MMDEW: An Adaptive Multi-Metric Entropy-Based Objective Function for Stable and Energy-Efficient RPL Routing in Rural Areas

Aditya Wijayanto
School of Data Science, Mathematics and Informatics, IPB University, Bogor, Indonesia | Software Engineering Study Program, Telkom University, Purwokerto Campus, JL. DI Panjaitan, Central Java, Indonesia
adityawijayanto@apps.ipb.ac.id (corresponding author)

Imas Sukaesih Sitanggang
School of Data Science, Mathematics and Informatics, IPB University, Bogor, Indonesia
Imas.sitanggang@apps.ipb.ac.id

Sri Wahjuni
School of Data Science, Mathematics and Informatics, IPB University, Bogor, Indonesia
my_juni04@apps.ipb.ac.id

Hendra Rahmawan
School of Data Science, Mathematics and Informatics, IPB University, Bogor, Indonesia
hrahmawan@apps.ipb.ac.id

Received: 1 October 2025 | Revised: 10 December 2025, 24 December 2025, and 29 December 2025 | Accepted: 30 December 2025

 DOI: https://doi.org/10.48084/etasr.15261

ABSTRACT

The swift expansion of the Internet of Things (IoT) has accelerated the adoption of Wireless Sensor Networks (WSNs) in various fields, particularly in Low-Power and Lossy Networks (LLNs). The Routing Protocol for LLNs (RPL) is an IPv6-based standard for such environments; however, its default objective function, the Minimum Rank with Hysteresis Objective Function (MRHOF), relies solely on the Expected Transmission Count (ETX), which is often suboptimal in dense and dynamic topologies. This study proposes a novel Multi-Metric Dynamic Entropy Weighting (MMDEW) objective function that integrates four key metrics: ETX, Signal-to-Noise Ratio (SNR), ΔCPU Energy Consumption (EC), and handover frequency, using an adaptive entropy weighting scheme derived from local historical variations. This mechanism enables lightweight, distributed, and self-adaptive parent selections. The proposed approach was evaluated in Contiki-NG under multiple transmission intervals (10, 20, and 30 s) and node densities (12, 15, and 20). Compared with MRHOF, MMDEW reduces handovers by up to 89% and CPU energy consumption by 10–43%, while maintaining a comparable Packet Delivery Ratio (PDR) and End-to-End (E2E) delay. Paired t-test results ($p < 0.05$, $|dz| > 0.8$) confirmed MMDEW's significant improvement in route stability and energy efficiency. These findings demonstrate that MMDEW provides a lightweight, distributed, and energy-aware routing solution suitable for long-term rural IoT deployments, such as wildfire early warning systems.

Keywords-RPL; objective function; multi-metric; entropy weighting; LLNs

I. INTRODUCTION

The swift expansion of the Internet of Things (IoT) has accelerated the adoption of Wireless Sensor Networks (WSNs) in various fields, including precision agriculture, smart environmental monitoring, and industrial automation. However, the characteristics of Low-Power and Lossy Networks (LLNs) introduce major challenges owing to limited energy resources and susceptibility to link degradation [1]. Recent research has explored the integration of optimization and learning in IoT environments. Several studies have also explored lightweight optimization and learning techniques to improve IoT performance, including Particle Swarm Optimization (PSO) and genetic algorithms [2] and the Dove

Optimization Algorithm (DOA) [3]. Similarly, PSO-assisted multitask learning [4] and adaptive non-Reinforcement Learning (RL) optimization models [5] have been proposed in the literature. These developments highlight the need for efficient, low-overhead mechanisms suitable for resource-limited IoT devices. In addition, several studies have highlighted the importance of improving the security and efficiency of IoT networks through optimized wireless protocols [6].

To address these challenges, the Internet Engineering Task Force (IETF) standardized the Routing Protocol for LLNs (RPL) as the dominant IPv6-based routing protocol [7, 8]. RPL uses an objective function to guide parent selection and routing metric priorities [9]. The widely deployed Minimum Rank with Hysteresis Objective Function (MRHOF) depends solely on Expected Transmission Count (ETX) [10, 11], which captures link reliability but ignores delay, residual energy, and stability aspects. As shown in prior studies, this single-metric design can degrade Packet Delivery Ratio (PDR) and energy efficiency in dense and dynamic topologies [11-14].

To address these limitations, researchers have proposed multi-metric RPL enhancements that combine ETX with delay, energy, Received Signal Strength Indicator (RSSI)/Signal-to-Noise Ratio (SNR), or queue metrics via fuzzy logic [15], entropy weighting with TOPSIS [16-18], or RL techniques [19, 20]. Although these approaches improve reliability and stability, many rely on region weight computation [18], require high computational costs [21], or incur long convergence times, which are typical of RL-based routing [22-24]. Security-oriented extensions, including anomaly detection [25], and group-key-based secure routing [26], further underscore the complexity of robust RPL operation.

Despite these advances, there are still significant gaps. EM-RPL requires global synchronization overhead [16]; CQARPL improves congestion handling but increases memory and control complexity [18]; DDSA-RPL achieves adaptive learning at the cost of heavy computation and long convergence [17]. Q-Learning-based RPL variants (CAQL-RI-RPL) enhance stability but remain impractical for real-time LLNs because of iteration costs and exploration overheads [19, 21]. Trust or energy-aware extensions [22-24] require frequent state exchanges, increasing energy usage. These limitations highlight the need for a routing mechanism that is lightweight, fully distributed, and capable of local adaptation without global coordination.

Table I summarizes the representative RPL extensions, their metrics, adaptation scopes, and main limitations. Although these methods offer improvements, most still depend on centralized coordination, static weighting, and high computational costs. This persistent gap motivates a new locally adaptive objective function capable of capturing short-term variations in link quality and energy while remaining lightweight for LLN motes. Field-based studies on forest fire patrol systems have also demonstrated that rural IoT deployments require routing strategies that remain stable, scalable, and energy-efficient under highly dynamic link conditions [27]. Additionally, authors in [28] showed that IoT-based operations in rural plantation areas require reliable communication to ensure stable real-time data acquisition. Authors in [29] also showed that incorporating bandwidth and latency into routing decisions enhances overall network reliability.

TABLE I. COMPARATIVE SUMMARY OF RPL OBJECTIVE FUNCTIONS

Ref.	Method	Metric used	Weighting/learning strategy	Adaptive scope	Mobility support	Key strength	Main limitations
[10]	MRHOF	ETX	–	Static	Static	Standard RPL, objective function, simple and lightweight	Ignores delay, energy, and stability, prone to frequent parent changes
[15]	Fuzzy-RPL	EXT, delay, energy	Fuzzy logic rules	Node (local)	Static	Multi-criteria reasoning improves reliability and energy balance	Requires rule tuning, weights remain static once defined
[16]	EM-RPL	ETX, RSSI, energy, timers	Multi-metric objective function with mobility timers	Node + local	Dynamic	Improves mobility handling and stability under node movement	Moderate delay, requires node-type preconfiguration
[18]	CQARPL	ETX, queue length	Heuristic / queue-aware weighting	Node	Static	Reduces congestion and maintains QoS	Higher memory and control message overhead
[17]	DDSLA-RPL	ETX, delay, energy, SNR	Learning automata	Node + global	Static	Adaptive routing through learning automata	Heavy computation cost, long convergence time
[19, 22]	Q-RPL (CAQL/RI-RPL)	ETX, energy, trust	Q-learning	Node distributed	Dynamic	Learns optimal routers adaptively under mobility	High iteration cost and exploration overhead
[23, 30]	QU-RPL / SL-RPL	ETX, energy, delay, reliability	Q-learning and supervised learning	Node distributed	Static	Learns optimal routes via utility and experience-based trust metrics	Requires model training and reward tuning, moderate computation cost
This work	MMDEW	ETX, SNR, ΔCPU EC, and handover	Entropy-based adaptive weighting using local standard deviation	Per-parent (Local)	Static and Dynamic	Stable, energy-efficient, and lightweight, suitable for mobile and rural IoT nodes	Slight increase in delay under high-density networks

Based on this insight, this study introduces the Multi-Metric Dynamic Entropy Weighting (MMDEW) objective function. Unlike global weighting or RL-based methods, MMDEW derives adaptive weights from the standard deviation of the local per-parent history, enabling responsiveness to link and energy variations without global synchronization or iterative training. This design preserves computational efficiency while improving stability and energy performance in rural and dynamic IoT Environments.

The main contributions of this study are summarized as follows:

1. A novel multi-metric RPL objective function (MMDEW) integrating reliability, signal quality, energy efficiency, and stability.
2. An adaptive entropy-based weighting mechanism driven by local historical variations.
3. Comprehensive performance evaluation under static and mobile scenarios using multiple routing metrics.

II. METHODOLOGY

The proposed MMDEW objective function is designed to overcome the limitations of single-metric RPL routing, such as MRHOF, which relies solely on ETX [10]. The MMDEW integrates four core metrics, ETX, SNR, Δ CPU Energy Consumption (EC), and handover, with an adaptive weighting mechanism based on the standard deviation of local historical values for each candidate parent. The selection of these metrics is supported by prior research emphasizing the importance of combining transmission reliability, energy efficiency, physical link quality, and route stability in LLNs [16, 31].

The overall parent selection process in the MMDEW is illustrated in Figure 1 and consists of five main blocks: initialization, data acquisition, adaptive weighting, rank calculation, and parent update decision.

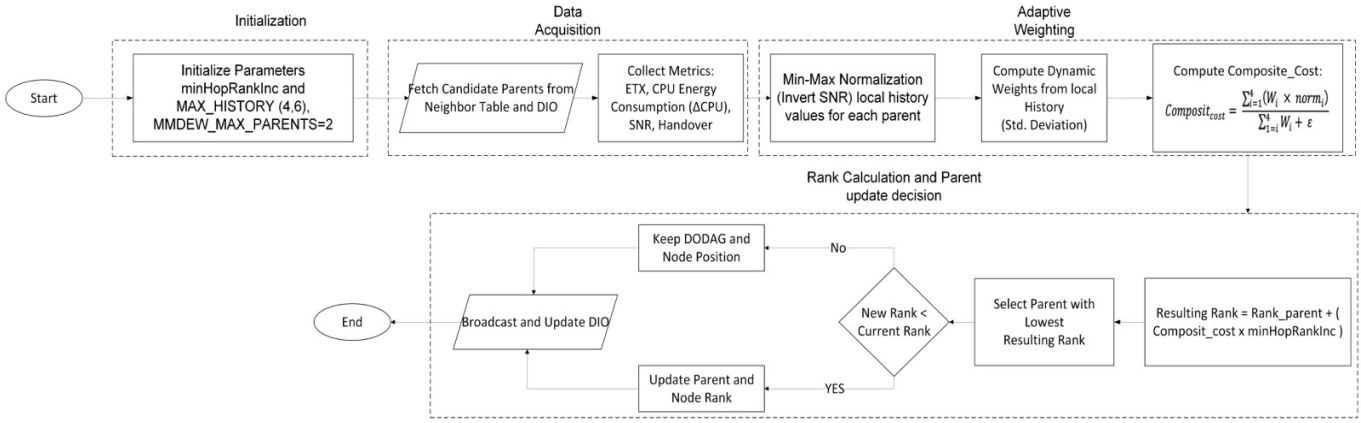

Fig. 1. Flowchart of the proposed MMDEW objective function illustrating the main phases of initialization, data acquisition, adaptive weighting, rank calculation, and parent update decision used for route optimization in RPL.

A. Initialization

At this stage, the initial parameters were configured to ensure consistency, and minHopRankInc was set to 128, following the RPL standard [8]. MAX_HISTORY was set to four and six entries to evaluate the trade-off between responsiveness (H4) and stability (H6). MMDEW_MAX_PARENTS is limited to two candidate parents to reduce the memory overhead of resource-constrained LLN nodes [32].

B. Data Acquisition

Before parent selection, the MMDEW collects four key metrics from neighboring nodes. This information is obtained from the neighbor table and DODAG Information Object (DIO) messages [8, 10].

1) Expected Transmission Count

ETX calculates the typical number of transmissions, including any necessary retransmissions, required to successfully deliver a packet from the sender to the receiver [9]. The ETX formula is given by (1) [16]:

$$\mathrm{ETX} = \frac{1}{\mathrm{df} \times \mathrm{dr}} \quad (1)$$

where df is the delivery ratio, and dr is the reverse delivery ratio.

2) ΔCPU Energy Consumption

ΔCPU EC represents the internal processing overhead of the objective function and is measured using the Energest module in Contiki-NG. This is derived from the difference in the active CPU ticks between parent selection iterations [32]. Based on the Zolertia Z1 mote specifications, with an active current I_{CPU} = 1.8mA and supply voltage V = 3.0 V. The CPU EC formula is presented in (2):

$$\Delta\mathrm{CPU\ EC} = \left(\frac{\Delta t_{CPU}}{f_{CPU}}\right) \times I_{CPU} \times V \quad (2)$$

where Δt_{CPU} is the difference in active CPU ticks between iterations, f_{CPU} is the system frequency, I_{CPU} is the active CPU current, and V is the supply voltage. This study provides an estimate of the delta energy consumption in millijoules (mJ).

3) Signal-to-Noise Ratio

SNR reflects the ratio between the received signal power and the noise power, providing a more reliable indicator of the physical link quality than the RSSI [31]. The SNR formula is given by (3):

$$\mathrm{SNR_{dB}} = \mathrm{P_{signal}} - \mathrm{P_{noise}} \quad (3)$$

where $\mathrm{P_{signal}}$ is the strength of the received signal and $\mathrm{P_{noise}}$ is the strength of the received noise. A constant noise floor of -85 dBm is used in the simulation, in line with the field measurements reported in [31].

4) Handover

Handover quantifies route instability by measuring the parent-switching frequency [14]. The handover is calculated cumulatively based on the changes in the parent node Link Layer Address (LLADR) and its formula is given by (4):

$$\mathrm{HO(t)} = \mathrm{HO(t-1)} + \delta_{\mathrm{switch}}(\mathrm{t}) \quad (4)$$

where $\mathrm{HO(t)}$ is the total number of parent handovers that have occurred at a node up to time t, $\mathrm{HO(t-1)}$ is the cumulative value of the previous handover before time t, and $\delta_{\mathrm{switch}}(\mathrm{t})$ is an indicator of the occurrence of parent switching at time t, defined as in Eq. (5):

$$\delta_{\mathrm{switch}}(\mathrm{t}) = \begin{cases} 1, & \text{if } \mathrm{LLADR_{new}(t)} \neq \mathrm{LLADR_{prev}} \\ 0, & \text{otherwise} \end{cases} \quad (5)$$

where $\mathrm{LLADR_{new}(t)}$ is the link-layer address (MAC address) of the new parent selected at time t, and $\mathrm{LLADR_{prev}}$ is the link-layer address of the parent that was previously used.

The MMDEW updates a per-parent circular history buffer (H4/H6) containing ETX, SNR, ΔCPU EC, and HO values. Each new metric overwrites the oldest entry, enabling short-term variation modeling for weight computation. The combined normalized metric forms a composite cost, which is scaled using minHopRankInc and added to the parent's rank to generate the final rank. HO donates a change in the preferred parent and serves as a stability indicator. ΔCPU EC ticks are converted to mJ using device-specific current and voltage parameters.

C. Adaptive Weighting

MMDEW applies local variation-based weights in real time and is decentralized. The weight of each metric is calculated dynamically using the standard deviation from the local history. Metrics with higher variations are considered to have greater effects on route stability. This differs from conventional entropy methods, which are global [18].

Each metric is normalized to ensure a comparable scale [33]. The normalization formula is given by (6):

$$\mathrm{X'} = \frac{\mathrm{X} - \mathrm{X_{min}}}{\mathrm{X_{max}} - \mathrm{X_{min}}} \quad (6)$$

where $\mathrm{X'}$ is the normalized metric value, X denotes the original value, and $\mathrm{X_{max}}$, $\mathrm{X_{min}}$ represent the maximum and minimum values.

For metrics such as the SNR, inversion is applied for consistency. The inversion formula is expressed in (7):

$$\mathrm{X'_{invert}} = 1 - \mathrm{X'} \quad (7)$$

where $\mathrm{X'}$ is the normalized value and $\mathrm{X'_{invert}}$ is the inversion result.

The weight of each metric ($\mathrm{W_i}$) is calculated dynamically using the standard deviation of the local history (with history lengths of four and six). A higher standard deviation indicates a greater influence of a metric [34]. The weight calculation is given by (8):

$$\mathrm{SD_i} = \sqrt{\frac{1}{\mathrm{N}} \sum_{\mathrm{i=1}}^{\mathrm{N}} (\mathrm{x_i} - \bar{\mathrm{x}}_\mathrm{i})^2}, \ \bar{\mathrm{x}}_\mathrm{i} = \frac{1}{\mathrm{N}} \sum_{\mathrm{i=1}}^{\mathrm{N}} \mathrm{x_i}$$

$$\mathrm{W_i} = \frac{\mathrm{SD_i}}{\sum_\mathrm{j} \mathrm{SD_j}}, \ \mathrm{i} \in \{\mathrm{ETX}, \Delta\mathrm{CPU\ EC}, \mathrm{SNR}, \mathrm{HO}\} \quad (8)$$

where:

- $\mathrm{SD_i}$: standard deviation of metric i
- $\bar{\mathrm{x}}_\mathrm{i}$: mean value of metric i across N history entries
- $\mathrm{x_i}$: value of metric i
- N: history size (4 or 6 in this study)
- $\mathrm{W_i}$: weight of metric i

To reflect the dominance of each metric, the composite cost was computed as the weighted average of all normalized metrics, as shown in (9):

$$\mathrm{Composit_{cost}} = \frac{\sum_{\mathrm{i=1}}^{4} (\mathrm{W_i} \times \mathrm{norm_i})}{\sum_{\mathrm{1=i}}^{4} \mathrm{W_i} + \varepsilon} \quad (9)$$

where:

- $\mathrm{Composit_{cost}}$: total composite cost of the candidate parent
- $\mathrm{W_i}$: weight of metric i
- $\mathrm{norm_i}$: normalization value metric i
- $\sum_{\mathrm{i=1}}^{4} (\mathrm{W_i} \times \mathrm{norm_i})$: total weighted contribution of all metrics
- $\sum_{\mathrm{1=i}}^{4} \mathrm{W_i} + \varepsilon$: normalization factor (with a small ε to avoid division by zero)

MMDEW effectively becomes O(n) computational complexity in RPL, as all input dimensions are strictly bounded and implemented as constant size structures, whereas RL retains a multiplicative O(|S||A||I|) complexity due to recursive temporal difference updates.

D. Rank Calculation and Parent Update Decision

The evaluation of rank in RPL follows [8], where minHopRankInc is used as a scaling factor applied to the composite cost. The most commonly used value is 128 [7]. The composite cost formula is presented in (10):

$$\mathrm{R} = \mathrm{Composit_{cost}} \times \mathrm{minHopRankInc} \quad (10)$$

where:

- R: rank increment representing the quality of the selected route
- $\text{Composit}_{\text{cost}}$: derived from the normalized and weighted metrics
- minHopRankInc: minimum rank increment constant per hop (128)

The preferred parent is selected based on its lowest rank. This approach aligns with the RPL standard, which uses ranks to indicate the path [8, 10]. The final ranking is determined using (11):

$$\text{Rank}_{\text{node}} = \text{Rank}_{\text{parent}} + \text{R} \quad (11)$$

where $\text{Rank}_{\text{parent}}$ is the updated rank of the node based on the selected parent node.

If the rank is lower than the current rank, the node updates its preferred parent, recalculates its rank, and increments the global and local handover counters. Otherwise, the current parent is retained. After the decision is made, the node broadcasts an updated DIO to inform the neighboring nodes of the rank change. Algorithm 1 summarizes the complete MMDEW parent-selection procedure, including metric acquisition, normalization, weight calculation, composite cost evaluation, and ranking updates.

In the implemented MMDEW, the metric weights are updated under two conditions. First, during normal operation, each node recalculates its adaptive weight on-demand whenever RPL triggers a rank update or parent reselection event, typically coinciding with data transmission or DIO processing. Second, a reactive update occurs immediately when a handover (HO) is detected, ensuring that recent link and energy variations are reflected in the parent selection process. This event-driven design allows the MMDEW to remain lightweight while maintaining its responsiveness to changes in topology.

```
Algorithm 1. Adaptive Weighting in MMDEW
Input:
     ETX, SNR, EC, HO metrics from
     neighbor list
Output:
     Updated rank R for each node

0:   When RPL triggers a rank update or
     Handover (HO)event, do:
1:   For each parent node p in the
     neighbor list, do:
2:   Read metrics (ETX, SNR, ΔCPU EC, HO)
     from local history
3:   Normalize each metric using (6)
4:   Compute standard-deviation-based
     weight using (8)
5:   Calculate CompositeCost using (9)
6:   Update parent rank R(p) = R_parent +
     CompositeCost x minHopRankInc(128)
7:   End for
8:   Select parent with minimum rank
9:   Optionally update HO counters and
     broadcast DIO advertisement
```

III. EXPERIMENTAL DESIGN

A. Simulation Environment

The experiment was conducted using the Contiki-NG platform on the Cooja simulator, which ran on Ubuntu 22.04 LTS. Each sensor node was configured as a Zolertia Z1 mote equipped with an MSP430 microcontroller, 8 KB RAM, and a CC2420 radio transceiver supporting IEEE 802.15.4 at 2.4 GHz, which is a widely adopted platform in WSN research [35]. MRHOF was chosen as the single baseline since it is the standard RPL objective function [10] and the foundation of subsequent extensions such as CQARPL and EM-RPL. Comparing MMDEW directly with these derived methods would obscure their independent contributions, as their enhancements are already built upon MRHOF. Therefore, the MRHOF provides the most neutral and widely accepted reference point for the evaluation. The Energest module was used to record the energy consumption, whereas a UDP client-server application generated traffic and acted as a data sink.

B. Simulation Parameters

The experiments were performed using UDP (IPv6/6LoWPAN). Table II presents the simulation parameters for packet transmission intervals of 10, 20, and 30 s [36]. Each packet contained a node ID and timestamp to measure the PDR and End-to-End (E2E) delay. MMDEW was configured with two history lengths: H4 and H6.

TABLE II. SIMULATION PARAMETERS

Component	Configuration
Interval	10 s, 20 s, and 30 s
MAC protocol	CSMA
Transmission range	70 m
Simulation time	600 s
Number of nodes	12 (2 moving nodes), 15 (5 moving nodes), and 20 (10 moving nodes)
Simulation area	150 m × 150 m
MAX_HISTORY	4 and 6
minHopRankInc	128
Simulation speed	20×
Repetitions	2 runs per scenario
Random seed	Fixed at 123456
Mobility speed	1–4 m/s, with 2 s pause time
Evaluation metrics	PDR, E2E delay, HO, ΔCPU EC
Hardware spec	AMD Ryzen 3, 8 GB RAM

Table III summarizes the experimental scenarios, which combined variations in node density (12, 15, and 20 nodes), transmission interval (10, 20, and 30 s), and history length (H4 and H6). Each scenario was tested using both MRHOF and MMDEW. To ensure statistical reliability, each configuration was executed twice (two repetitions) under identical settings, resulting in a total of 144 simulation runs. All simulations were conducted with a fixed random seed value of 123456 on an AMD Ryzen 3, 8 GB RAM environment using Contiki-NG with the Cooja simulator.

A test scenario is illustrated in Figure 2, which depicts the placement of nodes based on the groundwater observation well pattern [37]. The yellow nodes represent static sensors deployed for Groundwater Level (GWL) monitoring, and the mobile nodes are shown in green. The mobility model used is the Random Waypoint (RWP) model [19, 22]. Mobile nodes moved randomly toward a target node (purple) within the simulation area at speeds ranging from 1 to 4 m/s, with a pause time of 2 s. Although simple, the RWP model effectively represents field patrol movements in wildfire monitoring scenarios. An example of node mobility with two mobile nodes is shown in Figure 3.

TABLE III. SUMMARY OF SIMULATION SCENARIOS

Number of Nodes	Mobility variations	Interval (s)	History	Tested objective functions
12	V1 and V2	10, 20, and 30	4 and 6	MRHOF, MMDEW
15	V1 and V2	10, 20, and 30	4 and 6	MRHOF, MMDEW
20	V1 and V2	10, 20, and 30	4 and 6	MRHOF, MMDEW

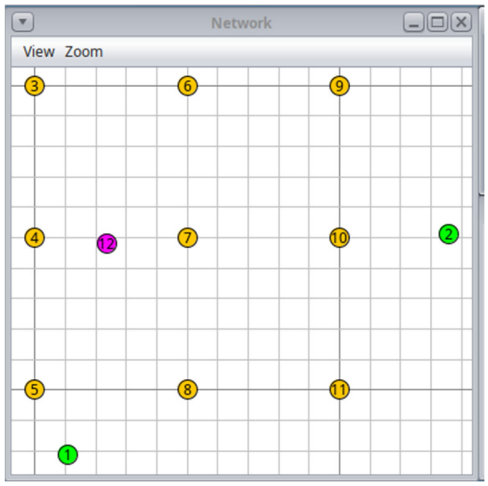

Fig. 2. Simulation topology with 12 nodes, of which two are mobile nodes, following a groundwater-level deployment pattern.

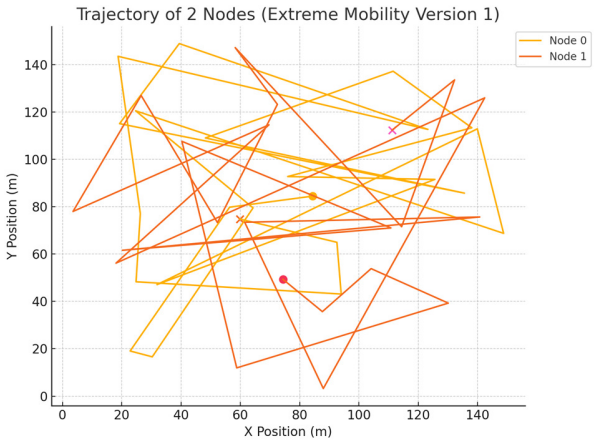

Fig. 3. Example of an RWP mobility pattern with two mobile nodes moving between waypoints at 1–4 m/s, representing field patrol movements in rural wildfire monitoring scenarios.

IV. RESULTS AND DISCUSSION

The performance of routing in LLNs can be explained by three main components of network delay: data, transmission, and processing [35]. First, queuing delay is primarily influenced by packet size and transmission interval. Shorter intervals (10 s) increase the packet arrival rate, which in turn raises the packet backlog at intermediate nodes, leading to longer waiting times. Second, transmission delays arise mainly from retransmissions, which occur more frequently when link quality degrades, as indicated by high ETX values and low SNR. These conditions reduce link reliability and consequently increase E2E latency [36, 38]. Third, processing delay reflects the computational cost incurred by the nodes when updating historical records, calculating weights, and selecting preferred parents.

A. Reliability

To illustrate the reliability characteristics of the evaluated routing strategies, the performance trends for PDR and E2E delay under varying network sizes and traffic loads are summarized in Figures 4 and 5. Figure 4 shows that MRHOF consistently achieves higher PDR than the proposed MMDEW, with values reaching up to 100% in the 15-node, 30 s scenario. This behavior is expected, as MRHOF directly optimizes ETX, prioritizing highly reliable links. Nevertheless, MMDEW maintains a PDR of at least 85% across all configurations, demonstrating robust data delivery performance. Furthermore, the configuration with a longer history window (H6) consistently outperforms H4, indicating better smoothing of short-term metric fluctuations. Figure 5 indicates that MRHOF provides lower E2E delay, which is expected because MRHOF optimizes ETX, favoring shorter and more reliable hops. Although MMDEW introduced a moderate increase in delay (maximum 320.95 ms), this remains within the acceptable bounds for real-time IoT applications, as delays above 420 ms are tolerable according to prior WSN studies [39].

B. Stability

To highlight the impact of the proposed objective function on routing persistency, the stability behaviors of MMDEW and MRHOF across different configurations are presented in Figure 6. Figure 6 demonstrates that the MMDEW significantly reduces handover events, achieving up to 89% fewer parent changes than the MRHOF. This confirms the stability benefit of using local standard deviation-based weighting. Increasing the history window (H6) further enhances stability, which is consistent with earlier findings from mobility-aware RPL enhancements, such as EM-RPL [16] and CQARPL [18].

C. Energy Efficiency

To further assess the resource utilization of the protocol, the comparative trends in energy consumption between MMDEW and MRHOF are summarized in Figure 7. Figure 7 shows that MMDEW decreases Δ CPU EC by 10–43%, reflecting the efficiency of its lightweight computation. These results align with earlier energy-focused RPL improvements, such as AC-RPL [40] and Adaptive Congestion-Aware RPL [41], which emphasize CPU efficiency as a major contributor to network lifetime.

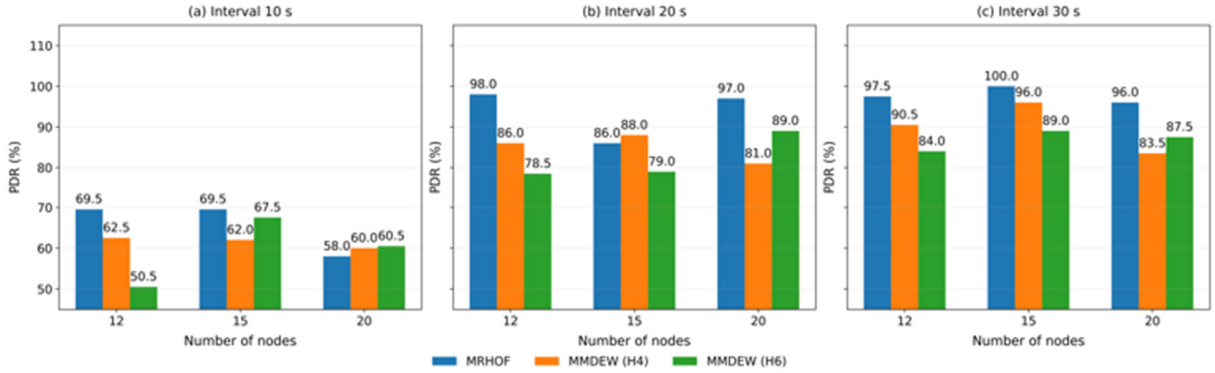

Fig. 4. PDR comparison between MRHOF and MMDEW (H4, H6) under varying numbers of nodes (12, 15, and 20) and intervals (10,20, and 30 s).

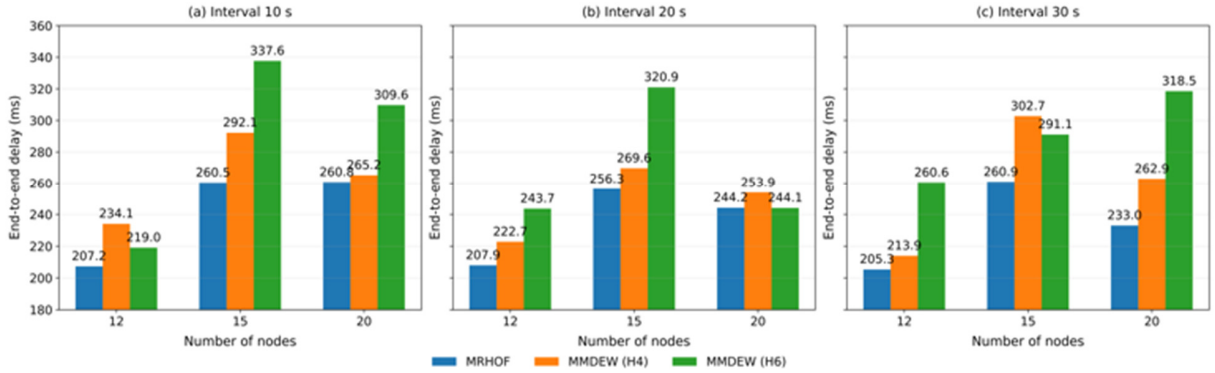

Fig. 5. E2E delay comparison between MRHOF and MMDEW (H4, H6) under varying numbers of nodes (12, 15, and 20) and intervals (10,20, and 30 s).

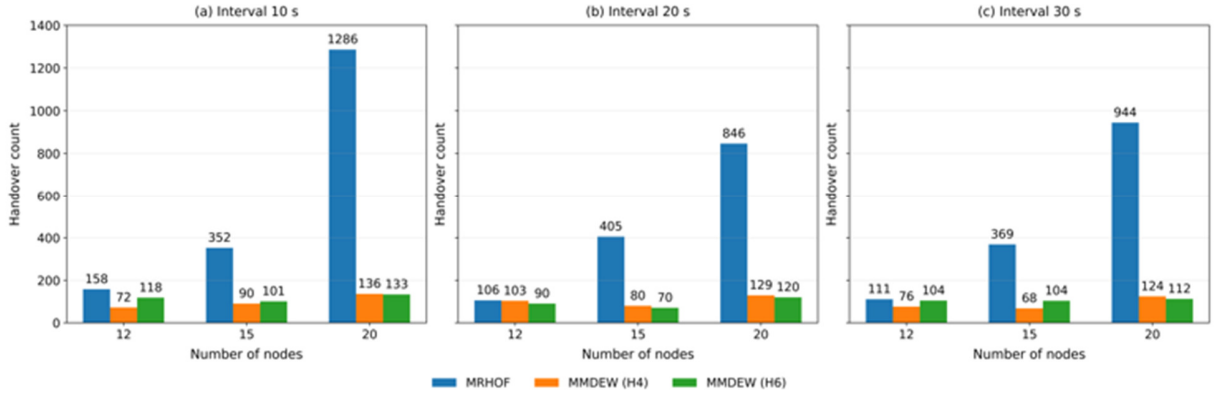

Fig. 6. Handover comparison between MRHOF and MMDEW (H4, H6) under varying numbers of nodes (12, 15, and 20) and intervals (10,20, and 30 s).

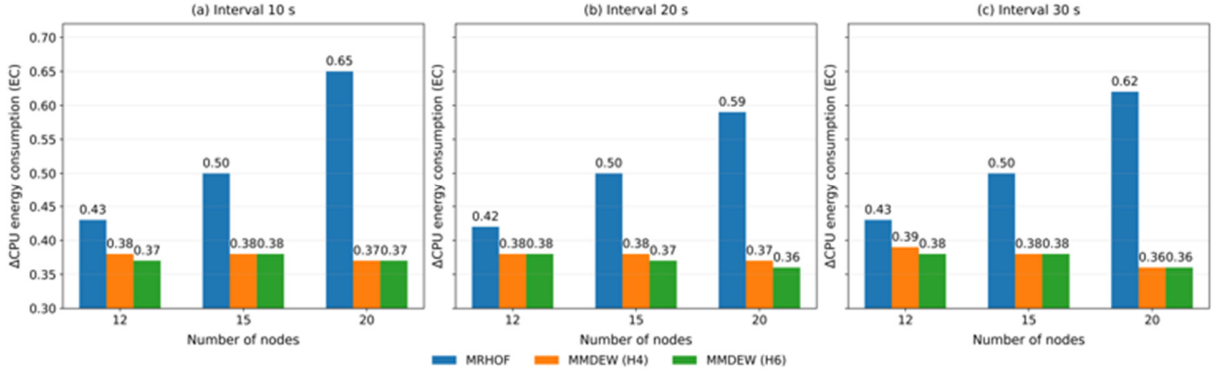

Fig. 7. ΔCPU EC comparison between MRHOF and MMDEW (H4, H6) under varying numbers of nodes (12, 15, and 20) and intervals (10,20, and 30 s).

D. Numerical Performance Summary

To provide a consolidated view of the experimental outcomes, detailed numerical measurements for all evaluated scenarios are summarized in Table IV. Table IV presents the numerical results (mean ± SD) for all node densities, intervals, and history configurations. The trends are consistent with the graphical results, and MMDEW provides substantial improvements in stability and energy. MRHOF remains superior in raw reliability (PDR and E2E delay). The standard deviations were generally small, indicating reproducible behavior across repeated runs.

TABLE IV. NUMERICAL RESULTS OF MMDEW VS. MRHOF (MEAN ± SD)

Interval	Objective function	Nodes	Mobile nodes	History	PDR (%)	E2E delay (ms)	HO (count)	ΔCPU EC (mJ)
10	MRHOF	12	2	-	69.50 ± 3.5	207.25 ± 0.8	157.75 ± 4.2	0.43 ± 0.00
		15	5	-	69.50 ± 2.1	260.50 ± 1.1	352.50 ± 7.2	0.50 ± 0.01
		20	10	-	58.00 ± 1.4	260.75 ± 1.6	1,286.15. ± 7.2	0.65 ± 0.01
	MMDEW	12	2	4	62.50 ± 3.5	234.05 ± 6.1	71.90 ± 2.1	0.38 ± 0.01
		15	5	4	62.00 ± 4.2	292.15 ± 6.6	90.0 ± 4.8	0.38 ± 0.01
		20	10	4	60.00 ± 0.0	265.95 ± 7.5	135.95 ± 3.2	0.37 ± 0.00
	MMDEW	12	2	6	50.50 ± 0.7	219.00 ± 3.0	118.0 ± 3.2	0.37 ± 0.01
		15	5	6	67.50 ± 1.7	337.65 ± 8.4	100.80 ± 2.8	0.38 ± 0.01
		20	10	6	60.50 ± 4.9	309.65 ± 6.8	133.30 ± 3.1	0.37 ± 0.00
20	MRHOF	12	2	-	98.00 ± 2.8	207.95 ± 2.1	105.70 ± 2.4	0.42 ± 0.00
		15	5	-	88.50 ± 3.5	256.30 ± 1.2	405.35 ± 2.7	0.50 ± 0.01
		20	10	-	90.50 ± 4.6	244.20 ± 1.8	846.10 ± 3.5	0.59 ± 0.02
	MMDEW	12	2	4	86.00 ± 2.1	222.70 ± 3.4	103.25 ± 2.1	0.38 ± 0.00
		15	5	4	88.00 ± 3.5	269.65 ± 4.5	79.80 ± 3.2	0.38 ± 0.01
		20	10	4	81.00 ± 4.2	253.95 ± 6.1	129.05 ± 2.8	0.37 ± 0.01
	MMDEW	12	2	6	78.50 ± 2.3	243.65 ± 3.2	90.20 ± 3.2	0.38 ± 0.01
		15	5	6	79.00 ± 3.1	320.95 ± 4.3	70.45 ± 3.2	0.37 ± 0.01
		20	10	6	89.00 ± 1.4	244.05 ± 2.9	119.80 ± 3.0	0.36 ± 0.01
30	MRHOF	12	2	-	97.50 ± 2.1	205.30 ± 1.5	110.85 ± 2.3	0.43 ± 0.00
		15	5	-	100.00 ± 0.0	260.95 ± 1.2	368.80 ± 2.4	0.50 ± 0.01
		20	10	-	96.00 ± 2.8	233.00 ± 1.9	944.00 ± 3.1	0.62 ± 0.01
	MMDEW	12	2	4	83.50 ± 3.5	213.90 ± 3.1	76.10 ± 2.3	0.39 ± 0.01
		15	5	4	84.00 ± 2.8	302.70 ± 3.5	67.95 ± 2.2	0.38 ± 0.00
		20	10	4	83.50 ± 3.5	262.90 ± 4.1	124.25 ± 2.9	0.36 ± 0.00
	MMDEW	12	2	6	84.00 ± 2.8	260.60 ± 3.1	104.35 ± 2.8	0.38 ± 0.00
		15	5	6	89.00 ± 1.4	291.05 ± 3.2	103.60 ± 2.1	0.38 ± 0.00
		20	10	6	87.50 ± 2.1	318.50 ± 3.5	112.25 ± 3.0	0.38 ± 0.00

a. Values represent mean ± standard deviation (SD) from two simulation runs under identical configurations.

E. Statistical Validation (t-test)

To verify statistical significance, a paired t-test was performed using nine matching MRHOF and MMDEW scenarios (H6 only). The results in Table V show that all p-values are less than 0.05. Negative t-values confirm that MMDEW has lower HO and ΔCPU EC. Positive delay t-values reflected the expected trade-off of adaptive multi-metric weighting. Effect sizes $|d_z| > 0.8$ indicate large practical significance.

TABLE V. PAIRED T-TEST VALIDATION OF MMDEW VS. MRHOF (H6)

Metric	t	p (two-tailed)	Effect size (d_z)	Interpretation
PDR (%)	-3.56	0.0037	-1.19	Significant, MRHOF has higher PDR
E2E delay (ms)	4.74	0.00073	1.58	Significant, MMDEW delay slightly higher
HO (count)	-2.96	0.009	-0.99	Significant, MMDEW reduces handovers
ΔCPU EC (mJ)	-4.93	0.0005	-1.64	Highly significant, MMDEW reduces ΔCPU EC

a. Negative t indicates that MMDEW < MRHOF. Family-wise error controlled via Holm–Benferronu ($\alpha = 0.05$).

F. Correlation Analysis

The Spearman correlation heatmap in Figure 8 shows that HO is positively correlated with ΔCPU EC (0.40), suggesting that frequent parent switching directly increases CPU load. In contrast, the correlation between PDR and E2E delay was relatively weak (-0.24), confirming that a higher PDR generally corresponds to a lower delay, although the effect is limited. Overall, compared with the MRHOF and representative multi-metric RPL objective functions reported in the literature, the proposed MMDEW demonstrates stronger improvements in routing stability and energy efficiency. MMDEW reduces handover frequency by up to 89% and lowers ΔCPU EC by 10–43%, exceeding the gains typically reported with weighting methods. These improvements were statistically significant based on paired t-test analysis ($p < 0.05$, $|d_z| > 0.8$). Although MMDEW introduces a moderate E2E delay increase of approximately 10%, the PDR remains within 5–10% of the MRHOF baseline across all evaluated scenarios. In contrast to RL-based routing schemes, MMDEW achieves these improvements without iterative training, exploration overhead, or complex state-action modeling, making it particularly suitable for resource-constrained and dynamic rural IoT deployments.

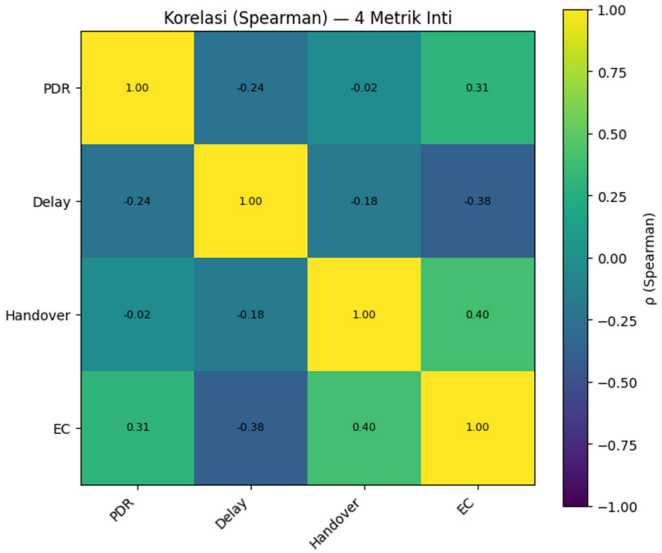

Fig. 8. Spearman correlation heatmap between performance metrics (PDR, Delay, ΔCPU EC, and HO).

V. CONCLUSION

The simulation results demonstrate that the proposed Multi-Metric Dynamic Entropy Weighting (MMDEW) objective function significantly improves routing stability and energy efficiency in Routing Protocol for Low-Power and Lossy Networks (RPL) networks while maintaining reliable data delivery. Compared with Minimum Rank with Hysteresis Objective Function (MRHOF), MMDEW reduced handover frequency by up to 89% and lowered Δ CPU Energy Consumption (EC) by 10–43%, while sustaining a Packet Delivery Ratio (PDR) within 5–10% of the MRHOF baseline and incurring only a moderate End-to-End (E2E) delay increase of approximately 10%.

Paired t-test analysis ($p < 0.05$, $|d_z| > 0.8$) confirmed that the observed improvements in routing stability and energy efficiency were statistically significant. Among the evaluated configurations, MMDEW with a longer history window (H6) achieved the best overall performance balance across 144 simulation runs, highlighting the effectiveness of entropy-based adaptation using local historical metric variations.

ACKNOWLEDGMENT

This research was funded by the Ministry of Education, Culture, Research and Technology, Republic of Indonesia (Grant No: 02021/BPPT/BPI.06/9/2023). The authors gratefully acknowledge the support of the Indonesian Education Scholarship (Beasiswa Pendidikan Indonesia/ BPI), the Center for Higher Education Funding and Assessment (Pelayanan Pembiayaan dan Asesmen Pendidikan Tinggi/ PPAPT), and the Endowment Fund for Education Agency (Lembaga Pengelola Dana Pendidikan/LPDP), Ministry of Finance, Republic of Indonesia.

REFERENCES

[1] T. Bekar, S. Görmüş, B. Aydın, and H. Aydın, "Q-Learning Algorithm Inspired Objective Function Optimization For IETF 6TiSCH Networks," in *2023 International Conference on Smart Applications, Communications and Networking (SmartNets)*, Istanbul, Turkiye, 2023, pp. 1–6, https://doi.org/10.1109/SmartNets58706.2023.10216124.

[2] A. M. Elshewey, A. A. Alhussan, D. S. Khafaga, E.-S. M. Elkenawy, and Z. Tarek, "EEG-based optimization of eye state classification using modified-BER metaheuristic algorithm," *Scientific Reports*, vol. 14, no. 1, Oct. 2024, Art. no. 24489, https://doi.org/10.1038/s41598-024-74475-5.

[3] A. M. Elshewey, "Enhancing crop yield prediction based on dove optimization algorithm and gradient boosting model," *Signal, Image and Video Processing*, vol. 19, no. 11, July 2025, Art. no. 951, https://doi.org/10.1007/s11760-025-04545-2.

[4] N. El-Rashidy, Z. Tarek, A. M. Elshewey, and M. Y. Shams, "Multitask multilayer-prediction model for predicting mechanical ventilation and the associated mortality rate," *Neural Computing and Applications*, vol. 37, no. 3, pp. 1321–1343, Jan. 2025, https://doi.org/10.1007/s00521-024-10468-9.

[5] J. Lei and J. Liu, "Reinforcement learning-based load balancing for heavy traffic Internet of Things," *Pervasive and Mobile Computing*, vol. 99, Apr. 2024, Art. no. 101891, https://doi.org/10.1016/j.pmcj.2024.101891.

[6] B. Mopuru and Y. Pachipala, "Advancing IoT Security: Integrative Machine Learning Models for Enhanced Intrusion Detection in Wireless Sensor Networks," *Engineering, Technology & Applied Science Research*, vol. 14, no. 4, pp. 14840–14847, Aug. 2024, https://doi.org/10.48084/etasr.7641.

[7] K. Ergun, R. Ayoub, P. Mercati, and T. Rosing, "Reinforcement learning based reliability-aware routing in IoT networks," *Ad Hoc Networks*, vol. 132, July 2022, Art. no. 102869, https://doi.org/10.1016/j.adhoc.2022.102869.

[8] A. Brandt *et al.*, "RPL: IPv6 Routing Protocol for Low-Power and Lossy Networks," Internet Engineering Task Force, Request for Comments RFC 6550, Mar. 2012. https://doi.org/10.17487/RFC6550.

[9] K. Pister, N. Dejean, and D. Barthel, "Routing Metrics Used for Path Calculation in Low-Power and Lossy Networks," Internet Engineering Task Force, Request for Comments RFC 6551, Mar. 2012. https://doi.org/10.17487/RFC6551.

[10] O. Gnawali and P. Levis, "The Minimum Rank with Hysteresis Objective Function," Internet Engineering Task Force, Request for Comments RFC 6719, Sept. 2012. https://doi.org/10.17487/RFC6719.

[11] A. Behal and G. Gupta, "Contiki based Anatomization of Routing Protocols for Low-Power and Lossy Networks In IoT," in *2022 International Conference on Futuristic Technologies*, Belgaum, India, 2022, pp. 1–5, https://doi.org/10.1109/INCOFT55651.2022.10094497.

[12] C. Carvalho *et al.*, "Entropy based routing for mobile, low power and lossy wireless sensors networks," *International Journal of Distributed Sensor Networks*, vol. 15, no. 7, July 2019, Art. no. 1550147719866134, https://doi.org/10.1177/1550147719866134.

[13] S.-W. Min, S.-H. Chung, H.-J. Lee, and Y.-V. Ha, "Downward traffic retransmission mechanism for improving reliability in RPL environment supporting mobility," *International Journal of Distributed Sensor Networks*, vol. 16, no. 1, Feb. 2020, Art. no. 1550147720903605, https://doi.org/10.1177/1550147720903605.

[14] A. Mohammadsalehi, B. Safaei, A. M. H. Monazzah, L. Bauer, J. Henkel, and A. Ejlali, "ARMOR: A Reliable and Mobility-Aware RPL for Mobile Internet of Things Infrastructures," *IEEE Internet of Things Journal*, vol. 9, no. 2, pp. 1503–1516, Jan. 2022, https://doi.org/10.1109/JIOT.2021.3088346.

[15] K. A. Darabkh, M. Al-Akhras, A. F. Khalifeh, I. F. Jafar, and F. Jubair, "An innovative RPL objective function for broad range of IoT domains utilizing fuzzy logic and multiple metrics," *Expert Systems with Applications*, vol. 205, Nov. 2022, Art. no. 117593, https://doi.org/10.1016/j.eswa.2022.117593.

[16] S. Sanshi and C. D. Jaidhar, "Enhanced mobility routing protocol for wireless sensor network," *Wireless Networks*, vol. 26, no. 1, pp. 333–347, Jan. 2020, https://doi.org/10.1007/s11276-018-1816-y.

[17] M. H. Homaei, S. S. Band, A. Pescape, and A. Mosavi, "DDSLA-RPL: Dynamic Decision System Based on Learning Automata in the RPL

Protocol for Achieving QoS," *IEEE Access*, vol. 9, pp. 63131–63148, 2021, https://doi.org/10.1109/ACCESS.2021.3075378.

[18] F. Kaviani and M. Soltanaghaei, "CQARPL: Congestion and QoS-aware RPL for IoT applications under heavy traffic," *The Journal of Supercomputing*, vol. 78, no. 14, pp. 16136–16166, Sept. 2022, https://doi.org/10.1007/s11227-022-04488-2.

[19] A. J. Ahmed *et al.*, "Congestion Aware Q-Learning (CAQL) in RPL Protocol – WSN based IoT Networks," in *2022 5th International Conference on Engineering Technology and its Applications*, Al-Najaf, Iraq, 2022, pp. 429–435, https://doi.org/10.1109/IICETA54559.2022.9888322.

[20] P. Chithaluru *et al.*, "An enhanced opportunistic rank-based parent node selection for sustainable & smart IoT networks," *Sustainable Energy Technologies and Assessments*, vol. 56, Mar. 2023, Art. no. 103079, https://doi.org/10.1016/j.seta.2023.103079.

[21] C. L. D. Santos, A. M. Mezher, J. P. A. León, J. C. Barrera, E. C. Guerra, and J. Meng, "Q-RPL: Q-Learning-Based Routing Protocol for Advanced Metering Infrastructure in Smart Grids," *Sensors*, vol. 24, no. 15, July 2024, Art. no. 4818, https://doi.org/10.3390/s24154818.

[22] N. Zahedy, B. Barekatain, and A. A. Quintana, "RI-RPL: a new high-quality RPL-based routing protocol using Q-learning algorithm," *The Journal of Supercomputing*, vol. 80, no. 6, pp. 7691–7749, Apr. 2024, https://doi.org/10.1007/s11227-023-05724-z.

[23] A. Musaddiq, R. Ali, J.-G. Choi, B.-S. Kim, and S.-W. Kim, "Collision Observation-Based Optimization of Low-Power and Lossy IoT Network Using Reinforcement Learning," *Computers, Materials & Continua*, vol. 67, no. 1, pp. 799–814, Jan. 2021, https://doi.org/10.32604/cmc.2021.014751.

[24] A. Musaddiq, R. Ali, S. W. Kim, and D.-S. Kim, "Learning-Based Resource Management for Low-Power and Lossy IoT Networks," *IEEE Internet of Things Journal*, vol. 9, no. 17, pp. 16006–16016, Sept. 2022, https://doi.org/10.1109/JIOT.2022.3152929.

[25] M. A. Alqarni and S. H. Chauhdary, "A Security Scheme for Statistical Anomaly Detection and the Mitigation of Rank Attacks in RPL Networks (IoT Environment)," *Engineering, Technology & Applied Science Research*, vol. 13, no. 6, pp. 12409–12414, Dec. 2023, https://doi.org/10.48084/etasr.6433.

[26] S. Othmen, W. Mansouri, and S. Asklany, "Robust and Secure Routing Protocol Based on Group Key Management for Internet of Things Systems," *Engineering, Technology & Applied Science Research*, vol. 14, no. 3, pp. 14402–14410, June 2024, https://doi.org/10.48084/etasr.7115.

[27] I. S. Sitanggang *et al.*, "Indonesian Forest and Land Fire Prevention Patrol System," *Fire*, vol. 5, no. 5, Sept. 2022, Art. no. 136, https://doi.org/10.3390/fire5050136.

[28] A. Mulyana, T. Djatna, S. Wahjuni, and H. Sukoco, "Precision tea picking ecosystem based on internet of things and edge computing: design and implementation," *International Journal of Information Technology*, Aug. 2025, https://doi.org/10.1007/s41870-025-02685-9.

[29] E. K. Adiyanto, S. Wahjuni, and H. Rahmawan, "Modification of Load Calculation in The Dijkstra Algorithm to Achieve High Throughput and Low Latency on 5G Networks," *Journal of Applied Engineering and Technological Science*, vol. 5, no. 2, pp. 1182–1198, June 2024, https://doi.org/10.37385/jaets.v5i2.4705.

[30] A. Musaddiq, Y. B. Zikria, Zulqarnain, and S. W. Kim, "Routing protocol for Low-Power and Lossy Networks for heterogeneous traffic network," *EURASIP Journal on Wireless Communications and Networking*, vol. 2020, no. 1, Jan. 2020, Art. no. 21, https://doi.org/10.1186/s13638-020-1645-4.

[31] R. A. R. Antayhua, M. D. Pereira, N. C. Fernandes, and F. R. de Sousa, "Exploiting the RSSI Long-Term Data of a WSN for the RF Channel Modeling in EPS Environments," *Sensors*, vol. 20, no. 11, May 2020, Art. no. 3076, https://doi.org/10.3390/s20113076.

[32] Zolertia. "Z1 datasheet Revision C." GitHub. https://github.com/Zolertia/Resources/blob/master/Z1/Hardware/Revision%20C/Datasheets/Zolertia%20Z1%20datasheet%20Revision%20C.pdf.

[33] N. Halloum, Y. Darmani, and A. Ahmadi, "Learning-Based Routing Policy for Wireless Sensor Networks," in *2024 32nd International Conference on Electrical Engineering*, Tehran, Iran, 2024, pp. 1–7, https://doi.org/10.1109/ICEE63041.2024.10668208.

[34] H. Hassani, M. Ghodsi, and G. Howell, "A note on standard deviation and standard error," *Teaching Mathematics and its Applications: An International Journal of the IMA*, vol. 29, no. 2, pp. 108–112, June 2010, https://doi.org/10.1093/teamat/hrq003.

[35] J. Kurose and K. Ross, *Computer Networking: A Top-Down Approach*, 7th ed. Boston, MA, USA: Pearson, 2017.

[36] W. Mardini, S. Aljawarneh, A. Al-Abdi, and H. Taamneh, "Performance evaluation of RPL objective functions for different sending intervals," in *2018 6th International Symposium on Digital Forensic and Security*, Antalya, Turkiye, 2018, pp. 1–6, https://doi.org/10.1109/ISDFS.2018.8355323.

[37] B. Triadi and P. Simanungkalit, "Monitoring Dan Upaya Mengendalikan Muka Air Pada Perkebunan Di Lahan Rawa Gambut Di Indonesia," *Jurnal Teknik Hidraulik*, vol. 9, no. 1, pp. 53–68, Sept. 2018, https://doi.org/10.32679/jth.v9i1.475.

[38] S. R. Lalani, B. Safaei, A. M. Hosseini Monazzah, and A. Ejlali, "PEARL: Power and Delay-Aware Learning-based Routing Policy for IoT Applications," in *2022 CPSSI 4th International Symposium on Real-Time and Embedded Systems and Technologies*, Tehran, Iran, 2022, pp. 1–8, https://doi.org/10.1109/RTEST56034.2022.9849862.

[39] J. A. C. Correa, S. B. S. Mora, B. M. Delgado, C. D. E. Amado, and D. G. Ibarra, "A forest fire monitoring and detection system based on wireless sensor networks," *Scientia et Technica*, vol. 27, no. 2, pp. 89–96, June 2022, https://doi.org/10.22517/23447214.24784.

[40] H. Farag and Č. Stefanovič, "Congestion-Aware Routing in Dynamic IoT Networks: A Reinforcement Learning Approach," in *2021 IEEE Global Communications Conference*, Madrid, Spain, 2021, pp. 1–6, https://doi.org/10.1109/GLOBECOM46510.2021.9685191.

[41] D. Z. Fawwaz and S.-H. Chung, "Adaptive Parent Change and Cell Usage Aware Objective Function for RPL in 6TiSCH Network," in *2024 International Conference on Information Networking*, Ho Chi Minh City, Vietnam, 2024, pp. 1–6, https://doi.org/10.1109/ICOIN59985.2024.10572170.