SvedbergOpen
DISSEMINATION OF KNOWLEDGE

International Journal of Artificial Intelligence and Machine Learning

Publisher's Home Page: https://www.svedbergopen.com/

Research Paper Open Access

Event Triggered Rx Window Scheduling for RPL-Based IoT Networks

Nitin P. Chawande[1], Manoj M. Deshpande[2]

[1]Department of Computer Engineering, A. C. Patil College of Engineering, Kharghar, Navi Mumbai, Maharashtra, India.
E-mail: npchawande@acpce.ac.in
[2]Department of Computer Engineering, A. C. Patil College of Engineering, Kharghar, Navi Mumbai, Maharashtra, India.
E-mail: mmdeshpande@acpce.ac.in

Abstract

Energy efficiency remains a fundamental challenge in battery-operated, resource-constrained sensor nodes, where the combination of IEEE 802.15.4 with 6LoWPAN and the Routing Protocol for Low-Power and Lossy Networks (RPL) has become the standard networking architecture. Prolonged idle listening is a major source of energy loss in constrained sensor nodes. Existing energy-efficient communication strategies primarily optimize routing or duty-cycling independently, without coordinating receiver activity with application-layer event generation. The proposed Event-Triggered Receive Window (ET-RxW) targets receiver-side energy consumption. Receiver activation is aligned with anticipated publication events rather than periodic wake-up schedules. As a result, unnecessary receiver activity during idle periods is eliminated, substantially reducing radio duty-cycling overhead while preserving reliable connectivity and packet delivery. ET-RxW was integrated into the Contiki-NG networking stack and evaluated in the Cooja simulator using Zolertia Z1 motes. Experimental results show a 96% reduction in Radio-Rx energy consumption compared with baseline RPL, while maintaining a 100% Packet Delivery Ratio (PDR) and a convergence time of 14 seconds. A formal analytical energy model was also developed to estimate receiver energy consumption from duty-cycle parameters. Its close agreement with the simulation results confirms the validity of the proposed approach, supporting long-term energy assessment and practical deployment of battery-powered IoT networks.

Keyword: Internet of Things, Radio Duty Cycling, RPL Routing Protocol, Publish-Subscribe Architecture, Idle Listening Elimination, Low-Power and Lossy Networks, Contiki-NG.

1. Introduction

Over the past decade, the Internet of Things (IoT) has seen widespread adoption in critical infrastructure. Its applications now extend to industrial monitoring, smart building management, and environmental sensing. The increasing deployment of these systems is driven by the need for sensing solutions that are scalable, reliable, and economically viable [1,2,3,4,5]. Most of these applications operate over Low-Power and Lossy Networks (LLNs), where sensor nodes must function with limited energy, memory, and processing capability. To support reliable multi-hop communication under these constraints, the Routing Protocol for Low-Power and Lossy Networks (RPL) has become the standard routing protocol for LLNs [6,7,8,9,10,11].

Despite the efficiency of RPL, the limited energy available in battery-powered sensor nodes continues to restrict the operational lifetime of LLN deployments [12,13,14,15]. To reduce communication overhead, many IoT applications employ the Publish–Subscribe (Pub–Sub) communication model, where data is transmitted only when an event of interest occurs instead of being sent periodically [16,17,18,19]. This event-driven communication model is naturally suited to resource-constrained networks because it avoids unnecessary transmissions while maintaining scalability for large IoT deployments [20,21,22].

Although event-driven communication reduces transmission activity, a significant amount of energy is still consumed at the Medium Access Control (MAC) layer. The radio receiver must periodically wake up to check for incoming packets, even when no communication takes place. Under sparse traffic conditions, idle listening, where the receiver remains active without receiving any data, has consistently been identified as one of the largest contributors to node energy consumption [12,23,24,25]. Experimental studies on event-driven networks further show that receiver operation alone accounts for a substantial portion of the overall energy budget, often exceeding the energy spent on packet transmission [26,27].

Several studies have attempted to improve the energy efficiency of RPL by optimizing routing metrics, objective functions, parent selection, and traffic distribution strategies [28,29,30,31,32,33]. While these approaches improve routing performance, they do not address the receiver-side energy consumed due to periodic duty cycling at leaf nodes.

The impact of this overhead becomes more pronounced in Pub–Sub applications, where long intervals may exist between successive events. To address this limitation, receiver activation is confined to predefined listening windows synchronized with anticipated publication events through the proposed Event-Triggered Receive Window (ET-RxW). During the remaining time, the radio stays in low-power sleep mode, thereby eliminating unnecessary idle listening. The proposed mechanism is implemented in the Contiki-NG operating system and evaluated using Energest-based power measurements. The evaluation showed a substantial reduction in Radio-Rx energy consumption while maintaining Packet Delivery Ratio (PDR) and network convergence performance [10,30].

The key contributions of this work are summarized below.

- Development of the ET-RxW approach: A deterministic scheduling technique is introduced for RPL leaf nodes, where radio activation is synchronized with application-layer events instead of periodic channel polling. This significantly reduces idle listening without modifying the existing RPL routing framework [8,9,11,32].
- Development of a cross-layer analytical energy model: A mathematical framework is formulated to establish the relationship between receiver duty cycling, event publication timing, and cumulative energy consumption. The proposed model enables systematic estimation of node lifetime and provides a theoretical basis for improving energy efficiency in sparse-traffic IoT networks [11,15,33].
- Implementation and simulation-based validation: The ET-RxW mechanism is integrated into the Contiki-NG networking stack and evaluated using Zolertia Z1 motes in the Cooja simulator. Evaluating the proposed work on the Z1 platform—a system constrained by a 16-bit MSP430 processor and tight memory limits—provides a realistic benchmark for actual, low-power IoT deployment conditions. The results show that ET-RxW introduces negligible computational overhead while remaining suitable for highly constrained sensor nodes [10,34,35,36,37,38].
- Experimental demonstration of reliable and energy-efficient operation: Energest-based measurements show that the proposed mechanism reduces Radio-Rx energy consumption by more than 80% compared with conventional duty cycling while maintaining a Packet Delivery Ratio (PDR) of 100% and rapid network convergence. These results demonstrate the practical applicability of ET-RxW for energy-efficient event-driven monitoring systems

2. Related Work

In 6LoWPAN IEEE 802.15.4 RPL networks, energy conservation remains a critical bottleneck, with idle listening, the state where a radio receiver remains active despite no incoming traffic accounting for more than half of the total energy consumption in certain sensor nodes. To address this challenge, prior work has generally followed two directions. Some studies intervene directly at the MAC and PHY layers, while others focus on routing-level optimizations.

2.1. Direct Solutions for Idle Listening

Recent studies have explored ways to reduce unnecessary radio activity by controlling the radio state and transmission schedule. PRIL-M [39], developed for multihop Time-Slotted Channel Hopping (TSCH) networks, places sleep commands in data frames through Information Elements (IEs). Based on these commands, a receiver can turn off its radio for a predicted number of upcoming cells. The approach reduces energy consumption by about 75% compared with standard TSCH. However, it is mainly intended for periodic traffic and increases end-to-end latency by nearly twenty times.

MILD [26] takes a different approach for higher-bandwidth IEEE 802.11 (Wi-Fi) networks. It uses PHY-aided down-clocking, where specialized preambles help identify packet arrivals. A device can therefore remain at a lower clock rate when the channel is idle and return to normal operation when traffic is detected. Reported energy savings reach 23.6% with only a small throughput penalty. However, because the detection mechanism relies on subcarriers, the approach is not directly suited to low-power IEEE 802.15.4 sensor nodes.

At the MAC layer, IEE-MAC [27] combines transmitter-initiated Low Power Listening (LPL) with adaptive sleep scheduling. The sleep interval is adjusted according to the remaining energy of the node, which can improve packet delivery and extend network lifetime under changing operating conditions. A limitation, however, is that receiver activity is still not synchronized with unpredictable application-layer events.

2.2. Routing and Topology Optimizations

Another line of research has focused on improving the energy efficiency of RPL itself [31]. LEA-RPL combines Particle Swarm Optimization (PSO) and Long Short-Term Memory (LSTM) techniques for parent selection and Trickle Timer optimization. The reported results show a reduction of up to 66% in convergence time, along with lower energy consumption through the consideration of QoS-related objectives. However, the energy savings in this approach mainly result from improvements at the routing layer. It does not specifically address the energy consumed by the radio receiver while a node remains in the idle-listening state.

Node mobility has also been considered in the design of energy-aware RPL protocols. In [30], BE-RPL uses RSSI-based monitoring to detect mobility and distribute the routing load among stationary parent nodes. The protocol reacts to changes in link conditions by selecting alternative parents before a link is broken, thereby reducing unnecessary control overhead and maintaining a higher packet delivery ratio (PDR) when nodes move. Although this approach improves the robustness of the routing topology, its primary concern is mobility-related route maintenance rather than the receiver energy spent during periods in which no packet is available for reception.

Table 1 Comparison of Existing Approaches and Limitations

Reference	Key Contribution	Limitation Relative to ET-RxW
Scanzio et al. (2024) [39]	**PRIL-M:** Proactive reduction of idle listening in multihop TSCH networks using sleep commands.	Specifically targets periodic traffic in TSCH networks; increases end-to-end latency by approximately 20 ×.
Park et al. (2025) [26]	**MILD:** Reduces idle listening energy via a PHY-aided down-clocking state.	Designed for IEEE 802.11 (Wi-Fi) and mobile devices; not applicable to IEEE 802.15.4 LLNs.
Bengheni (2025)[27]	**IEE-MAC:** Uses transmitter-initiated Low Power Listening (LPL) and adaptive sleep scheduling.	Relies on general duty cycles and lacks synchronization with specific application Pub–Sub events.
Mokrani et al. (2025) [31]	**PSO-based LEA-RPL:** Optimizes parent selection and Trickle Timers using Particle Swarm Optimization.	Focuses on reducing routing overhead and convergence time rather than node-level receiver scheduling.
Hussain et al. (2023)[30]	**BE-RPL:** Monitors mobility via RSSI and balances load across parent nodes.	Primarily addresses handover stability and path loss; does not address idle listening during event gaps.
Khan et al. (2022) [15]	**Clustering using FSO:** Bio-inspired Fish Swarm Optimization for optimal Cluster Head selection.	Topology-level management to prevent "energy holes," leaving physical idle listening unaddressed.
Kim et al. (2017) [10]	**Comprehensive RPL Survey:** Topics include upward/downward routing, security, and interoperability.	Extensive survey of protocol challenges but lacks a proposed mechanism for receiver-side duty cycling.
ET-RxW (Proposed)	**Event-Triggered RX-Window Scheduling:** Synchronizes receiver activation with Pub–Sub events.	**Eliminates idle listening** by aligning wake-windows with application events without specialized hardware.

Another approach was presented in [15], where Fish Swarm Optimization (FSO) was applied for cluster-head selection in 6LoWPAN networks. The FSO-based approach aims to distribute the energy consumption among the nodes and thereby reduce the formation of energy holes. This helps avoid excessive energy depletion at particular nodes by selecting cluster heads more effectively.

However, the approach does not address the energy consumed by leaf nodes while their receivers are idle. With sporadic packet arrivals, the receiver can remain active between two communication events even though no packet is being received

2.3. Comprehensive Standard Challenges

The broader challenges associated with the RPL standard have also been extensively discussed [10]. The survey identifies RPL as the de facto routing standard for LLNs, while pointing out several continuing issues, including downward routing, interoperability, and the complexity introduced by its numerous optional features. It also reports that a large part of the RPL research has been evaluated through simulation, with relatively few large-scale deployments in real-world environments.

2.4. Research Gap

The literature reviewed above shows a difference between how energy consumption is addressed at the routing and MAC layers and how the application generates traffic. MAC-layer approaches such as PRIL-M and IEE-MAC mainly use periodic traffic patterns or general duty-cycling mechanisms. Such mechanisms may not work efficiently when communication events occur sporadically, as in Publish–Subscribe (Pub–Sub) applications.

LEA-RPL and BE-RPL mainly deal with routing decisions. LEA-RPL focuses on parent selection and related routing optimization, while BE-RPL considers mobility and changes in the routing topology. These mechanisms can improve network performance and reduce some sources of energy consumption. However, they do not control the receiver according to when an application-layer event is expected. Thus, a leaf node may keep its receiver active even during a period when no data packet is available.

This observation motivates the proposed ET-RxW protocol. ET-RxW uses an event-aligned scheduling mechanism in which receiver windows are coordinated with application-layer Pub–Sub events. The objective is to reduce unnecessary receiver-on periods in the 6LoWPAN stack while avoiding the need for specialized hardware and the high latency associated with keeping the receiver continuously synchronized through proactive scheduling

3. System Model and Problem Formulation

The network under consideration consists of an RPL based Internet of Things (IoT) deployment composed of resource-constrained sensor nodes organized in a multi-hop topology. Within this framework, leaf nodes function as Publishers in a Publish-Subscribe communication model, transmitting sensed data to a central subscriber only when a monitored parameter exceeds a predefined threshold. Consequently, packet generation follows a stochastic, event driven pattern rather than a deterministic periodic schedule.

To balance connectivity with power conservation, each node employs Radio Duty Cycling (RDC) [40]. The radio transceiver operates in four states: transmission (TX), reception (RX), CPU-active, and low-power sleep mode (LPM) [36,37,38,41,42]. In conventional RPL operations, nodes periodically activate their receivers to perform channel assessments, regardless of whether a communication event has been triggered. In sparse, event-driven applications, this periodic activation leads to significant idle listening, which remains the dominant factor in energy depletion.

3.1. Energy Consumption Model

For an observation interval T, during which sensing events occur with an average rate λ, the total energy consumption E_{total} of a leaf node is formulated as in Equation (1), is the sum of the transmission, reception, CPU, and low-power mode energies.

$$E_{\text{total}} = E_{\text{tx}} + E_{\text{rx}} + E_{\text{cpu}} + E_{\text{lpm}} \quad (1)$$

where E_{tx} is the energy consumed during the transmission state; E_{rx} is the energy consumed during the receiver state, encompassing both active reception and idle listening; E_{cpu} is the energy consumed during the CPU-active state; and E_{lpm} is the energy consumed during the low-power sleep mode [36,37,38].

3.2. Problem Statement

In monitoring scenarios where sensing events are infrequent (λ≪1), the energy expenditure associated with repeated receiver activations without actual packet exchange (E_{rx}) becomes the dominant contributor to the total energy budget [33,40] Consequently, minimizing this idle listening behaviour is critical for extending node longevity. The primary objective of this research is to minimize E_{rx} by eliminating periodic idle listening and precisely aligning receiver activation with discrete communication events, while simultaneously ensuring a high Packet Delivery Ratio (PDR) and minimal network convergence delay [26,30].

3.3. Baseline Energy Consumption Analysis

In standard RPL-based deployments, nodes follow asynchronous RDC protocols to maintain network reachability [23,41,42]. In this mechanism, the radio transceiver is periodically activated to perform channel checks (Clear Channel Assessments) to detect incoming transmissions. This approach works well when traffic is periodic because it helps maintain routing connectivity. In event-driven Publish-Subscribe systems, however, communication is sparse and irregular [19,33,40]. As a result, periodic channel checks increase receiver energy consumption.

3.4. Mathematical Formulation of Baseline Rx Energy

In a baseline RDC setup, energy consumption depends mainly on how often the radio checks the channel and how long the radio stays on during each check. Let T represent the total observation interval and f_{check} denote the channel check frequency (typically 8 Hz or 16 Hz in standard configurations). The periodic channel check interval is defined as $T_{\text{check}} = 1/f_{\text{check}}$. The total number of receiver activations during the interval T, denoted as N_{base}, is calculated as:

$$\boldsymbol{N}_{\text{base}} = \boldsymbol{T} \cdot \boldsymbol{f}_{\text{check}} = \frac{\boldsymbol{T}}{\boldsymbol{T}_{\text{check}}} \quad (2)$$

The cumulative receiver energy consumption in baseline operation, $E_{\text{rx,base}}$, is formulated as:

$$\boldsymbol{E}_{\text{rx,base}} = \boldsymbol{N}_{\text{base}} \cdot \boldsymbol{T}_{\text{listen}} \cdot \boldsymbol{P}_{\text{rx}} \quad (3)$$

where T_{listen} is the fixed duration of each radio-on phase during a channel check, and P_{rx} is the power consumption of the radio in receive mode (e.g., 18.8 mA at 3.3 V for the CC2420 transceiver) [36,37,38].

3.5. Analysis of Idle Listening Overhead

In Publish–Subscribe communication, sensing events occur infrequently. As a result, the number of receiver checks is much higher than the number of data packets actually transmitted [32,33]. Consequently, the term $E_{\text{base,rx}}$ is dominated by idle listening, energy spent searching for a carrier signal that does not exist.

Experimental data obtained through Energest-based power tracking confirms that in baseline operations, the Radio-Rx state represents the largest fraction of the total node energy budget [12,34,35]. This disproportionate consumption shortens the operational lifetime of leaf nodes, necessitating a transition from periodic, agnostic wake-ups to the event-aligned strategy proposed in this study [30,31].

4. Proposed Rx-Window Scheduling Model (ET-RxW)

To mitigate the idle-listening overhead identified in Section 3, the proposed Energy-Aware Rx-Window (ET-RxW) strategy transitions from the asynchronous, time-driven wake-up pattern characterized by Equations (2) & (3) to a deterministic, event-aligned activation model. Under this framework, the radio receiver is activated only within a constrained temporal window, Tw, which is strictly coupled with a sensed publication event rather than with a fixed system clock.

4.1. Mathematical Formulation of ET-RxW Energy

Unlike $\boldsymbol{E_{base,rx}}$ in Equation (3) , where receiver activation is a function of elapsed time T through the check count N_{base}, the proposed model defines receiver activity as a function of discrete sensing events Ne. If Ne represents the total number of sensing events detected during the observation interval T, the proposed receiver energy consumption $E_{rx,prop}$ is formulated as:

$$\boldsymbol{E}_{\text{rx,prop}} = \boldsymbol{N_e} \cdot \boldsymbol{T_w} \cdot \boldsymbol{P}_{\text{rx}} \tag{4}$$

where Ne is the total number of discrete sensing events (publications) occurring within T, Tw is the duration of the receiver activation window per event, and $\boldsymbol{P_{rx}}$ retains the same receive-mode power value used in Equation (3).

4.2. Synchronization and Temporal Guard Bands

In the implemented ET-RxW prototype, the receiver window Tw is distributed around the transmission timestamp, comprising a short synchronization probe phase followed by a post-transmission listening phase.

This structure provides a temporal buffer to accommodate processing delays, hardware interrupt latencies, and clock drift between the leaf node and its parent router, while also allowing the node to solicit and receive routing updates through a DODAG Information Solicitation (DIS) and DODAG Information Object (DIO) exchange immediately before transmission.

The proposed configuration sets Tw=750 ms, comprising a 250ms pre-transmission synchronization phase followed by a 500ms post-transmission Rx window, as specified in Algorithm (1) and justified analytically in Section 5. During the prolonged idle intervals between successive Ne events, the radio transceiver remains in low-power mode. Consequently, the proposed ET-RxW model is designed to eliminate the dominant receiver energy overhead identified in Section 3. While a fixed window duration is utilized in the experimental validation to ensure high reliability for downward acknowledgments and routing updates, the formulation in Equation (4) allows dynamic tuning of Tw to balance aggressive energy conservation against network responsiveness.

4.3. Energy Saving Ratio

A performance metric, the Energy Saving Ratio η, is defined as the fractional reduction in receiver energy achieved by ET-RxW relative to the baseline established in Equation (3):

$$\boldsymbol{\eta} = \mathbf{1} - \frac{\boldsymbol{E}_{\mathbf{rx,prop}}}{\boldsymbol{E}_{\mathbf{rx,base}}} \tag{5}$$

Substituting Equation (4) and (3) into Equation (5), and noting that the common factor P_{rx} cancels, yields the closed-form efficiency expression:

$$\boldsymbol{\eta} = \mathbf{1} - \frac{\boldsymbol{N_e} \cdot \boldsymbol{T_w}}{\boldsymbol{N}_{\mathbf{base}} \cdot \boldsymbol{T}_{\mathbf{listen}}} = \mathbf{1} - \frac{\boldsymbol{N_e} \cdot \boldsymbol{T_w}}{\boldsymbol{T} \cdot \boldsymbol{f}_{\mathbf{check}} \cdot \boldsymbol{T}_{\mathbf{listen}}} \tag{6}$$

This expression shows that the magnitude of energy conservation is governed primarily by the sparsity of sensing events (N_e) relative to the baseline channel-check frequency (f_{check}), consistent with the idle-listening dominance argument of Section 3.5.

4.4. Theoretical Performance Bounds

In applications where sensing events are highly infrequent ($N_e/T \to 0$), consistent with $\lambda \ll 1$ in Section 3, the saving ratio η approaches unity, representing the near-total elimination of idle listening overhead. Equation (6), however, treats Tlisten as a single fixed duration corresponding to an idle channel clear-assessment (CCA) check, consistent with the baseline formulation of Equation (3).

This simplification does not capture a second mode of receiver behaviour present in the ContikiMAC implementation: whenever a CCA check detects channel activity from a neighbouring node's transmission or routing control traffic, the receiver extends its listening duration beyond the idle-check time in order to receive the full frame. Section 4.5 refines the baseline model to account for this second mode, establishing the bounds within which the achievable energy saving must lie.

4.5. Channel-Activity-Aware Refinement of the Baseline Model

The ContikiMAC implementation defines two distinct receiver on durations relevant to T_{listen} in Equation (3): an idle-channel check duration, T_{idle}, incurred when a CCA finds the channel clear, and an extended listening duration, T_{detect} corresponding to the LISTEN_TIME_AFTER_PACKET_DETECTED parameter of the ContikiMAC source incurred whenever a CCA detects activity and the receiver must remain on to capture the incoming frame. The effective per-check listening duration is therefore modelled as a probability-weighted mixture:

$$\boldsymbol{T}_{\text{listen,eff}} = \boldsymbol{p} \cdot \boldsymbol{T}_{\text{detect}} + (\mathbf{1} - \boldsymbol{p}) \cdot \boldsymbol{T}_{\text{idle}} \tag{7}$$

where p is the probability that a given baseline channel check detects activity rather than finding the channel idle. Substituting Equation (7) for T_{listen} in Equation (6) gives the refined efficiency expression:

$$\boldsymbol{\eta} = \mathbf{1} - \frac{\boldsymbol{N_e} \cdot \boldsymbol{T_w}}{\boldsymbol{T} \cdot \boldsymbol{f}_{\text{check}} \cdot (\boldsymbol{p} \cdot \boldsymbol{T}_{\text{detect}} + (\mathbf{1} - \boldsymbol{p}) \cdot \boldsymbol{T}_{\text{idle}})} \tag{8}$$

The refined model collapses to Equation (6) at $p = 0$ (no background traffic) and reaches its maximum predicted saving at $p = 1$, where every channel check coincides with detected activity.

Algorithm 1 ET-RxW Leaf Node Operation

```
Require: EVENT INTERVAL, HEARTBEAT INTERVAL, CO2 THRESHOLD, T_sync, T_rx,win
Ensure: UDP publication(s) to the fog broker via the RPL root
1:  seq ← 0; battery ← 100
2:  RegisterUDPConnection
3:  while ¬Routing.IsReachable do                                   ▷ RPL bootstrap
4:  SendDIS; wait 5 s
5:  end while
6:  wait guard delay; Radio.Off
7:  sample timer ← EVENT INTERVAL; heartbeat timer ← HEARTBEAT INTERVAL
8:  loop
9:  wait for next ProcessEvent
10: if sample timer.expired ∨ heartbeat timer.expired then          ▷ Tier-1 wake CPU only, radio off
11: co2 ← ReadSensor
12: trigger ← None
13: if co2 > CO2_THRESHOLD then
14: trigger ← Co2Event
15: else if heartbeat_timer.expired then
16: trigger ← Heartbeat
17: end if
18: if trigger≠ None then                                           ▷ Tier-2 wake radio on
19: Radio.On; SendDIS
20: wait T_sync (250 ms)
21: if Routing.IsReachable ∧ GetRootAddr(root) then
22: seq ← seq + 1
23: topic ← (trigger = Co2Event) ? 1:99
24: payload ← Encode (node id, topic, seq, co2, temp, hum, battery)
25: Send(payload, root)
26: wait T_rx,win (500 ms)                                          ▷ Rx window ACK / downlink
27: heartbeat_timer ← HEARTBEAT_ INTERVAL                           ▷ Reset on any successful TX
28: end if
29: Radio.Off
30: end if
31: sample timer ← EVENT_INTERVAL
32: end if
33: end loop
```

4.6. Multi-Tier Timing Model

The leaf-node implementation distinguishes two independent wake tiers, which together define the scope of applicability of the energy model above. In the first tier, the microcontroller wakes from low-power mode every $EVENT\_INTERVAL$ (20s) to sample the sensor and evaluate the threshold condition introduced in Section 3; the radio subsystem is not invoked at this tier and remains off, so this activity contributes to Ecpu in Equation (1) rather than to E_{rx}. In the second tier, the radio is explicitly activated only once the first-tier evaluation confirms a trigger condition, a threshold violation or heartbeat expiry to perform DIS solicitation, transmission, and the post-transmission Rx window, before returning to the off state. Algorithm (1) formalizes this two-tier procedure as implemented.

4.7. Heartbeat Semantics

Algorithm (1) (lines 13–17) evaluates the heartbeat condition only when the sensing threshold has not been exceeded, and line 27 re-arms heartbeat timer upon any successful transmission, irrespective of whether the triggering condition was a threshold event or heartbeat expiry. This establishes the heartbeat mechanism as a parent-liveness beacon: its role is to guarantee that the node demonstrates reachability to its parent at least once every HEARTBEAT_INTERVAL (300 s), rather than to enforce a periodic sensor report that competes with event-driven publication.

4.8. Simulation Parameters and Calculated Energy Saving Ratio

The analytical validation reuses the CC2420 receive power $P_{\mathrm{rx}} = 62.04$ mW (18.8 mA at 3.3 V) and baseline check frequency $f_{\mathrm{check}} = 8$ Hz introduced in Section 3, together with an observation interval $T = 3600$ s, a radio activation window $T_w = 0.75$ s per Algorithm (1), an idle channel-check duration $T_{\mathrm{idle}} = 2.5$ ms, and an activity-detection listening duration $T_{\mathrm{detect}} = 12.5$ ms derived from the ContikiMAC LISTEN_TIME_AFTER_PACKET_DETECTED constant. The event rate $N_e = 12$ events per hour, corresponding to the 300 s heartbeat interval, represents a sparse-event, heartbeat-only lower bound on transmission frequency,

Table 2 Reconciliation of the analytical model with the measured energy saving ratio.

Scenario	p	$T_{\text{listen,eff}}$	η
Idle-channel only (lower bound)	0%	2.50 ms	87.5% (analytical)
Implied by experimental measurement	≈53%	7.81 ms	96% (measured)
Fully traffic-loaded (upper bound)	100%	12.50 ms	97.5% (analytical)

since threshold-triggered transmissions occur in addition to the heartbeat and are evaluated separately against the experimental results in Section 6. Under these parameters, ET-RxW reduces the number of discrete radio activations by a factor of

$$\frac{f_{\text{check}} \times T}{N_e} = \frac{8 \times 3600}{12} = 2400$$

relative to N_{base} in Equation (2), while the application layer continues uninterrupted monitoring via the Tier-1 sampling cycle. Evaluating Equation (8) at its two limiting cases, $p = 0$ and $p = 1$, establishes the theoretical envelope for η:

4.9. Reconciliation with Experimental Results

Energest-based energy instrumentation in Contiki-NG Section 6. measured a receiver-side energy reduction of 96% for the ET-RxW configuration relative to the baseline of Equation (3), falling within the theoretical envelope of Table 2 and close to its upper bound. Solving Equation (8) for p using the measured $\eta = 96\%$ recovers the implied probability that a baseline channel check encounters activity in this deployment Equation (9).

$$T_{\text{listen,eff}} = \frac{N_e \cdot T_w}{T \cdot f_{\text{check}} \cdot (1-\eta)} = 7.81 \text{ ms} \Longrightarrow p \approx 53\% \qquad (9)$$

An implied activity-detection probability of approximately 53% is physically consistent with the evaluated topology, which comprises eight nodes sharing a single personal area network, with forwarder nodes operating at 99.6–99.9% radio-on time as measured independently in Section 6 . This indicates that Equation (6) correctly establishes a conservative lower bound on the achievable saving, consistent with the idle-listening dominance identified analytically in Section 3, and that the refined model of Equation (8) accounts for the measured 96% reduction through realistic background channel activity, without requiring any modification to the ET-RxW mechanism itself.

5. Design Trade-Off Analysis of Rx-Window Duration

The choice of T_w affects both energy consumption and communication reliability. A shorter listening window keeps the radio on for less time, which can save energy. However, the receiver then has less time to detect an incoming packet, and packet loss can occur because of clock drift, MAC-layer contention, or processing delays. Based on this trade-off, T_w=750 ms is used throughout Section 4 and is also specified in Algorithm (1).

5.1. Reliability and Synchronization Constraints

To sustain a high packet delivery ratio, Tw must accommodate the total temporal uncertainty between the leaf node and its parent router. As specified in Algorithm (1) (lines 19–26), this window comprises a 250 ms pre-transmission synchronization phase, triggered immediately on radio activation and allowing a single DIS solicitation and routing state verification before the packet is dispatched, followed by a 500 ms post-transmission RX window providing a dedicated interval for receiving link-layer acknowledgments and downward control traffic from the broker.

5.2. Analytical Impact of Window Scaling

The total energy consumed during a single communication event is proportional to the sum of the constituent phase durations a shown in Equation (10) and Equation (11)

$$E_{\text{event}} = T_{\text{sync}} \cdot (P_{\text{rx}} + P_{\text{cpu}}) + T_{\text{tx}} \cdot (P_{\text{tx}} + P_{\text{cpu}}) + T_{\text{rx,win}} \cdot (P_{\text{rx}} + P_{\text{cpu}}) \qquad (10)$$

which can be expressed more compactly as:

$$E_{\text{event}} = T_w \cdot P_{\text{cpu}} + T_{\text{sync}} \cdot P_{\text{rx}} + T_{\text{tx}} \cdot P_{\text{tx}} + T_{\text{rx,win}} \cdot P_{\text{rx}} \qquad (11)$$

where $T_w = T_{\text{sync}} + T_{\text{tx}} + T_{\text{rx,win}}$ is the total active window duration, P_{cpu} is the CPU active-mode power, and P_{tx} and P_{rx} denote the transmit and receive power of the radio respectively. On the CC2420 transceiver used in this work, P_{rx} and P_{tx} are of comparable order of magnitude at the configured transmit power level, and P_{cpu} is substantially smaller than both radio power terms, so the synchronization and RX-window phases account for the majority of E_{event}. Experimental observations on the target platform indicate that a total window of 750 ms constitutes an effective guard band, sufficient to absorb cumulative clock drift and software interrupt latency in the network stack, confirming $T_w = 0.75$ s as the configuration adopted throughout Section 4

5.3. Adaptability to Traffic Sparsity

In the considered setup, sensing takes place every 20 s, whereas the heartbeat communication interval is 300 s. Therefore, the additional radio-on time associated with a 750 ms activation window has a relatively small effect on the overall energy consumption. The longer window is used to provide sufficient time for reliable reception of event-driven publications while still retaining the energy advantage of the Publish–Subscribe approach. The

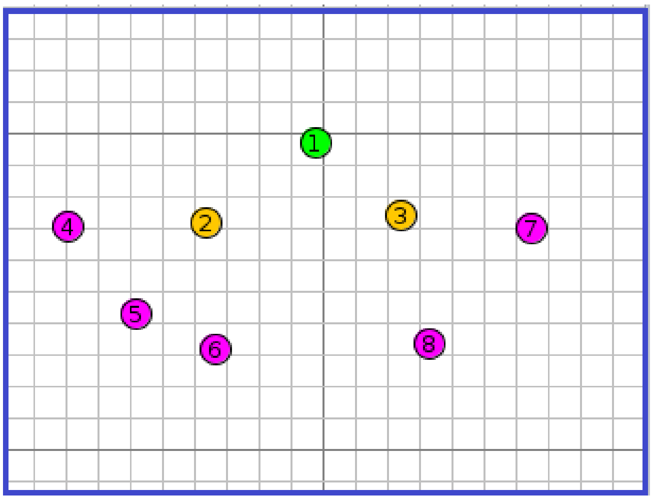

Figure 1 Simulated 5-2-1 IoT Network Topology

calculation in Section 4.8 gives a conservative lower-bound energy saving of 87.5%. The experimental results in Section 6 show an even larger reduction of 96%.

5.4. Reliability and Synchronization Constraints

To sustain a high packet delivery ratio, T_w must accommodate the total temporal uncertainty between the leaf node and its parent router. As specified in Algorithm (1) (lines 19-26), this window comprises a 250 ms pre-transmission synchronization phase, triggered immediately on radio activation and allowing a single DIS solicitation and routing-state verification before the packet is dispatched, followed by a 500 ms post-transmission Rx window providing a dedicated interval for receiving link-layer acknowledgments and downward control traffic from the broker.

6. Performance Evaluation

To validate the effectiveness of the proposed ET-RxW scheduling mechanism, two experiments were conducted using the Contiki-NG operating system within the Cooja network simulator. The primary objective was to quantify the reduction in receiver-side energy consumption while maintaining identical network traffic conditions.

6.1. Network Topology and Architecture

The experimental setup models a multi-hop, three-tier IoT network implemented in the Cooja simulation environment. Figure 1 shows the simulated eight-node topology arranged in a 5–2–1 hierarchy.

- **Leaf Publisher Nodes (5):**

Five leaf publisher nodes (Nodes 4–8) generate event-driven sensor data and operate using the proposed ET-RxW approach, keeping their radio transceivers in a low-power state until a predefined sensing threshold is exceeded or heartbeat expiry, as formalized in Algorithm (1)

- **Forwarder Nodes (2):**

Nodes 2 and 3 participate fully in the RPL DODAG as non-root routers, acting as intermediate relays to maintain multi-hop connectivity between the leaf tier and the fog broker. These nodes maintain a continuous receiver duty cycle to ensure packet forwarding availability at all times.

- **Fog Node, RPL Root and Broker (1):**

Node 1 serves as the central sink, RPL DODAG root, and publish–subscribe broker, aggregating all sensor publications from the leaf tier via the forwarder tier.

This 5–2–1 hierarchical topology, whose operational and simulation parameters are summarized in Table 3 and Table 4 ensures that energy optimization is primarily applied at the network edge (leaf nodes), while routing stability is preserved within the network core (forwarder nodes). The cluster-based structure limits the number of leaf nodes per forwarder to a maximum of six, consistent with the queue-buffer and neighbor-table constraints of the target platform.

6.2. Comparative Study Design

To rigorously evaluate the impact of the proposed scheduling mechanism, two experimental scenarios were defined. All parameters—including topology, traffic patterns, and RPL configuration—were kept identical to ensure a fair comparison. Scenario

6.2.1. Scenario A: Baseline (Standard RDC)

In the baseline scenario, nodes follow the same sensing (20 seconds) and heartbeat (300 seconds) logic. However, radio operation is governed by the standard RDC mechanism operating at approximately 8 Hz . As a result, the receiver performs periodic channel checks irrespective of actual communication events, leading to continuous idle listening

Table 3 Simulation Parameters

Parameter	Value
Simulator	Cooja
Operating System	Contiki-NG
Hardware Platform	Zolertia Z1 (MSP430/CC2420)
Simulation Duration	3600 seconds (1 hour)
Leaf Nodes	5
Forwarder Nodes	2
Fog/Root Node	1
Energy Monitoring	Energest

6.2.2. Scenario B: Optimized (ET-RxW Enabled)

In the optimized scenario, identical sensing and traffic conditions are maintained. However, the radio remains in a low-power state by default and is activated only during scheduled communication events. Specifically, the receiver is enabled for a fixed 750 ms RX window (Tw), eliminating unnecessary channel checks and restricting radio usage to actual communication instances.

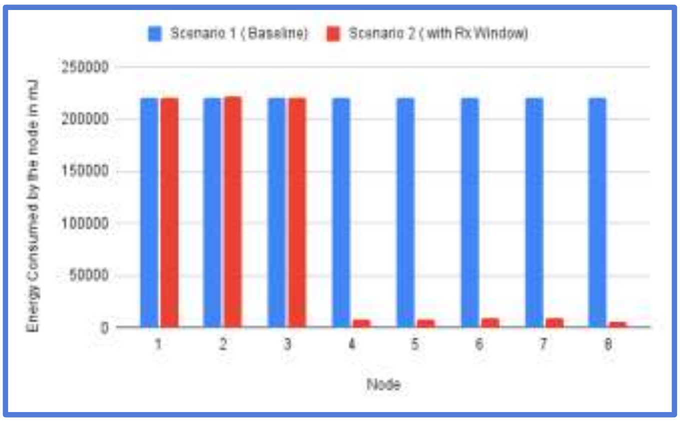

Figure 2 Total energy consumed by the node in Scenario 1 and Scenario 2

7. Results and Discussion

The performance of the proposed ET-RxW mechanism was evaluated against the baseline RDC approach using a 3600 second simulation. The analysis focuses on leaf nodes (Nodes 4–8), as Node 1 acts as the sink and Nodes 2–3 function as Forwarders, which are not expected to benefit from Rx window optimization due to their continuous routing responsibilities. Figure 2 shows Total energy consumption of all 8 nodes in scenario 1 and 2. It is observed that there is significant energy saving in Scenario 2 (ET-RxW)in leaf nodes 4,5,6,7 and 8.

7.1 Radio-RX Energy Reduction

The Radio-RX state is the dominant contributor to energy consumption in RDC-based systems due to continuous idle listening]. Based on the data from Table 5 and Table 6 the average Radio-RX energy for leaf nodes was reduced from $\approx 219{,}580 mJ$ to a range of 5,622–8,260 mJ. This results in an average reduction of approximately 96%, demonstrating the efficacy of aligning radio activity with actual communication events.

7.2. Reliability and Communication Performance

A key concern with aggressive duty cycling is the potential degradation of communication reliability. However, ET-RxW maintains a 100% Packet Delivery Ratio (PDR) for all leaf nodes. The 750ms RX window (Tw) effectively accommodated the cumulative clock drift and processing latencies inherent in the Contiki-NG stack, ensuring that reliability is not compromised for energy savings

Table 4 ET-RXW Operational Parameters

Parameter	Value	Functional Justification
Sensing Interval	120 Seconds	Periodic CPU activation for sensing.
Heartbeat Interval	300 Seconds	Periodic keep-alive for session persistence.
Event Threshold	CO_2 > 450 ppm	Triggers asynchronous event-driven transmission.

Table 4 ET-RXW Operational Parameters

Parameter	Value	Functional Justification
Sync Guard Band	250 ms	Pre-transmission synchronization margin .
Post-TX RX Window	500 ms	Reception window for ACKs and downstream packets.
Total RX Window (Tw)	750 ms	Total radio ON duration per communication event.

Table 5 Baseline Radio Energy (Leaf Nodes)

Node	CPU (mJ)	LPM (mJ)	Tx (mJ)	Rx (mJ)	Total (mJ)	PDR (%)	Pkts
1	556	267	14	219,606	220,443	N/A	N/A
2	659	267	76	219,538	220,540	N/A	N/A
3	494	268	57	219,559	220,377	N/A	N/A
4	375	268	31	219,587	220,261	100	134
5	444	268	36	219,583	220,330	100	139
6	373	268	30	219,588	220,260	100	124
7	379	268	33	219,585	220,265	100	144
8	373	268	33	219,586	220,259	100	138

Table 6 Optimized Radio Rx Energy With Rx Scheduling

Node	CPU (mJ)	LPM (mJ)	Tx (mJ)	Rx (mJ)	Total (mJ)	PDR (%)	Pkts
1	773	2767	16	219,604	220,660	N/A	N/A
2	1423	265	379	219,209	221,277	N/A	N/A
3	903	267	288	219,308	220,765	N/A	N/A
4	412	268	43	7,625	8,347	100	114
5	503	268	58	7,735	8,563	100	122
6	614	267	151	7,874	8,906	100	129
7	375	268	42	8,260	8,945	100	137
8	340	268	41	5,622	6,271	100	52

7.3. Total Energy Consumption and Node Lifetime

Beyond Rx energy, total node energy includes CPU activity and Radio-TX operations. ET-RxW achieved a total energy reduction of approximately 96% by decoupling the sensing plane (20 s interval) from the communication plane (300 s heartbeat). Battery model observations indicate that while baseline nodes retained approximately 96.2% of their capacity after one hour, ET-RxW nodes retained approximately 99.1%, suggesting that this strategy can significantly extend node longevity in practical IoT deployments

7.4. Behavior of Non-Leaf Nodes

Nodes 1–3 behave differently in terms of energy consumption. Node 1 (Sink) keeps its radio active for data reception, whereas Nodes 2–3 (Forwarders) are involved in multi-hop packet forwarding. Hence, the Rx energy reduction at these nodes is limited. The main energy saving of ET-RxW is expected at the leaf nodes, where communication events are sparse and the receiver can remain inactive between events.

7.5. Discussion of Key Findings

The results validate that traditional RDC mechanisms are inefficient for Publish–Subscribe IoT systems due to the dominance of idle listening. By aligning radio activity with application-layer events, ET-RxW achieves a near-optimal energy profile suitable for event-driven environments like industrial monitoring

8. Conclusion

This work investigated the energy-efficiency limitations of conventional Radio Duty Cycling (RDC) in RPL-based Publish–Subscribe IoT networks, where periodic channel probing introduces unnecessary receiver energy consumption. The proposed ET-RxW mechanism separates the frequent CPU sensing cycle (20 s) from the comparatively infrequent application-driven radio communication cycle (300 s), making it well suited for event-driven applications such as environmental monitoring.

Experimental evaluation in the Contiki-NG/Cooja environment using a 5–2–1 network topology shows that ET-RxW reduces Radio-Rx energy consumption of leaf nodes by approximately 96% compared with conventional RDC. This improvement is achieved by eliminating periodic idle listening and activating the radio only during application-defined communication windows.

The reduction in receiver energy consumption is achieved without affecting network reliability. The proposed approach maintains a Packet Delivery Ratio (PDR) of 100% while preserving stable RPL operation. The selected 750 ms receive window also provides sufficient tolerance for synchronization uncertainty and normal network dynamics. Consequently, the overall reduction in energy consumption is expected to extend node lifetime, making the proposed mechanism suitable for long-term deployments in resource-constrained IoT applications.

The results indicate that incorporating application awareness into receiver scheduling can significantly improve the energy efficiency of event-driven IoT networks. Since the proposed mechanism operates without modifying the RPL routing framework, it can be integrated into existing RPL-based systems with minimal changes while reducing unnecessary receiver activity.

References

1. F.-J. Ferrández-Pastor, H. Mora, A. Jimeno-Morenilla, and B. Volckaert, "Deployment of IoT Edge and Fog Computing Technologies to Develop Smart Building Services," *Sustainability*, vol. 10, no. 11, p. 3832, 2018, doi: 10.3390/su10113832.
2. P. E. Gabriel, G. Piñeres-Espitia, S. A. Butt, E. O. Francisco, A. Cama-Pinto, and Y. Maleh, "Performance Analysis of 6LoWPAN Protocol for a Flood Monitoring System," *EURASIP J. Wirel. Commun. Netw.*, vol. 2022, no. 1, p. 16, 2022, doi: 10.1186/s13638-022-02098-3.
3. M. Chernyshev, Z. Baig, O. Bello, and S. Zeadally, "Internet of Things (IoT): Research, Simulators, and Testbeds," *IEEE Internet Things J.*, vol. 5, no. 3, pp. 1637–1647, 2018, doi: 10.1109/JIOT.2017.2786639.
4. N. Barros *et al.*, "SchoolAIR: A Citizen Science IoT Framework Using Low-Cost Sensing for Indoor Air Quality Management," *Sensors*, vol. 24, no. 1, p. 148, 2024, doi: 10.3390/s24010148.
5. M. Benammar, A. Abdaoui, S. H. M. Ahmad, F. Touati, and A. Kadri, "A Modular IoT Platform for Real-Time Indoor Air Quality Monitoring," *Sensors*, vol. 18, no. 2, p. 581, Feb. 2018, doi: 10.3390/s18020581.
6. P. S. Chia, N. H. Kamis, S. F. Abdul Razak, S. Yogarayan, W. Yassin, and M. F. Abdollah, "Routing Protocols Performance on 6LoWPAN IoT Networks," *IoT*, vol. 6, no. 1, p. 12, 2025, doi: 10.3390/iot6010012.
7. N. H. M. Yusoff, N. A. Zakaria, and A. H. Rosli, "Design and Implementation of 6LoWPAN Application: A Performance Assessment Analysis," *Int. J. Adv. Comput. Sci. Appl.*, vol. 11, no. 8, pp. 263–270, 2020, doi: 10.14569/IJACSA.2020.0110834.
8. A. Brandt *et al.*, "RPL: IPv6 Routing Protocol for Low-Power and Lossy Networks," RFC Editor, RFC 6550, 2012. doi: 10.17487/RFC6550.
9. J. Hui and J. P. Vasseur, "The Routing Protocol for Low-Power and Lossy Networks (RPL) Option for Carrying RPL Information in Data-Plane Datagrams," RFC Editor, 6553, Mar. 2012. doi: 10.17487/RFC6553.
10. H.-S. Kim, J. Ko, D. E. Culler, and J. Paek, "Challenging the IPv6 Routing Protocol for Low-Power and Lossy Networks (RPL): A Survey," *IEEE Commun. Surv. Tutor.*, vol. 19, no. 4, pp. 2502–2525, 2017, doi: 10.1109/COMST.2017.2751617.
11. J. Hui, J. P. Vasseur, D. Culler, and V. Manral, "An IPv6 Routing Header for Source Routes with the Routing Protocol for Low-Power and Lossy Networks (RPL)," RFC Editor, 6554, Mar. 2012. doi: 10.17487/RFC6554.
12. J. Schandy, L. Steinfeld, and F. Silveira, "Average Power Consumption Breakdown of Wireless Sensor Network Nodes Using IPv6 over LLNs," in *2015 International Conference on Distributed Computing in Sensor Systems*, 2015, pp. 242–247. doi: 10.1109/DCOSS.2015.37.
13. S. Kharche and S. Pawar, "Node Level Energy Consumption Analysis in 6LoWPAN Network Using Real and Emulated Zolertia Z1 Motes," in *2016 IEEE International Conference on Advanced Networks and Telecommunications Systems (ANTS)*, 2016, pp. 1–5. doi: 10.1109/ANTS.2016.7947870.
14. R. Arya and S. C. Sharma, "Optimization Approach for Energy Minimization and Bandwidth Estimation of WSN for Data Centric Protocols," *Int. J. Syst. Assur. Eng. Manag.*, vol. 9, pp. 2–11, 2018, doi: 10.1007/s13198-015-0397-8.
15. M. Khan and others, "Clustering Based Optimal Cluster Head Selection...," *Energies*, vol. 15, no. 13, p. 4528, 2022, doi: 10.3390/en15134528.
16. B. Motamedi and B. Villányi, "A Reliable Publish–Subscribe Mechanism for Internet of Things-Enabled Smart Greenhouses," *Appl. Sci.*, vol. 14, no. 15, p. 6407, Jan. 2024, doi: 10.3390/app14156407.
17. M. Nast, F. Golatowski, and D. Timmermann, "Design and Performance Evaluation of a Standalone MQTT for Sensor Networks (MQTT-SN) Broker," in *2023 IEEE 19th International Conference on Factory Communication Systems (WFCS)*, 2023, pp. 1–8. doi: 10.1109/WFCS57264.2023.10144241.
18. MQTT.org, "MQTT Specification." 2024. [Online]. Available: https://mqtt.org/mqtt-specification/

19. A. Banks, E. Briggs, K. Borgendale, and R. Gupta, "MQTT Version 5.0." OASIS Standard, Mar. 2019. [Online]. Available: https://docs.oasis-open.org/mqtt/mqtt/v5.0/os/mqtt-v5.0-os.html
20. R. Zitouni, J. Petit, A. Djoudi, and L. George, "IoT-Based Urban Traffic-Light Control: Modelling, Prototyping and Evaluation of MQTT Protocol," in *2019 International Conference on Internet of Things (iThings) and IEEE Green Computing and Communications (GreenCom) and IEEE Cyber, Physical and Social Computing (CPSCom) and IEEE Smart Data (SmartData)*, 2019, pp. 182–189. doi: 10.1109/iThings/GreenCom/CPSCom/SmartData.2019.00051.
21. T. Anitha, S. Manimurugan, S. Sridhar, S. Mathupriya, and G. C. P. Latha, "A Review on Communication Protocols of Industrial Internet of Things," in *2022 2nd International Conference on Computing and Information Technology (ICCIT)*, 2022, pp. 418–423. doi: 10.1109/ICCIT52419.2022.9711544.
22. B. Mishra and A. Kertesz, "The Use of MQTT in M2M and IoT Systems: A Survey," *IEEE Access*, vol. 8, pp. 201071–201086, 2020, doi: 10.1109/ACCESS.2020.3035849.
23. S. Ponde and S. Lomte, "An Energy-Efficient MAC Protocols for Wireless Sensor Networks," in *Computing in Engineering and Technology*, B. Iyer, P. S. Deshpande, S. C. Sharma, and U. Shiurkar, Eds., Singapore: Springer Singapore, 2020, pp. 177–187. doi: 10.1007/978-981-32-9515-5_17.
24. B. R. S. Devi, A. George, and A. K. Thomas, "Performance investigation of low power radio duty cycling MAC for resource constrained WSN," *Int. J. Eng. Technol.*, vol. 7, no. 1.9, pp. 93–98, 2018.
25. J.-W. Kim, J. Kim, and J. Lee, "Cross-Layer MAC/Routing Protocol for Reliability Improvement of the Internet of Things," *Sensors*, vol. 22, no. 23, p. 9429, 2022, doi: 10.3390/s22239429.
26. J.-H. Park, Y.-J. Suh, D. Kim, H. Lee, H. Ahn, and Y. D. Park, "MILD: Minimizing Idle Listening Energy Consumption via Down-Clocking for Energy-Efficient Wi-Fi Communications," *Sensors*, vol. 25, no. 4, p. 1155, 2025, doi: 10.3390/s25041155.
27. A. Bengheni, "An Improved Energy-Efficient MAC Protocol for WSNs," in *Smart Computing and Control Renewable Energy Systems*, M. Hatti, Ed., Cham: Springer Nature Switzerland, 2025, pp. 672–683. doi: 10.1007/978-3-031-80301-7_70.
28. P. Thubert, "Objective Function Zero for the Routing Protocol for Low-Power and Lossy Networks (RPL)," RFC Editor, 6552, Mar. 2012. doi: 10.17487/RFC6552.
29. S. Poorana Senthilkumar and B. Subramani, "Enhanced Objective Function ETX Metric in Routing Protocol for Low-Power and Lossy Networks (RPL)," *Int. J. Comput. Netw. Appl.*, vol. 9, no. 4, pp. 462–475, Aug. 2022, doi: 10.22247/ijcna/2022/214507.
30. S. J. Hussain and M. Roopa, "BE-RPL: Balanced-load and Energy-efficient RPL," *Comput. Syst. Sci. Eng.*, vol. 45, no. 1, pp. 785–801, 2023, doi: 10.32604/csse.2023.030393.
31. S. Mokrani, M. Belkadi, T. Sadoun, J. Lloret, and R. Aoudjit, "LEA-RPL: Lightweight Energy-Aware RPL Protocol for Internet of Things Based on Particle Swarm Optimization," *Telecommun. Syst.*, vol. 88, no. 1, p. 14, 2025, doi: 10.1007/s11235-024-01254-y.
32. H. A. A. Al-Kashoash, H. Kharrufa, Y. Al-Nidawi, and A. H. Kemp, "Congestion control in wireless sensor and 6LoWPAN networks: toward the Internet of Things," *Wirel. Netw.*, vol. 25, no. 8, pp. 4493–4522, 2019, doi: 10.1007/s11276-018-1743-y.
33. S. Tadigotla and J. K. Murthy, "A Comprehensive Study on RPL Challenges," in *2020 Third International Conference on Advances in Electronics, Computers and Communications (ICAECC)*, 2020, pp. 1–6. doi: 10.1109/ICAECC50550.2020.9339505.
34. G. Oikonomou, S. Duquennoy, A. Elsts, J. Eriksson, Y. Tanaka, and N. Tsiftes, "The Contiki-NG open source operating system for next generation IoT devices," *SoftwareX*, vol. 18, p. 101089, 2022, doi: 10.1016/j.softx.2022.101089.
35. Contiki-NG Project, "Official GitHub Repository and Documentation." [Online]. Available: https://github.com/contiki-ng/contiki-ng
36. Texas Instruments, *CC2420 2.4 GHz IEEE 802.15.4 / ZigBee-Ready RF Transceiver Datasheet*. Dallas, TX, USA: Texas Instruments, 2007. [Online]. Available: https://www.ti.com/lit/ds/symlink/cc2420.pdf
37. Texas Instruments, *MSP430F261x, MSP430F241x Mixed-Signal Microcontrollers Datasheet*. Dallas, TX, USA: Texas Instruments, 2007. [Online]. Available: https://www.ti.com/lit/ds/symlink/msp430f2617.pdf
38. Zolertia S.L., *Zolertia Z1 Mote Datasheet, Revision C*. Zolertia S.L., 2010. [Online]. Available: https://github.com/Zolertia/Resources/blob/master/Z1/Hardware/Revision%20C/Datasheets/Zolertia%20Z1%20datasheet%20Revision%20C.pdf
39. S. Scanzio, F. Quarta, G. Paolini, G. Formis, and G. Cena, "Ultralow Power and Green TSCH-Based WSNs With Proactive Reduction of Idle Listening," *IEEE Internet Things J.*, 2024, doi: 10.1109/JIOT.2024.3406646.
40. S. Homayouni and R. Javidan, "ERA-ContikiMAC: An Adaptive Radio Duty Cycling Layer in Internet of Things," in *2018 9th International Symposium on Telecommunications (IST)*, 2018, pp. 74–79. doi: 10.1109/ISTEL.2018.8661144.
41. P. Thubert and J. Hui, "Compression Format for IPv6 Datagrams over IEEE 802.15.4-Based Networks," RFC Editor, 6282, Sep. 2011. doi: 10.17487/RFC6282.
42. G. Montenegro, N. Kushalnagar, J. Hui, and D. Culler, "Transmission of IPv6 Packets over IEEE 802.15.4 Networks," RFC Editor, 4944, Sep. 2007. doi: 10.17487/RFC4944.