CRETA: Cross-layer RPL with Efficient Trickle and Adaptive Radio Duty Cycle Designed for Mobile IoT Application

Soumya Vastrad, *Member, IEEE,* and Shobha K. R., *Senior Member, IEEE.*

Original scientific article

***Abstract*—The literature on IPv6 Routing Protocol for Low Power and Lossy Networks (RPL) makes it abundantly evident that there is a trade-off between convergence time and power usage. To mitigate this effect, we present CRETA: Cross-layer RPL with Efficient Trickle and Adaptive RDC. In this case, information is shared between adjacent layers. The DODAG Information Object (DIO) count obtained from the network layer is used by the proposed method to define the data link layer. We obtained the DIO count at the network layer using our earlier work, the Power Efficient Trickle Algorithm (PETA). The data link layer modifies the radio duty cycle based on this count. For both constant and dynamic traffic rates, the Random Way Point mobility model is used to assess RPL's performance. BonnMotion is used to create mobile traces. The performance of CRETA is assessed at 3 Kmph, 11 Kmph, and 19 Kmph to guarantee effectiveness in a variety of user scenarios. With the Contiki OS/Cooja simulator, CRETA is compared to benchmark algorithms, conventional RPL, PETA, and MSAT-RPL. Our results show that CRETA works better than normal RPL and MSAT-RPL, using 40% less power at 3 Kmph, 26% at 11 Kmph, and 18% at 19 Kmph.**

***Index Terms*—BonnMotion, Contiki OS/Cooja, Cross-Layer Approach, Internet-of-Things, LLNs, Mobile RPL, Trickle Algorithm, WSNs.**

I. INTRODUCTION

WIRELESS Sensor Networks (WSNs) are the main building block of IoT [1]. WSNs are specific Low Power Lossy Networks (LLNs) suitable for many IoT applications. Though they are suitable for a wide range of applications, they are also inherited with many challenges such as routing, memory, and power constraints. To meet these challenges, the Internet Engineering Task Force (IETF) proposed a lightweight routing protocol namely IPv6 Routing Protocol for Low Power and Lossy Networks (RPL) [2]. Decades ago, IETF projected RPL mainly for routing in LLNs. Since then, RPL has gained research interest all over the globe. To meet the needs of IoT, enormous research works have been carried out to enhance the performance of RPL in static as well as mobile environments. Although wide range of research is carried out in both static as well as mobile scenarios, always there is a scope for realising IoT-oriented RPL [3]. To meet the requirements of IoT applications it is ideal to consider mobility-supported RPL with both constant and dynamic traffic rates.

Real-time mobile IoT encompasses many applications such as health care, agriculture robotics, target detection, wildlife foraging, and many more [4]. These applications demand energy efficiency and minimal delay to ensure efficient operation [5]. Thus, while designing an efficient algorithm we must consider two key parameters such as energy consumption and delay. A survey of existing works related to RPL shows that there is a trade-off between power consumption and convergence time [6]. Work presented in [7] demonstrates that the duty cycle has a strict bearing on the RPL performance in mobile scenarios. Thus, there is an enduring necessity to:

- Propose an efficient algorithm to mitigate the trade-off between power consumption and convergence time.
- Design an approach to consider both control overhead and adaptive RDC to address the challenges of mobile RPL.
- Develop an efficient algorithm suitable for IoT applications with varying node speeds and dynamic traffic rates.

Thus, the goal of this paper is to address the aforementioned challenges associated with mobile RPL suitable for IoT applications. Hence, in this work, we present three major contributions that advance the routing algorithm RPL in the mobile context for mobile IoT applications.

The main contributions of this work are:

- We propose a novel framework CRETA: Cross-layer RPL with Efficient trickle and Adaptive RDC that mitigates the trade-off between power consumption and convergence time thus addressing a significant gap in the literature.
- RDC is made adaptive based on the DIO count to have a quick response to topology changes.
- We considered dynamic features of mobile IoT applications by considering both dynamic traffic rates and constant traffic rates for evaluation, to provide empirical evidence from the simulation experiments. We also validated the performance of the proposed protocol CRETA by evaluating the performance for different speeds. Together,

Manuscript received January 16, 2025; revised February 24, 2025. Date of publication September 30, 2025. Date of current version September 30, 2025. The associate editor prof. Renata Lopes Rosa has been coordinating the review of this manuscript and approved it for publication.

S. Vastrad is currently with the Department of Electronics and Communication Engineering, BMS Institute of Technology and Management, Bengaluru, Karnataka, India (e-mail: soumyav4@gmail.com).

Shobha K. R. is with the Department of Electronics and Communication Engineering, Dayananda Sagar College of Engineering, Bengaluru, Karnataka, India (e-mail: shobhashankarmsr@gmail.com).

Digital Object Identifier (DOI): 10.24138/jcomss-2024-0094

these contributions pave the way for future research in the field of low-power and lossy networks.

The remaining portion of the paper is structured as follows: Section II briefs the literature related to mobile RPL. The proposed work and methodology are explained in section III. Results and analysis of the simulation experiments are done in section IV. The paper is concluded with future directions in section V.

II. Overview of the Literature

Pitfalls of RPL in mobile scenarios have gained the interest of the researchers and enormous research is being carried out on IoT inclined mobile RPL. In Table I, we have briefed the existing works on mobile RPL where majority of the works have IoT application as the objective.

It is evident from the literature listed in Table I that significant efforts were made to maximise RPL performance in mobile contexts. Different approaches have a different purpose. Various performance metrics were improved to enhance the performance of RPL. Few of them have focused on maintaining reliable routing topology using the Received Signal Strength Indicator (RSSI) to improve the Packet Delivery Ratio (PDR). While few have optimized the performance of RPL by reducing power consumption by selecting the optimum parent many have augmented RPL by optimizing the trickle algorithm. Few others have paid attention to improving RPL in mobile scenarios using a cross-layer approach.

A. Literature Gap

Though there are many existing works on optimizing the behaviour of RPL in mobile scenarios it is noteworthy that very few have focused on mitigating the effect of power consumption on convergence time. Many existing works have evaluated the performance for either low-speed or high-speed. While speed requirements of IoT vary based on the use cases. Since different IoT services demand data to be delivered at varied traffic rates, it is necessary to consider dynamic traffic rates [26] along with constant traffic rates. Thus, it is essential to propose an efficient algorithm which can address these gaps.

III. Proposed Work and Methodology

Unlike static scenarios, mobile IoT requires quicker action if there is any topology change as nodes are on the fly. To facilitate this, if more DODAG Information Object (DIO) messages are disseminated, then the network will be flooded with a huge number of DIOs in mobile scenarios. This leads to an increase in the power consumption. In contrast, if we try to reduce the power consumption by suppressing DIOs, convergence time will increase. Thus, in this paper, we propose a Cross-layer RPL with an Efficient Trickle algorithm and Adaptive radio duty cycle (CRETA). CRETA uses multilayer data to mitigate the effect of convergence time on power consumption.

In cross-layer design implementations, layer collaboration is application-specific. If a designer would like to reduce the power consumption and increase the lifetime then MAC and network layer collaborations are apt. To optimize transmit power and efficient bandwidth distribution collaboration between MAC, network and physical layers is recommended. Similarly for congestion avoidance transport and network layer collaboration is recommended and for security applications, MAC and physical layers collaboration is used [27], [28].

In CRETA, we use network and data link layer collaboration as we are focusing on reduction in power consumption. We propose a novel idea of using DIO count at the network layer to adjust the Radio Duty Cycle (RDC) at the data link layer to enable quick changes in the algorithm based on the dynamic changes in the network.

In standard RPL, the trickle algorithm is used at the network layer to communicate the network updates and trickle maintain the count called as DIO Count for every DIO transmission. To facilitate less power consumption, instead of the standard trickle algorithm [29], CRETA uses our previous work Power Efficient Trickle Algorithm (PETA) [30] at the network layer. At this layer, the threshold value for DIO count is calculated using (1). Based on this threshold value Th_Count, the radio duty cycle is adjusted at the data link layer. RDC protocol ContikiMAC [31] is used at data link layer. If the DIO count exceeds the threshold value Th_Count, the radio is turned ON otherwise it behaves the same as that of standard RPL. The busier the network is, the more often the radio is turned ON. Since the radio responds quickly to the network activities, convergence time is reduced. Though the radio is turned ON more often, power consumption is handled meticulously by controlling the dissemination of DIOs using PETA at the network layer. Thus, with CRETA both power consumption and convergence time are managed effectively.

Algorithms used by the proposed methodology CRETA are as given in Algorithm 1 and Algorithm 2. Algorithm 1, will be running at the network layer and Algorithm 2 at the data link layer. Cross-layer RPL, CRETA uses DIO count obtained at the network layer to adjust the RDC at the data link layer. DIO count is computed using PETA [30] at the network layer and based on this input RDC is adjusted at the data link layer. In LLNs, data link layer uses 2 protocols. Radio duty cycle is handled at the data link layer by the RDC protocol ContikiMAC [31], while transmission and retransmission are handled by the MAC protocol CSMA. After receiving consistent DIO at the network layer, the protocol will check for the joined DAG (Directed Acyclic Graph) in step 1 of Algorithm 1. If DAG is joined then increment the DIO Count and go to step 2 and note down the new DIO count (N_Count), old count (O_Count) and calculate threshold count(Th_Count) using (1). If DAG is not joined then update the parent info and go to step 4 to trigger the rank recalculation at the network layer. After calculating the threshold value Th_Count, check if the N_Count is exceeding Th_Count as in step 3. This is done to check the network's stability. Increased network activity is correlated with changes in the network topology. Thus, it expects rapid response to ensure the efficient operation of the RPL protocol in mobile scenarios.

As indicated by step 3 of Algorithm 1, if N_Count exceeds Th_Count then it checks the status of the receiver. If the receiver is in sleep mode then RDC has adjusted at the data link layer such that the receiver enters the wake-up state else follows the duty cycle as per the protocol described in Algorithm 2.

Algorithm 1 Algorithm used by CRETA at Network Layer

Step 1: Received Consistent DIO. (Process running at Network Layer)
if Joined the DAG. **then**
  Increment the DIO_Count and Go to Step 2.
else
  Update the Parent info and go to Step 4.
end if
Step 2: Calculate N_Count, O_Count, and Th_Count.
Step 3: Check the threshold value.
if N_Count $>$ Th_Count **then**
  if Sleep mode (Check for sleep mode at Data Link Layer) **then**
    Wake up from sleep mode.
  else
    Do nothing.
  end if
else
  Follow the Duty Cycle as per the protocol described in Algorithm 2.
end if
Step 4: Trigger rank recalculation at Network Layer.

Threshold DIO count, Th_Count is calculated using (1) where α = 90, and Scale = 100. Assigning different weights to N_Count and O_Count, helps in balancing their impact on resulting Th_Count. Since, the new DIO count N_Count, is having more significance than old DIO count O_Count, N_Count is assigned with higher weightage than O_Count. In the sense, to prioritise the most recent count over the historical, we assigned the highest weight to N_Count. α is a configurable parameter that determines how quickly one responds to the changes to the transient fluctuations. Thus, to have rapid response to the fleeting network variations, in our simulation we assigned α with 90. To provide standardization and flexibility scale is chosen as 100.

$$Th\_Count = \frac{(N\_Count \times \alpha) + (O\_Count \times (Scale - \alpha))}{Scale}. \quad (1)$$

where,

N_Count= New DIO count, updated after every change in the topology.

O_Count=Old DIO count, noted when the network is inactive.

A change in the DIO count indicates network activity and a higher value of DIO count indicates higher network activity. Since at the network layer, CRETA uses our previous work Power Efficient Trickle Algorithm PETA [30], the number of DIOs is minimized in turn reducing the power consumption. This DIO count obtained at the network layer is utilized at the data link layer to adjust the RDC and helps to keep the convergence time in check indeed mitigating the trade-off between the power consumption and convergence time.

Algorithm 2 Algorithm used by CRETA at Data Link Layer

Step 1: Sleep Mode
Step 2: Check CCA is positive or negative to Know Channel is Busy or Idle
if CCA returns negative, Channel is Busy, Nodes are in Receiving Mode **then**
  if Unicast Reception **then**
    if Is the node intended receiver **then**
      Send Acknowledgement (ACK) and Go to Step 1
    else
      Go to Step 1.
    end if
  else
    Broadcast Reception. No ACK. Go to Step 1
  end if
else
  CCA positive, Channel is free. Nodes are in Transmit Mode
  if Broadcast **then**
    Send Broadcast Packet, switch-off radio during packet interval
    if Wake-up interval reached **then**
      Go to Step 1
    else
      Send Broadcast packet
    end if
  else
    Unicast. Send Packet. With Phase Lock Optimization (PLO), be in sleep mode until neighbor's wake-up phase
  end if
  if ACK received **then**
    Update wake-up phase. Go to sleep Mode.
  else
    if PLO threshold reached **then**
      Remove the wake-up phase. Go to Step 1.
    else
      Send Unicast packet
    end if
  end if
end if

IV. Results and Analysis

We used the Contiki OS / Cooja simulator to carry out the experiments. To check the scalability we evaluated the performance of RPL for 10, 20 and 30 nodes. In our simulation experiments, we restricted the network size to 30 due to two reasons. First, we evaluated the performance of RPL using dynamic traffic rate along with constant traffic rate due to this reason network is flooded with heavy traffic. Second, except the sink node all nodes are mobile, hence more control messages are disseminated in the network. Due to these two reasons, if we increase the size of the network beyond 30, the simulation tool at our machines will not be able to handle the heavy load. Thus, for these experiments, we restrict the size of the network to 30 nodes. As the performance of CRETA is

TABLE I
REVIEW OF THE EXISTING WORKS ON MOBILE RPL

Reference	Methodology Adopted	Limitation	Mobility Models Used
QUERA [8]	It uses quality-aware and mobility measures, such as RSSI, ETX, and Time-to-Reside (TTR). Moreover, QUERA employs its neighbour table management policy to investigate and uphold steady candidates. These two factors lessen the energy wasted when retransmitting information because of packet loss.	The Q-learning-based approach struggles to rapidly adapt to highly dynamic network topologies, resulting in potential inefficiencies in mobile IoT environments.	Random Walk, Manhattan Grid
RPL* [9]	An unsupervised Explainable Artificial Intelligence (XAI) approach is used to identify any departure from the network's typical behaviours as anomalies. Through the transmission of the DIS and DIO via only pertinent nodes, the innovative proactive mobility management system, which has been carefully constructed, guarantees that the communication overhead is minimised.	It improves decision transparency but does not inherently address rapid topology changes and frequent route recalculations.	Random Way Point, Manhattan, Nomadic
IoMT-FRPL [10]	The fuzzy interface system for mobile nodes in the network incorporates Transmission Count (ETX) data to save energy. It comprises three essential steps: The data transmission and motion investigation stage, followed by fuzzy-based prediction of a new static parent for the mobile node in the second stage and verification of the unique attachment point in the third step.	The inherent uncertainty and variability in mobile networks can challenge the effectiveness of fuzzy-logic-based approaches potentially leading to suboptimal performance under dynamic mobile IoT environments.	Physical Random Walk
QFS-RPL [11]	Fisheye State Routing protocol ideation and the Q-learning algorithm policy are used in this protocol. Large-scale networks can employ QFS-RPL because it solves the problem by broadcasting scores in the network space and does not require heavy learning with exponential time complexities as Q-learning does.	The protocol faces challenges in adapting quickly to topology changes, maintaining routing efficiency, and balancing energy consumption with real-time adaptability.	Random Way Point
RM-RPL [12]	To prevent loops from forming, RM-RPL allows mobile nodes to function as both routers and parents inside the network. It incorporates and modifies the behaviour of the protocol when nodes are stationary and presents a novel objective function that optimises the choice of parent nodes. In addition, it correctly identifies crucial packets.	Frequent route updates, complex parent selection, and additional loop avoidance mechanisms lead to higher control overhead and energy consumption.	Random Way Point.
LA RPL [13]	To improve performance, Learning Automata (LA) is integrated with the RPL routing system for the Internet of Things networks. In order to change routing path choices in response to shifting network conditions, the approach made use of the LA.	The protocol faces challenges in achieving optimal performance in rapidly changing mobile IoT environments due to the intricacy of learning automata and possible delays in convergence.	Random Way Path
Mobility Enhanced RPL [14]	Examines the difficulties brought up by mobility in the Internet of Things networks. It manages packets and processes the new parent selection, by keeping a watch on mobile nodes' energy usage. to improve RPL to appropriately integrate mobile nodes in the network.	This approach does not explicitly address the possible increase in control overhead that can be caused by frequent route updates and mobility-induced topology changes.	Random Way Path
MT-RPL [15]	It recommends using a cross-layer strategy to stop nodes from becoming disconnected for extended periods. This cross-layer method uses reports from the MAC protocol X-Machiavel at the MAC layer to initiate the network layer actions required to keep mobile nodes connected.	It introduces additional complexity in coordinating information exchange across layers, which may lead to processing delays, increased computational overhead, and challenges in resource-constrained IoT devices.	Modified Random Way Point.
MARPL [16]	In this cross-layer approach, the network layer determines a node's variability neighbourhood by using media access layer signal strength data.	This approach imposes a heavier computational load on resource-limited IoT devices, resulting in longer processing times.	Steady-state Random Way Point.
rpl-TotEg-Neighbors [17]	A new version of MRHOF with combinations of various metrics such as total energy, number of neighbours, and estimated transmission count (ETX), is presented for mobile IoT.	Higher computational complexity can strain resource-limited IoT devices, potentially reducing network efficiency.	Random Way Point, Reference Point Group, Nomadic.
mRPL [18]	To maintain backward compatibility with the standard protocol, it added a hand-off mechanism. Mobile nodes (MNs) were subjected to the smart-HOP hand-off method, which involved scheduling the control messages within the trickle algorithm.	Reactive handover approach relies on signal quality-based parent selection, which introduces delays in route convergence potentially impacting network stability in highly dynamic IoT environments.	Hard and soft hand-off model.
ARMOR [19]	Metric Time To Reside (TTR) is employed in this algorithm. TTR gives an estimate of the duration during which the nodes will be within each other's transmission range.	TTR metric calculation, frequent parent re-evaluations, and increased control message processing, impose higher processing and energy overhead on resource-constrained IoT devices.	Random Walk and Manhattan.
MobiRPL [20]	It emphasizes ensuring a reliable routing structure than reducing energy usage to handle network changes. Three new methods are included in it: the objective function based on hop distance, RSSI, connectivity management, and mobility detection.	Since RSSI is highly fluctuating, it can result in inefficient routing decisions.	Random Way Point.
MSE-RPL [21]	With a dynamic trickle timer, it utilizes a neighbour link quality table and a function to choose the optimal parent in the event of mobility, critical zones, and a blacklist.	Real-time RSSI-based rank adjustments demand additional processing power, which can burden resource-constrained IoT devices.	Random Walk Model.

Continued on next page

TABLE I - continued from previous page

Reference	Methodology Adopted	Limitation	Mobility Models Used
RMA-RP [22]	Updates next-hop nodes to adapt to changes in topology using a dynamic motion detection approach based on link quality. Adaptive timers are added to adjust the rate at which control messages are transmitted.	This approach relies on mobility awareness and location-based information, introducing high computational overhead and making it less suitable for resource-constrained devices or environments where precise location data is unavailable.	Random Way Point.
Mobility-Aware Parent Selection Algorithm [23]	It allows the random mobility of the nodes in the RPL and selects the optimal parent from the list of preferred parents using metrics such as the Euclidean distance between the mobile node and the selected parent node, the ETX, the expected lifetime, and RSSI. To overcome the extended listen-only period, they also presented a dynamic trickle method for a trickle timer, which dynamically allocates a timer based on a random set of neighbour nodes under mobility.	This approach can lead to instability in the routing process, with nodes constantly switching between parents, introducing routing overhead and increasing the chances of packet loss.	Random Way Point.
EKF-MRPL [24]	By using the Extended Kalman Filter to forecast its non-linear course, mobile nodes are provided with seamless communication. As it takes into consideration choosing a new attachment depending on the anticipated path, it minimises the amount of association adjustments.	In some cases, the selection of a new preferred parent can increase end-to-end delay through factors such as inaccurate predictions, parent-switching overhead, recalculations of movement models.	Random Walk.
MSAT-RPL [25]	It uses an adaptive trickle algorithm to reduce the delay. To conserve power it sends the nodes to sleep mode whenever necessary.	Due to the adaptive nature of the trickle timer, there can be a delay in the convergence. Impact of adaptive nature of trickle timer on convergence is not evaluated.	Random Way Point, Random Walk, Random Direction.

ETX-Expected Transmission Count, RSSI-Received Signal Strength, MRHOF- Minimum Rank Hysteresis Objective Function.

TABLE II
SIMULATION PARAMETERS

Parameters	Details
Simulator	Contiki OS 3.0 / Cooja Simulator.
Radio Environment	UDGM (Distance Loss).
Simulation Time	600000ms.
Topology	Random Topology.
Size of the Network	10, 20, and 30.
Transmission Range	50m.
Interference Range	100m.
Objective Function	MRHOF and OF0.
DIO Int Min	12.
DIO Int Max	20.
Redundancy Constant (K)	10.
Tx/Rx Ratio	100%.
Dynamic Traffic Rate	Min: 1ppm, Max: 60ppm.
Constant Traffic Rate	1ppm.
Mote Type used for Simulation	Zolertia - Z1 Mote.
Tools Used for Generation of Mobility Traces	BonnMotion.
Mobility Model Used	Random Way Point Mobility Model.
Speed	3Kmph, 11Kmph, 19Kmph.

UDGM- Unit Disk Graph Medium, m - metre, ms - milli seconds, and ppm - packet per minute.

TABLE III
BONNMOTION PARAMETERS

Parameters	Value
Mobility Model Used	Random Way Point Mobility Model
Area	100m x 100m
Minimum Speed of Mobile Nodes	0mps
Maximum Speed of Mobile Nodes	3Kmph, 11Kmph, and 19Kmph
Minimum Pause Time	0s
Maximum Pause Time	20s
Simulation Duration	600s

mps- metres per second, Kmph- kilometres per hour, s-second.

not influenced by network size, using advanced machines and increasing the Java heap space it is possible to increase the size of the network and run the simulation. Since the space complexity of CRETA is O(1) and it maintains constant time complexity we can assume that increased network size will not decline the performance of CRETA.

We used the BonnMotion [32] tool to generate the mobility traces for the Random Way Point Mobility model. These traces are imported to the Contiki OS/ Cooja simulator. Table II and Table III give a brief description of simulation parameters and BonnMotion parameters respectively.

As we mentioned, energy-critical applications such as animal foraging, target detection, and agriculture robotics can benefit from our proposed protocol CRETA. Hence, to realise the performance of CRETA, a suitable mobility model should be selected and the optimum speed must be set to mobile nodes as the mobility model and speed are application-dependent. Therefore, in our work, an entity-based Random Way Point (RWP) mobility model is selected as RWP is suitable for animal foraging and target detection applications [33]. Since the speed of an entity is also application-dependent we considered three different speeds to match with entities involved in aforementioned applications. Therefore, we considered 3Kmph, 11Kmph, and 19Kmph.

We have compared the performance of CRETA with 3 benchmark protocols standard RPL [2], MSAT-RPL [25], and our previous work PETA [30]. The working principle of MSAT-RPL uses an adaptive trickle algorithm to reduce the delay. To conserve power, it sends the nodes to sleep mode whenever necessary. Since the MSAT-RPL [25] algorithm serves a similar purpose as CRETA, we have chosen MSAT-RPL [25] as one of the benchmark protocols. In PETA [30], the time interval is adjusted dynamically during run time based on the idle time left after the transmission. With the conventional approach, PETA gave promising results for static scenarios but for mobile scenarios, power consumption is reduced at the cost of convergence time. Thus, to demonstrate how a cross-layer approach incorporated on top of PETA can effectively reduce the trade-off between convergence time and power consumption compared to conventional methods, we have included our previous work, PETA [30], as another benchmark protocol alongside the standard RPL [2].

A. Metrics used for Evaluating the Performance of RPL in Mobile Scenarios

To ensure the efficient operation of routing protocol it is necessary to consider key metrics to evaluate the performance of RPL. For the evaluation, we have considered a few of the important performance metrics such as control overhead, average power consumption, Packet Delivery Ratio (PDR) and convergence time as discussed in the following section.

1) Control Overhead (CO): Several control messages DIO, Destination Advertisement Object (DAO) and DODAG Information Solicitation (DIS) communicated in the network, contribute to control overhead. It is calculated as in (2).

$$CO = (No.\,of\,DIS + No.\,of\,DIO + No.\,of\,DAO). \quad (2)$$

2) Average Power Consumption (APC): APC is the total power consumed by all the nodes in the network in Low Power Mode (LPM), CPU mode, Transmission Mode (Tx mode) and Receiving Mode (Rx mode) [34]. It is calculated using (3) and (4).

$$PC = \frac{Energest\_Value \times Voltage \times Current}{(Run\_Time \times R\_TIMER)} \quad (3)$$

$$APC = \frac{Power\,Consumed}{Total\,No.\,of\,Nodes\,In\,the\,Network} \quad (4)$$

Energest value [34] that was indicated in equation (3) is the total number of ticks that the system has spent in a specific state. According to the Zolertia Z1 datasheet [35], the voltage is 3V, and the current values are as follows: Idle mode = 0.426mA, Power-down mode = 0.020mA, TX mode = 17.4mA, and RX mode = 18.8mA. The R_TIMER operates at 32678 ticks per second. Run_Time is 10 clock seconds, which is used in the power trace start function of the power trace application in the Contiki OS/Cooja simulator.

3) Packet Delivery Ratio (PDR): PDR is the ratio of packets received successfully at the destination to the total packets sent from the source. It is calculated as in (5).

$$PDR = \frac{Packets\,Received\,by\,Destination\,Node}{Total\,Packets\,Sent\,by\,the\,Source\,Node} \quad (5)$$

4) Convergence Time(CT): Convergence time is the time taken by a node to converge to form a DODAG [36]. It is calculated using (6).

$$\begin{aligned} CT = {} & Time\,at\,Which\,the\,Last\,Node\,Joined\,the\,DAG - \\ & Time\,at\,Which\,the\,First\,DIO\,was\,Sent \end{aligned} \quad (6)$$

B. Analysis of RPL Performance in Mobile Scenarios with Focus on Different Metrics

A rise in node velocity could necessitate more frequent handovers, which could cause brief disruptions or even packet loss while switching between nodes [37]. Thus, to know the impact of speed on the behaviour of protocols, a comprehensive analysis of the performance of protocols RPL, MSAT-RPL, PETA and CRETA is carried out in mobile scenarios at different speeds. Each protocol's drawbacks and findings are thoroughly examined. To monitor changes in the protocol's performance and its parameters at varying speeds, we have devoted this section to four distinct subsections 1, 2, 3 and 4 with an emphasis on control overhead, APC, PDR and convergence time respectively.

1) Analysis of RPL Performance in Mobile Scenarios with Focus on Control Overhead for Different Speeds: A number of control messages exchanged to construct the DODAG contribute largely to the protocol performance. The reason is, that as the number of control messages increases control overhead increases leading to power consumption in the network. Since power is the main constraint in LLNs it is essential to control the number of control messages. The higher the power consumption, the lower the lifetime of the network. Thus, control overhead plays a vital role in assessing the protocol performance. This section discusses solely about control overhead in CRETA and three benchmark protocols at different speeds.

The process of selecting parents in the Objective Function Zero (OF0) [38] only considers the path with the minimum hop count and ignores link quality. This metric is insufficient because it ignores long hops, which hurt QoS and energy consumption. While Minimum Rank with Hysteresis Objective Function (MRHOF) [39] only considers the ETX metric without taking other metrics into account, which has a negative impact on network performance [40]. Thus, we have considered both objective functions namely OF0 [38] and MRHOF [39] with both constant and dynamic traffic rates for evaluating all four protocols.

Graphs plotted in Fig. 1, Fig. 2 and Fig. 3 depict the control overhead at 3Kmph, 11Kmph and 19Kmph respectively. From the graphs depicted in Fig. 1 to Fig. 3, we can observe that as the speed of the mobile nodes increases control overhead decreases with all four protocols. This is due to the fact that at higher speeds, mobile nodes spend very less time with their neighbouring nodes hence, are unable to establish a connection with neighbours. Thus, the number of control messages decreases as speed increases.

From all the plots depicted in the above figures from Fig. 1 to Fig. 3, we can see that CRETA has lesser control overhead than RPL and MSAT-RPL. This is due to controlled DIOs. As discussed earlier, in the proposed methodology CRETA, we have utilized PETA as a trickle algorithm. This will control the number of DIOs reducing the control overhead of CRETA compared to standard RPL and MSAT-RPL. While in cross-layer CRETA we have adjusted the RDC based on DIO count thereby switching from sleep state to wake-up state whenever necessary. Thus, there is a slight increase in the

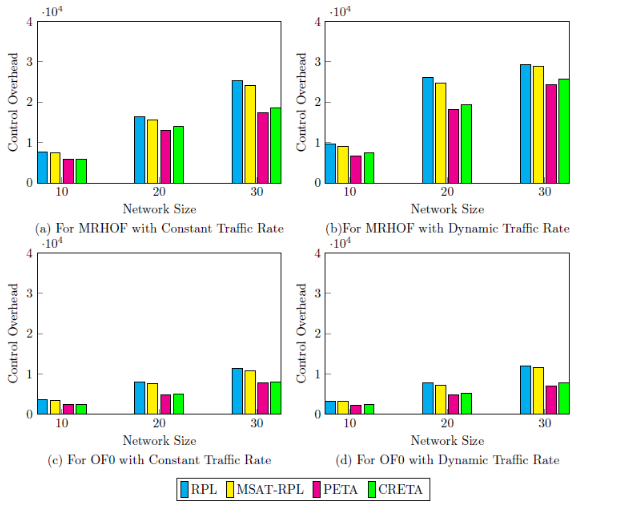

Fig. 1. Analysis of RPL Performance in Mobile Scenarios at 3Kmph with Focus on Control Overhead

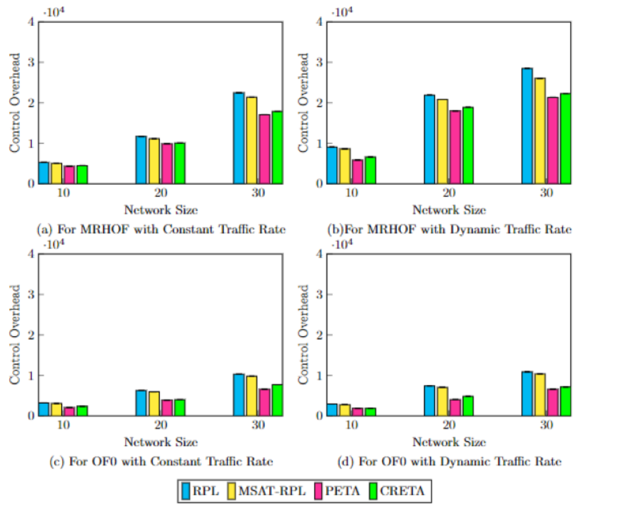

Fig. 2. Analysis of RPL Performance in Mobile Scenarios at 11Kmph with Focus on Control Overhead

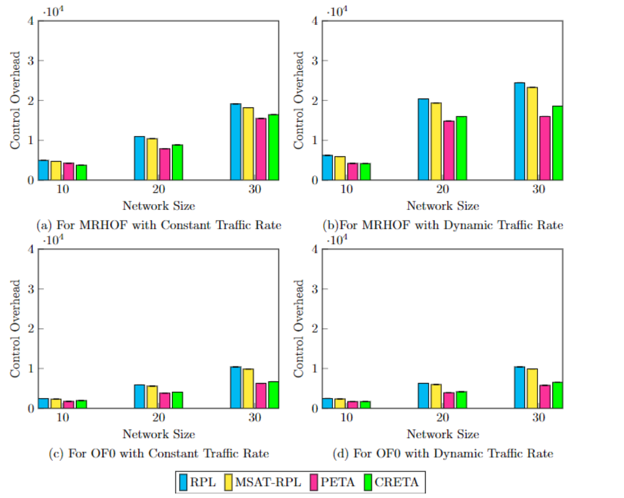

Fig. 3. Analysis of RPL Performance in Mobile Scenarios at 19Kmph with Focus on Control Overhead

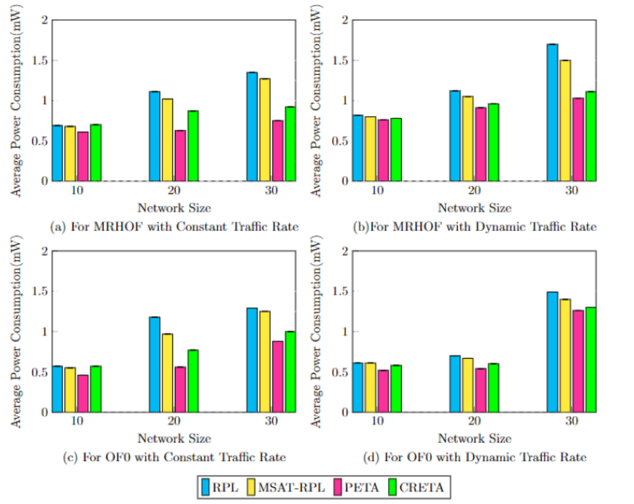

Fig. 4. Analysis of RPL Performance in Mobile Scenarios at 3Kmph with Focus on Average Power Consumption

control overhead in CRETA compared to PETA. However, CRETA performs better than standard RPL and MSAT-RPL in terms of control overhead.

2) Analysis of RPL Performance in Mobile Scenarios with Focus on Average Power Consumption for Different Speeds: Since power and battery life are the main limitations in IoT applications, Average Power Consumption (APC) is the crucial parameter one has to consider while evaluating the performance of a routing protocol. APC is calculated using (3) and (4). Figures 4 through 6 display graphs that illustrate the average power consumption of a network with mobile nodes. As explained in the preceding section, CRETA has lesser control overhead than standard RPL and MSAT-RPL. Since, the number of control messages exchanged impacts the power consumed by the nodes, the power consumed by mobile nodes at CRETA is lesser than the standard RPL and MSAT-RPL. Control overhead of PETA is slightly lesser than CRETA as number of control messages are less in PETA. We can say that the APC of CRETA is lesser than standard RPL and MSAT-RPL. It is significant to note that as the network size grows, CRETA performs better in terms of APC than standard RPL and MSAT-RPL proving the potential ability of CRETA to handle the larger network size.

3) Analysis of RPL Performance in Mobile Scenarios with Focus on Packet Delivery Ratio for Different Speeds: From the previous sections, it is illustrated that CRETA outperforms standard RPL and MSAT-RPL in terms of control overhead and APC. Subsequently, it is crucial to evaluate the performance of protocol in terms of PDR to know the influence of reduced control overhead on PDR. From figures Fig. 7, Fig. 8, and Fig. 9 we can observe that, though the number of DIOs is regulated in CRETA, PDR remains comparable with all the benchmark protocols. This is because in case of active network RDC was adjusted in such a way that, updates were not missed ensuring seamless data transmission. Only redundant DIOs were limited from the transmission. Consequently, the connectivity among the mobile nodes was maintained throughout. Thus, it is notable that CRETA is as

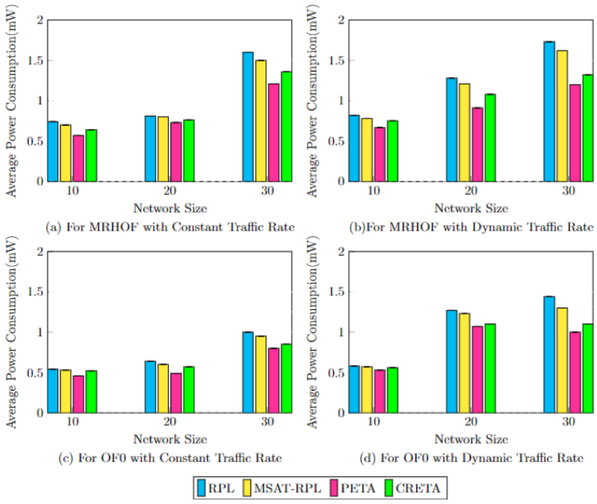

Fig. 5. Analysis of RPL Performance in Mobile Scenarios at 11Kmph with Focus on Average Power Consumption

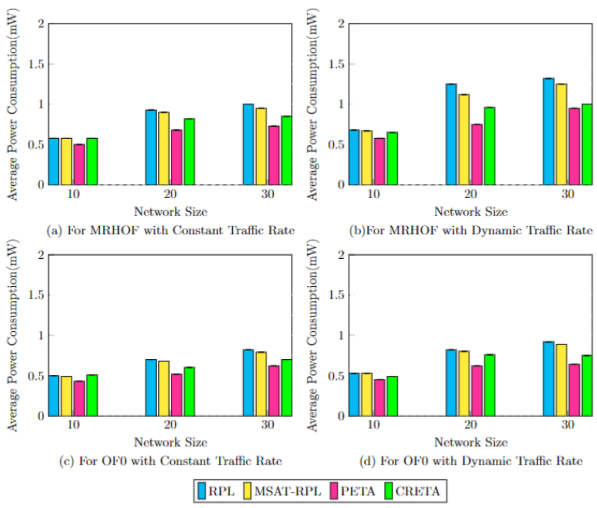

Fig. 6. Analysis of RPL Performance in Mobile Scenarios at 19Kmph with Focus on Average Power Consumption

good as other benchmark protocols in terms of PDR.

4) Analysis of RPL Performance in Mobile Scenarios with Focus on Convergence Time for Different Speeds: Convergence time is one more key metric to be considered to know the consequence of reducing power consumption. When it comes to convergence time and power consumption, there is always a trade-off between convergence time and power consumption, requiring a balance between the two. If we try to improve the one the other will decline. Hence to mitigate this we came up with CRETA a cross-layer approach. Figures from Fig. 10 to Fig. 12 demonstrate how quickly standard RPL and MSAT-RPL converge. In contrast, PETA's convergence time increases significantly. Since in CRETA, we use a combination of adaptive RDC and DIO count, convergence time is minimized. By noting the DIO count, CRETA makes the RDC adaptable based on the network conditions. This quality of CRETA makes it exceptional by ensuring better network responsiveness. It is evident from the figures Fig. 10 to Fig. 12 that at higher speeds network responded quickly

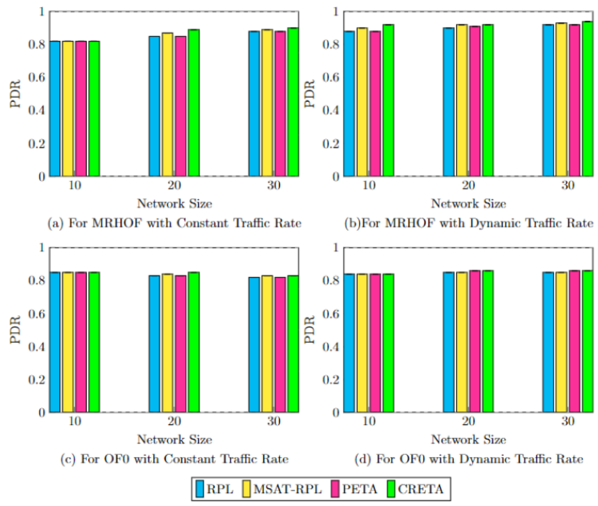

Fig. 7. Analysis of RPL Performance in Mobile Scenarios at 3Kmph with Focus on Packet Delivery Ratio.

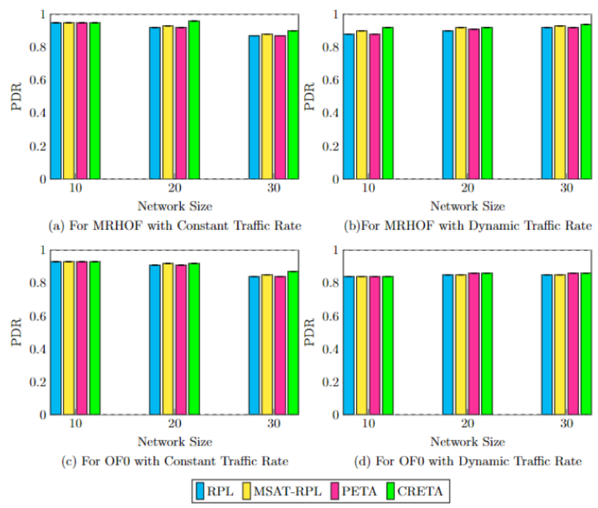

Fig. 8. Analysis of RPL Performance in Mobile Scenarios at 11Kmph with Focus on Packet Delivery Ratio.

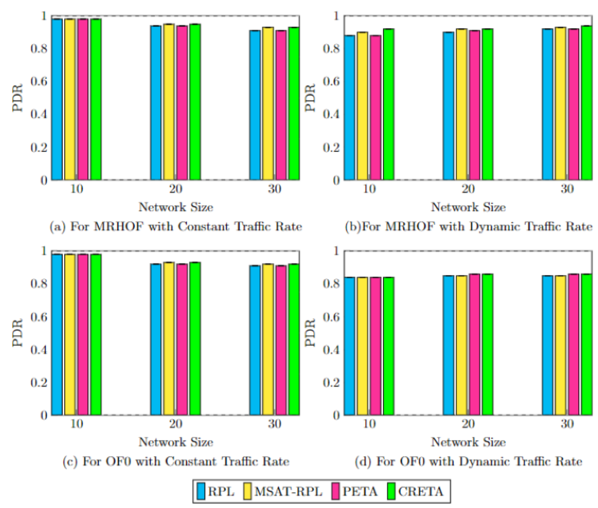

Fig. 9. Analysis of RPL Performance in Mobile Scenarios at 19Kmph with Focus on Packet Delivery Ratio.

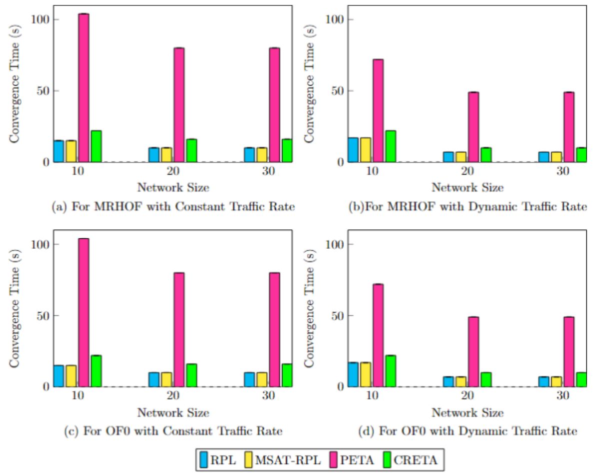

Fig. 10. Analysis of RPL Performance in Mobile Scenarios at 3Kmph with Focus on Convergence Time..

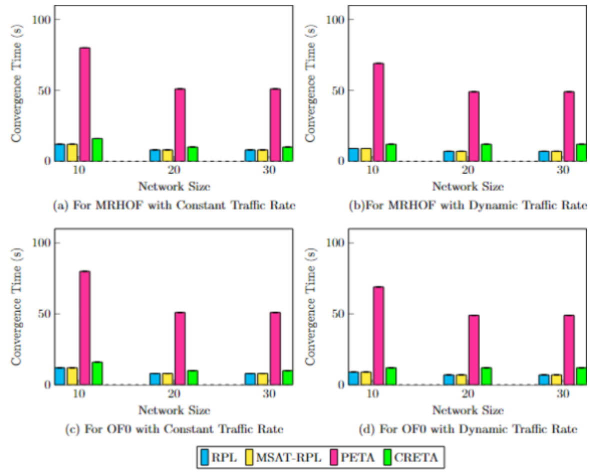

Fig. 11. Analysis of RPL Performance in Mobile Scenarios at 11Kmph with Focus on Convergence Time.

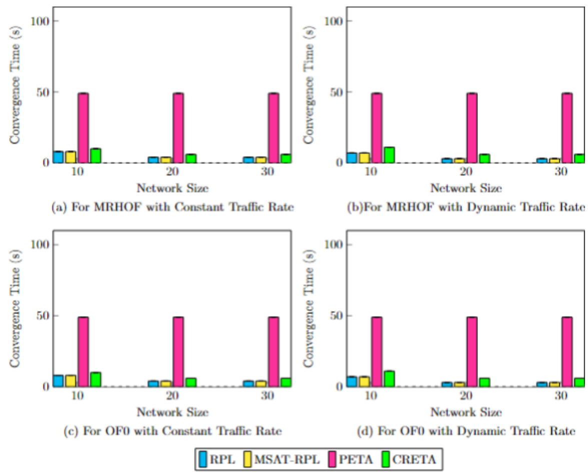

Fig. 12. Analysis of RPL Performance in Mobile Scenarios at 19Kmph with Focus on Convergence Time.

to the topology changes by adjusting the RDC dynamically. Thus, it is observed that CRETA performs better in terms of convergence time at higher speeds. From the figures, we can see that the convergence time of CRETA is way lesser than PETA but slightly higher than standard RPL and MSAT-RPL. Considering the reduction in APC and increase in the lifetime of the network, a small increase in the convergence time is acceptable and we say that CRETA is highly suitable for energy-critical applications in mobile applications operating at various speeds, where a slight increase in the convergence time is considered tolerable. Therefore, the proposed work CRETA is well-suited for IoT applications such as wildlife monitoring and smart agriculture, and environmental monitoring among others.

The extensive analysis mentioned above makes it evident that in all the scenarios with different speeds, with a minor increase in the convergence time, CRETA performs better than its benchmark protocols in terms of control overhead, and APC with the same PDR. We therefore state that CRETA minimises the trade-off between power consumption and convergence time.

Since energy is the most important resource in IoT-based heterogeneous WSNs [41],[42], it is imperative to provide an energy-efficient protocol while taking into account the heterogeneity of the IoT ecosystem. Therefore, we can conclude that the suggested protocol CRETA is a good fit for the heterogeneous IoT ecosystem because it is energy-efficient and strikes a balance between power consumption and convergence time. Furthermore, we took into account several scenarios with dynamic traffic and varying speeds of mobility. Therefore, we say that it encourages diversity.

V. Conclusion and Future Directions

The main pitfall of existing RPL optimizations in mobile IoT is a trade-off between power consumption and convergence time. Thus, in our work to mitigate the impact of power consumption on convergence time, we have proposed a cross-layer protocol CRETA. At the network layer, CRETA uses our previous work, optimized trickle algorithm PETA to compute the threshold DIO count. The threshold DIO count is provided as input to control the RDC at the data link layer. The usage of multilayer data helps to bring the balance between two important metrics, power consumption and convergence time.

From the prior studies, it is visible that most of the works on RPL in mobile scenarios have considered either low speeds or high speeds. However, a protocol that performs better at low speed is not necessarily suitable for high-speed applications. Thus, to evaluate the suitability of CRETA for IoT applications operating at various speeds we have analysed the performance of the proposed methodology at 3Kmph, 11Kmph and 19Kmph for 10, 20 and 30-node networks. Mobility traces for entity-based Random Way Point mobility model were generated using BonnMotion tool and these traces were imported to Contiki OS/Cooja simulator. CRETA outperforms benchmark protocols standard RPL, and MSAT-RPL in terms of control overhead and power consumption. 40% less power is consumed in scenarios with 3Kmph speed and 26%, and 18% less in 11Kmph and 19Kmph respectively while maintaining the comparable PDR. In future work we plan to test the protocol CRETA for different group mobility models.

REFERENCES

[1] A. A. Agarkar, M. Z. Hussain, J.T. Raja et al. “Analysis of taskable mobile IoT sensing systems for coverage and throughput.” Int J Syst Assur Eng Manag (2023). https://doi.org/10.1007/s13198-023-01872-w.

[2] T. Winter , P. Thubert , A. Brandt, T. Clausen, J. Hui , R. Kelsey, P. Levis, K. Pister, R. Struick, and J. Vasseur, “RPL: IPv6 Routing Protocol for Low-Power and Lossy Networks,” IETF RFC 6550, March 2012.

[3] I. S. Alsukayti and M. Alreshoodi, “Toward an understanding of recent developments in RPL routing,” IET Netw., vol. 8, no. 6, pp. 356–366, Nov. 2019, https://doi:10.1049/iet-net.2018.5167.

[4] G. Gokilakrishnan et al. “A Review of Applications, Enabling Technologies, Growth Challenges and Solutions for IoT/IIoT.” 2023 9th International Conference on Advanced Computing and Communication Systems (ICACCS). Vol. 1. IEEE, 2023. https://doi: 10.1109/ICACCS57279.2023.10112825.

[5] Y. Yang, “Adaptive switching and routing protocol design and optimization in internet of things based on probabilistic models” , International Journal of Intelligent Networks, Volume 5, 2024, Pages 204-211, ISSN 2666-6030, https://doi.org/10.1016/j.ijin.2024.05.001.

[6] S.-T. Liu and S.-D. Wang, “Improved Trickle Algorithm Toward Low Power and Better Route for the RPL Routing Protocol,” IEEE Access, vol. 10, pp. 83322–83335, 2022, https://doi: 10.1109/ACCESS.2022.3196693.

[7] J. R. Cotrim, and J. H. Kleinschmidt, “Performance evaluation of RPL on a mobile scenario with different ContikiMAC radio duty cycles”, 2017 IEEE 18th International Conference on High Performance Switching and Routing (HPSR). IEEE, 2017, https://doi: 10.1109/HPSR.2017.7968675.

[8] S. R. Lalani, B. Safaei, A. M. H. Monazzah, H. Taghizadeh, J. Henkel and A. Ejlali, “QUERA: Q-Learning RPL Routing Mechanism to Establish Energy Efficient and Reliable Communications in Mobile IoT Networks,” in IEEE Transactions on Green Communications and Networking 2024, https://doi: 10.1109/TGCN.2024.3399455.

[9] S. Budania, and M. V. Shenoy, “RPL*: An Explainable AI-based routing protocol for Internet of Mobile Things,” Internet of Things, Volume 27, 2024, 101283, ISSN 2542-6605, https://doi.org/10.1016/j.iot.2024.101283.

[10] Z. Ghanbari, N. J. Navimipour, M. Hosseinzadeh, H. Shakeri, and A. Darwesh, “A New Lightweight Routing Protocol for Internet of Mobile Things Based on Low Power and Lossy Network Using a Fuzzy-Logic Method,” Pervasive and Mobile Computing, Volume 97, 2024, 101872,ISSN 1574-1192, https://doi.org/10.1016/j.pmcj.2023.101872

[11] M. Alilou, A. B. Sangar, K. Majidzadeh, and M. Masdari, “QFS-RPL: mobility and energy aware multi path routing protocol for the internet of mobile things data transfer infrastructures”, Telecommun Syst 85, 289–312 (2024), https://doi.org/10.1007/s11235-023-01075-5

[12] A. Seyfollahi et al, “RM-RPL: reliable mobility management framework for RPL-based IoT systems”, Cluster Comput 27, 4449–4468 (2024), https://doi.org/10.1007/s10586-023-04199-0.

[13] C. Thiagarajan and P. Samundiswary, “Learning Automata -Based RPL Routing Protocol for IoT with Mobility (LA -RPL - Mobility),” 2023 International Conference on System, Computation, Automation and Networking (ICSCAN), PUDUCHERRY, India, 2023, pp. 1-5, doi: 10.1109/ICSCAN58655.2023.10395632.

[14] C. Thiagarajan and P. Samundiswary, “Enhanced RPL-Based Routing with Mobility Support in IoT Networks,” 2023 Second International Conference on Advances in Computational Intelligence and Communication (ICACIC), Puducherry, India, 2023, pp. 1-4, doi: 10.1109/ICACIC59454.2023.10435227.

[15] C. Cobarzan, J. Montavont, and T. Noel, “MT-RPL: a cross-layer approach support in RPL”, European Alliance Innovat. 2016; 2(5): 151712, https://doi.org/10.4108/eai.1-12-2016.151712.

[16] J. Kniess, and V. de Figueiredo Marques, “MARPL: A crosslayer approach for IoTs based on neighbor variability for mobility support in RPL”, Transactions on Emerging Telecommunications Technologies 31.12 (2020): e3931, https://doi.org/10.1002/ett.3931.

[17] H. Echoukairi et al., “A New Objective Function for RPL Based on Combined Metrics in Mobile IoT”, J. Commun. 18.5 (2023): 301-309. https://doi:10.12720/jcm.18.5.301-309.

[18] H. Fotouhi, D. Moreira, and M. Alves, “mRPL: Boosting mobility in the Internet of Things”, Ad Hoc Networks 26 (2015): 17-35. https://doi.org/10.1016/j.adhoc.2014.10.009.

[19] M. Aliasghar et al., “ARMOR: a reliable and mobility-aware RPL for mobile Internet of Things infrastructures”, IEEE Internet of Things Journal 9.2 (2021): 1503-1516. https://doi:10.1109/JIOT.2021.3088346.

[20] H. Kim, H.-S. Kim, and S. Bahk, “MobiRPL: Adaptive, robust, and RSSI-based mobile routing in low power and lossy networks”, Journal of Communications and Networks 24.3 (2022): 365-383 https://doi:10.23919/JCN.2022.000004

[21] A. Vaezian, and Y. Darmani, “MSE-RPL: mobility support enhancement in RPL for IoT mobile applications”, IEEE Access 10 (2022): 80816-80832. https://doi:10.1109/ACCESS.2022.3194273.

[22] H. Farag et al., “RMA-RP: A reliable mobility-aware routing protocol for industrial iot networks”, 2019 IEEE Global Conference on Internet of Things (GCIoT). IEEE, 2019. https://doi: 10.1109/GCIoT47977.2019.9058396.

[23] S. Murali, and A. Jamalipour, “Mobility-aware energy-efficient parent selection algorithm for low power and lossy networks”, IEEE Internet of Things Journal 6.2 (2018): 2593-2601. https://doi:10.1109/JIOT.2018.2872443.

[24] M. Bouaziz, A. Rachedi, and A. Belghith, “EKF-MRPL: Advanced mobility support routing protocol for internet of mobile things: Movement prediction approach”, Future Generation Computer Systems 93 (2019): 822-832. https://doi.org/10.1016/j.future.2017.12.015.

[25] JNVR S. Kumar, and D. Suresh, “Design and implementation of a mobility support adaptive trickle algorithm for RPL in vehicular IoT networks”, International Journal of Ad Hoc and Ubiquitous Computing 40.1-3 (2022): 38-49. https://doi.org/10.1504/ijahuc.2022.12352.

[26] A. Seyfollahi and A. Ghaffari, “A lightweight load balancing and route minimizing solution for routing protocol for low-power and lossy networks,” Computer Networks, vol. 179, p. 107368, Oct. 2020, https://doi: 10.1016/j.comnet.2020.107368.

[27] D. K. Sah, and T. Amgoth, “Parametric survey on cross-layer designs for wireless sensor networks,” in Computer Science Review, 2018 vol. 27, pp. 112-134, ISSN 1574-0137, https://doi.org/10.1016/j.cosrev.2017.12.002.

[28] S. Parween, and S. Z. Hussain, “A review on cross layer design approach in WSN by different Techniques,” in Advances in Science, Technology & Engineering Systems Journal, 2020, vol. 5, no.4, pp 741-754, https://doi: 10.25046/aj050488.

[29] P. Levis, T. Clausen, J. Hui, O. Gnawali, and J. Ko, “The Trickle Algorithm,” IETF RFC 6206, March 2011.”

[30] S. Vastrad, and K. R. Shobha, “Power Efficient Trickle Algorithm with Dynamic Adjustment of Idle Interval for Internet of Things Application,” in IEEE Sensors Journal, https://doi: 10.1109/JSEN.2023.3341426.

[31] A. Dunkels, “The ContikiMAC Radio Duty Cycling Protocol,” Swedish Institute of Computer Science, Kista, Sweden, 2011.

[32] N. Aschenbruck, R. Ernst, E. Gerhards-Padilla, & M. Schwamborn, “BonnMotion: A mobility scenario generation and analysis tool,” Proceedings of the 3rd International ICST Conference on Simulation Tools and Techniques, 2010., https://doi:10.4108/icst.simutools2010.86.

[33] B. Safaei et al., “Impacts of Mobility Models on RPL-Based Mobile IoT Infrastructures: An Evaluative Comparison and Survey,” in IEEE Access, vol. 8, pp. 167779-167829, 2020, https://doi: 10.1109/ACCESS.2020.3022793.

[34] S. Goyal and T. Chand, “Improved Trickle Algorithm for Routing Protocol for Low Power and Lossy Networks,” IEEE Sens. J., vol. 18, no. 5, pp. 2178–2183, Mar. 2018, https://doi: 10.1109/JSEN.2017.2787584.

[35] Zolertia Z1 datasheet, march 2011,available online http://zolertia.sourceforge.net/wiki/images/e/e8/Z1 RevC - Datasheet.pdf.”

[36] S. S. Solapure and H. H. Kenchannavar, “Design and analysis of RPL objective functions using variant routing metrics for IoT applications,” Wireless Netw, vol. 26, no. 6, pp. 4637–4656, Aug. 2020, https://doi: 10.1007/s11276-020-02348-6.

[37] V. C. Diniesh et al., “Addressing Mobility Based RPL Routing for Low Power Lossy Networks in IoV Networks,” 2024 Second International Conference on Emerging Trends in Information Technology and Engineering (ICETITE), Vellore, India, 2024, pp. 1-6, https://doi: 10.1109/icETITE58242.2024.10493629.

[38] P. Thubert, “Objective Function Zero for the Routing Protocol for Low Power and Lossy networks (RPL),” IETF RFC 6552, March 2012.”

[39] O. Gnawali and P. Levis, “The Minimum Rank with Hysteresis Objective Function,” IETF RFC 6719, Sep 2012.“

[40] N. Hadaya and S. Alabady, “New RPL Protocol for IoT Applications,” in Journal of Communications Software and Systems, vol. 18, no. 1, pp. 72-79, March 2022, doi: https://doi.org/10.24138/jcomss-2021-0094

[41] S. K. Chaurasiya, S. Mondal, A. Biswas, A. Nayyar, M. A. Shah and R. Banerjee, “An Energy-Efficient Hybrid Clustering Technique (EEHCT) for IoT-Based Multilevel Heterogeneous Wireless Sensor Networks,” in IEEE Access, vol. 11, pp. 25941-25958, 2023, doi: 10.1109/ACCESS.2023.3254594.

[42] S. S. Vidhya, S. Mathi, V. Ananthanarayanan and G. Neelakanta Iyer, "IP-RPL: An Intelligent Power-Aware Routing Protocol for Next-Generation Low-Power Networks," in IEEE Sensors Journal, vol. 25, no. 2, pp. 3640-3648, 15 Jan.15, 2025, doi: 10.1109/JSEN.2024.3506816.

Soumya Vastrad is currently working as an Assistant Professor at the department of Electronics and Communication Engineering, BMS Institute of Technology and Management, Bengaluru. She carried out her research on full-time basis at Ramaiah Institute of Technology, Bengaluru, Karnataka, India with the Department of Electronics and Telecommunication Engineering in the university, Visvesvaraya Technological University (VTU), Belagavi, Karnataka, India. She received her Master of Technology degree in Digital Communication from VTU. She is awarded PhD from VTU. Her research interests include Wireless Communication, Wireless Sensor Networks, Internet of Things, and Machine Learning. She is a student member of IEEE.

Shobha K R received her M.Tech. degree in Digital Communication Engineering from Bengaluru University, Karnataka, India, and a Ph.D. from Visvesvaraya Technological University. She is currently working as the Head of the Department and Professor in the Department of Electronics and Communication Engineering at Dayananda Sagar College of Engineering, Bengaluru.Her research areas include Mobile Ad hoc Networks, IoT and Cloud Computing. She has many publications to her credit in reputed conferences and journals. She is a Senior IEEE Member and excom member of IEEE - Bangalore Section, Sensors Council Chapter - Bangalore Section, and IEEE Communication Society - Bangalore Section. She is a fellow of IETE and President of ISoC Bengaluru Chapter.