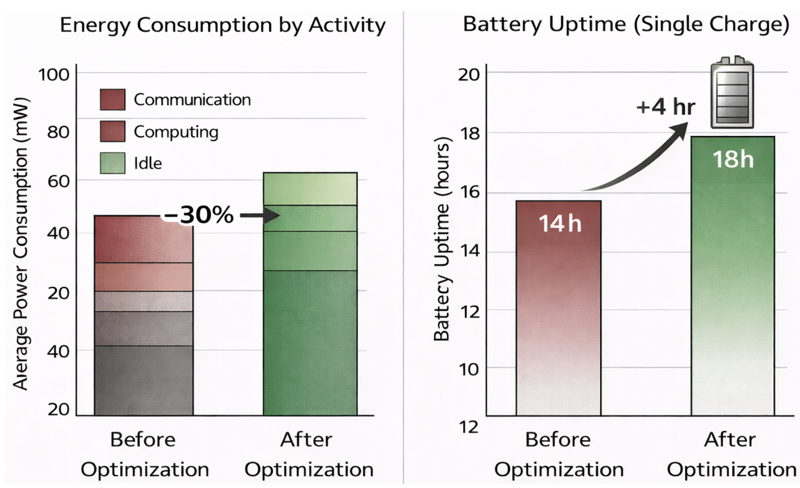

Figure 12: Average power consumption by activity.

The first subplot indicates a significant decline in average power utilization after optimization, saving roughly 25% for BLE, 33% for LoRaWAN, 30% for MQTT, and 32% for CoAP. These cuts show that protocol aware adaptation through dynamic duty cycling, changing the data rate, and selective transmission scheduling can cut down on unnecessary radio activity while keeping the connection.

The second subplot shows that the battery life has improved. Each methodology has a different basic efficiency, but they all make a single charge last four to six hours longer. The finest one makes a node last 16 to 18 h longer. Optimization helps LoRaWAN in many ways. This illustrates how easy it is to modify the time between broadcasts and set the time frames for reception. Protocols that are really basic, like BLE, have gone a long way. This is crucial because it demonstrates that adaptive control works even when the power is extremely low.

The first subplot in Fig. 13 indicates how much energy each activity consumes on average. Communication uses the greatest energy before optimization. The new system will use around 30% less energy to talk to other nodes, but it will still require roughly the same amount of energy while it's not being utilized or when it's performing arithmetic. This illustrates that the optimization only looks at the major energy constraint.

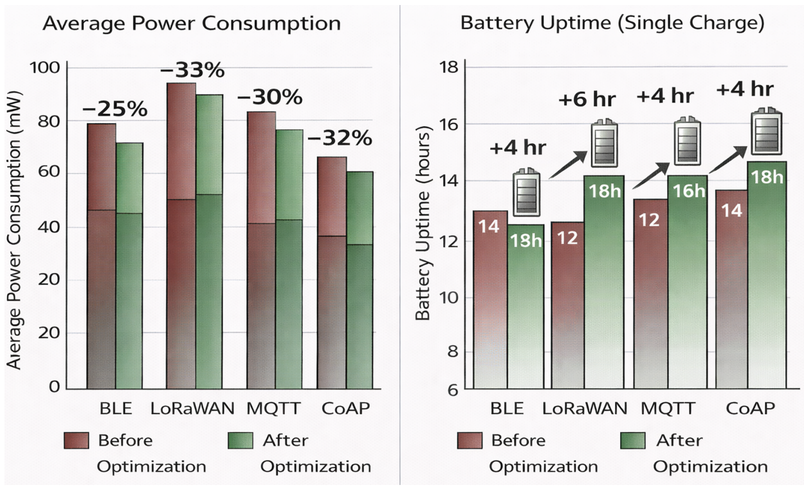

Figure 13: Power consumption.

The second subplot in Fig. 13 shows that the battery lasts longer as time goes on. The battery lasts roughly four hours longer once it has been optimized. This result illustrates that the energy savings from each activity may endure for a long period. It also backs up the premise that IoT systems that know how much energy they consume or conserve energy are conceivable.

The results demonstrate that the proposed adaptive framework reduces energy consumption, operates protocol-agnostically, helps batteries last longer, and consistently mitigates communication-related power waste. The approach is viable for energy-constrained IoT and swarm robotic systems as the optimizations apply uniformly across protocols.

Fig. 14 demonstrates how adaptive processing, duty-cycle management, and protocol aware communication all work together to help the system consume less power and survive longer on battery power.

The arrangement of the CPU shows that the number of cores and the operating frequency have a big effect on how much energy is utilized. Using a single core with a reasonable clock frequency of 1.47 GHz lowers power utilization between 6.4 to 4.15 W. However, with flexible scaling in a multi core design, it decreases even more, to 3.24 W. This illustrates that when dealing with IoT devices that must conserve energy, a smart CPU arrangement is more critical than a large processing capability.

The most efficient way to conserve energy is to change the duty cycle. Decreasing this duty cycle from constant to 10%–20% can lower average power consumption by up to 74%, making it acceptable for most IoT applications. This shows that activity-based scheduling might assist accomplish functional objectives while reducing costs for outage and communication.

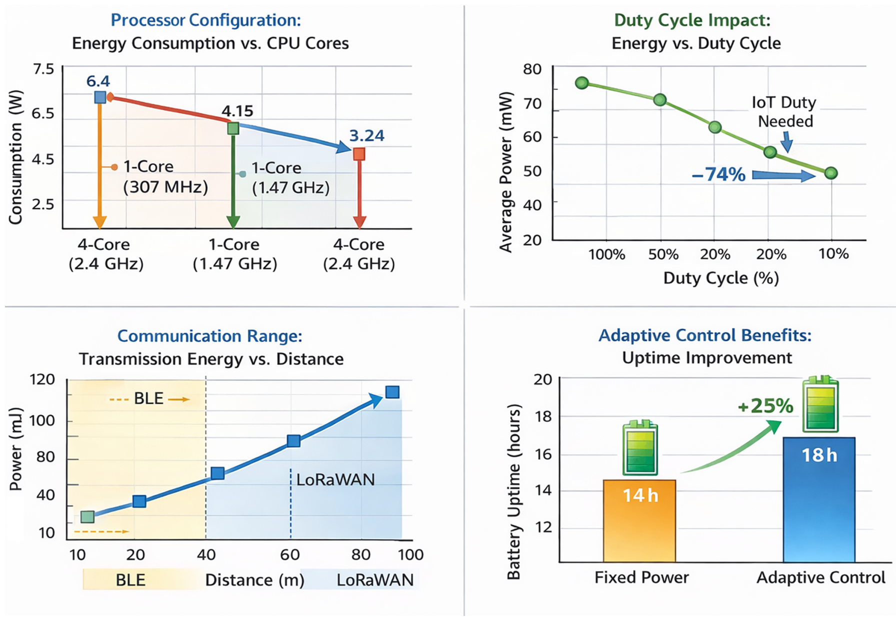

Figure 14: Summary of results.

The research of transmission energy shows that the length of a discussion and the quantity of power used are related. LoRaWAN can carry data over greater distances than BLE, although BLE consumes less power over shorter distances (less than 30–40 m). But sending data outside of this range uses more energy. During this transition period, the adaptive framework chooses the protocol that transfers the least amount of energy per bit, taking into account distance and channel constraints.

The batteries endure a long time, which shows how effectively adaptive control approaches operate. With adaptive control, the duration it can run rises increased from 14 to 18 h, which is a 25% increase. This illustrates that changing the processor, duty cycle, and communication levels all at once is highly advantageous for the system.

The findings reveal that the proposed adaptive framework works effectively in real-time to control duty cycles, communication strategies, and computational effort. This lets IoT and swarm robotic installations operate all the time and consume less power.

The reported 25% improvement in battery lifetime is explicitly bounded with respect to the chosen baseline.

The improvement is computed relative to a fixed power transmission baseline, in which the node operates with static transmit power, fixed duty cycling, and no adaptive optimization. Let T_{fixed} and T_{adaptive} denote the battery lifetime under the fixed baseline and the proposed adaptive framework, respectively. The relative improvement is defined as:

$$\Delta T(\%) = \frac{T_{\text{adaptive}} - T_{\text{fixed}}}{T_{\text{fixed}}} \times 100. \tag{69}$$

In our experimental setup, we observe:

$$\Delta T \approx 25\%, \tag{70}$$

stating that a non adaptive, rigid architecture has a shorter working lifetime than a dynamic control framework.

It is important to remember that:

- This improvement is not a universal gain across all possible baselines, but specifically reflects comparison against a fixed power strategy.
- When compared to more advanced baselines (e.g., rule based or partially adaptive schemes), the relative improvement is reduced but remains positive.
- The reported gain depends on workload characteristics, channel conditions, and system parameters such as duty cycle and traffic patterns.

The suggested adaptive communication system improves battery lifespan by up to 25% compared to a static power baseline under reviewed experimental settings.

To give a more plausible perspective on battery life cycle, we explicitly describe the battery model characteristics as well as substitute the implicit linear lifespan assumption with a nonlinear discharge formulation.

The studies presume a chargeable Li-ion battery with the following characteristics:

- Nominal Capacity: $C_{\text{bat}} = 5000$ mAh
- Nominal Voltage: $V_{\text{nom}} = 3.7$ V
- Maximum Voltage: $V_{\text{max}} = 4.2$ V
- Cutoff Voltage: $V_{\text{cut}} = 3.0$ V

The total nominal energy is approximated as:

$$E_{\text{bat}} = C_{\text{bat}} \cdot V_{\text{nom}} = 18.5 \text{ Wh}. \tag{71}$$

Instead of assuming linear discharge, we adopt a non linear voltage state of charge (SoC) relationship typical of Li-ion batteries. The discharge behavior can be approximated as:

$$V_{\text{bat}}(\text{SoC}) = V_{\text{nom}} - k_1(1 - \text{SoC}) - k_2(1 - \text{SoC})^2, \tag{72}$$

where SoC $\in [0, 1]$ and k_1, k_2 are empirical constants capturing the voltage drop profile.

The instantaneous available energy is then:

$$E(t) = \int_{t_0}^{t} V_{\text{bat}}(\text{SoC}(t)) \cdot I(t)\, dt, \tag{73}$$

where $I(t)$ is the current draw.

Battery lifetime is determined by the time at which:

$$V_{\text{bat}}(\text{SoC}) \le V_{\text{cut}}. \tag{74}$$

The optimal usable energy varies with discharge rate as well as load circumstances, in opposition to the linear models. Because of the effects of internal resistance, high current demand induces voltage to decrease more quickly, reducing useable capacity.

This nonlinear model has been used to modify the stated lifetime gains:

- Energy savings from adaptive control delay the onset of rapid voltage decline near low SoC.
- Reduced peak power consumption lowers instantaneous current draw, mitigating voltage sag.
- Lifetime extension is therefore not strictly proportional to average power reduction.

Battery lifetime improvements are evaluated using a non linear discharge model, where adaptive energy savings extend the usable operating region by delaying the approach to the cutoff voltage, rather than assuming a linear scaling between power reduction and lifetime.

This formulation provides a more realistic explanation of battery management in IoT devices. It underlines that adaptive communication systems reduce mean energy consumption while increasing effective energy use in non linear discharge circumstances.

5.2 Discussion on Embedded Deployment Constraints and MCU-Level Resource Feasibility

Although the major experimental assessment was carried out on the NVIDIA Jetson TX2 architecture for regulated power profiling as well as heterogeneous protocol testing, we specifically address deployment feasibility for resource constrained embedded devices.

The suggested optimization framework was created with lightweight execution restrictions in mind, using:

- reduced action-space dimensionality,
- event-triggered policy updates,
- lightweight constrained DQL inference,
- parameter sharing,
- and duty-cycle-aware scheduling.

To improve practical relevance, we present a comparative resource analysis between the proposed framework and representative embedded IoT hardware platforms, as summarized in Table 4.

Table 4: Comparison of embedded resource constraints and proposed framework requirements.

Platform	RAM	Flash	CPU Freq.	Deployment Feasibility
ARM Cortex-M4	256 KB	1 MB	80–120 MHz	Feasible with quantized model
ESP32	520 KB	4 MB	160–240 MHz	Fully feasible
STM32L4	128–320 KB	512 KB–1 MB	80 MHz	Feasible with lightweight policy
Raspberry Pi Pico W	264 KB	2 MB	133 MHz	Feasible for inference-only deployment
Jetson TX2 (Experimental Platform)	8 GB	32 GB	2 GHz	Supports full training and profiling
Proposed Lightweight DQL Controller	≈150–300 KB	<1 MB	>80 MHz recommended	Suitable for TinyML deployment

We highlight that the restricted DQL controller is not designed to provide continuous onboard training for ultra low-power systems. Instead, the framework provides the following:

- offline training,
- compressed policy deployment,
- federated parameter updates,
- and lightweight inference execution.

To further decrease embedded overhead, we addressed different optimization mechanisms:

- model pruning,
- parameter quantization,
- fixed-point arithmetic,
- and event-driven inference scheduling.

The computational complexity per inference step is approximately as follows:

$$C_{\text{inf}} = O(|A| + d),$$

where $|A|$ denotes the action-space size and d represents the compressed parameter dimension.

Our analysis emphasizes that communication adaptation and duty-cycle scheduling remain largely platform-independent, allowing the proposed optimization strategy to scale across heterogeneous embedded hardware.

6 Conclusion

We developed a complete approach for reducing the communication energy usage of IoT equipment. This method combines protocol evaluation, a dynamic algorithm, as well as energy optimization. Our main contributions are (i) a comparison of low-power IoT protocols and MAC schemes that avoid collisions, (ii) real world examples of how to use ambient energy and backscatter to save energy, (iii) adaptive power control and scheduling techniques, including a DQL based agent that changes transmission parameters based on energy state, and (iv) an experimental prototype that shows significant energy savings.

Our architecture makes energy use much more efficient without making communication worse, as the results show. Our adaptive method made devices about 25% more reliable and longer lasting than the default (static) settings. To get this benefit, we should always use low-power, short-range communications when we can. When we're not using them, we should use ambient energy, and we should change the settings on our hardware (like the CPU and GPU) based on the energy environment.

Our research shows that IoT systems that will last need to have integrated design. By integrating energy sources, power control, and communication protocols, we might reduce the energy consumption of IoT swarms. By adding advanced learning based flexibility (such predicting energy availability), investigating other approaches, and addressing security issues, future research will enhance this framework. In brief, IoT and robotic systems require ways to talk to one other that may evolve over time and don't need any maintenance. Our analysis offers a clear way to get there.

In some respects, integrating the Internet of Things (IoT) to everyday equipment makes it easier to conserve energy, but in others, it makes it more difficult. It's critical to spend energy wisely as more devices connect to one another. Modifying laws and regulations which were not designed having the Internet of Things consideration is a major worry. We need to modify energy conservation laws so that they assist the environment while saving customers money.

Optimizing communication protocols for IoT devices is critical. Many legacy communication protocols prioritized data throughput and bandwidth at the expense of increased power consumption. Most Wi-Fi protocols target high-power devices and can connect to other devices across great distances. Novel low-power communication protocols, such as 6LoWPAN, are needed to achieve high data rates with minimal power consumption. These protocols are essential for ensuring effective operation of energy-efficient IoT systems.

In the future, all IoT devices and networks will need to adopt technology that saves energy. Organizations can achieve substantial energy savings with emerging technologies such as LTE-M and Narrowband IoT. This will extend battery lifetime and reduce overall energy consumption. This is particularly significant for smart city initiatives, as reduced energy consumption substantially benefits environmental sustainability. Establishing these standards is essential for proper operation of energy-efficient IoT systems.

Edge computing is increasingly critical for IoT systems. Edge computing reduces network traffic by processing data at the source rather than forwarding it to centralized servers. This enables real-time data access. This enhancement benefits latency-sensitive applications such as autonomous vehicles (AVs). Organizations are expected to achieve greater efficiency and reduced energy consumption through integrated deployment of IoT devices, edge computing, and 5G technology.

In conclusion, our study demonstrates that energy-efficient communication among autonomous IoT devices is feasible for practical applications. We have set up a way to make long lasting, maintenance free IoT systems by carefully designing the parts of the system and using flexible strategies to save energy and cut down on usage. The results of this study contribute to wider efforts to make the Internet of Things more efficient, reliable, and environmentally friendly.

Acknowledgement: We are grateful to our collaborators for their contributions to this publication. Their discussions, ideas, and constructive criticism have significantly enhanced the outcomes of this research.

Funding Statement: The authors received no specific funding for this study.

Author Contributions: The authors confirm contribution to the paper as follows: conceptualization, Amir Ijaz and Hashem Haghbayan; methodology, Amir Ijaz; software, Amir Ijaz; validation, Amir Ijaz and Abdul Malik; formal analysis, Amir Ijaz; investigation, Amir Ijaz; resources, Juha Plosila; data curation, Abdul Malik; writing—original draft preparation, Amir Ijaz; writing—review and editing, Ethiopia Nigussie; visualization, Ethiopia Nigussie; supervision, Hashem Haghbayan; project administration, Juha Plosila. Please turn to the CRediT role descriptors–CRediT for the term explanation. All authors reviewed and approved the final version of the manuscript.

Availability of Data and Materials: Not applicable.

Ethics Approval: Not applicable.

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

References

1. Ziouzios D, Baras N, Dasygenis M, Karayannis V, Tsanaktsidis C. Energy efficiency optimization in swarm robotics for smart photovoltaic monitoring. Energies. 2025;18(7):1587. doi:10.3390/en18071587.
2. Ijaz A, Haghbayan H, Malik A, Nigussie E, Plosila J. Towards optimizing communication cost in energy efficient IoT devices for swarm robotics. Procedia Comput Sci. 2025;265(1):49–56. doi:10.1016/j.procs.2025.07.155.
3. BenSaleh M, Sulaiman MS, Saida R, Hadj Kacem Y, Abid M. Wireless sensor network design methodologies: a survey. J Sens. 2020;2020(1):9592836. doi:10.1155/2020/9592836.
4. Ketshabetswe LK, Zungeru AM, Mangwala M, Chuma JM, Sigweni B. Communication protocols for wireless sensor networks: a survey and comparison. Heliyon. 2019;5(5):e01591. doi:10.1016/j.heliyon.2019.e01591.
5. Modieginyane KM, Letswamotse BB, Malekian R, Abu-Mahfouz AM. Software-defined wireless sensor networks application opportunities for efficient network management: a survey. Comput Netw. 2018;146(2):72–90. doi:10.1016/j.compeleceng.2017.02.026.
6. Kori GS, Kakkasageri MS, Manvi SK. Computational intelligent techniques for resource management schemes in wireless sensor networks. In: Recent trends in computational intelligence enabled research. Amsterdam, The Netherlands: Elsevier; 2021. p. 41–59. doi:10.1016/B978-0-12-822844-9.00023-2.

7. Deng X, Guan P, Hei C, Li F, Liu J, Xiong N. An intelligent resource allocation scheme in energy harvesting cognitive wireless sensor networks. IEEE Trans Netw Sci Eng. 2021;8(2):1900–12. doi:10.1109/tnse.2021.3076485.
8. Anastasi G, Conti M, Di Francesco M, Passarella A. Energy conservation in wireless sensor networks: a survey. Ad Hoc Netw. 2009;7(3):537–68. doi:10.1016/j.adhoc.2008.06.003.
9. Tyagi SKS, Mukherjee A, Pokhrel SR, Hiran KK. An intelligent and optimal resource allocation approach in sensor networks for smart agri-IoT. IEEE Sens J. 2020;21(16):17439–46. doi:10.1109/jsen.2020.3020889.
10. Farhan L, Hameed RS, Ahmed AS, Fadel AH, Gheth W, Alzubaidi L, et al. Energy efficiency for green internet of things networks: a survey. Network. 2021;1(3):279–314. doi:10.3390/network1030017.
11. Moore SJ, Nugent CD, Zhang S, Cleland I. IoT reliability: a review leading to five key research directions. CCF Trans Pervasive Comput Interact. 2020;2(3):147–63. doi:10.1007/s42486-020-00037-z.
12. Heinzelman WR, Chandrakasan A, Balakrishnan H. Energy-efficient communication protocol for wireless microsensor networks. In: Proceedings of the 33rd Annual Hawaii International Conference on System Sciences; 2000 Jan 7; Maui, HI, USA. doi:10.1109/HICSS.2000.926982.
13. Adil M, Khan R, Almaiah MA, Binsawad M, Ali J, Al Saaidah A. An efficient load balancing scheme of energy gauge nodes to maximize the lifespan of constraint-oriented networks. IEEE Access. 2020;8:148510–27. doi:10.1109/ACCESS.2020.3015941.
14. Adil M, Khan R, Ali J, Roh BH, Ta QTH, Almaiah MA. An energy proficient load balancing routing scheme for wireless sensor networks to maximize lifespan in operational environments. IEEE Access. 2020;8:163209–24. doi:10.1109/ACCESS.2020.3020310.
15. Liu X, Wu J. A method for energy balance and data transmission optimal routing in wireless sensor networks. Sensors. 2019;19(13):3017. doi:10.3390/s19133017.
16. Wang K, Yu CM, Wang LC. DORA: a destination-oriented routing algorithm for energy-balanced wireless sensor networks. IEEE Internet Things J. 2020;8(3):2080–91. doi:10.1109/JIOT.2020.3025039.
17. Patel NR, Kumar S, Singh SK. Energy- and collision-aware WSN routing protocol for sustainable and intelligent IoT applications. IEEE Sens J. 2021;21(22):25282–92. doi:10.1109/JSEN.2021.3076192.
18. Kour H, Sharma AK. Performance evaluation of HEED and H-HEED protocol for realistic models in WSN. In: Proceedings of the 2015 International Conference on Computer, Communication and Control (IC4); 2015 Sep 10–12; Indore, India. p. 1–5. doi:10.1109/IC4.2015.7375715.
19. Kumar M, Zhang X, Liu L, Wang Y, Shi W. Energy-efficient machine learning on the edges. In: Proceedings of the 2020 IEEE International Parallel and Distributed Processing Symposium Workshops (IPDPSW); 2020 May 18–22; New Orleans, LA, USA. p. 912–21. doi:10.1109/IPDPSW50202.2020.00153.
20. Kim S, Yoo Y. Contention-aware adaptive data rate for throughput optimization in LoRaWAN. Sensors. 2018;18(6):1716. doi:10.3390/s18061716.
21. Reynders B, Wang Q, Tuset-Peiro P, Vilajosana X, Pollin S. Improving reliability and scalability of LoRaWANs through lightweight scheduling. IEEE Internet Things J. 2018;5(3):1830–42. doi:10.1109/jiot.2018.2815150.
22. Benkhelifa F, Qin Z, McCann JA. User fairness in energy harvesting-based LoRa networks with imperfect spreading factor orthogonality. IEEE Trans Commun. 2021;69(7):4319–34. doi:10.1109/tcomm.2021.3068304.
23. Farhan L, Kaiwartya O, Alzubaidi L, Gheth W, Dimla E, Kharel R. Toward interference-aware IoT framework: energy- and geolocation-based modeling. IEEE Access. 2019;7:56617–30. doi:10.1109/ACCESS.2019.2913899.
24. Naik N. Choice of effective messaging protocols for IoT systems: MQTT, CoAP, AMQP and HTTP. In: Proceedings of the 2017 IEEE International Systems Engineering Symposium (ISSE); 2017 Oct 11–13; Vienna, Austria. p. 1–7. doi:10.1109/SysEng.2017.8088251.
25. Al-Fuqaha A, Guizani M, Mohammadi M, Aledhari M, Ayyash M. Internet of things: a survey on enabling technologies, protocols, and applications. IEEE Commun Surv Tutor. 2015;17(4):2347–76. doi:10.1109/COMST.2015.2444095.
26. Mekki K, Bajic E, Chaxel F, Meyer F. A comparative study of LPWAN technologies for large-scale IoT deployment. ICT Express. 2019;5(1):1–7. doi:10.1016/j.icte.2017.12.005.

27. Ratasuk R, Vejlgaard B, Mangalvedhe N, Ghosh A. NB-IoT system for M2M communication. In: Proceedings of the 2016 IEEE Wireless Communications and Networking Conference; 2016 Apr 3–6; Doha, Qatar. p. 1–5. doi:10.1109/WCNC.2016.7564708.
28. Ijaz A, Haghbayan H, Nigussie E, Malik A, Plosila J. Optimizing communication and computational cost for IoT devices in energy-efficient swarm robotics. Green Tech and Sust. 2026;4(3):100409. doi:10.1016/j.grets.2026.100409.