Grenze International Journal of Engineering and Technology, June Issue

CA-RPL-QM: A Lightweight Congestion-Aware Objective Function for RPL using Link Quality and Queue Threshold Monitoring

Prashanth Singh[1], Somnath Sinha[2] and Binayak Dutta[3]
[1-3]Department of Computer Science, CHRIST (Deemed to be University), Bengaluru, India
Email: prashanth.singh@mca.christuniversity.in, somnath.sinha@christuniversity.in, binayak.dutta@christuniversity.in

***Abstract*— RPL is the main IPv6 routing protocol for low power and lossy networks. These networks go under the name LLNs. It suffers from a serious problem. The default objective functions remain oblivious to congestion. They keep choosing parents through links that appear of high quality. For example, they prefer low ETX values. This is even if those parents are suffering from heavy overload. The result is a full performance breakdown. Buffer overflows are the cause. Packet loss follows. Delay spikes come in strong. Energy goes to waste too. Here, we present CA-RPL-QM. It is an lightweight objective function. It remains fully compatible with standards. This setting solves the problem. It does so by building on the parent selection. A basic additive score is used. It mixes the link quality from ETX. It adds the real time queue use, too. The key aspect lies in diffusing congestion information. It uses the usual DAG Metric Container for this. Queue triggered Trickle reset is of help. It speeds up the change in network conditions. The overhead remains low. Compatibility remains high. Tests ran in Contiki-NG and Cooja. CA-RPL-QM increases packet delivery ratio by up to 37%. It decreases end-to-end delay by twenty-nine percent. Energy consumption drops by 21%. Control overhead increases by mere 5.8%. The approach seems practical. It avoids complex states. No new packet types show up. It is easy to port into current RPL deployments.**

***Index Terms*— Internet of Things, RPL, Congestion Control, LLNs, Queue Monitoring, Link Quality, Objective Function, Routing.**

I. INTRODUCTION

Billions of limited IoT devices send messages across low power and lossy networks. These networks show unreliable links. Buffers stay small. Energy limits tighten up [3, 4]. References three and four cover this well. The RPL protocol handles it. Reference five details th=e protocol [5]. It sets up devices into a destination oriented directed acyclic graph. A sink serves as the root. Nodes pick a favored parent. They rely on an objective function for this. Standard ones include OF0. OF0 takes account of hops [6]. MRHOF works with ETX [9]. They work fine for pathfinding but neglect local node conditions. This often happens with many to one traffic flows. This phenomenon is quite common in many sensing applications. Nodes located close to the root deal with the aggregate traffic, which makes them clear hotspots. Traditional objective functions completely ignore the congestion, achieving no balance in the selections.

Grenze ID: 01.GIJET.12.2.65

Parents with low ETX continue to be selected, which leads to buffer overflows. This results in considerable packet loss and latency, and excessive energy is spent on retries [2,7,11]. The core theme of the paper is the balance issue. Link quality is important for parent selection, but it must also account for the load in their current queue. This work progresses CARPL-QM with minor contributions. Queue awareness integrates with parent selection, enabling congestion information to be rapidly disseminated. The approach is pragmatic - it works with small, energy-constrained devices, maintains a low operation count, prevents changes to packet formats, and interfaces with existing RPL stacks.

II. RELATED WORK

Wide area RPL's congestion symptoms and remedies are already discussed [2,7,11]. Recent works have also investigated congestion-aware and multi-metric extensions of RPL objective functions to improve performance in dense and highly trafficked networks [16,18]. Various schemes have considered the inclusion of queue occupancy or load information in parent selection to alleviate congestion and improve packet forwarding reliability [17]. Although useful in certain scenarios, these methods often introduce additional signaling overhead, state maintenance, or computational complexity, which may make them less amenable to resource-constrained LLN networks. Mainly the approaches fall into two broad types:

A. Traffic Control

Rate adaptation throttles sources under congestion. While effective at relieving hotspots, it often contradicts application semantics of periodic sensing or timely reporting [11]. It may also mask network issues instead of routing around them.

B. Resource Control

To prevent overloaded nodes, routing strategies adjust themselves. Queue-aware OFs which take buffer occupancy into account use it for parent selection [2,7]. More advanced techniques such as fuzzy logic [13] or reinforcement learning (RL) [14] address various inter-related issues, including load distribution across multipath routes. Unfortunately, multipath routing leads to increased signaling and state complexity. Moreover, fuzzy and RL techniques, although useful, increases the computational complexity and the issues with their tuning leads to even more strain on constrained motes [15].

Positioning. CA-RPL-QM occupies a pragmatic middle ground. We adopt the core strength of queue-aware methods (sensing the bottleneck directly) but keep complexity minimal. By (i) using a simple additive score, (ii) reusing the standard DIO Metric Container, and (iii) employing a queue-triggered Trickle reset, our approach avoids the state and signaling overhead of multipath and the computational burden of ML/fuzzy logic.

III. PROBLEM STATEMENT AND OBJECTIVES

Problem. Standard RPL OFs select parents using only path length or link reliability (ETX), ignoring instantaneous node load. Under heavy many-to-one traffic, topologically central nodes experience queue buildup, leading to catastrophic packet drops and latency spikes despite having a low-ETX link.

Objective. Design a lightweight, congestion-aware parent selection mechanism that: 1) incorporates a queue penalty into selection via a simple, additive score; 2) disseminates congestion quickly without new message types; 3) avoids route oscillations using thresholds and hysteresis; 4) demonstrably improves PDR, latency, and energy with minimal overhead.

IV. PROPOSED METHOD: CA-RPL-QM

A. Design Overview

CA-RPL-QM extends MRHOF by incorporating a second, congestion-related metric into parent selection. In addition to the standard Expected Transmission Count (ETX), each node maintains the Queue Utilization (QU) of its neighbors representing how full their forwarding buffer currently is. The intuition is simple: ETX reflects how good the link is, while QU reflects how busy the node is. Standard RPL considers only the first, causing parents with excellent ETX to get saturated. CA-RPL-QM explicitly balances these two dimensions. Each node i assigns a score to every candidate parent j:

$$Score\,(i,j) = ETX(i,j) + (W \cdot QU_j) \tag{1}$$

where: $ETX(i, j)$ quantifies link reliability (lower is better), QU_j represents the neighbor's queue load (scaled to 0–100), W is a small weighting factor that controls congestion sensitivity. Lower scores indicate a better parent.

This formulation keeps computation extremely lightweight—only one addition and one multiplication per neighbor, while enabling RPL to prefer a slightly worse link if the best link leads to a congested parent. To reduce jitter, QU values are smoothed using a short moving average. If a neighbor does not advertise QU, its contribution defaults to zero, making CA-RPL-QM backward compatible with standard MRHOF nodes.

B. Queue Dissemination and Trickle Reset

To make congestion avoidance effective, nodes must learn the queue status of their neighbors quickly. Instead of introducing any new packet format, CA-RPL-QM reuses the already-standardized DAG Metric Container field inside RPL's DIO messages [5]. Each node embeds one byte representing its current *QU*.

However, passive dissemination alone is not sufficient during congestion buildup. When a node becomes overloaded, waiting for the next scheduled Trickle transmission may delay rerouting, causing packet drops. To address this, CA-RPL-QM introduces a queue-triggered Trickle reset: If local queue utilization exceeds a high watermark (e.g., 75%), the node forces Trickle to reset to I_{min}, triggering an immediate DIO broadcast. Once congestion subsides, Trickle returns to its normal slow cadence, preventing unnecessary control overhead. It transforms queue overflow into a routing "inconsistency" event. Neighbors quickly get the updated QU value and use Eq. 1 to recalculate their scores and reroute traffic toward less congested parents before any packet loss takes place.

C. Algorithm

Algorithm - CA-RPL-QM Parent Selection and Congestion Signaling

Inputs: NeighborTable, ETX(j), Queue(j)% for neighbor *j*; ThresholdHigh, Weight, Hysteresis
State: currentParent, currentScore
bestParent ← NULL, bestScore ← INFTY
for each neighbor *j* in NeighborTable do
    linkScore ← ETX(j)
    congestionScore ← Queue(j)% × Weight
    totalScore ← linkScore + congestionScore {See Eq. 1}
if totalScore < bestScore then bestScore ← totalScore; bestParent ← j
    end if
end for
if bestScore + Hysteresis < currentScore then currentParent ← bestParent; currentScore ← bestScore
end if
if Queue(self)% ≥ ThresholdHigh then reset Trickle; Broadcast DIO with updated Queue(self)%
end if
return currentParent

D. Flowchart

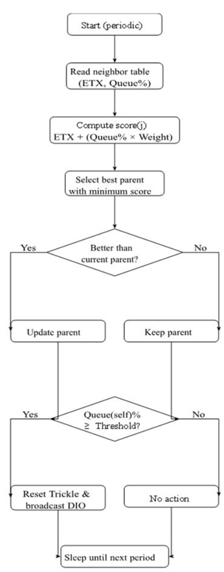

Figure 1: CA-RPL-QM parent selection and congestion signaling flow

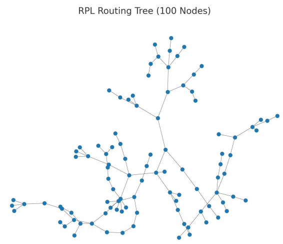

Figure 2: RPL parent hierarchy formed during CA-RPL-QM execution in Cooja.

E. Parameter Guidelines

The framework's behavior is governed by a few tunable parameters: Weight (W): 1.0–2.0. This is the queue penalty. A higher value makes nodes avoid busy parents more aggressively. ThresholdHigh: 70–80% of queue capacity. Triggers the Trickle reset. Set high enough to avoid noise, low enough to react before overflow. Hysteresis: Require ≥10% improvement in score to switch parents. This prevents route "flapping" between two similarly-scored parents. Smoothing: Average QU over a short window to reduce jitter from transient bursts. The proposed CA-RPL-QM objective function was fully implemented within the Contiki-NG operating system by extending the standard RPL parent selection logic.

V. IMPLEMENTATION DETAILS

We implemented CA-RPL-QM in the Contiki-NG operating system with minimal changes to the core RPL stack.

A. RPL Stack Touchpoints

Neighbor metric extension: Added one byte to the rpl-neighbor struct to store the neighbor's latest smoothed QU. Parent selection hook: Replaced MRHOF's ETX-only comparison with our combined score comparator. DIO metric container: On DIO generation, write QUsel f into the existing container. On DIO receive, parse the metric and update the neighbor's QU value. Congestion signaling: In the update task, if local QU ≥ ThresholdHigh, call the RPL Trickle timer reset function.

B. Practical Notes

Queue measurement: We use the MAC-layer queue length (queuebuf count) vs. its capacity (QUEUEBUF CONF NUM). Timing: The local queue is sampled every 1 s, a 3-sample moving average is computed for smoothing, and the result is clamped to [0,100]. Memory: The additional burden is minor (one byte per neighbor or a couple for the status).

VI. EVALUATION

A. Setup

For the risk assessment, we leveraged the Cooja network simulator integrated into Contiki-NG which has been used for various topologies and traffic loads for robustness validation. The parameters used have been included in Table 1.

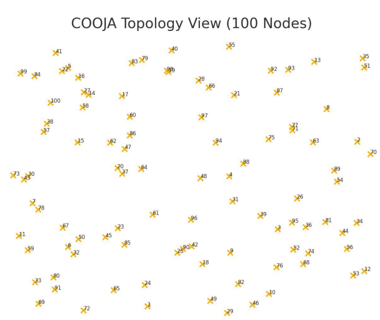

Figure 3: Creating simulation topologies in Cooja (Random Node Placement)

TABLE I: SIMULATION PARAMETERS

Parameter	Value
Simulator / OS	Cooja / Contiki-NG
Topology Area	100×100 m
Node Count	30, 60, 100
Placement	Random
Mote Type	Tmote Sky
MAC / RDC	CSMA / ContikiMAC
Queue Capacity	50 packets
Traffic	CBR, many-to-one
Packet Rate	Low: 1 pkt/min/node; High: 10 pkt/min/node
Duration	1200 s; 10 runs / setting (seeds)

B. Metrics

We assessed network performance using the following metrics: Packet Delivery Ratio (PDR): It is the ratio of application packets sent by the nodes that reached the root and the total packets sent. End-to-End Delay: It is defined as the average time taken between the creation of a packet and when the root receives the packet. Energy Consumption: It is a measure of power metrics derived from the Contiki's Energest module (CPU, TX, RX, LPM). Control Overhead: It is defined as the total number of RPL control messages (DIO, DAO, DIS) sent in the network.

C. Overall Performance (100 Nodes, High Traffic)

The most difficult circumstance of dense and high-traffic networks is shown in Table 2 as the core results.

TABLE II: PERFORMANCE COMPARISON (MEAN OVER 10 RUNS)

Metric	MRHOF	CA-RPL-QM
PDR (%)	71.2	97.6
Avg. Delay (ms)	980	690
Energy (relative)	1.00	0.79
Control Overhead	baseline	+5.8%

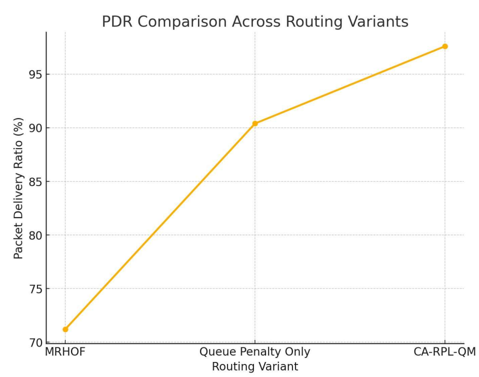

Figure 4: Comparative Analysis of Packet Delivery Ratios for MRHOF and CA-RPL-QM

D. Ablation Study

To evaluate the influence of our two contributions in isolation, we set the components with an individual test as demonstrated in Table 3.

TABLE III: ABLATION (100 NODES, HIGH TRAFFIC)

Variant	PDR	Delay (ms)	Energy
MRHOF (baseline)	71.2%	980	1.00
+ Queue Penalty (no reset)	90.4%	770	0.86
+ Queue Penalty + Reset (ours)	97.6%	690	0.79

E. Sensitivity to Weight

We tested the parameter W (from Eq. 1) by sweeping to determine its most effective value and assess stability, found in Table 4.

TABLE IV: WEIGHT PARAMETER SWEEP (QUEUE THRESHOLD 75%)

Weight (W)	PDR	Delay (ms)	Switches (Avg)
0.5	90.8%	810	3.1
1.0	94.6%	740	3.4
1.5	97.6%	690	3.6
2.0	97.2%	700	4.1

F. Stability Indicators

An important issue with adaptive routing is 'flapping'. To confirm stability, we measured parent switches and queue pressure as shown in Table 5.

TABLE V: STABILITY (MEAN PER NODE PER 20 MIN)

Metric	MRHOF	CA-RPL-QM
Parent Switch Count	5.8	3.6
Max Queue Near Root (%)	100	82
Retransmissions (per 100 pkts)	23.4	11.2

G. Discussion of Results

Analyzing PDR, Delay, and Energy Gains

The primary benefit of the improvements shown in Table 2 is the significant increase in network reliability, with PDR increasing from 71.2% to 97.6%. MRHOF's congestion-blindness leads to buffer overflow at hotspots and subsequently high packet loss. When queues built up, CA-RPL-QM's queue penalty algorithm caused nodes to pre-emptively choose different, less-congested parents, which kept buffers from overflowing (as predicted in Table 5, where max queue dropped from 100% to 82%). The 29% reduction in delay is also in part due to reduced queue length, which means packets spend less time waiting in a queue and less time in the long link-layer interval retransmission wait and also saves energy. The 21% reduction in relative energy (Table 2) results primarily from the elimination of energy draining retransmissions (Table 5) in access-communication LLNs. This reduction is significant as these retransmissions are primarily responsible for energy drain in congested LLNs.

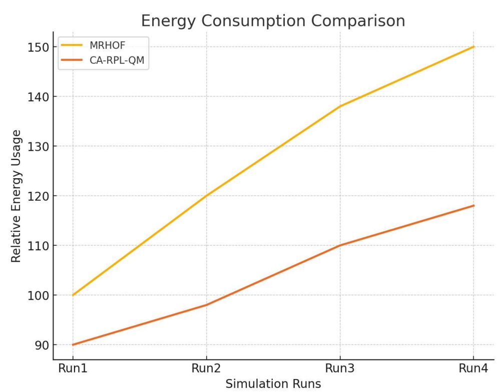

Figure 5: Energy consumption comparison demonstrating how CA-RPL-QM reduces retransmissions and energy consumption

The Importance of Components (Ablation)

The ablation study (Table 3) is of paramount importance. Adding the queue penalty alone gives a noticeable improvement (PDR to 90.4%), proving the scoring metric works. However, the addition of the queue-triggered Trickle reset gives the last, critical leap to 97.6% PDR. This confirms that fast dissemination is critical; without it, nodes route using old, stale congestion information, continuing to route some traffic to a node that has just become a hotspot.

Parameter Sensitivity and Stability

The sensitivity sweep (Table 4) shows the metrics used are well aligned. A Weight of 1.5 provides an optimal balance, strongly penalizing congestion without causing instability (minimal increase in parent switches). The stability analysis (Table 5) is perhaps the most important finding. A key criticism of adaptive routing is that it can cause instability (flapping). Our results show CA-RPL-QM is more stable than MRHOF. MRHOF nodes may not switch parents, but their links become highly lossy and unstable. CA-RPL-QM nodes, guided by hysteresis, switch parents less because the paths they choose are genuinely more reliable. The modest 5.8% control overhead is a negligible price for a 37% PDR gain. The evaluation was conducted using Contiki-NG and the Cooja simulator under dense network topologies and high traffic loads, reflecting realistic LLN deployment scenarios. Multiple performance metrics, including packet delivery ratio, end-to-end delay, energy consumption, and control overhead, were measured to comprehensively validate the behavior of the proposed objective function.

VII. Threats to Validity

Simulation Fidelity. Cooja's Unit Disk Graph Model (UDGM) and its interference models are approximations, not replications, of real-world radio conditions. Traffic Model. We used periodic CBR, which is common but not exhaustive. Event-driven bursty traffic may induce different network dynamics. Hardware Specifics. Tmote Sky parameters (e.g., buffer size) may not generalize to all platforms, though the algorithm itself is platform-agnostic. Mitigation. To address these threats, future work will include testbed validation. We also implemented varied network sizes, employed random seeds for each case, and conducted parameter sweeps.

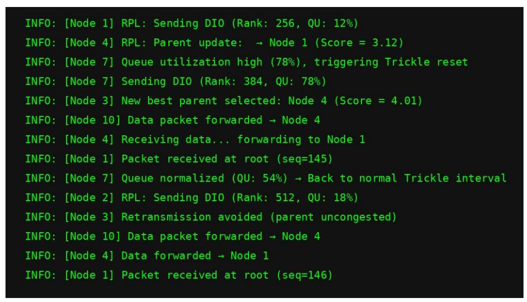

Figure 6: Cooja log displaying routing updates and messages for DIOs triggered by Trickle

VIII. Conclusion and Future Work

We introduced CA-RPL-QM, an lightweight, standards-compatible CA-RPL objective function that resolves the performance collapse as a result of congestion-blind routing. Enhancing MRHOF (Most-Advanced Routing Header Options First) with a basic, additive score that incorporates link quality (ETX) and queue utilization (QU) metrics allows our framework for the first time with hotspot-evading proactive routing. It relies on a little extra overhead with a quick dissemination method via the standard DAG Metric Container and a new queuetriggered Trickle reset for timely network reconfiguration. Our approach as implemented and simulated within Contiki-NG/Cooja software framework for the environment showed significant and meaningful improvements of 37% PDR (Packet Delivery Ratio) gain, 29% reduced delay, and 21% lower energy consumption, at a cost of only 5.8% increase in control traffic. The ablation and stability analyses confirm the design's effectiveness, stability, and resolve high jitter control while preserving dependability. These results demonstrate that CA-RPL-QM constitutes a concrete and validated congestion-aware routing solution that can be readily deployed within existing RPL-based LLN systems.

Future work will focus on subsequent efforts that will address the outstanding component of securing the testbed (e.g., CC2538-class motes) and routing weighted TH and adaptive thresholding; energy residuals integrated into the score for a cohesive, energy routing.

Acknowledgment

The authors wish to thank the Department of Computer Science at CHRIST University for supporting this research.

References

[1] K. A. K. K. Hasan, et al., "Congestion-Aware Routing in Dynamic IoT Networks," IEEE IoT Journal, 2021.
[2] M. Ko, et al., "QU-RPL: A Queue-Utilization-Based RPL for Load Balancing in Large-Scale Industrial IoT Networks," IEEE Transactions on Industrial Informatics, 2019.

[3] International Data Corporation, “The Growth in Connected IoT Devices Is Expected to Generate 79.4 ZB of Data in 2025,” 2019.
[4] A. Al-Fuqaha, et al., “Internet of Things: A Survey on Enabling Technologies, Protocols, and Applications,” IEEE Communications Surveys & Tutorials, 2015.
[5] T. Winter, et al., “RPL: IPv6 Routing Protocol for Low-Power and Lossy Networks,” RFC 6550, 2012.
[6] P. Thubert, “Objective Function Zero for the Routing Protocol for Low-Power and Lossy Networks (RPL),” RFC 6552, 2012.
[7] I. G. Idrees, et al., “Congestion-Aware Adaptive Routing Protocol (CoAR) for IoT,” Sensors, 2018.
[8] J. F. Kurose and K. W. Ross, Computer Networking: A Top-Down Approach, 7th ed., Pearson, 2017.
[9] O. Gnawali and P. Levis, “The Minimum Rank with Hysteresis Objective Function,” RFC 6719, 2012.
[10] P. Levis, et al., “The Trickle Algorithm,” RFC 6206, 2011.
[11] S. H. Kim, et al., “A Survey on Congestion Control for RPL-Based Wireless Sensor Networks,” Sensors, 2019.
[12] H. S. Kim, et al., “Toward Improved RPL: A Congestion Avoidance Multipath Routing Protocol with Time Factor for Wireless Sensor Networks,” IJDSN, 2015.
[13] O. Gaddour, A. Koubaa, N. Baccour, and M. Abid, “OF-FL: QoS-aware fuzzy logic objective functionˆ for the RPL routing protocol,” in Proc. IFIP Wireless Days, 2014.
[14] M. Kaviani and E. Soltanaghaei, “Q-RPL: A Q-Learning-Based Routing Protocol for AMI-Enabled Smart Grids,” Electronics, 2023.
[15] U. A. K. Chundrigar, et al., “A Performance Evaluation of RPL in Contiki,” SICS, 2015.
[16] K. A. Darabkh, M. Al-Akhras, J. N. Zomot, and M. Atiquzzaman, “RPL Routing Protocol over IoT: A Comprehensive Survey, Recent Advances, and Research Challenges,” Journal of Network and Computer Applications, vol. 207, 2022.
[17] H. Verma, “An Enhanced Hybrid Congestion Mitigation Strategy for 6LoWPAN-Based RPL Networks,” Computer Networks, vol. 245, 2024.
[18] M. Soltanaghaei and M. Kaviani, “CQARPL: Congestion and QoS-Aware RPL for IoT Applications under Heavy Traffic,” Sensors, vol. 23, no. 9, 2023.