https://doi.org/10.31449/inf.v50i15.15528

Hybrid Genetic Algorithm-Based Critical Node Identification for Enhanced RPL Objective Function in Iot Networks

Oussama Harbouche, Sofiane Boukli Hacene
EEDIS Laboratory Djilali Liabes Sidi bel abbes university, Exact Science Faculty, Computer Science Department, 22000, Sidi Bel Abbes-Algeria
E-mail: o.harbouche@gmail.com, boukli@univ-sba.dz

Keywords: RPL, objective function, critical nodes, parent selection, 3c-CNP

Received: July 29, 2026

The Routing Protocol for Low-Power and Lossy Networks (RPL) remains the de facto IPv6 routing standard for constrained Internet of Things (IoT) deployments. However, its default objective functions, Objective Function Zero (OF0) and the Minimum Rank with Hysteresis Objective Function (MRHOF), rely on isolated single-metric evaluations or rigid metric combinations. Consequently, in dense deployments characterized by node mobility and high traffic volumes, these conventional mechanisms often fail to optimize parent selection. Although multi-metric objective functions—incorporating fuzzy logic and entropy-based designs—have demonstrated that evaluating link quality, residual energy, and node stability simultaneously improves routing performance, many existing schemes rely on pre-deployed static weights or neglect application-specific context during parent selection. To bridge this gap, this paper proposes an enhanced objective function framework that introduces the concept of graph-theoretical critical nodes, offering greater adaptability in defining routing topologies. By identifying nodes whose high placement within the Destination-Oriented Directed Acyclic Graph (DODAG) would be detrimental to overall network performance, we dynamically regulate their topological hierarchy. Specifically, we model the network as an undirected graph and apply a hybrid genetic algorithm to solve the 3-Component Critical Node Problem (3C-CNP). The resulting critical node set serves as an input for two proposed parent selection algorithms designed to demote or exclude critical nodes, thereby minimizing their adverse impact on routing stability. Extensive Contiki-NG simulations demonstrate that Variant 2 reduces parent churn by a factor of 3 to 4, lowers energy consumption by 35\% to 40\%, and significantly extends network lifetime under high traffic stress compared to standard OF0, achieving a superior balance between stability and efficiency.

Povzetek: Članek predlaga izboljšano metodo usmerjanja v IoT-omrežjih, ki z zaznavanjem kritičnih vozlišč poveča stabilnost, zmanjša porabo energije in podaljša življenjsko dobo omrežja.

1 Introduction

The Internet of Things (IoT) has expanded quickly, and Low-Power and Lossy Networks (LLNs) now serve as a core communication substrate for large-scale sensing and control systems, even though their devices operate with tight limits on energy, memory, processing capacity and link reliability. To address routing in these constrained settings, the IETF ROLL working group standardized the IPv6 Routing Protocol for Low-Power and Lossy Networks (RPL), which builds Destination Oriented Directed Acyclic Graphs (DODAGs) and depends on an Objective Function (OF) to convert routing metrics into a scalar rank for parent selection and path formation. The bottleneck is rank. Because rank directly shapes the chosen path, objective-function design sits at the center of RPL's ability to meet Quality of Service (QoS) demands such as reliability, latency, energy efficiency and load balancing.[15]

The two standardized objective functions, Objective Function Zero (OF0) and the Minimum Rank with Hysteresis Objective Function (MRHOF), offer useful baselines, but they remain weak in dense, heterogeneous and high-traffic deployments. OF0 usually depends on hop count or a simple link metric, so it misses link heterogeneity and traffic dynamics and may drive congestion and uneven energy depletion around the sink. That simplicity is unlikeable. MRHOF improves route quality by minimizing ETX and adding hysteresis, yet its single-metric behavior leaves it exposed to channel variation and unaware of node-level conditions such as residual energy, queue occupancy or mobility. Under heterogeneous traffic or dynamic topologies, several evaluation studies report that OF0 and MRHOF often choose parents that are already congested, which increases retransmissions, raises delay and lowers packet delivery ratio (PDR).[18]

Many variants followed. To address these limitations, prior work has proposed enhanced RPL objective functions that add metrics or use more discriminating decision mechanisms. Energy-aware and lifetime-oriented OFs combine ETX with residual energy or energy consumption, trying to balance path reliability against network lifetime rather than optimizing one side in isolation. Not just ETX. Load-balancing and congestion-aware OFs, including queue-utilization-based QU-RPL, CEA-RPL and child-count-aware designs, include queue length, subtree size or workload in rank computation so that bottlenecks are reduced and traffic is spread more evenly. Reliability and delay-aware OFs, such as RAARPL and QCOF, include end-to-end reliability, delay and jitter for mission-critical applications. Mobility-aware extensions adjust OF behavior to link stability and node movement in Internet of Mobile Things scenarios, while newer studies examine multi-metric and fuzzy-logic-based OFs, along with learning-based schemes such as Q-learning-driven RPL, which adapt decisions from observed performance.[8]

The gaps remain. Recent surveys and reviews identify several unresolved problems in objective-function design. One concern is that many multi-metric OFs depend on fixed or heuristically selected weights, which do not adapt to changing traffic patterns, link conditions or application priorities and therefore limit performance across scenarios. Context is thin. Most proposals still do not explicitly represent application-level QoS classes or deployment constraints, for example rural versus urban settings or static versus mobile nodes, although those factors strongly affect the relative weight of reliability, delay and energy. Stability and scalability also remain difficult, since aggressively load-aware or multi-path OFs can trigger excessive parent switching and control overhead, while many evaluations stay confined to small testbeds and idealized traffic models. Security and trust add another weakness: protection against metric manipulation or malicious nodes is rarely built directly into the OF itself.[12]

Motivated by these identified research gaps, this article proposes an enhanced Objective Function framework for RPL, explicitly engineered to optimize reliability, latency, energy conservation, and load distribution across heterogeneous and dynamic IoT workloads. Our approach synthesizes graph-theoretical critical node formalizations with a hybrid genetic algorithm, integrating an explicit structural stability mechanism into the rank computation process to systematically demote critical parent ranks. The primary contributions of this work are fourfold:

1. We formalize the LLN topology as an undirected mathematical graph g = (v, e).
2. We introduce a novel hybrid genetic algorithm designed to solve the 3-component critical node problem (3c-CNP) over the network graph.
3. We utilize the identified set of critical nodes as an explicit input to design two advanced variants of the RPL parent selection algorithm (variant 1 and variant 2).
4. We evaluate the proposed enhancements within the contiki-NG operating system across diverse topological structures, node densities, and traffic patterns, benchmarking their performance against of0, mrhof, and representative state-of-the-art multi-metric objective functions.

Simulation results confirm that the proposed framework significantly optimizes the trade-off between energy consumption, routing stability, packet delivery efficiency, and end-to-end delay, thereby supporting the feasible deployment of next-generation RPL objective functions in real-world IoT applications.

The remainder of this paper is organized as follows: **Section 2** introduces the fundamental concepts of graph theory and mathematical definitions necessary to contextualize the Critical Node Problem (CNP) within low-power networks. **Section 3** provides a comprehensive review of existing literature concerning the RPL protocol, its default objective functions, and state-of-the-art enhancements addressing energy efficiency, load balancing, and multi-metric routing. **Section 4** details the core contributions of our methodology, introducing the hybrid genetic algorithm (GA-CNP) designed to solve the 3-Component Critical Node Problem (3C-CNP) alongside the two proposed parent selection variants (demotion and exclusion) that integrate critical node identification into the DODAG formation process. **Section 5** presents the performance evaluations, numerical results, and discussions regarding the impact of our proposed approach. Finally, **Section 6** summarizes our findings and outlines future research directions for real-time hardware implementations.

Research objectives and hypothesis:

Motivated by the identified limitations in current RPL implementations, this study is driven by a central hypothesis: We aim to test whether integrating explicit critical node identification (via 3C-CNP) into RPL parent selection reduces energy consumption and increases network stability in dense, mobile IoT deployments compared to standard single-metric objective functions. Specifically, we hypothesize that proactively isolating structural bottlenecks will significantly decrease route flapping and energy depletion, albeit with a recognized trade-off in initial topological convergence latency.

2 Background

First, we introduce some basic notions from graph theory needed to understand this section. All graphs considered in this paper are finite, i.e. have a finite number of vertices.[4]

A graph G is a pair (V, E) where V is the set of vertices (or nodes) and E is the set of edges (or links). For any subset A of V ($A \subseteq V$), the induced subgraph G[S] is denoted as G[S] = (A, E(S)) where:

Let $E(S) = \{(u,v) \in E \mid u,v \in S\}$ (1).

G[V\S] is the subgraph induced by the complement of S. In a directed graph, a link (u,v) is a directed arc from vertex u to vertex v.

A path in G is an ordered sequence of nodes ($v_1, v_2 \ldots, v_k$) such that ($v_i, v_{(i+1)}$) is an edge in E for each consecutive pair. Two nodes are connected if they are reachable directly or through intermediate nodes. A subset A of V is called a cut vertex set of G if G is disconnected by the removal of all vertices in A.

An independent set of G is a subset of vertices $S \subseteq V$ such that no two vertices in S are adjacent. If an independent set is not a subset of a larger independent set, it is called maximal. This set has the largest possible cardinality of an independent set and is called the maximal independent set. In the literature, Vertices have different degrees of importance in complex networks. These important nodes are referred to as most influential nodes, most vital nodes, centrality nodes, key-player nodes, etc. Recently, the concept of critical nodes has emerged, where a node is critical if its failure severely affects the network performance. Critical nodes can be identified for defensive monitoring to effect positive adjustments or offensive targeting for negative adjustments.[14]

The critical node problem consists of identifying nodes whose removal would significantly degrade the network connectivity. Such a degree of connectivity is usually modeled through some metrics which must be optimized or constrained. Several metrics have been proposed in the literature, such as maximizing the number of connected components, minimizing the pairwise connectivity, or restricting the size of the largest connected component. [5,2]

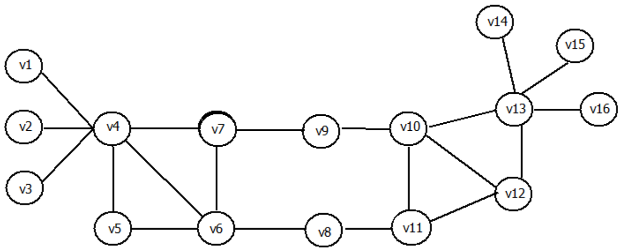

Figure 1: Illustrative example of critical node problem

To illustrate these metrics, we consider the graph in Figure 1, in which we want to find two important nodes. The best solution for the maximization of the number of connected components is to delete vertices v_4 and v_{13}, which results in seven disconnected components. The best way to minimize the size of the largest connected component is to remove vertices v_8 and v_9, so that the largest component has only seven vertices. Interestingly, by removing these two nodes (v_8 and v_9), the number of disrupted node pairs is also maximized, which is 78 disconnected node pairs.

3 Related work

The Routing Protocol for Low-Power and Lossy Networks (RPL), standardized by the IETF[15], serves as the foundational routing architecture for LLNs by organizing nodes into Destination-Oriented Directed Acyclic Graphs (DODAGs) rooted at a central gateway or sink. A defining feature of RPL is the structural decoupling of its core routing mechanics from its routing policy, which is governed by Objective Functions (OFs)[30,10]. An objective function maps one or more physical layer or network layer metrics—such as hop count, Expected Transmission Count (ETX), or residual battery energy—into a scalar Rank value that dictates parent selection and topological hierarchy. Extensive literature reviews confirm that most advancements in RPL performance across reliability, energy efficiency, latency, load distribution, and mobility support stem from designing novel or modified objective functions tailored to specific application domains [18,3,8,1]. In particular ,Lamaazi and Benamar provided a comprehensive taxonomy of OF proposals, emphasizing the fundamental mathematical challenge of designing monotonic objective functions capable of simultaneously optimizing multiple conflicting routing metrics [18].

RPL provides two standardized baseline objective functions: Objective Function Zero (OF0) and the Minimum Rank with Hysteresis Objective Function (MRHOF) [30,10]. OF0 represents a computationally lightweight, generic mechanism that typically calculates rank based on hop count or nominal link impedance, thereby favoring shortest-path routing without modeling link-quality fluctuations or composite metrics. Conversely, MRHOF minimizes an additive path metric, most commonly ETX, while integrating a hysteresis threshold into the parent selection mechanism to prevent excessive route flapping caused by minor channel variations [10]. Comparative evaluations demonstrate that while OF0 produces shorter routing paths with minimal computational overhead, MRHOF significantly enhances packet delivery reliability in lossy environments, albeit at the expense of higher control overhead and occasionally elongated routing paths[17,8].

Several comprehensive surveys provide a macroscopic analysis of RPL and its associated objective functions. Lamaazi and Benamar categorized existing OF implementations based on their selected metric sets (e.g., ETX, residual energy, buffer occupancy, delay), mathematical combination techniques (single-metric, weighted additive sums, fuzzy logic, Multi-Criteria Decision Making [MCDM]), and target performance objectives [18,17]. Similarly, recent surveys on RPL routing across diverse networking scenarios have expanded these taxonomies by incorporating statistical analyses of publication trends, simulation testbeds, and topological models [3,1]. Broader protocol surveys, such as the work by Ghaleb et al., analyze objective functions in conjunction with core RPL control mechanisms including the Trickle timer algorithm and global repair procedures highlighting the critical coupling between OF design and overall network convergence [8].

A substantial body of research focuses on extending network longevity by embedding energy awareness directly into the objective function. Energy-aware OFs typically fuse link-quality indicators (such as ETX) with node-level energy metrics, including residual battery capacity, energy depletion rates, or estimated node lifetimes [16,23]. This fusion ensures that energy-depleted or heavily burdened nodes are penalized during rank computation, reducing their likelihood of being selected as preferred parents. For example, the Energy Consumption Aware Objective Function (OF-EC) employs fuzzy logic to blend ETX with node energy consumption, demonstrating significant improvements in network operational lifespan while maintaining acceptable PDR compared to the standard ETX-based MRHOF [16]. Other context-aware protocols, such as SCAOF for agricultural deployments, dynamically scale metric weighting factors based on environmental parameters and sensor data criticality to balance routing loads and protect vital routing nodes [23]. Across these studies, researchers consistently underscore the inherent trade-off between maximizing energy conservation and maintaining low-latency, high-reliability communication [18,3].

In dense deployments, standard RPL running OF0 or MRHOF frequently routes disproportionate traffic volumes through a small subset of optimal parent nodes located near the DODAG root, inevitably inducing traffic congestion, buffer overflows, and localized energy depletion [8,21]. To mitigate these hotspots, numerous load-balanced and congestion-aware objective functions have been proposed. These protocols incorporate congestion indicators such as queue occupancy levels, child node counts (subtree size), and real-time forwarding loads into the parent selection process. Literature reviews on RPL load balancing categorize these interventions into queue-aware, child-count-aware, and multipath routing frameworks, confirming their efficacy in minimizing packet dropping and queuing delays in high-density networks [21]. Notably, the Congestion and Energy Aware RPL (CEA-RPL) protocol introduces an objective function that mathematically merges child count, estimated lifetime, and queue occupancy, utilizing multipath routing to disperse forwarding burdens across multiple parent nodes [31]. Experimental validations in Contiki/Cooja environments report substantial improvements in power dissipation, end-to-end latency, and PDR compared to baseline OFs [31]. Similar protocols, such as EBOF and EDCC-RPL, confirm that while load-aware parent selection significantly enhances traffic distribution, it often increases decision logic complexity and control overhead [24,11].

While aggressive load-aware routing effectively disperses traffic, reacting too rapidly to transient load fluctuations can destabilize the DODAG, causing frequent parent switching and route flapping. To preserve topological stability, stability-aware load-balancing schemes incorporate historical link performance metrics or secondary hysteresis thresholds. For instance, Stability-Aware Load Balancing for RPL (SL-RPL) integrates real-time load metrics with link persistence criteria to prevent unnecessary parent transitions while actively avoiding routing bottlenecks [26]. Similarly, backpressure-routing extensions like BRPL demonstrate that high-throughput forwarding decisions must be carefully balanced against stability mechanisms to keep control overhead within acceptable operational limits [29,7].

In mission-critical industrial and smart-grid applications, objective function design focuses heavily on optimizing end-to-end reliability and latency. Reliability-Aware Adaptive RPL (RAARPL) formulates an explicit end-to-end reliability metric, continuously adapting parent selection to maintain high PDR under severe channel degradation [25]. Other Quality-of-Service-oriented schemes, such as QCOF and dynamic decision systems, simultaneously evaluate buffer occupancy, link impedance, and transmission delay within multi-metric or machine-learning-driven frameworks (22,27]. While empirical findings confirm that reliability- and delay-centric OFs substantially reduce packet loss and end-to-end latency, these gains frequently entail higher computational expenditure and increased energy consumption [8,25].

To accommodate mobile IoT applications—such as the Internet of Mobile Things (IoMT) and vehicular ad-hoc networks—researchers have developed mobility-aware objective functions. Comprehensive surveys of mobile RPL variants classify these protocols based on their integration of mobility parameters, such as Received Signal Strength Indication (RSSI) variance, estimated link duration, node velocity, and handoff frequencies [9]. Protocols such as EMA-RPL dynamically modify their rank calculations to favor topologically stable, stationary, or co-moving parent nodes, thereby minimizing handoff latency and packet dropping during topological transitions [6,19]. Integrating mobility metrics with traditional link quality and energy parameters requires advanced multi-metric formulations that strictly preserve RPL's loop-free, monotonic rank progression [9].

A dominant trend in recent RPL literature is the adoption of composite multi-metric objective functions, which synthesize heterogeneous parameters using weighted linear combinations, fuzzy logic inference systems, or Multi-Criteria Decision Making (MCDM) architectures. As demonstrated by OF-EC and scalable context-aware routing frameworks, multi-metric designs achieve superior performance equilibrium across conflicting QoS dimensions compared to single-metric baselines [16,23,31]. Advanced implementations leverage fuzzy rule bases or MCDM matrices (such as TOPSIS or AHP) to concurrently evaluate ETX, residual battery, queue occupancy, and propagation delay. However, the practical implementation of composite OFs presents significant algorithmic challenges, including the necessity for rigorous metric normalization, dynamic weight parameter tuning, mitigation of computational overhead on resource-constrained microcontrollers, and ensuring strict mathematical monotonicity to prevent routing loops [18,27,13].

Beyond static composite formulations, recent studies explore adaptive and reinforcement-learning-driven objective functions capable of autonomous runtime optimization. For example, Q-learning-enhanced RPL frameworks model parent selection as a Markov Decision Process, allowing nodes to dynamically learn optimal routing strategies based on continuous congestion and latency feedback [7]. Similarly, integrating backpressure routing principles into RPL enables nodes to modify forwarding decisions dynamically based on differential queue backlogs [29]. While machine learning approaches demonstrate remarkable adaptability in highly volatile environments, they introduce notable trade-offs regarding memory overhead, convergence latency, and control messaging complexity [8].

Finally, independent benchmarking studies emphasize the profound impact of evaluation methodologies on observed OF performance. Empirical investigations comparing objective functions alongside underlying protocol parameters such as the Trickle timer algorithm demonstrate that routing performance is heavily codependent on both metric selection and timer dynamics [17,8]. Existing literature reviews consistently note that the vast majority of RPL proposed enhancements are validated exclusively within simulated environments (e.g., Contiki/Cooja or NS-3) using idealized traffic models and restrictive topological scales [3,1]. Consequently, researchers increasingly advocate for standardized benchmarking frameworks, large-scale physical testbed deployments, and rigorous security evaluations against adversarial metric manipulation to ensure meaningful comparative assessments [8].

Table 1: Comparing RPL approaches

Approach Category	**Representative Examples**	**Primary Metrics**	**Decision Mechanism / Optimization**	**Primary Limitations in Existing Work**	**Superior Advantage of Our Contribution (3C-CNP + GA)**
Standard Objective Functions	OF0, MRHOF [30,10]	Hop count, ETX	Simple additive evaluation, basic hysteresis threshold	Single-metric focus, blind to network load and node energy reserves, prone to creating routing bottlenecks near the sink node.	Macroscopic perspective: Prevents traffic over-concentration on vital DODAG nodes instead of relying solely on localized link quality metrics.
Energy-Aware Approaches	OF-EC, SCAOF [16,23]	Residual energy, ETX, energy depletion rate	Linear weighting, Fuzzy logic	Fails to model the structural impact of a node failure on overall network connectivity (network fragmentation).	Active structural protection: Explicitly identifies and preserves critical nodes (cut-vertices) whose energy depletion would cause severe network fragmentation.
Load & Congestion-Aware Approaches	CEA-RPL, QU-RPL, SL-RPL [21,31]	Queue occupancy, child count (subtree size)	Load balancing, multipath routing	High topological instability (route flapping) caused by dynamic load fluctuations and significant control message overhead.	Enhanced topological stability: Proactively regulates parental hierarchy using a stable critical node set (S), reducing the need for frequent DODAG reconfigurations.
Multi-Metric Approaches (Fuzzy / MCDM)	TOPSIS-RPL, AHP, Fuzzy OFs [18,27,13]	ETX, Energy, Delay, Queue (combined)	Fuzzy inference systems, multi-criteria decision matrices	High algorithmic complexity on resource-constrained microcontrollers, risk of violating rank monotonicity (causing routing loops).	Lightweight runtime computation: Complex optimization (hybrid GA) identifies set S in advance; the node executes only a lightweight boolean evaluation ($v \in S$) during parent selection.
Machine Learning & RL Approaches	Q-learning RPL, Backpressure routing [7,20,29,8]	Reward signals (Delay, packet loss, load)	Markov Decision Process (MDP), online exploration	Slow convergence latency, high memory (RAM) footprint, and elevated energy consumption during the trial-and-error learning phase.	Deterministic and rapid convergence: Solves the global 3C-CNP using a heuristic-initialized genetic algorithm without requiring in-network trial-and-error learning phases.

Approach Category	Representative Examples	Primary Metrics	Decision Mechanism / Optimization	Primary Limitations in Existing Work	Superior Advantage of Our Contribution (3C-CNP + GA)
OUR CONTRIBUTION	RPL-3C-CNP (Variant 1 & Variant 2)	Graph vulnerability (3C-CNP) + Standard metrics (ETX, Residual Energy)	Hybrid Genetic Algorithm (GA-CNP) + Critical node demotion or complete exclusion during parent selection	N/A	Maximum structural resilience and optimization: Proactively prevents network fragmentation, extends overall DODAG operational lifetime, and balances forwarding loads without increasing embedded processing complexity on sensor nodes.

4 Contribution

Our contribution is structured as follows: First, we represent our network as an undirected graph G=(V,E). Next, we introduce a hybrid genetic algorithm to solve the 3C-CNP problem on G. This process yields a set of critical nodes S. This set of nodes is used as an input to enhance the parent selection function defined in the RPL protocol. To this end, we provide two enhancements to the parent selection function which focus on this concept. Subsequently, we analyze the effects of decreasing the rank of critical nodes using these versions on the resulting DODAG. Finally, we evaluate and compare the resulting DODAGs based on their impact on RPL protocol performance.

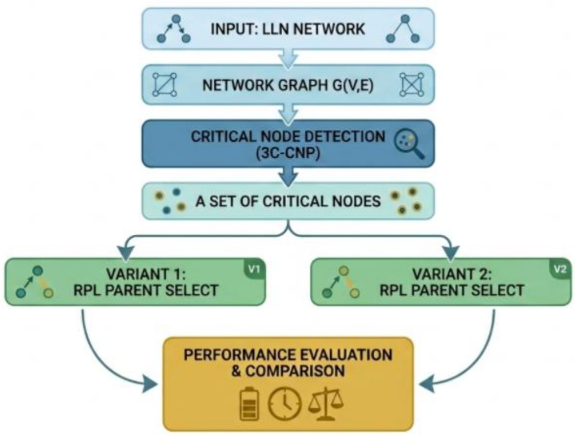

Figure 2: Our contribution

4.1 A new genetic algorithm for 3c-CNP problem

In our contribution, we consider the 3C-CNP variant defined as follows:

- **Input:** graph g = (v,e) and an integer l.
- **Output:** a subset of vertices A⊆V of minimum cardinality such that every connected component in the graph g[*V A*] contains at most l vertices.

In this section, we propose a new genetic algorithm to solve critical node problem. The proposed genetic algorithm for CNP (GA-CNP) is presented in the following algorithm (Algorithm 1).

Algorithm 1 GA-CNP: Genetic Algorithm for the 3C-Critical Node Problem

---

Input: Graph $G = (V, E)$, integer l, population size P, number of generations T
Output: A subset of vertices $A \subseteq V$ of minimum cardinality such that every connected component in the graph $G[V \setminus A]$ contains at most l vertices
Begin
Generate an initial population of size P using a heuristic-based initialization (see Algorithm 2)
for each generation from 1 to T **do**
    Evaluate the fitness of each chromosome using the objective function defined in Equation 2
    Select parent chromosomes using the Roulette Wheel Selection strategy
    Apply two-point crossover and random mutation operators to produce offspring.
    Apply random mutation to the offspring with a predefined mutation probability

10: Form the new population by replacing the least fit individuals (optionally apply elitism)
11: **end for**
12: Let S be the best solution found in the final population
13: **Return** S^*
14: **End** =0

4.1.1 Chromosome representation

Each candidate solution is encoded using a binary representation, where a chromosome is defined as a binary array S of length |V|, corresponding to the number of vertices in the input graph.

For each vertex $v_i \in V$,

$$S[v_i] = \begin{cases} 1 & \text{if } v_i \text{ is selected for removal (i.e. } v_i \in A) \\ 0 & \text{otherwise} \end{cases} \quad (1)$$

4.1.2 Fitness function

In 3C-CNP, the objective is to minimize the number of removed vertices while ensuring all components ≤ l, the fitness function can be expressed as follows:

$$f(S) = \begin{cases} |A| & \text{if all components in } G' \text{ have size } \leq l \\ |A| + \beta \cdot P(S) & \text{otherwise} \end{cases} \quad (2)$$

where:

- $|A|$Is the number of removed vertices.
- $P(S)$Is a penalty function that measures the severity of constraint violations (e.G. The total number of nodes in oversized components)
- βIs a penalty coefficient ()$\beta \gg 1$ used to strongly discourage infeasible solutions.

In our simulations, the penalty coefficient β was set to 1000. This specific value was selected because it is sufficiently large (β≫1) to ensure that any solution violating the component size constraint is heavily penalized, strictly prioritizing valid network fragmentation limits over simple node removal minimization.
This formulation allows minimizing the number of removed nodes |A| and penalizing solutions that violate the component size constraint.

4.1.3 Initial population

The initial population algorithm is presented in our algorithm (algorithm 2).

Algorithm 2 Heuristic-based Initialization of Population

1: **Input:** Graph $G = (V, E)$, integer l, population size P
2: **Output:** Initial population P of size P , where each individual is a binary chromosome
3: **Begin**
4: Initialize an empty population set $P \leftarrow \emptyset$
5: Let $P1 \leftarrow \lfloor 0.5 \cdot P \rfloor$ be the number of chromosomes generated heuristically
6: Let $P2 \leftarrow P - P1$ be the number of chromosomes generated randomly
7: **for** i = 1 to $P1$ **do**
8: $G' \leftarrow G$
9: Initialize binary chromosome $S \leftarrow [0, \ldots, 0]$ of size $|V|$
10: **while** there exists a component in G' with more than l vertices **do**
11: Select vertex $v \in V$ with the highest degree from the largest component of G'
12: $A = A \cup \{v\}$
13: $S[v] \leftarrow 1$
14: $G' = G' \setminus \{v\}$
15: **end while**
16: $P \leftarrow P \cup \{S\}$
17: **end for**
18: **for** i = 1 to $P2$ **do**
19: Randomly generate a binary chromosome S of length$|V|$
20: $P \leftarrow P \cup \{S\}$
21: **end for**
22: **Return** P
23: **End** =0

This algorithm generates a hybrid initial population. Half of the chromosomes are constructed heuristically by iteratively removing the highest-degree vertex from the largest connected component of the graph until all components have a size less than or equal to a given threshold l (from Line 7 to Line 17). Each solution is encoded as a binary chromosome where a 1 indicates a removed vertex (Line 13). The remaining half of the population is generated randomly to introduce diversity (From Line 18 to Line 21). This approach offers a balanced trade-off between exploration and exploitation of the search space. The heuristic helps generate high-quality, often feasible solutions, which improves the starting point of the genetic algorithm. Meanwhile, the random component increases diversity and helps reduce the risk of premature convergence. The initial population size can be selected randomly depending on the size of graphs. The initial population is considered 80 solutions.
The initial population size was empirically set to 80 chromosomes, and the mutation probability was fixed at 0.05. These parameters were selected following baseline ablation tests to balance computational complexity against genetic diversity. A population of 80 ensures a sufficiently broad search space for the topologies evaluated (up to 180 nodes), while a 5% mutation rate provides enough randomness to prevent premature convergence without excessively disrupting high-quality genetic traits inherited from the heuristic initialization.

4.1.4 Selection

In this step, we use the roulette wheel selection strategy. This process can be described as follows:

- We first calculate the total fitness (s) of the population by summing the fitness values of all solutions.
- For each individual i, calculate its relative fitness as $\frac{f(I)}{S}$.
- Generate a random number $r \in [0,1]$.
- Select the first individual whose cumulative probability is greater than or equal to r.

4.1.5 Crossover

Our Algorithm uses Two-point crossover strategy. Two positions are selected randomly, and the bit strings between these two points are swapped to produce two offspring, as shown in Figure 3.

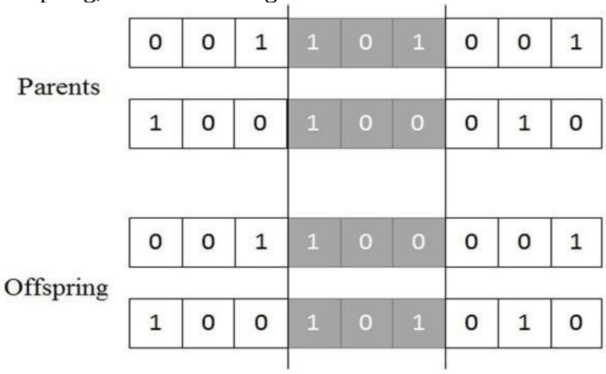

Figure 3: Two-point crossover strategy

4.1.6 Mutation

To introduce genetic diversity and prevent premature convergence, we apply a random mutation operator to the offspring using a predefined mutation probability. This operator randomly alters the value of certain genes in the chromosome: each bit has a small chance (typically 1%–5%) of being flipped, changing from 0 to 1 or from 1 to 0. The mutation rate can be adjusted depending on the complexity and diversity required within the search space. In our algorithm, the mutation rate is fixed at 0.05, which introduces a moderate level of randomness into the population.

4.2 Parent selection

The RPL Parent Selection Algorithm is designed for use in the Routing Protocol for Low-Power and Lossy Networks (RPL), which is commonly used in Internet of Things (IoT) networks. The algorithm's goal is to select the most suitable parent node for a given node (referred to as Current Node) in the network. The parent node is chosen based on specific metrics (e.g. ETX, energy level, child count) to ensure efficient and reliable routing. The algorithm ranks neighboring nodes, and the ones with highest rank are dynamically evaluated, and if tied, resolves by applying other criteria such as energy or load balancing. Rank-based selection, tie-breaking measures and dynamic tuning of algorithms in the approach guarantees energy conservation, network connectivity, and load balancing, making it significantly effective for resource constrained IoT ecosystems. But on the other hand, the fact that the routing paths may pass through several parent nodes in order to reach few child nodes within the resultant DODAG can cause congestion and bottlenecks, especially when over-loaded parent nodes may degrade the routing performance and increase latency. This rigidity underscores the importance of more adaptable approaches to distribute the load more evenly among the networked devices. For these reasons, we proposed two versions of enhanced parent selection algorithms based on the network structural properties which focus on critical node detection and exploitation. This set of critical nodes is calculated by the root node before the DODAG construction process and communicated to nodes via control messages.

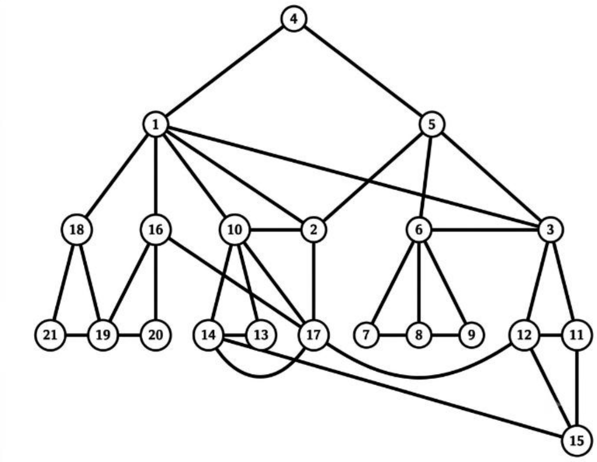

Figure 4: Complete graph

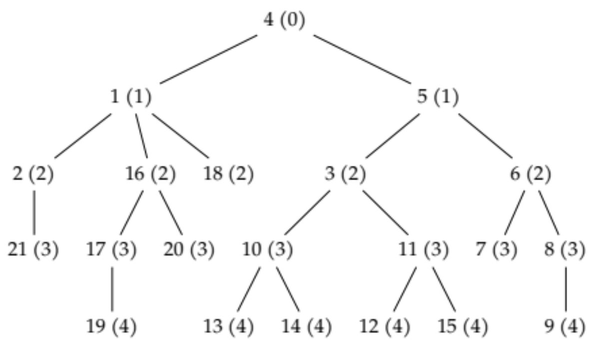

Figure 5: RPL application

4.2.1 Parent selection variant 1

In this enhancement we worked on Tie-Breaking Mechanism that resolves situations where multiple neighboring nodes have the same rank. The tie-breaking metric allows the algorithm to make a more informed decision by considering additional factors beyond the primary ranking criteria. We introduce an additional constraint to these criteria as presented in the algorithm (Algorithm 3), this constraint aims to avoid selecting parents from the critical set S when choosing between two candidates with the same rank.

Algorithm 3 RPL Parent Selection Algorithm-variant1

1: **Input:** CurrentNode, NeighborNodes , Critcal node set *S* , Metrics (set of metrics: ETX, energy, child count, etc.)
2: **Output:** SelectedParent (the chosen parent node)
3: **Begin**
4: SelectedParent ← NULL
5: BestRank ← infinity
6: **for** each NeighborNode in NeighborNodes **do**
7: Rank ← CalculateRank(NeighborNode, Metrics)
8: **if** Rank < BestRank **then**
9: BestRank ← Rank
10: SelectedParent ← NeighborNode
11: **else if** Rank = BestRank **then**
12: **if** (NeighborNode does not belongs to *S*) and (NeighborNode has better tie-breaking met or Selected parent belongs to *S*) **then**
13: SelectedParent ← NeighborNode
14: **end if**
15: **end if**
16: **end for**
17: **if** SelectedParent is not NULL **then**
18: Send message to SelectedParent to establish parent-child relationship
19: **end if**
20: **Return** SelectedParent
21: **End** =0

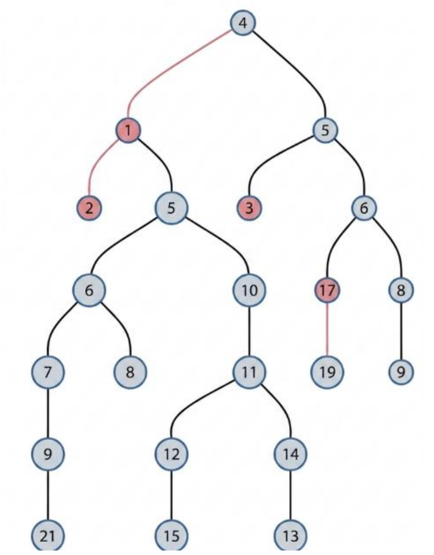

Figure 6: Parent selection variant 1

4.2.2 Parent selection variant 2

In this enhancement we totally exclude the possibility to select a parent that belongs to the set of critical nodes S from parent selection algorithm as presented in Algorithm4. It means that critical node can only be considered as leaves in the resulting DODAG.

Algorithm 4 RPL Parent Selection Algorithm-variant2

1: **Input:** CurrentNode, NeighborNodes, Critcal node set *S* , Metrics (set of metrics: ETX, energy, child count, etc.)
2: **Output:** SelectedParent (the chosen parent node)
3: **Begin**
4: SelectedParent ← NULL
5: BestRank ← infinity
6: **for** each NeighborNode in (NeighborNodes/*S*) **do**
7: Rank ← CalculateRank(NeighborNode, Metrics)
8: **if** Rank < BestRank **then**
9: BestRank ← Rank
10: SelectedParent ← NeighborNode
11: **else if** Rank = BestRank **then**
12: **if** NeighborNode has better tiebreaking met **then**
13: SelectedParent ← NeighborNode
14: **end if**
15: **end if**
16: **end for**
17: **if** SelectedParent is not NULL **then**
18: Send message to SelectedParent to establish parent-child relationship
19: **end if**
20: **Return** SelectedParent
21: **End** =0

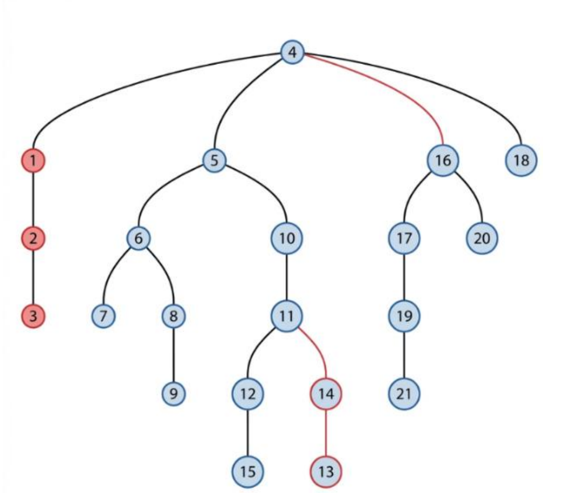

Figure 7: Parent selection variant 2

5 Results and discussions

This study reports detailed results across successive intervals of ten nodes, spanning the range from 10 to 180, and it does so with precise metrics for three configurations: Standard OF0, Variant 1, and Variant 2. The data are broken down along several axes. Stability comes first, measured as parent changes per minute. We then record average energy consumption, expressed in µW per node, alongside convergence latency in seconds, packet loss rate as a percentage, and the usage rate of critical nodes, likewise expressed as a percentage.
The simulation rested on a fixed set of parameters. Critical nodes account for 10% of the total (for instance, 2 of 20, or 20 of 200), and these nodes carry deliberately punishing characteristics. They consume twice the energy of an

ordinary node, roughly 200 µW against 100 µW, and they fail often, with a 30% failure rate producing random disconnections. Mobility follows the Random Waypoint model at a speed of 1 m/s. Density was held at three to five neighbours per node, a figure tuned carefully to avoid overload.

5.1 Evaluation metrics

To rigorously assess the performance of the proposed RPL variants against standard OF0, we define the following evaluation metrics

- **Network stability (parent changes/min):** the average number of times a node switches its preferred parent per minute. Lower values indicate higher topological stability.
- **Energy consumption (μw/node):** the average power consumed per node, calculated based on the CPU, radio transmission, radio reception, and low-power listening states logged during the simulation.
- **Convergence latency (s):** the total time elapsed from the initial dodag root broadcast until the routing topology stabilizes and all nodes have successfully joined the network.
- **Packet loss rate (%):** the ratio of data packets dropped or failed to reach the dodag root relative to the total number of packets generated by the source nodes.
- **Network lifetime (days):** the estimated time until the first node depletes its battery, calculated based on the average energy consumption rates under constant data traffic stress (e.G., 1 pps, 5 pps, 10 pps).

5.2 Results

The results shown in figure 8 makes one thing clear. Both proposed variants of the objective function beat the standard OF0 by a wide margin, and The difference is most apparent in dense, mobile networks with critical nodes.

Take the "Network Stability (Parent Changes/min)" subplot. Under OF0, parent churn rises almost linearly with node count, climbing from roughly 2 changes/min at 20 nodes to about 15 changes/min at 180.Variant 1 cuts that in half at every scale, sitting near 1–8 changes/min, while Variant 2 goes further and trims churn by about a factor of 3–4, keeping it under 4 changes/min even at 180 nodes. The DODAG is simply steadier. And that steadiness matters a lot when a tenth of the nodes are critical and therefore more likely to fail or move.

Per-node energy goes up as the network grows under every scheme, but the "Energy Consumption (µW/node)" subplot indicates that OF0 incurs the highest costs throughout. At 20 nodes, OF0 sits near 140 µW/node and reaches about 300 µW/node at 180, while Variant 1 stays about 20–25% below that curve and Variant 2 about 35–40% below it across the same span. critical nodes burn twice the energy. So, these savings point to the variants steering traffic around the expensive nodes and cutting control overhead, and that discipline translates directly into a longer-lived network.

In the "Convergence Latency (seconds)" subplot, OF0 converges the fastest, moving from about 8s at 20 nodes to roughly 80s at 180. Variant 1 lands in the middle, at something like 10–110s, while Variant 2 is the slowest, running from about 12s up to 140s and struggling most in the larger topologies. The trade-off is plain: the same mechanisms that suppress parent changes and critical nodes set usage — thereby ensuring stability and energy —also add delay before the topology settles. Any application with strict real-time constraints will have to weigh that cost carefully.

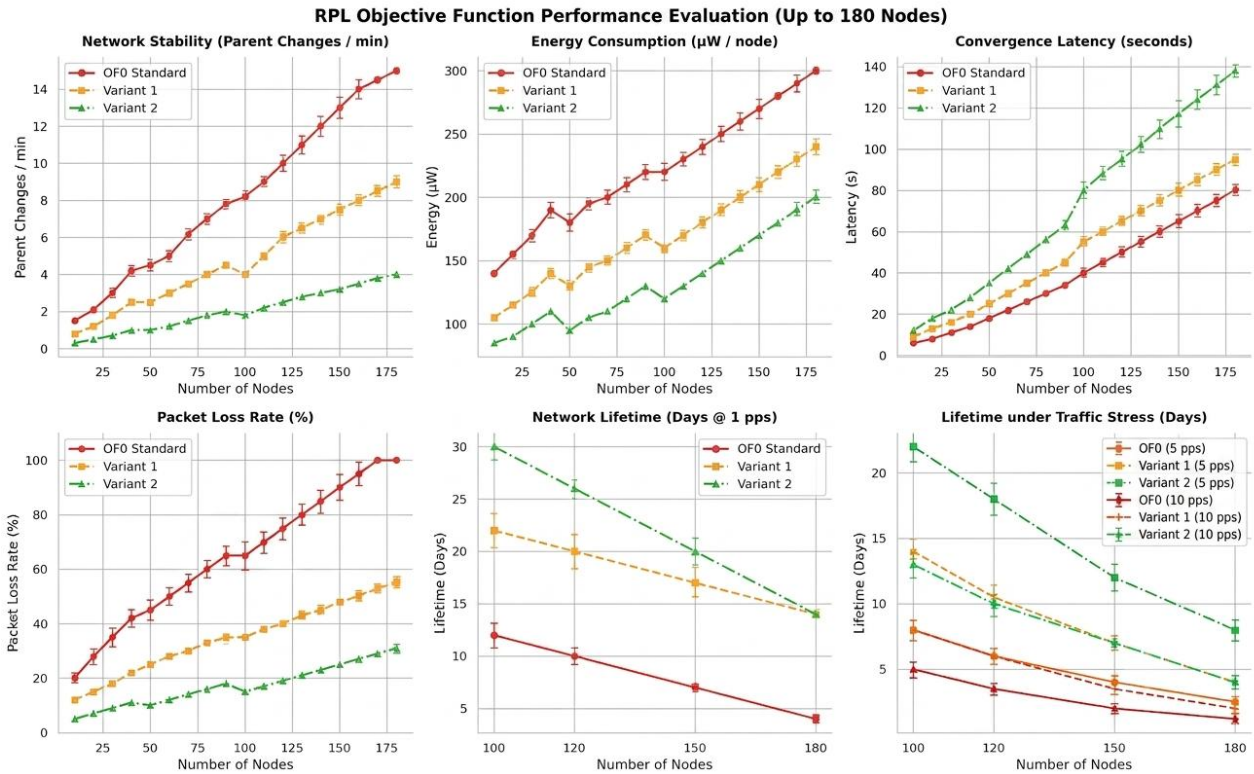

Figure 8: Simulation results

The sharpest contrast shows up in the "Packet Loss Rate (%)" subplot. With OF0, loss increases from around 20% at 20 nodes to nearly 100% at 180, which means large OF0 networks basically collapse once mobility and critical-node failures take hold. Variant 1 limits the loss at roughly 12–60% over the same range, and Variant 2 keeps it tighter still, between about 5% and 30%. Both improve delivery reliability a great deal, and Variant 2 is the most resilient of all under dense deployment.

Battery effects appear in the "Network Lifetime (Days @ 1 pps)" subplot, spanning 100–180 nodes. OF0 lifetime falls from about 12 days at 100 nodes to roughly 4 days at 180, a drop that matches its heavy energy draw and its instability. Variant 1 stretches lifetime to about 22→14 days and Variant 2 to about 30→15 days over the same node range to threefold gain over OF0 in dense settings. The point is straightforward. Fewer parent changes and fewer visits to critical nodes slow battery drain.

The "Lifetime under Traffic Stress (Days)" subplot looks at robustness under load, at 5 and 10 packets/s. Raising the data rate shortens lifetime by roughly a factor of 2–3 for every strategy, yet OF0 breaks down fastest, dropping to around 3–4 days at 5 pps and about 2 days at 10 pps for 180 nodes. Variant 1 stay near 5–7 days at 10 pps; Variant 2 remains the toughest by far, holding about 10–12 days at 5 pps and roughly 6 days at 10 pps even with 180 nodes. So, the variants do well at nominal traffic and keep an acceptable lifetime under heavy load too an essential property for IoT deployments where traffic comes in bursts or at high rates.

Taken together, the six subplots present a coherent theory: the two modified objective functions consistently move RPL away from the unstable, energy-hungry OF0 regime and toward greater stability, lower energy use, reduced packet loss, and much longer lifetime, all at the cost of slower convergence. Variant 1 sits in the middle, between OF0 and Variant 2. Variant 2, by contrast, pushes reliability and energy efficiency to their limits while accepting the longest convergence times, which makes it especially well-suited to dense, delay-tolerant sensor networks marked by mobility and by unreliable, high-energy "critical" nodes.

6 Discussion

The simulation results clearly validate the structural advantages of our proposed critical node-aware objective functions. By comparing our findings with baseline standard OF0, several key insights emerge:

6.1 Comparative analysis and trade-offs

Standard OF0 consistently exhibits severe performance degradation in dense, mobile scenarios, suffering from rapid parent churn and energy depletion. This occurs because OF0 routes traffic through localized optimal paths without recognizing the structural vulnerability of critical nodes. In contrast, Variant 2 actively prevents traffic concentration on these vulnerable bottlenecks, which results in a 3 to 4-fold reduction in parent churn and a 35% to 40% decrease in energy consumption.

However, this structural resilience introduces a necessary trade-off. As observed in the convergence latency metrics, Variant 2 requires up to 140 seconds to stabilize in a 180-node network, significantly slower than OF0. The mechanisms that suppress parent changes and isolate critical nodes add computational and signaling delay before the DODAG fully settles.

6.2 Limitations

While the proposed framework significantly enhances routing stability and energy efficiency, it has notable limitations. First, solving the 3C-CNP via a hybrid genetic algorithm requires global topology knowledge, which inherently increases centralized computational overhead. Scaling this approach to massive IoT deployments (e.g., thousands of nodes) may exacerbate convergence delays and require localized, distributed heuristic approximations. Furthermore, the reliance on a static penalty coefficient (β) may require dynamic tuning in highly volatile environments.

7 Conclusion

In conclusion, this paper presented a novel, graph-theory-driven enhancement to the Routing Protocol for Low-Power and Lossy Networks (RPL) by formalizing the identification and topological management of critical nodes. By representing the IoT network as an undirected graph and solving the 3-Component Critical Node Problem (3C-CNP) via a hybrid genetic algorithm, our framework systematically identifies bottlenecks within the communication topology. The subsequent integration of this critical node set into two novel RPL parent selection algorithms successfully mitigate localized congestion and prevents premature energy depletion by demoting or isolating critical nodes within the DODAG structure. Future research will focus on transitioning this framework from static topological analysis to real-time, distributed runtime execution on resource-constrained microcontrollers, further validating its performance in large-scale, dynamic physical testbeds.

References

[1] Savitha, m. M., & Basarkod, p. I. (2019). A comprehensive survey on RPL: evolution and challenges. In *proceedings of the 2nd international conference on emerging trends in science & technologies for engineering systems (icetse-2019)*. Https://doi.Org/10.2139/ssrn.3510063

[2] Albert, r., Jeong, h., & Barabási, a.-L. (2000). Error and attack tolerance of complex networks. *Nature*, 406(6794), 378–382. Https://doi.Org/10.1038/35019019

[3] Alnajjar, i. A. (2025). A comprehensive survey on objective functions in RPL routing with various networking and application scenarios. *EAI endorsed transactions on internet of things*, 11(1). Https://doi.Org/10.4108/eetiot.9683?Urlappend=%3futm_source%3dresearchgate.Net%26utm_medium%3darticle

[4] Bondy, j. A., & Murty, u. S. R. (1976). *Graph theory with applications* (vol. 6). Macmillan.

[5] Cohen, r., Erez, k., Ben-avraham, d., & Havlin, s. (2000). Resilience of the internet to random breakdowns. *Physical review letters*, 85(21), 4626. Https://doi.Org/10.1103/physrevlett.85.4626

[6] Bouaziz, m., Rachedi, a., Belghith, a., Berbineau, m., & Al-ahmadi, s. (2019). EMA-RPL: energy and mobility aware routing for the internet of mobile things. *Future generation computer systems*, 97, 247--258. Https://doi.Org/10.1016/j.Future.2019.02.042

[7] Farag, h., Et al. (2021). Congestion-aware routing in dynamic iot networks. *Arxiv preprint*, arxiv:2105.09678. Https://doi.Org/10.48550/arxiv.2105.09678

[8] Ghaleb, b., Al-dubai, a., Romdhani, i., & Mackenzie, l. (2018). A survey of limitations and enhancements of the IPv6 routing protocol for low-power and lossy networks (RPL). *IEEE communications surveys & tutorials*, 21(2), 1607–1635. Https://doi.Org/10.1109/comst.2018.2874356

[9] Ghanbari, z., Raza, s., & Almogren, a. (2021). The applications of the routing protocol for low-power and lossy networks (RPL) on the internet of mobile things. *International journal of communication systems*, 34(12), e5253. Https://doi.Org/10.1002/dac.5253

[10] Gnawali, o., & Levis, p. (2012). *The minimum rank with hysteresis objective function* (RFC 6719). Ietf.

[11] Habib, m. A., Et al. (2026). Edcc-RPL: a novel energy-efficient and load-balanced objective function for RPL-based iot networks. *Plos ONE*. Https://doi.Org/10.1371/journal.Pone.0346827

[12] Kim, h. S., Ko, j., Culler, d. E., & Paek, j. (2017). Challenging the IPv6 routing protocol for low-power and lossy networks (RPL): a survey. *IEEE communications surveys & tutorials*, 19(4), 2502–2525. Https://doi.Org/10.1109/comst.2017.2751617

[13] Kuwelkar, s., & Virani, h. G. (2023). Of-FZ: an optimized objective function for the IPv6 routing protocol for LLNs. *Iete journal of research*, 69(9), 6101--6119. Https://doi.Org/10.1080/03772063.2021.1990139

[14] Lalou, m., Tahraoui, m. A., & Kheddouci, h. (2018). The critical node detection problem in networks: a survey. *Computer science review*, 28, 92–117. Https://doi.Org/10.1016/j.Cosrev.2018.02.002

[15] Winter, t., Thubert, p., Brandt, a., Hui, j., Kelsey, r., Leiserson, p., Pister, k., Struik, r., Vasseur, j. P., & Alexander, r. (2012). RPL: IPv6 routing protocol for low-power and lossy networks (ietf RFC 6550). Internet engineering task force.

[16] Lamaazi, h., & Benamar, n. (2018). Of-EC: a novel energy consumption aware objective function for RPL based on fuzzy logic. *Journal of network and computer applications*, 117, 42--58. Https://doi.Org/10.1016/j.Jnca.2018.05.015

[17] Lamaazi, h., & Benamar, n. (2019). A novel approach for RPL assessment based on the objective function and trickle optimizations. *Wireless communications and mobile computing*, 2019, 4605095. Https://doi.Org/10.1155/2019/4605095

[18] Lamaazi, h., & Benamar, n. (2020). A comprehensive survey on enhancements and

limitations of the RPL protocol: a focus on the objective function. *Ad hoc networks*, 96, 102001. Https://doi.Org/10.1016/j.Adhoc.2019.102001

[19] Urama, i. H., Fotouhi, h., & Abdellatif, m. M. (2017). Optimizing RPL objective function for mobile low-power wireless networks. In proceedings *of the IEEE 41st annual computer software and applications conference (compsac)* (pp.678--683). Https://doi.Org/10.1109/compsac.2017.185

[20] Zhang, j., Chen, j., Dai, y., Wang, s., & Qi, y. (2025). RL-tree: a reinforcement learning-based adaptive and secure routing protocol for wireless sensor networks. *Informatica*, 49. Https://doi.Org/10.31449/inf.V46i23.11214

[21] Pancaroglu, d., & Sen, s. (2021). Load balancing for RPL-based internet of things: a review. *Ad hoc networks*, 119, 102548. Https://doi.Org/10.1016/j.Adhoc.2021.102491

[22] Ben aissa, y., Grichi, h., Khalgui, m., Koubâa, a., & Bachir, a. (2019). Qcof: new RPL extension for qos and congestion-aware in low power and lossy network. In *proceedings of the 14th international conference on software technologies (icsoft)* (pp.560--569). Https://doi.Org/10.5220/0007978805600569

[23] Chen, y., Chanet, j.-P., Hou, k.-M., Shi, h., & De sousa, g. (2015). A scalable context-aware objective function (scaof) of routing protocol for agricultural low-power and lossy networks (rpal). Sensors, 15(8), 19507-19540. Https://doi.Org/10.3390/s150819507

[24] Rana, p. J., Bhandari, k. S., & Lee, k. (2020). Ebof: a new load balancing objective function for low-power and lossy networks. *Ieie transactions on smart processing and computing*, 9(3), 244--254. Https://doi.Org/10.5573/ieiespc.2020.9.3.244

[25] Shahbakhsh, p., Ghafouri, s. H., & Bardsiri, a. K. (2023). Raarpl: end-to-end reliability-aware adaptive RPL routing protocol for internet of things. *International journal of communication systems*. Https://doi.Org/10.1002/dac.5445

[26] Wang, f., Babulak, e., & Tang, y. (2020). SL-RPL: stability-aware load balancing for RPL. *Transactions on machine learning and data mining,* 13(1), 27--39.

[27] Solapure, s. S., & Kenchannavar, h. H. (2020). Design and analysis of RPL objective functions using variant routing metrics for iot applications. *Wireless networks*, 26, 4637--4656. Https://doi.Org/10.1007/s11276-020-02348-6

[28] Taghizadeh, s., Bobarshad, h., & Elbiaze, h. (2018). Clrpl: context-aware and load balancing RPL for iot networks under heavy and highly dynamic load. *IEEE access*, 6, 21577–23291. Https://doi.Org/10.1109/access.2018.2817128

[29] Tahir, y., Yang, s., & Mccann, j. (2018). Brpl: backpressure RPL for high-throughput and mobile iots. *IEEE transactions on mobile computing*, 17(1),29--43. Https://doi.Org/10.1109/TMC.2017.2705680

[30] Thubert, p. (2012). *Objective function zero for the routing protocol for low-power and lossy networks (RPL)* (RFC 6552). Ietf.

[31] Venugopal, k., & Basavaraju, t. G. (2023). Congestion and energy aware multipath load balancing routing for LLNs (CEA-RPL). *International journal of computer networks & communications*, 15(3). Https://doi.Org/10.5121/ijcnc.2023.15305