Cluster Computing (2024) 27:4449–4468
https://doi.org/10.1007/s10586-023-04199-0

RM-RPL: reliable mobility management framework for RPL-based IoT systems

Ali Seyfollahi[1] · Md Mainuddin[2] · Tania Taami[2] · Ali Ghaffari[3]

Received: 25 July 2023 / Revised: 17 October 2023 / Accepted: 1 November 2023 / Published online: 27 November 2023

Abstract
This paper represents the Reliable Mobility Management of RPL (RM-RPL) protocol, specifically developed to overcome the limitations of the Routing Protocol for Low-Power and Lossy Networks (RPL) in mobile IoT environments. RM-RPL incorporates a sophisticated mechanism to prevent the formation of loops, enabling mobile nodes to operate as both routers and parents within the network. It introduces a novel objective function that optimizes the selection of parent nodes and includes a mechanism to adjust the protocol's behavior when nodes are stationary. Furthermore, an algorithm is devised to acknowledge critical packets properly. The proposed model provides superior support for mobility, efficient routing, and dependable data transmission, rendering it highly suitable for diverse IoT applications. Through comprehensive evaluations, RM-RPL demonstrates exceptional performance in challenging scenarios characterized by large-scale networks, high density, and dynamic conditions. Comparative analysis reveals that RM-RPL significantly enhances the packet delivery ratio and exhibits commendable power consumption, end-to-end delay, and handover delay.

Keywords Internet of things (IoT) · Mobility management · Packet delivery ratio · Reliable · RPL routing protocol

1 Introduction

Developing Wireless Sensor Networks (WSN) as a distributed system of ambient sensor nodes owes much to military applications [1]. The Internet of Things (IoT) ecosystem was introduced in 1999 as a heterogeneous set of different communication technologies. It connects broad-spectrum intelligent devices with an embedded network interface to provide a better user experience and service by interaction, sharing information, saving resources, and real-time communication between almost 42 billion things [2, 3]. Radio frequency identification system (RFID) technology enables automatic identification, tracking, and monitoring of objects globally through radio waves, making it a fundamental IoT component [4]. IoT integrates concepts such as autonomous computing, WSN, Pervasive and Mobile Computing, and Cyber-Physical Systems (CPS) [5, 6].

Each thing can contain one or more sensors that sense the environment and use limited resources to process, store, and communicate with other things [7]. Numerous embedded and autonomous devices are reserved for handling multiple tasks and constructing Low-Power and Lossy Networks (LLNs). Node resource constraint is considered one of the significant challenges in LLNs, so many standard Internet communication protocols are not usable and exacerbate the need to lighten or even introduce new protocols [8, 9]. Internet Engineering Task Force (IETF) customized the IPv6 protocol, which has a much wider addressing space for billions of IoT [10, 11] objects than IPv4, thus developing the IPv6 over Low-Power Wireless Personal Area Networks (6LoWPAN) [12] protocol. The IEEE 802.15.4 standard defines two physical and data link layers for IoT. 6LoWPAN is deployed between these two

✉ Ali Seyfollahi
stu.seyfollahi@iaut.ac.ir

✉ Ali Ghaffari
a.ghaffari@iaut.ac.ir

[1] Department of Computer Engineering, Tabriz Branch, Islamic Azad University, Tabriz, Iran

[2] Department of Computer Science, Florida State University, Tallahassee, FL, USA

[3] Department of Computer Engineering, Faculty of Engineering and Natural Science, Istinye University, Istanbul, Türkiye, Turkey

layers after compressing IPv6 data packets [13, 14]. Conventional IPv6-based routing mechanisms [15] are primarily intended for personal computers [2], not LLN. So, attempts were made to introduce a new protocol for 6LoWPAN networks [16]. The IETF ROLL (Routing Over Low-power and Lossy Networks) developed the RPL protocol for LLNs, which is highly compatible with LLN nodes' resource constraints, high loss rates, link instability, and low data rates [17].

Mobility means overcoming the location dependence of the addressing mechanism and effectively exchanging data from anywhere in the wireless network [18]. Intelligent entities like vehicles and wearables are highly mobile in mobile scenarios. Many IoT services depend on user mobility, leading to problems such as disconnections, delays, and loss of user data. The dynamic topology that enables node mobility in the network was not considered in the RPL's RFC (Request For Control), meaning that RPL is suitable for static node-based applications [19, 20]. Detecting out-of-coverage nodes and fast connecting to the new parent are two crucial challenges in discussing RPL mobility. Recent developments have been introduced for RPL that provide node mobility. The Reverse Trickle Timer-based RPL (RTT-RPL) mechanism [21] performs better in different scenarios and represents a higher Packet Delivery Ratio (PDR). It also has lower overhead, power consumption, and End-to-End Delay (E2ED) than standard RPL. But this mechanism also has several disadvantages. First, only leaf nodes are considered mobile, not parents or routers. Second, RTT-RPL does not provide a mechanism for selecting the optimal parent based on a suitable Objective Function (OF) for mobility. Third, this mechanism's assumption to reduce Mobile Node (MN)'s stability and increase the probability of MN leaving the parent coverage is not necessarily correct. Finally, there is no guarantee of delivering critical packets for applications such as smart hospitals.

Therefore, this study aims to identify the strategies to eliminate the RTT-RPL mechanism's weaknesses while maintaining efficiency. In the proposed RM-RPL protocol, this research presents a new way to avoid RPL loops to provide the infrastructure so that MNs in the network can be selected as the parent of other nodes. A mechanism is also proposed to change the protocol behavior to reduce the protocol overhead when MNs are fixed at regular intervals. Finally, a solution is provided to ensure critical packet delivery on the network. The RM-RPL protocol has a higher delivery ratio for data packets than the RTT-RPL mechanism and RPL. Also, in the case of critical packets sent to the network by MNs, the delivery rate for the proposed protocol is much better than RTT-RPL, which indicates that the proposed solution provides a good guarantee for the delivery of critical packets. E2ED, power consumption, and handover delay for RM-RPL are consistent or better than RTT-RPL, indicating that it has eliminated RTT-RPL weaknesses while maintaining proper performance.

The rest of this paper is organized as follows. Section 2 is devoted to the basic concepts required and introduces some features of the RPL routing protocol. The latest schemes for improving mobility management in RPL are reviewed in Sect. 3. Section 4 formulates the system design, the RTT-RPL and standard RPL mechanisms' weaknesses, and the proposed mechanism for loop prevention in mobile parent nodes. Section 5 states our RM-RPL proposed protocol. In Sect. 6, after introducing the simulation setup and evaluation metrics, the RM-RPL is compared with RPL and other methods under different scenarios. Finally, Sect. 7 concludes the study, and suggestions for the future are made to expand the proposed mechanism.

2 Research background

This section first describes the essential features of the RPL routing protocol. The following discusses the mechanism and algorithm of RTT-RPL and its disadvantages. At the end of this section, we review some of the most recent and essential research works that focus on mobility management in RPL.

2.1 Routing protocol for low-power and lossy networks (RPL)

RPL is a proactive, distance-vector, tree-like collector, and destination-oriented routing protocol that forms a directed acyclic structure called Destination-Oriented Directed Acyclic Graph (DODAG) between nodes [22]. One or more DODAGs can exist in the network, members of an instance with a unique ID. Each node can also be a member of several instances simultaneously, but one DODAG within each instance. The objective function (OF) is defined based on constraints, network topological properties, and metrics such as link properties, hop counts, etc., identifying this DODAG. In each DODAG, data is generated by host or leaf nodes, and after routing by router nodes, it is sent to the root or LLN Border Router (LBR) that aggregates the traffic and forwards it to the Internet [23, 24].

RPL uses a rank mechanism that describes each node's position relative to the others in DODAG to detect and prevent network resource loops. Nodes close to the root have a lower value in rank and are a better candidate to choose as the preferred parent for the joining nodes [25]. Each node holds the preferred parent and the other

candidates' information in its routing table. RFC 6550 introduces two default OF0 and The Minimum Rank with Hysteresis Objective Function (MRHOF) objective functions. OF0 calculates the rank based on the information received from the DODAG Information Object (DIO) packets and MRHOF using Expected Transmission Count (ETX). It obtains the best route based on the link characteristics and the number of successful submissions [26].

2.1.1 RPL control messages' structure

The RPL control packets' structure is based on Internet Control Message Protocol for the Internet Protocol Version 6 (ICMPv6), where each packet has a header and the main body section. The Type field contains the "155" value for control packets. The Code field describes the control packet type (DODAG Information Solicitation (DIS), DIO, Destination Advertisement Object (DAO), and DAO-ACK), and the Checksum field controls the error [27]. Each node starts broadcasting the DIS packet when connecting to the network structure or receiving a new root configuration. Neighbors, in response, send a DIO packet to it. The root node also starts to broadcast DIO when constructing a DODAG. Nodes stated in the root coverage receive the DIO, update their routing table information, and re-broadcast. Each DIO contains network-updating data, and the recipient can decide to connect to a new instance or parent.

The joining node dispatches a DAO to its current parent towards the root. DAO receivers discover upward paths in DODAG by entering their addresses and sending the DAO packet to the root. Intermediate routers store DAO information if it is configured to be in storing mode. Otherwise, it will be ignored. Point-to-point communication between two nodes is possible if the DODAG is constructed as a storing trend. There is no need for root involvement in routing in storing mode, and intermediate nodes can help the forward source data to the destination by preserving route information [17]. Otherwise, the root only retains the route data. The DAO receiver or root, in turn, dispatches a DAO-ACK to the DAO transmitter if the network is storing or non-storing [28].

2.1.2 The DODAG formation in RPL

The DODAG formation process in RPL begins with broadcasting the DIO messages by the root. If the DIO received by the neighbors follows the metric defined in the RPL and the new rank calculated based on the received rank is better than the current rank of the node, the routing table is updated, and the node is joined to the new parent. Finally, the receiver re-broadcasts the DIO. Otherwise, it maintains its current position in DODAG. The packet will be dropped if the advertised DIO metric does not match the receiver's metric. Then, the connection between the nodes and their preferred parent and the DAO exchanges with the root is completed. Thus, the path upwards to the root is created, and the DODAG is completed. Figure 1 shows the DODAG construction process based on control packets exchanged in the RPL.

2.1.3 RPL maintenance and repairing mechanisms

The trickle timer mechanism [29] is used as one of the embedded RPL designs to control the frequency of DIO packets. This scheme works well in static networks, but its application in RPL mobility management scenarios is challenging. Equation (1) produces the minimum value of this timer ($I_{minimum}$) in the nth interval, the value at the start of the algorithm. If the network is stable, this value is doubled (I_d) and increases exponentially to reach the maximum value ($I_{maximum}$), according to Eq. (2) [30]. Again, when any network incompatibility occurs, its value is reset to $I_{minimum}$. These values are adjustable according to the type of application and requirements. A balance must be struck between selecting the minimum or maximum values to increase response time or improve energy consumption [31, 32].

$$I_{minimum} = 2^n \tag{1}$$

$$I_{maximum} = 2^{n+I_d} \tag{2}$$

RPL also uses local and global mechanisms to troubleshoot the network. First, it tries to replace the defective route with another. Otherwise, the root's global repair starts, and the DODAG structure is restored. RPL uses

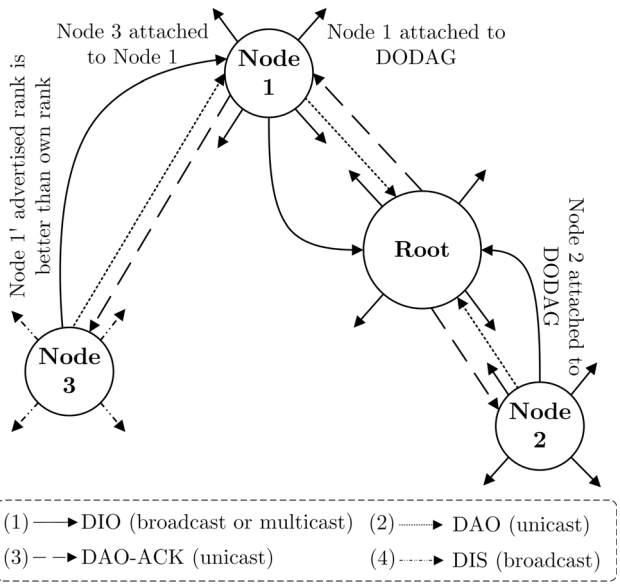


Fig. 1 The DODAG construction in RPL

three mechanisms to avoid the loop: (1) preventing a parent selection with a higher rank for each node, (2) announcing the infinite rank to others for not selecting it as a parent and avoiding the loop, and (3) using the Max-depth rule and do not selecting a parent with a rank higher than the lowest rank of each node and fixing the infinity count when loops occur. Also, by adjusting the flag control values in each packet and changing them intermittently, RPL partially avoids getting stuck in the loop, moving the packets forward correctly, and detecting some loops [33, 34].

Algorithm 1 Pseudocode of the RTT-RPL mechanism

```
Begin
If (DAO_delivered && MNF==1) Then
    mobile_node = True_value;
EndIf
While (mobile_node == True) Do
    I_temp = I_max;
    While (I_temp/2 > I_min) Do
        I = Select a random value between [I_temp/2, I_temp];
        Configure the next DIO using the interval I;
        I_temp = I_temp/2;
    EndWhile
    I = Select a random value between [I_min, I_temp];
    DTSN++;
    Configure the next DIO using the interval I;
    If (!DAO_delivered || DAO_delivered && MNF == 0) Then
        mobile_node = False_value;
    EndIf
EndWhile
Reset to the standard RPL trickle algorithm;
End
```

2.2 Mobility limitations in RPL and an overview of the reverse trickle mechanism and its problems

RPL suffers from poor mobility management support [35] in the DODAG structure. First, node mobility causes it to lose connection with its parent temporarily. Thus, MN begins to discover potential neighbors by sending new DIS. At the same time, no specific timing or interpretation of DIS and DIO exchanges for mobility management is provided in RFC 6550. Second, MN is disconnected from the network until it receives a new DIO that advertises a better rank or DODAG topology changes. Therefore, long delays may be imposed on the network to mobile and even static nodes. Third, the RPL documentation did not clearly describe when the preferred parent or candidates were removed from the routing table due to mobility. Therefore, RPL is generally incapable of high-mobility scenarios because it slowly detects the inaccessibility of MNs to the network structure until it receives a proper DIO, resets the rank, and excessive MN delays.

The RTT-RPL mechanism [21] tries to solve the inefficient RPL mobility management by quickly detecting node mobility, reducing delays, and introducing a new trickle schedule. In this method, MNs should declare their mobility status in DAO packets by announcing a Mobile Node Flag (MNF) control flag. When the parent detects node mobility, it immediately changes its timer to RTT. The thinking behind this approach is based on the fact that the timer initially has the highest value (I_{max}). Over time, the increased probability of the MN leaving the range decreases exponentially rather than detecting another movement due to resetting the timer to I_{max}. The RTT-RPL mechanism imposes less overhead by involving only the nodes in the MN range. Only the leaf nodes are considered mobile to prevent looping. They don't advertise a DIO not to be selected as a parent. Algorithm 1 shows the RTT-RPL mechanism. The parent starts a reverse trickle algorithm by detecting an MN after receiving a DAO packet containing the flag $MNF = 1$.

Initially, the timer value is I_{max}, and after each DIO sending, its value is halved to reach I_{min}. Once the timer is I_{min}, the parent increases the Destination Advertisement Trigger Sequence Number (DTSN) value and requests a new DAO from MN. If the MNF value is still *1*, the timer is reset to I_{max}. Otherwise, the algorithm will switch to the standard RPL trickle timer. An MN can limit the maximum delay time by defining a DIO threshold from a new or current parent. Once the threshold is met, MN removes its present parent and ranks it infinitely. Therefore, it waits to receive a new DIO with better features by sending a new DIS.

3 Related work

RPL suffers from the rapid detection of parent unavailability mechanisms in mobility scenarios, which depends on external solutions. These mechanisms, such as neighbor detection in IPv6 [36] or Media Access Control (MAC)-based schemes [37–39], are complex and do not satisfy acceptable performance in high mobility [40]. Also, selecting an MN as a parent can lead to loops in the network.

Using the Corona mechanism presented in [41], O. Gaddour et al. [42] try to provide Quality of Service (QoS) based on reliability, latency, and energy while improving mobility support for RPL. The proposed Co-RPL mechanism divides the network into coronas to implement mobility. The parent selection is based on the Link Quality Indicator (LQI) and the maintenance of the hop-count metrics. DIO exchanges with the discovered neighbor are carried out quickly regardless of the trickle mechanism. Also, a few repairing mechanisms have been proposed to replace the defective route with a new one. Y. Tahir et al. [43] proposed an adaptive backpressure-based mechanism

for RPL that calculates a weight metric based on node queue backlog, link quality, and channel capacity and considers it in parent selection and DODAG construction. It can switch from RPL to backpressure routing in heavy traffic loads and node mobility scenarios. H. Kharrufa et al. [44] proposed a new trickle mechanism based on the Received Signal Strength Indicator (RSSI) to support mobility in dynamic scenarios for diverse applications. Their "D-RPL" solution optimally detects mobility and inconsistency in the network by designing a new OF for parent selection and controlling overhead and loops.

M. Bouaziz et al. [45] propose an RSSI-based solution for predicting the following location of nodes in the mobility process and frequent monitoring of the distance between them. Their proposed mechanism optimizes connections between MNs in healthcare scenarios and reduces energy consumption. If the communication quality between nodes is high, packet losses are lower, and they feel less need for mobility and parent switching. Relying on this knowledge, S. Hoghooghi and R. Javidan [46] try to force the parent to monitor the connection quality with its child node, energy consumption, and overhead. In their proposed solution, like the idea presented in [45], MN replaces its parent based on the residual energy, ETX, and RSSI. A new memory maintains a stable parent list. K. Manikannan and V. Nagarajan [47] used the Firefly algorithm [48] to optimize mobility management and energy consumption in RPL. Firefly falls under the categories of swarm intelligence, metaheuristics, and nature-inspired algorithms [49, 50]. The proposed mechanism optimizes the preferred parent selection by predicting random node movement, energy consumption, Expected Lifetime (ELT), RSSI, and ETX metrics. The cross-layer mechanism named "MARPL" proposed by J. Kniess and V. de Figueiredo Marques [51] seeks to prevent parent inaccessibility by monitoring RSSI changes using the MAC layer to detect neighbors' mobility and calculate mobility based on their variability. The main idea is to provide a reactive pattern to reduce the control overhead when the node is in its parent range.

B. Safaei et al. [52] conducted a comprehensive experimental study to assess the impact of various mobility models on the performance of a mobility-aware RPL. The study evaluates the network performance and IoT devices in mobile RPL-based applications, considering different mobility models from perspectives such as power consumption, reliability, latency, and control packet overhead. The findings aim to guide researchers in designing application-specific or standardized versions of RPL suitable for mobile IoT scenarios. Software-defined networking (SDN) has excellent potential to outsource heavy algorithmic computing to controllers, better monitor the network view, and predict node mobility. A. Mohammadsalehi et al. [53] believe that the criteria proposed for optimal mobility management in RPL do not lead to selecting sustainable paths in the long run. Random parent selection in RPL exacerbates this problem. They use a new Time-to-Reside metric to estimate non-stochastic and more reliable routes, which has improved reliability by more than 2.5x and 4x compared to [54] and standard RPL, respectively. The proposed mechanism by S. Murali and A. Jamalipour [54] offers a new "D_Trickle" timer to solve the long listening problem and assign it in a dynamic and random scheme to mobile nodes. They optimize [55–57] energy consumption and maximize parent selection based on RSSI, ETX, distances between the nodes, and ELT.

By focusing on this advantage, I. Rabet et al. [58, 59] seek to improve the node connection quality in mobility scenarios, keep routing information up to date, reduce control overheads, and provide more accurate localization by monitoring RSSI changes using the Particle and Kalman filter to identify reliable links. T. Hussain et al. [60] propose an innovative strategy to enhance source location privacy. They introduce a hybrid phantom technique that combines phantom nodes with multi-path routes, aiming to achieve more robust privacy protection while minimizing energy usage. The phantom nodes are selected using the Analytic Hierarchical Process (AHP), considering factors such as energy level, distance, heterogeneity, and neighbor list. The results demonstrated the successful implementation of this approach in safeguarding the confidentiality of source location data within social IoT networks.

Almost all outlined solutions for improving mobility management in RPL suffer from the lack of critical data for reliability-related applications such as healthcare. The proposed RM-RPL mechanism seeks to provide a reliable solution for IoMT-based healthcare applications by designing a new model.

4 System model

The four weaknesses of RTT-RPL are: (1) In RTT-RPL, it is assumed that MNs cannot be selected as the parent because loops may be created in the network. This assumption imposes a relatively significant constraint, and it is better to provide a mechanism to avoid the loops so that MNs can also play the role of routers. (2) In RTT-RPL, the proximity of that node is not considered and is based on its rank and ETX; in other words, when an MN in RTT-RPL realizes that it is no longer within its parent range. RTT-RPL does not provide a mechanism for selecting the optimal parent and suffices with the OF provided in the RPL. (3) RTT-RPL has assumed that MN's stability probability decreases over time. In other words, MN leaving probability from the current parent range increases

over time. However, this assumption is not always valid because the node may be exceedingly mobile initially but then remain stable for a long time. For example, suppose a patient in a hospital is present in the network as an MN but takes long breaks after a period of mobility. (4) Neither RPL nor RTT-RPL guarantees the reception of critical packets. Consider an intelligent hospital with patients who are considered as MNs. It may even cause the patient to die if critical packets about their health are not delivered to the server. The proposed RM-RPL model addresses these weaknesses and provides a reliable solution for delivering critical packets for IoMT-based healthcare applications.

In standard RPL and RTT-RPL, if the node is mobile, its rank changes to infinite, so it cannot be selected as a parent. However, it will choose any other node, including its child nodes, as a parent, leading to loops. RM-RPL first sets a flag bit for each of its neighbors to determine if it is a child. Second, children are prohibited from sending a DIO to a mobile parent DIS. These two mechanisms are complementary. Suppose we prevent children from sending DIOs to their previous mobile parents. This is prone to attack because the malicious node may intentionally send DIO packets to its previous parents, increasing the likelihood of loops. Also, the second mechanism prevents additional overhead in the network.

In RM-RPL, a parent MN sends a DIO if it receives DIS from another MN, so it cannot select an MN as its parent if it remains static. This phenomenon will make the network more stable and reduce unnecessary delays. If an MN is the parent of another, it uses RTT to send DIOs. Otherwise, sending DIO in it is subject to receiving DIS from MN. In Fig. 2, in the left DODAG, which belongs to the RTT-RPL protocol, MN gives up its previous parent range, node 1. Because MN has not received a DIO from its parent for a long time, it changes its rank to infinite. It selects its child, node 6, as the new parent, forming a loop. On the right DODAG, which belongs to RM-RPL, node six is registered as a child in the MN's routing table and is no longer selected as the new parent. Node 3 is considered the new parent, and the loop is prevented.

5 Reliable mobility management of RPL (RM-RPL)

The proposed RM-RPL protocol provides a simple mechanism to prevent network loops if MN is selected as a parent. It designs a new OF for optimal parent selection, modifies protocol behavior when MN has no mobility to optimize power consumption, and ensures delivery of critical packets. Our scheme first suggests a novel OF to select the preferred parent. In the following, we will talk about changing the static behavior of the node and discovering the mobility. Then, the RM-RPL ensures the delivery of critical packets to the server and introduces the critical packet structure. Finally, it examines the algorithm for guaranteeing the delivery of critical data and the optimal selection of its parameters.

5.1 A novel OF for optimal preferred parent selection

The RTT-RPL does not define an efficient OF for the optimal selection of a new parent for MN outside the parent range. RM-RPL offers a new OF based on how close each MN is to its neighbors and its length within its parent coverage. RM-RPL, such as RTT-RPL, distinguishes between a mobile and static node in transmitted DIS by assigning one value to the *MNF* bit. It then asks the neighbors to send a five-times DIO [61]. According to Eq. (3), MN waits as much as t_d to receive the DIO from the neighbors and inserts the RSSI packet for each received DIO. The t_d value is adjustable depending on the network topology.

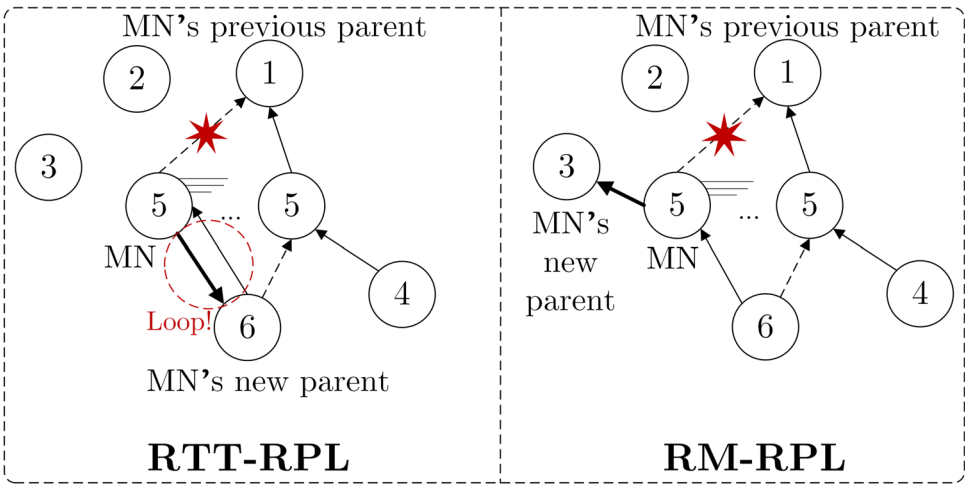


Fig. 2 Loop prevention mechanism in RM-RPL

$$t_d = MaxOneHopDelay \times 5 \quad (3)$$

Then, the average RSSI of the five received packets for each neighbor (*AvgRSSI*) is calculated. The lower the value, the lower the chances of a neighbor becoming a parent. *MaxOneHopDelay* is the maximum one-step packet hop calculated using offline network execution. Algorithm 2 is used to obtain the RM-RPL OF.

5.2 Detecting node mobility and changing its behavior when it is static

One of the RTT-RPL weaknesses is that the node is always considered mobile throughout the algorithm. But suppose in a hospital scenario, the patient is not always mobile and is resting for a long time. RM-RPL changes the behavior of the MN node at static times. If MN stays still for more than t seconds, the *MNF* bit will be declared zero in DAO and DIO packets to the neighbors to prevent the parent from sending unnecessary DIOs to maintain the routes. Once mobility starts, this bit is reset to one again and broadcasted through DIO packets.

Algorithm 2 Pseudocode of the proposed objective function

```
Begin
send DIS with mobility flag 1;
wait for td second to receive 5 DIOs from neighbors;
For each neighbor n
    AvgRSSIn = averageOf 5 received DIO's RSSI from n;
EndFor
Choose preferred parent with minimum AvgRSSI;
End
```

5.3 Guaranteeing the delivery of critical packets to the server

Unlike RTT-RPL, RM-RPL offers an algorithm to ensure the delivery of critical packets to the server with minimal delay. If MN is out of parent coverage, not connected to the new one, or is in the parent range but unavailable, packet losses occur. In this section, we first describe the critical packet's structure and then the algorithm.

Algorithm 3 Pseudocode of receiving a critical packet in a node

```
Incoming CRITICAL packet Algorithm
Input: Received Data Packet: DataPacket, Sender router: S
Begin
If (DataPacket.CRITICAL = 0)
    Use the normal procedure for data packets;
Else
    Put DataPacket in priority and send it before any regular packets;
    Create a CRITICAL-ACK packet and send it to S;
    Go to "Forward CRITICAL packet algorithm" with CRITICALDataID as an Input argument;
EndIf
End
```

5.3.1 The structure of critical packets and acknowledgment

Figure 3 shows the structure of critical data packets in RM-RPL. The *CRITICAL* flag is used to detect critical packets. The one-byte *CRITICALDataID* field expresses the packet ID number. The *RPLInstanceID* field is the RPL instance identifier. The *Sender Rank* is set to zero for the data transmitter, and each router is assigned its rank. The *O* flag's value for critical packets is set to 0 because, in healthcare applications, the data is usually sent upward to the root or server and is assigned to one value for downward paths. In a rank error or a duality between the *O* bit and the *Sender Rank*, the *R* flag is set to 1. The *F* flag is also involved in detecting DAO packet inconsistencies. Figure 4 shows the acknowledgment structure for critical packets or CRITICAL-ACK.

5.3.2 Algorithm for guaranteeing the receipt of critical packets

Initially, MN assigns 1 to the *CRITICAL* flag in the packet header, and the parent prioritizes sending the packet with this attribute. This priority continues to the root so that the packet is received step by step on the route and received at the destination. Finally, MN will ensure the successful delivery of the packet by receiving the CRITICAL-ACK and reading its ID. Algorithm 3 shows this process.

MN waits for t_c until receiving the acknowledgment. If there is a problem in the path and it is lost, MN will retransmit it, and the critical packet repetition number (*CRITICALCounter*) for that packet will increase. If the acknowledgment is still not received during the t_c, the counter will meet the *CRITICALCounterMax* threshold. The t_c parameter can be expressed as Eq. (4). DTx_{CPR} is the E2ED of critical packet delivery by root, DRx_{CACK} is the E2ED of critical packet acknowledgment delivery by the transmitter, and rt_n is the times the critical packet sender retransmits the packet is configurable by the network designer.

$$t_c = (DTx_{CPR} + DRx_{CACK}) \times rt_n \quad (4)$$

Suppose no acknowledgment is received after the threshold. MN immediately broadcasts a DIS, and the first

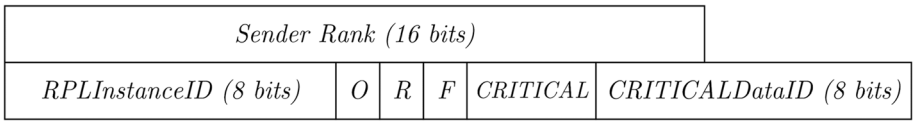


Fig. 3 Structure of data packets with the possibility of defining critical packets

DIO sender is directly selected as the preferred parent to send the critical packet. Algorithm 4 describes this process, and the algorithm continues until the packet reaches the final destination or root to ensure critical packet delivery.

Algorithm 4 Pseudocode of forwarding a critical packet to the preferred parent

```
1:  Forward CRITICAL packet algorithm
2:  Input: Emergency packet: CRITICALPacket, Emergency packet ID: CRITICALDataID, next hop (preferred parent): P
3:  CRITICALCounter = CRITICALCounterMax;
4:  Begin
5:  Forward CRITICALPacket to next-hop P;
6:  While (not received CRITICAL-ACK from P)
7:      Wait for t_c seconds and reforward the CRITICALPacket;
8:      CRITICALCounter--;
9:      If (CRITICALCounter == 0)
10:         While (no new preferred parent selected yet)
11:             Send broadcast DIS;
12:             Wait 1 second for the incoming DIO to select a new preferred parent;
13:         EndWhile
14:         Go to "Forward CRITICAL packet algorithm" with the new next-hop;
15:     EndIf
16: EndWhile
17: End
```

5.3.3 Proper selection of algorithm parameters

In the previous section, two primary and configurable parameters were introduced: (1) t_c, which is how long the sender of the critical message has been waiting to receive the *CRITICAL-ACK*, and (2) *CRITICALCounterMax*, which is the maximum number of attempts to retransmit the critical message.

If t_c is selected too large, the message transmission delay to the destination will be increased. In a small case, *CRITICAL-ACK* may be received after this time, which is not valid because the packet is retransmitted with a new ID or the node is looking for a new parent. Also, if the *CRITICALCounterMax* is large, there may be no chance of receiving a *CRITICAL-ACK* due to re-transmissions resulting from out-of-coverage of the preferred parent. Also, the energy overhead will increase with futile re-transmission attempts, and the delivery delay to the destination will increase. The node may still be within its parent range in the small case. Still, the parent is temporarily unable to receive the message due to congestion or problems, or the *CRITICAL-ACK* sent by it is lost due to topology changes. If it is small, the node changes the parent after much effort. In other words, the number of parent switches in the network increases, but the delivery delay decreases because MN does not wait for its parent's problem to be resolved.

In networks with high data rates, packet latency increases hop by hop, so a larger t_c must be selected. Vice versa, if the data rate is low, the t_c becomes smaller. If the packet loss ratio in the network is high, the critical packet will not be delivered to the parent, or the *CRITICAL-ACK* sent by the parent will be lost. Therefore, selecting the *CRITICALCounterMax* parameter with a higher value in these networks is better.

Fig. 4 The structure of the CRITICAL-ACK packet

5.4 Case study

Consider the scenario in Fig. 5. The MN has been outed from its parent coverage, node 1, over a distance. Now, MN sends a critical packet to the root (server). MN waits to receive *CRITICAL-ACK* for t_c configured for 100 s. Because MN is out of the parent range, it does not receive the acknowledgment packet. So, MN retransmits the critical packet for two times *CRITICALCounter*. Because, after this threshold, it fails to receive the ACK again and broadcasts a new DIS. Node 2 on the MN coverage

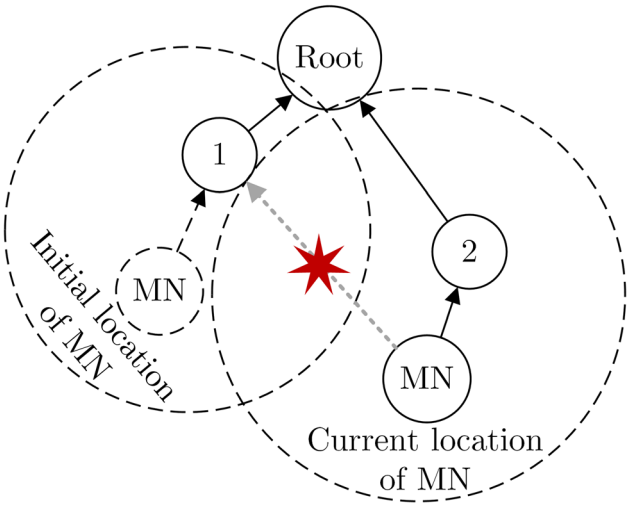


Fig. 5 A scenario for exchanging a critical message

receives the DIS and, in response, sends MN a DIO. MN selects Node 2 as the preferred parent and retransmits the critical packet. Node 2 prioritizes forwarding the critical message. It delivers this packet to its parent, the root, and waits to receive the DAO-ACK. After receiving the critical message, the root also sends a CRITICAL-ACK to Node 2. MN gets the ACK, and the critical packet delivery to the server is guaranteed.

6 Performance evaluation

The simulations in this research were performed using the Cooja simulator [62], which runs on the Contiki operating system [63]. Contiki supports IPv4, IPv6, 6LoWPAN, and RPL standards. Many Contiki-based systems are admirably low-power, and the Cooja provides node mobility through the mobility plugin.

6.1 Simulation setup

We use Tmote Sky nodes in the simulation process, and the UDGM model simulates radio communications. The coverage radius of each node is assumed to be 50 m, and the Random Waypoint Model (RWM) [64] simulates the displacement and velocity of the nodes over a time interval. In this popular and straightforward random routing model, each node moves linearly to a random destination at a minimum and maximum speed after stopping for a specified period. This behavior is repeated throughout the simulation runtime. Figure 6 shows an example of the random behavior of this model for the MN node at hypothetical surface A.

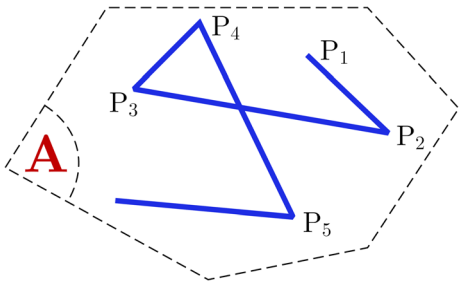


Fig. 6 An example of an RWM mobility pattern

Table 1 shows the common parameters between all simulations. This paper evaluates smart hospital scenarios in which nodes transmit normal data packets to the server at fixed *20*-second intervals and critical packets at *1-min* intervals. According to studies performed offline on different networks, the value of *200 (ms)* for the t_c parameter, i.e., the maximum hop-by-hop delay in different networks, has been selected. The value of $t_d = 5 \times 200\ (ms) = 1\ (s)$ is assumed. Also, considering the *5%* packet loss probability, the $CRITICALCounterMax = 1$ is configured. In each scenario, some fixed nodes are arranged so that the two adjacent nodes are within the communication range of each other, and the entire area covered by the static nodes is rectangular.

6.2 Performance metrics

The performance evaluation metrics analyzed in this paper are Power consumption, PDR, E2ED, and handover delay, which we will introduce.

6.2.1 Power consumption

PowerTrace plugin for Cooja calculates the power usage for each node, which expresses the average clock ticks spent processing, sending packets, and listening. According to Eq. (5), *Voltage* and *Current* are obtainable from the Tmote sky catalog. *RTIMER* denotes the clock ticks per second. *Runtime* stands for the simulation time in seconds. *Energest_Value* can describe three parameters: TX and RX, the ticks that the node's radio chip was sending or receiving, and the CPU, the ticks that the node was performing computational processes. Finally, n denotes the total number of nodes.

$$Power\ consumtion = \sum_{i=1}^{n} \left(\frac{Energest\_Value \times Current \times Voltage}{RTIMER \times Runtime} \right)_i \times \frac{1}{n} \tag{5}$$

Table 1 Simulation parameters

Parameter	Value
Sensing area	200 × 200 m
Mobility model	RWM
Simulation environment	Contiki/Cooja v3.0
Node stop time	10 min
The lowest speed of node mobility	1.5 m/s
The highest speed of node mobility	5 m/s
Mote type	Sky mote
Simulation time	1 h
Radio interface	UDGM Distance-loss
Radio	CC2420
Transmission range	50 m
Payload size	40 bytes
Data packet transmission ratio	20 s
Critical packet transmission ratio	60 s
Tx/Rx success ratio	95%
Network layer protocols	µIPv6 and RPL
Transmission layer protocol	User Datagram Protocol (UDP)
t_c	200 ms
t_d	1 s
CRITICALCounterMax	1

6.2.2 Packet delivery ratio

Equation (6) expresses the average PDR, where *DP* is the total successful packets delivered to the receiver and *TP* is the total packets sent by the transmitter.

$$PDR = \sum_{i=1}^{n} \frac{DP_i}{TP_i} \times \frac{1}{n} \tag{6}$$

6.2.3 End-to-end delay

Equation (7) describes the average E2ED, the time interval between received and transmitted packets. The *RT* and *TT* parameters are every packet's receiving and transmitting times. The *m* and *n* denote the total sent packets per node and the total nodes, respectively.

$$E2ED = \frac{\sum_{i=1}^{n}\sum_{j=1}^{m}\left(RT_j^i - TT_j^i\right)}{m} \times \frac{1}{n} \tag{7}$$

6.2.4 Handover delay

This parameter indicates the time interval between the mobile node (MN) leaving the current parent range and the connection to the new parent. This time is calculated by the position of the nodes and the distance to the preferred parent, which is recorded in the logs. Also, the connection time is recorded for all mobile nodes. In Eq. (8), the parameters *ct* and *et* indicate the connection to the new parent and exit from the previous parent coverage. The *q* also represents the total number of mobile nodes.

$$Handover\ Delay = \frac{\sum_{i=1}^{q}\left(ct_{MN}^i - et_{MN}^i\right)}{q} \tag{8}$$

6.3 Results and discussion

In this paper, simulations are performed during three different experiments. The proposed RM-RPL mechanism performance is evaluated compared to the standard RPL, RTT-RPL [38], D-RPL [44], Hoghooghi and Javidan (HJ-RPL) [46], and EMA-RPL [45] under four metrics introduced in Sect. 5.2. We designed three experiments based on the several network sizes, the distances between nodes, and the network dynamics. The static and mobile nodes are deployed in different scenarios based on the simulation parameters introduced in Sect. 5.1.

6.3.1 Network size analysis

The first experiment investigates the effect of network size on evaluation metrics. Approximately 20% of the nodes are mobile, and the fixed nodes are deployed at 40 m. The coverage area of each node is also 50 m. Table 2 defines the exact number of nodes in each simulation. As Fig. 7 shows, the handover delay for all protocols grows with

Table 2 The number of static and mobile nodes in the first experiment

No.	Total number of nodes	Number of static nodes	Number of mobile nodes
Scenario A	11	9	2
Scenario B	20	16	4
Scenario C	30	24	6
Scenario D	38	30	8
Scenario E	50	40	10
Scenario F	60	48	12
Scenario G	70	56	14
Scenario H	80	64	16

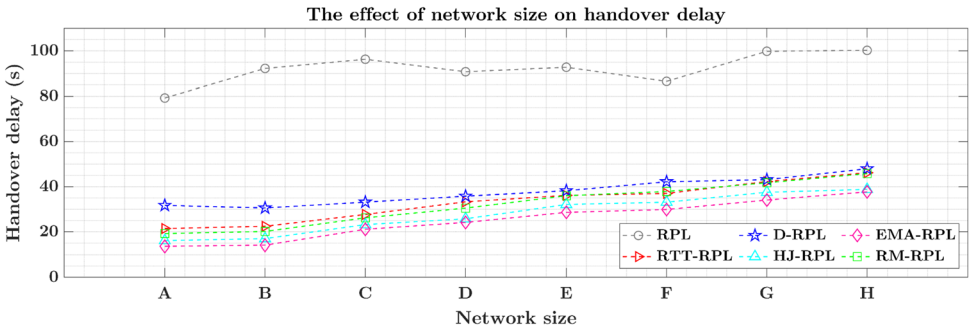


Fig. 7 Effect of network size on handover delay

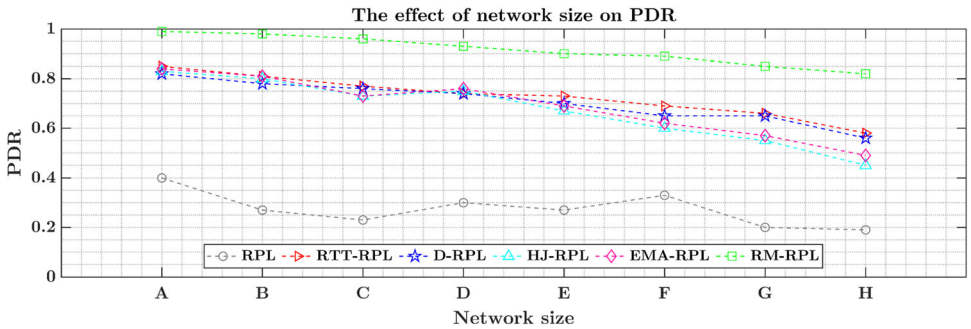


Fig. 8 Effect of network size on PDR

increasing network size, which is evident. Because the more nodes mobility, the more handover delays occur in the network. RPL experiences more handover delays due to a lack of mobility management mechanism and slow response to network changes. RM-RPL is roughly consistent with RTT-RPL, as the proposed mechanism does not change how MNs detect out-of-parent range detection.

As the number of nodes increases, especially mobile nodes, because D-RPL and RTT-RPL do not have a competent OF for optimal parent selection, instability in routes occurs more frequently and, therefore, more handover delays than RM-RPL. HJ-RPL and EMA-RPL share the same idea. They maintain connection continuity when nodes are mobile and transfer data even in handover situations. They face a minor delay than other protocols. As the nodes' count in the system grows, PDR decreases because the incidence of disconnections due to node mobility increases, and the packet loss rate also increases. Figure 8 depicts the PDR variations for the compared protocols. PDR for RM-RPL shows a significant improvement compared to RTT-RPL and other protocols.

The proposed mechanism provides an effective OF for optimally selecting the preferred parent, resulting in fewer MN nodes leaving their parent range and fewer losses. Delivery of critical packets is also guaranteed, which other protocols do not have. RPL experiences much higher losses due to no mobility management and has the worst PDR values. D-RPL uses the RTT mechanism like RM-RPL, but our proposed method only switches to this timer when mobile nodes are. Still, D-RPL changes its parent even in static nodes if the link quality deteriorates based on RSSI. Therefore, this mechanism experiences more losses due to more parent changes and has a lower PDR than RM-RPL. In HJ-RPL and EMA-RPL methods, a static node is always responsible for detecting mobility and reacting to it by changing parents. Therefore, it increases the computational

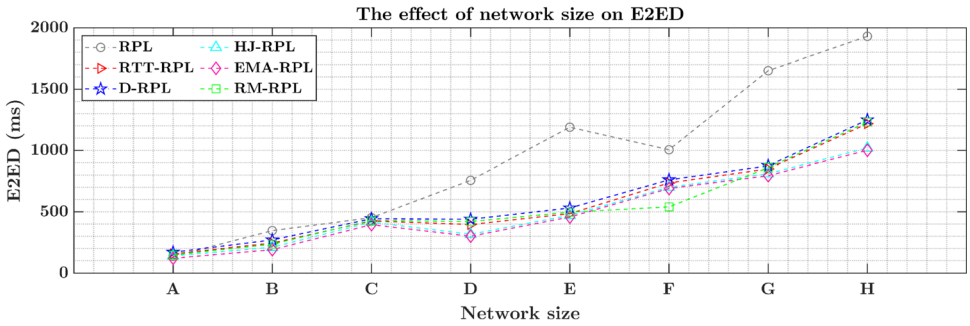


Fig. 9 Effect of network size on E2ED

overhead in these nodes, especially in dense and highly mobile networks. This phenomenon can lead to hotspot points in the network, more losses, and worse PDR.

When network size increases, E2ED deteriorates due to growing the hop counts to the root. Figure 9 presents the E2ED variations for the comparative protocols. E2ED for RPL is higher than other protocols because it realizes communication interruptions are too late due to node mobility. Therefore, the instability of the network topology in RPL is more significant and requires more changes in link replacement, resulting in more delays in delivering packets to the destination. E2ED changes for RM-RPL compared to RTT-RPL are almost the same or slightly longer for the proposed mechanism. In RM-RPL, critical packets have a higher priority for forwarding to the root. It has led to more delays for regular packets. D-RPL experiences approximately similar latencies to RTT-RPL and RM-RPL, indicating a suitable mechanism for managing mobility and routing packets to the root. In some cases, the delays for D-RPL have increased due to more instability of the links and more parent switches.

E2ED for RPL is insignificant in small networks compared to the others, as more nodes are probably in the direct root range. However, as the network size and the nodes' mobility grow, the nodes' distances increase. Due to the reactive nature of detecting the mobility behavior of the nodes, the delays also worsen. HJ-RPL and EMA-RPL are proactive in node mobility management. Due to leaf node selection as MN, these mechanisms experienced less E2ED due to no communication interruptions, parent changes when nodes are mobile, and data transmission even at mobility detection.

Figure 10 shows the effect of increasing nodes on power consumption under evaluation protocols. Power consumption for RM-RPL is slightly higher than RTT-RPL because MNs have to rely on five packets when selecting a parent and calculate RSSI to eliminate possible RSSI errors. Therefore, sending these packets by static nodes to MNs has increased the energy overhead. However, the proper mechanism for changing the MNs' behavior means that the proposed mechanism does not significantly increase power consumption compared to RTT-RPL. Excessive data loss in RPL due to the lack of a mobility management mechanism leads to increased re-transmissions and, consequently, high node energy loss.

D-RPL consumes less than RM-RPL because it does not use the RSSI error neutralization mechanism. But most parent switches in D-RPL increase power consumption, even for static nodes, because they detect a drop in link quality compared to the threshold value. Therefore, the proposed mechanism generally does not significantly increase compared to D-RPL. In EMA-RPL, because the parent node is responsible for detecting node mobility and the system monitors link interruptions and continuous data transmission, it suffers more energy overhead than the proposed mechanism, especially in dense and highly mobile networks. On the other hand, in HJ-RPL, MNs detect and announce mobility to parent nodes. Therefore,

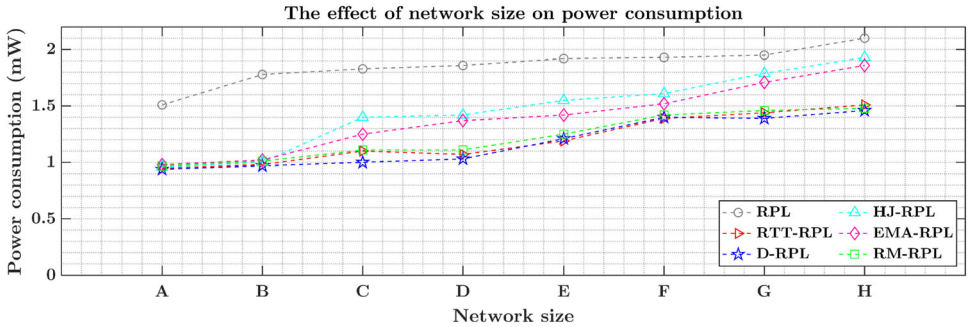


Fig. 10 Effect of network size on power consumption

Table 3 Simulation values were obtained for different protocols in the first experiment (network size)

Scenario	A	B	C	D	E	F	G	H
Handover delay (s)								
RPL	79.22	92.21	96.26	90.74	92.74	94.50	99.81	100.21
RTT-RPL	21.45	22.45	27.68	33.28	36.12	36.99	42.26	46.16
D-RPL	31.70	30.56	33.14	35.78	38.19	42.15	43.10	47.91
HJ-RPL	16.10	16.99	23.18	25.77	32.10	33.11	37.50	38.79
EMA-RPL	13.60	14.10	21.11	24.15	28.61	29.88	34.07	37.68
RM-RPL	19.22	20.16	26.18	30.50	35.97	37.86	41.70	45.77
PDR								
RPL	0.40	0.27	0.23	0.30	0.27	0.33	0.20	0.19
RTT-RPL	0.85	0.81	0.77	0.74	0.73	0.69	0.66	0.58
D-RPL	0.82	0.78	0.76	0.74	0.70	0.65	0.65	0.56
HJ-RPL	0.83	0.80	0.73	0.75	0.67	0.60	0.55	0.45
EMA-RPL	0.84	0.81	0.73	0.76	0.69	0.62	0.57	0.49
RM-RPL	0.99	0.98	0.96	0.93	0.90	0.89	0.85	0.82
E2ED (ms)								
RPL	135	346	452	757	1189	1245	1651	1930
RTT-RPL	156	244	425	396	485	736	850	1218
D-RPL	169	270	443	439	531	761	875	1249
HJ-RPL	139	210	415	316	471	701	811	1019
EMA-RPL	120	190	395	300	457	690	795	1001
RM-RPL	147	235	430	420	497	540	860	1231
Power consumption (mW)								
RPL	1.51	1.78	1.83	1.86	1.92	1.93	1.95	2.10
RTT-RPL	0.95	0.95	1.10	1.07	1.19	1.39	1.44	1.51
D-RPL	0.94	0.97	1.00	1.03	1.21	1.40	1.39	1.46
HJ-RPL	0.96	1.00	1.4	1.42	1.55	1.61	1.79	1.93
EMA-RPL	0.98	1.02	1.25	1.37	1.42	1.52	1.71	1.86
RM-RPL	0.97	1.01	1.10	1.11	1.25	1.42	1.46	1.48

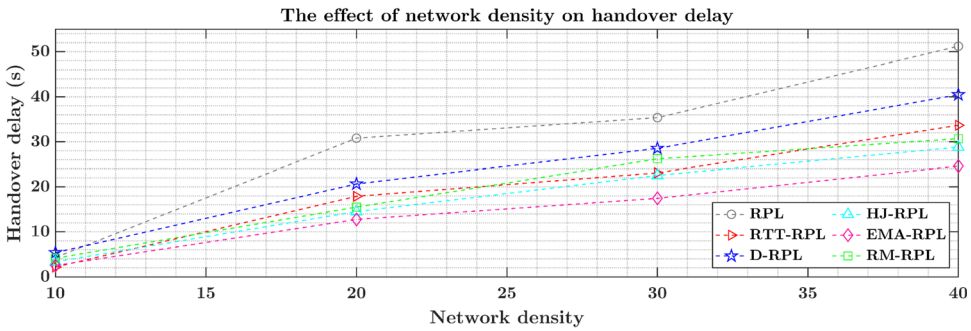


Fig. 11 Effect of network density on handover delay

more network instabilities are experienced, and more losses are imposed on the system. As a result, the power consumption of this protocol has been slightly higher than EM-RPL. Table 3 gives the various simulation outputs for all under-evaluation protocols for the first experiment.

6.3.2 Network density analysis

In the second experiment, the network performance at different densities is studied. Thirty-eight nodes are considered, 30 are fixed, and eight are mobile. Each simulation's distance between static nodes increases from 10 to 40 m. Figure 11 depicts the effect of the rising distance among nodes on handover delay. As the distance increases, this delay heightens in all protocols. When the density

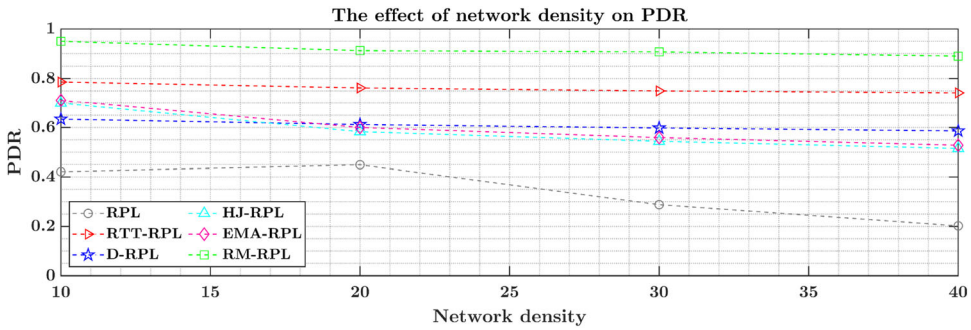


Fig. 12 Effect of network density on PDR

decreases, the number of candidate parents for each node and the likelihood of quickly joining the new parent will decrease. Also, the handover delay in RM-RPL is almost the same as in RTT-RPL because the proposed protocol does not change the MN mechanism to leave the parent range.

In RPL, handover latency is exacerbated by increasing distances between nodes. Lack of mobility management mechanism, slow response time to disconnected connections, and late detection of existing parent unavailability worsen the delay in reconnecting to a new parent at a lower density. The gradual decrease in network density due to increasing distances between nodes, lack of proper OF, and more stability of links in D-RPL cause more handover delay than RM-RPL. HJ-RPL and EMA-RPL also have fewer delays than other methods due to maintaining the cohesion of the connections due to the selection of leaf nodes as MN.

As the distance between nodes increases, the PDR will decrease, as the nodes' chances of quickly selecting a parent decrease and the handover delay rate increases. Figure 12 presents the effect of distance variations between nodes on PDR for different protocols. RM-RPL has better PDR than other methods because it ensures the delivery of critical packets; due to changes in protocol behavior, MNs leave their parent range less, and handover delay is reduced. RPL has the lowest PDR values due to its lack of mobility management. D-RPL has a lower delivery rate than RM-RPL because as the distances between nodes increase, they switch more between their parents, causing more losses. HJ-RPL and EMA-RPL impose more computational load on the monitor node by increasing node distances due to more handovers and packet losses.

Increasing the distance between nodes heightens E2ED in the network as more hops are created to forward the packet to the root. There is more priority in RM-RPL routing critical packets, so E2ED shows similar or slightly higher values than RTT-RPL. Due to the lack of mobility management in the RPL, the greater instability of the network structure, and the late detection of disconnections and node mobility, E2ED will be higher for this protocol than the others. D-RPL has caused similar or slightly longer delays than RTT-RPL and the proposed mechanism because most parent switches occur even in static nodes, which causes topology instability and imposes more delays.

Lower E2ED occurs in EMA-RPL and HJ-RPL due to their efforts to ensure uninterrupted communication when mobile nodes. Figure 13 shows the E2ED changes for under-consideration protocols. Increasing the distance between nodes causes more handover to select a new parent, thus imposing more energy overhead.

Figure 14 presents the impact of rising distance among nodes on power usage in different protocols. RM-RPL experiences more minor power consumption than RTT-RPL in parts of the graph due to sending more DIO packets to eliminate RSSI errors. Due to packet losses and no-mobility detection, RPL experiences more re-transmissions and power consumption. D-RPL does not have an RSSI error neutralization mechanism and, therefore, has less

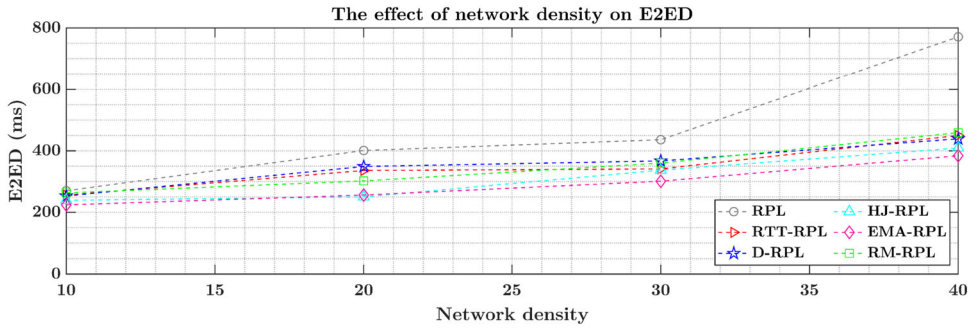


Fig. 13 Effect of network density on E2ED

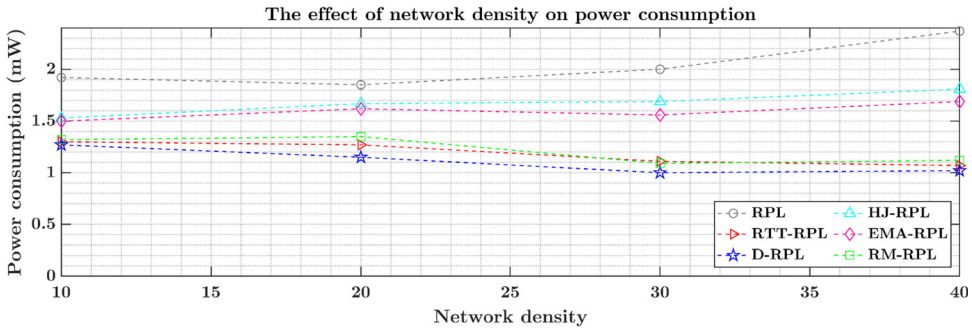


Fig. 14 Effect of network density on power consumption

Table 4 Simulation values were obtained for different protocols in the second experiment (network density)

Distance between the nodes	10	20	30	40
Handover delay (s)				
RPL	4.30	30.80	35.37	51.20
RTT-RPL	2.16	17.91	23.10	33.67
D-RPL	5.37	20.66	28.53	40.48
HJ-RPL	3.33	14.51	22.54	28.81
EMA-RPL	2.50	12.76	17.45	24.59
RM-RPL	4.12	15.51	56.2	30.69
PDR				
RPL	0.421	0.449	0.288	0.202
RTT-RPL	0.786	0.762	0.750	0.742
D-RPL	0.634	0.612	0.598	0.586
HJ-RPL	0.699	0.583	0.544	0.515
EMA-RPL	0.711	0.600	0.559	0.528
RM-RPL	0.950	0.912	0.907	0.890
E2ED (ms)				
RPL	270	401	436	771
RTT-RPL	256	336	341	450
D-RPL	253	349	367	439
HJ-RPL	239	250	337	409
EMA-RPL	224	256	301	384
RM-RPL	263	302	360	459
Power consumption (mW)				
RPL	1.92	1.85	2.00	2.37
RTT-RPL	1.30	1.27	1.11	1.07
D-RPL	1.27	1.15	1.00	1.02
HJ-RPL	1.53	1.67	1.69	1.81
EMA-RPL	1.50	1.62	1.56	1.69
RM-RPL	1.32	1.35	1.09	1.12

power consumption than RM-RPL. EMA-RPL and HJ-RPL methods cause hotspot points and impose more computational overhead, which has increased power consumption in nodes compared to the proposed mechanism. Table 4 shows the simulation outputs for all protocols in the second experiment.

6.3.3 Network dynamics analysis

The third experiment examines the exponential increase of mobile nodes on the evaluation metrics. This experiment is also performed with 38 static nodes deployed at a 40-meter fixed distance. The mobile nodes are 4, 8, 16, 24, and 32 in the simulations. Figure 15 depicts that increasing the mobile node ratio increases the network's handover delay. As more MNs occur, nodes do more handover to change the parent. The rate of change of these delays is worse for RPL than for other protocols. Also, the handover delay for the proposed mechanism has almost the same conditions due to the similarity of the MN leaving mechanism from the parent range. D-RPL imposes more handover delays than RM-RPL due to the lack of OF and the instability of communication protocols. EMA-RPL and HJ-RPL experience more minor handover delays than other mechanisms due to the node mobility monitoring system and link probability.

Figure 16 shows the effect of increasing the ratio of MNs for the evaluated protocols about PDR. Increasing MNs leads to more temporary outages from the parent range and packet losses. As a result, PDR is gradually reduced. RPL reliability is lower than other protocols due to a lack of mobility management, which, with increasing MNs, more losses resulted in lower PDR values. RM-RPL guarantees the delivery of critical packets. Therefore, it experiences the best PDR values among the various methods. The increase in MNs causes more parent switches in D-RPL to ensure link quality, resulting in a lower delivery rate than RM-RPL. The rise in packet losses in EMA-RPL and HJ-RPL with increasing MNs, due to more computational load and the possibility of more congestion in the monitor nodes, has caused a further decrease in PDR than RM-RPL.

As MNs increase, E2ED in the network increases due to more hops to forward packets to the destination. Figure 17 presents the ratio changes of MNs for different protocols. Lack of node mobility management and worsening network instability due to more interruptions have led to more delays in RPL than other mechanisms. D-RPL shows

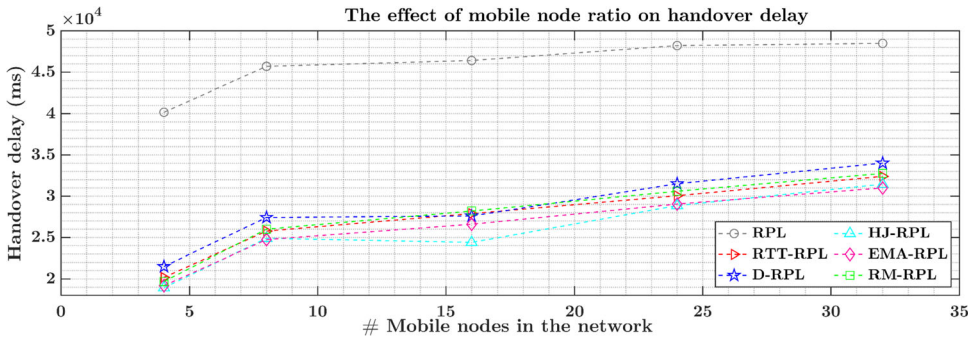


Fig. 15 Effect of network dynamics on handover delay

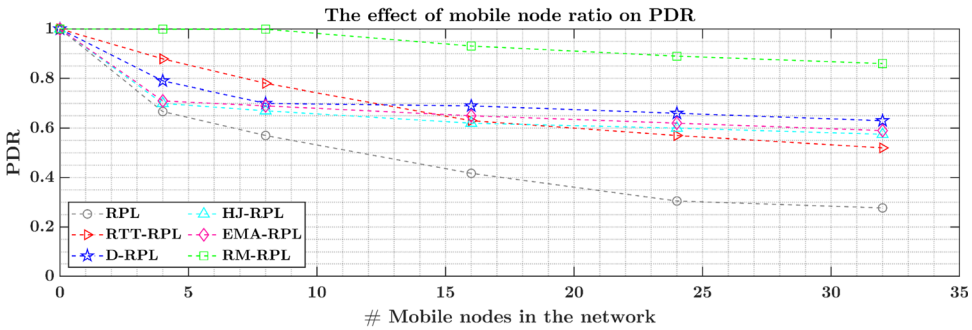


Fig. 16 Effect of network dynamics on PDR

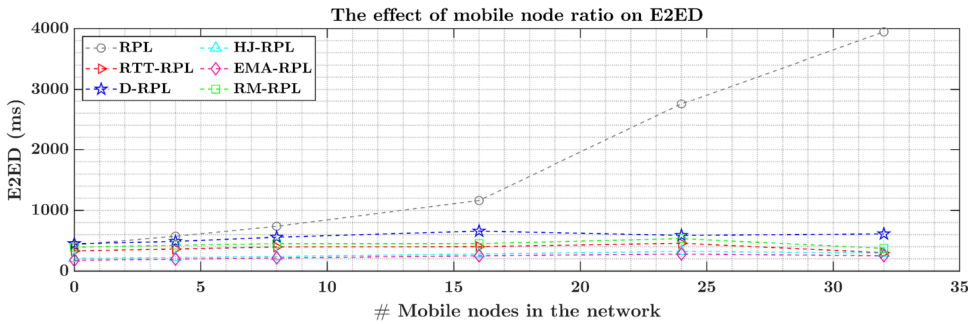


Fig. 17 Effect of network dynamics on E2ED

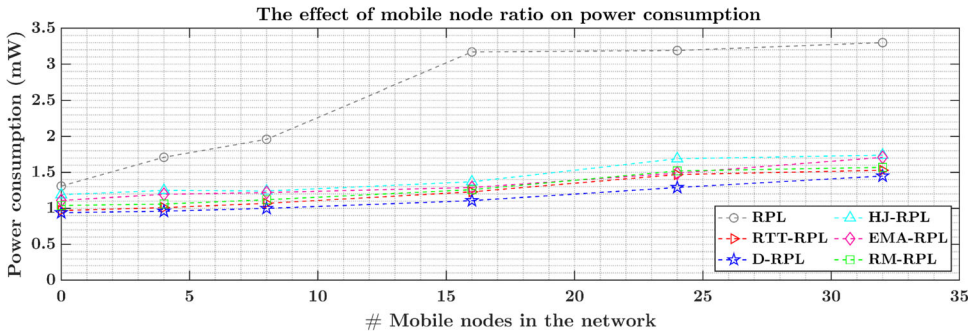


Fig. 18 Effect of network dynamics on power consumption

longer delays than RM-RPL. Prioritizing critical packets causes more delays in delivering regular data packets, so that RM-RPL will experience more similar delays than RTT-RPL. HJ-RPL and EMA-RPL also resulted in fewer outages due to the selection of leaf nodes as MN rather than parents, resulting in less E2ED than other protocols. Due to more handovers and calculations, power consumption is rising with the increase in mobile nodes.

Figure 18 depicts the MNs ratio's effect on power consumption. RM-RPL has experienced more power consumption than RTT-RPL due to more control packets eliminating RSSI errors. RPL requires more re-retransmissions due to the lack of mobility, which has led to more

Table 5 Simulation values were obtained for different protocols in the third experiment (network dynamics)

Number of the mobile nodes	0	4	8	16	24	32
Handover delay (ms)						
RPL	–	40,175	45,706	46,414	48,222	48,490
RTT-RPL	–	20,189	25,769	27,885	30,055	32,400
D-RPL	–	21,500	27,417	27,602	31,509	34,012
HJ-RPL	–	18,939	24,895	24,398	28,860	31,450
EMA-RPL	–	19,189	24,739	26,601	29,050	31,000
RM-RPL	–	19,700	26,000	28,201	30,600	32,756
PDR						
RPL	1	0.667	0.570	0.417	0.305	0.277
RTT-RPL	1	0.879	0.780	0.630	0.570	0.520
D-RPL	1	0.791	0.701	0.690	0.660	0.632
HJ-RPL	1	0.702	0.670	0.620	0.602	0.575
EMA-RPL	1	0.710	0.691	0.650	0.621	0.592
RM-RPL	1	0.999	0.998	0.931	0.890	0.861
E2ED (ms)						
RPL	429	573	737	1163	2755	3946
RTT-RPL	331	362	396	401	457	301
D-RPL	449	487	557	658	587	611
HJ-RPL	209	222	239	280	323	295
EMA-RPL	176	194	214	250	279	250
RM-RPL	391	424	450	451	534	372
Power consumption (mW)						
RPL	1.31	1.71	1.96	3.17	3.19	3.30
RTT-RPL	0.97	1.01	1.07	1.23	1.47	1.53
D-RPL	0.94	0.96	1.00	1.10	1.29	1.45
HJ-RPL	1.19	1.25	1.24	1.37	1.69	1.74
EMA-RPL	1.11	1.19	1.22	1.29	1.49	1.70
RM-RPL	1.04	1.06	1.12	1.26	1.52	1.57

packet losses and has imposed more power consumption. D-RPL has less power consumption than the proposed mechanism because it has not taken action to eliminate the link quality error. The other two methods have a higher power consumption than the RM-RPL due to the higher imposed computational overhead. Table 5 shows the simulation outputs for all protocols in the third experiment.

7 Conclusions and future studies

The increasing prevalence of diverse IoT applications emphasizes the need for robust mobility support. However, due to its limitations, the traditional RPL protocol is not well-suited for dynamic mobile environments. RPL's inability to handle node mobility and parent unavailability results in communication instability and network losses. Previous attempts to enhance RPL for mobile IoT applications have faced challenges such as low PDR, suboptimal parent selection, and excessive energy consumption. To address these issues, a proposed protocol called RM-RPL offers significant improvements. RM-RPL allows mobile nodes to act as parents within the network, adapting seamlessly to changing network topologies. It employs an efficient OF to identify optimal parent nodes, ensuring reliable communication paths with minimal energy consumption. Additionally, RM-RPL adjusts protocol behavior during static states to prevent energy depletion in mobile nodes. The superiority of RM-RPL over standard RPL has been confirmed through extensive evaluation, achieving an average 2.6X improvement in PDR and enhancing reliability by up to 4.2X. These results position RM-RPL as an ideal choice for critical healthcare IoT applications where reliable communication is crucial. Researchers can use these findings to design application-specific or standardized protocol versions, paving the way for further exploration and development in mobility support for IoT.

Our mechanism has addressed critical challenges in IoT mobility. However, there are still exciting opportunities for further research and development. Firstly, we intend to

devise a robust handover management mechanism to enable seamless transitions between network domains, minimizing service disruption and ensuring uninterrupted communication. We aim to optimize reconnection delays to enhance network responsiveness and improve user experience. Additionally, we plan to design a dynamic and adaptive mechanism that replaces the conventional reverse trickle timer, allowing compatibility with real-time applications and adaptability to varying network conditions. Lastly, we recognize the importance of minimizing control overhead and will focus on efficient control packet management, balancing reliability and resource efficiency.

Author contributions AS, MM, TT, and AG wrote the main manuscript text and TT prepared Figs. 1, 2 and 3. All authors reviewed the manuscript.

Funding The authors have not disclosed any funding.

Data availability Enquiries about data availability should be directed to the authors.

Declarations

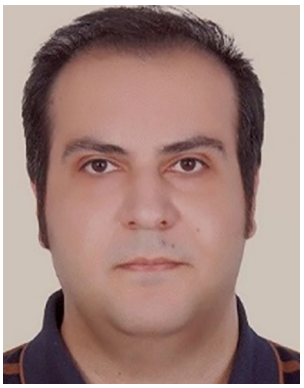
Competing interests The authors declare no competing interests.

References

1. Najarro, L.A.C., Song, I., Kim, K.: Differential evolution with opposition and redirection for source localization using RSS measurements in wireless sensor networks. IEEE Trans. Autom. Sci. Eng. **17**(4), 1736–1747 (2020). https://doi.org/10.1109/TASE.2020.2975287
2. Albreem, M.A., Sheikh, A.M., Alsharif, M.H., Jusoh, M., Yasin, M.N.M.: Green internet of things (GIoT): applications, practices, awareness, and challenges. IEEE Access **9**, 38833–38858 (2021). https://doi.org/10.1109/ACCESS.2021.3061697
3. Dai, X., et al.: Task co-offloading for d2d-assisted mobile edge computing in industrial internet of things. IEEE Trans. Industr. Inf. **19**(1), 480–490 (2022). https://doi.org/10.1109/TII.2022.3158974
4. Liang, X., Huang, Z., Yang, S., Qiu, L.: Device-free motion & trajectory detection via RFID. ACM Trans. Embedded Comput. Syst. (TECS) **17**(4), 1–27 (2018). https://doi.org/10.1145/3230644
5. Eshmawi, A.A., et al.: Deep learning with metaheuristics based data sensing and encoding scheme for secure cyber physical sensor systems. Clust. Comput. **26**(4), 2245–2257 (2023). https://doi.org/10.1007/s10586-022-03654-8
6. Luo, J., Zhao, C., Chen, Q., Li, G.: Using deep belief network to construct the agricultural information system based on internet of things. J. Supercomput. **78**(1), 379–405 (2022). https://doi.org/10.1007/s11227-021-03898-y
7. Sheibani, M., Barekatein, B., Arvan, E.: A lightweight distributed detection algorithm for DDAO Attack on RPL routing protocol in internet of things. Pervasive Mob. Comput. (2022). https://doi.org/10.1016/j.pmcj.2021.101525
8. Manvi, S., Shobha, K., Vastrad, S.: Performance analysis of routing protocol for low power and lossy networks (RPL) for IoT environment. In: Distributed Computing and Intelligent Technology: 19th International Conference, ICDCIT Bhubaneswar, India, January 18–22, 2023, Proceedings, 2023, pp. 341–348. Springer (2023). https://doi.org/10.1007/978-3-031-24848-1_25
9. Sahay, R., Geethakumari, G., Mitra, B.: Mitigating the worst parent attack in RPL based internet of things. Clust. Comput. **25**(2), 1303–1320 (2022). https://doi.org/10.1007/s10586-021-03528-5
10. Liu, C., Wu, T., Li, Z., Ma, T., Huang, J.: Robust online tensor completion for IoT streaming data recovery. IEEE Trans. Neural Netw. Learn. Syst. (2022). https://doi.org/10.1109/TNNLS.2022.3165076
11. Cheng, B., Wang, M., Zhao, S., Zhai, Z., Zhu, D., Chen, J.: Situation-aware dynamic service coordination in an IoT environment. IEEE/ACM Trans. Netw. **25**(4), 2082–2095 (2017). https://doi.org/10.1109/TNET.2017.2705239
12. Kushalnagar, N., Montenegro, G., Schumacher, C.: IPv6 over low-power wireless personal area networks (6LoWPANs): overview, assumptions, problem statement, and goals. (2007)
13. Al-Kashoash, H.A., Kharrufa, H., Al-Nidawi, Y., Kemp, A.H.: Congestion control in wireless sensor and 6LoWPAN networks: toward the internet of things. Wireless Netw. (2018). https://doi.org/10.1007/s11276-018-1743-y
14. Glissa, G., Meddeb, A.: 6LowPSec: an end-to-end security protocol for 6LoWPAN. Ad Hoc Netw. **82**, 100–112 (2019). https://doi.org/10.1016/j.adhoc.2018.01.013
15. Liu, G.: A Q-Learning-based distributed routing protocol for frequency-switchable magnetic induction-based wireless underground sensor networks. Futur. Gener. Comput. Syst. **139**, 253–266 (2023). https://doi.org/10.1016/j.future.2022.10.004
16. Almusaylim, Z.A., Alhumam, A., Jhanjhi, N.: Proposing a secure RPL based internet of things routing protocol: a review. Ad Hoc Netw. **101**, 102096 (2020). https://doi.org/10.1016/j.adhoc.2020.102096
17. Seyfollahi, A., Moodi, M., Ghaffari, A.: MFO-RPL: a secure RPL-based routing protocol utilizing moth-flame optimizer for the IoT applications. Comput. Stand. Interfaces **82**, 103622 (2022). https://doi.org/10.1016/j.csi.2022.103622
18. Zaidi, A., Farooq, H., Rizwan, A., Abu-Dayya, A., Imran, A.: A framework to address mobility management challenges in emerging networks. IEEE Wirel. Commun. (2023). https://doi.org/10.1109/MWC.015.2100666
19. Darabkh, K.A., Al-Akhras, M., Zomot, J.N., Atiquzzaman, M.: RPL routing protocol over IoT: a comprehensive survey, recent advances, insights, bibliometric analysis, recommendations, and future directions. J. Netw. Comput. Appl. (2022). https://doi.org/10.1016/j.jnca.2022.103476
20. Muzammal, S.M., Murugesan, R.K., Jhanjhi, N., Hossain, M.S., Yassine, A.: Trust and mobility-based protocol for secure routing in internet of things. Sensors **22**(16), 6215 (2022). https://doi.org/10.3390/s22166215
21. Cobarzan, C., Montavont, J., Noel, T.: Analysis and performance evaluation of RPL under mobility. In: 2014 IEEE Symposium on Computers and Communications (ISCC), pp. 1–6. IEEE (2014). https://doi.org/10.1109/ISCC.2014.6912471
22. Seyfollahi, A., Ghaffari, A.: A lightweight load balancing and route minimizing solution for routing protocol for low-power and lossy networks. Comput. Netw. **179**, 107368 (2020). https://doi.org/10.1016/j.comnet.2020.107368
23. Mishra, A.K., Singh, O., Kumar, A., Puthal, D.: Hybrid mode of operations for RPL in IoT: a systematic survey. IEEE Trans. Netw. Serv. Manag. **19**(3), 3574–3586 (2022). https://doi.org/10.1109/TNSM.2022.3159241

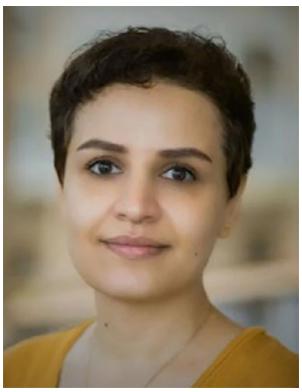
24. Garg, S., Mehrotra, D., Pandey, H.M., Pandey, S.: Static to dynamic transition of RPL protocol from IoT to IoV in static and mobile environments. Clust. Comput. **26**(1), 847–862 (2023). https://doi.org/10.1007/s10586-022-03689-x
25. Shah, Z., Levula, A., Khurshid, K., Ahmed, J., Ullah, I., Singh, S.: Routing protocols for mobile internet of things (IoT): a survey on challenges and solutions. Electronics **10**(19), 2320 (2021). https://doi.org/10.3390/electronics10192320
26. Garg, S., Mehrotra, D., Pandey, S.: A study on RPL protocol with respect to DODAG formation using objective function. In: Soft Computing: Theories and Applications, pp. 633–644. Springer, Singapore (2022)
27. Rojas, E., Hosseini, H., Gomez, C., Carrascal, D., Cotrim, J.R.: Outperforming RPL with scalable routing based on meaningful MAC addressing. Ad Hoc Netw. **114**, 102433 (2021). https://doi.org/10.1016/j.adhoc.2021.102433
28. Zhang, W., Han, G., Feng, Y., Lloret, J.: IRPL: an energy efficient routing protocol for wireless sensor networks. J. Syst. Architect. **75**, 35–49 (2017). https://doi.org/10.1016/j.sysarc.2017.03.006
29. Levis, P., Clausen, T., Hui, J., Gnawali, O., Ko, J. (2011) RFC 6206: the trickle algorithm. RFC. https://doi.org/10.17487/RFC6206
30. Kharrufa, H., Al-Kashoash, H.A., Kemp, A.H.: RPL-based routing protocols in IoT applications: a review. IEEE Sens. J. **19**(15), 5952–5967 (2019). https://doi.org/10.1109/JSEN.2019.2910881
31. Jiang, H., Xiao, Z., Li, Z., Xu, J., Zeng, F., Wang, D.: An energy-efficient framework for internet of things underlaying heterogeneous small cell networks. IEEE Trans. Mob. Comput. **21**(1), 31–43 (2020). https://doi.org/10.1109/TMC.2020.3005908
32. Cao, K., et al.: Improving physical layer security of uplink NOMA via energy harvesting jammers. IEEE Trans. Inf. Forensics Secur. **16**, 786–799 (2020). https://doi.org/10.1109/TIFS.2020.3023277
33. Niu, X.: Optimizing DODAG build with RPL protocol. Math. Problems Eng. (2021). https://doi.org/10.1155/2021/5579564
34. Vaziri, B.B., Haghighat, A.T.: Brad-OF: an enhanced energy-aware method for parent selection and congestion avoidance in RPL Protocol. Wireless Pers. Commun. **114**(1), 783–812 (2020). https://doi.org/10.1007/s11277-020-07393-0
35. Min, S.-W., Chung, S.-H., Lee, H.-J., Ha, Y.-V.: Downward traffic retransmission mechanism for improving reliability in RPL environment supporting mobility. Int. J. Distrib. Sens. Netw. **16**(1), 1550147720903605 (2020)
36. Narten, T., Nordmark, E., Simpson, W., Soliman, H.: RFC 4861: Neighbor discovery for IP version 6 (IPv6). (2007), https://doi.org/10.17487/RFC4861
37. Kuntz, R., Montavont, J., Noël, T.: Improving the medium access in highly mobile wireless sensor networks. Telecommun. Syst. **52**(4), 2437–2458 (2013). https://doi.org/10.1007/s11235-011-9565-6
38. Cobârzan, C., Montavont, J., Noel, T.: Integrating mobility in RPL. In: European Conference on Wireless Sensor Networks, pp. 135–150. Springer, Berlin (2015)
39. Fotouhi, H., Moreira, D., Alves, M., Yomsi, P.M.: mRPL+: a mobility management framework in RPL/6LoWPAN. Comput. Commun. **104**, 34–54 (2017). https://doi.org/10.1016/j.comcom.2017.01.020
40. Kamgueu, P.O., Nataf, E., Ndie, T.D.: Survey on RPL enhancements: a focus on topology, security and mobility. Comput. Commun. **120**, 10–21 (2018). https://doi.org/10.1016/j.comcom.2018.02.011
41. Gaddour, O., Koubäa, A., Rangarajan, R., Cheikhrouhou, O., Tovar, E., Abid, M.: Co-RPL: RPL routing for mobile low power wireless sensor networks using Corona mechanism. In: Proceedings of the 9th IEEE International Symposium on Industrial Embedded Systems (SIES 2014), pp. 200–209. IEEE (2014). https://doi.org/10.1109/SIES.2014.6871205
42. Gaddour, O., Koubâa, A., Abid, M.: Quality-of-service aware routing for static and mobile IPv6-based low-power and lossy sensor networks using RPL. Ad Hoc Netw **33**, 233–256 (2015). https://doi.org/10.1016/j.adhoc.2015.05.009
43. Tahir, Y., Yang, S., McCann, J.: BRPL: backpressure RPL for high-throughput and mobile IoTs. IEEE Trans. Mob. Comput. **17**(1), 29–43 (2017). https://doi.org/10.1109/TMC.2017.2705680
44. Kharrufa, H., Al-Kashoash, H., Al-Nidawi, Y., Mosquera, M.Q., Kemp, A.H.: Dynamic RPL for multi-hop routing in IoT applications. In: 13th Annual Conference on Wireless On-Demand Network Systems and Services (WONS), pp. 100–103. IEEE (2017). https://doi.org/10.1109/WONS.2017.7888753
45. Bouaziz, M., Rachedi, A., Belghith, A., Berbineau, M., Al-Ahmadi, S.: EMA-RPL: energy and mobility aware routing for the internet of mobile things. Futur. Gener. Comput. Syst. **97**, 247–258 (2019). https://doi.org/10.1016/j.future.2019.02.042
46. Hoghooghi, S., Javidan, R.: Proposing a new method for improving RPL to support mobility in the internet of things. IET Netw. **9**(2), 48–55 (2020). https://doi.org/10.1049/iet-net.2019.0152
47. Manikannan, K., Nagarajan, V.: Optimized mobility management for RPL/6LoWPAN based IoT network architecture using the firefly algorithm. Microprocess Microsyst. **77**, 103193 (2020). https://doi.org/10.1016/j.micpro.2020.103193
48. Yang, X.-S.: Firefly algorithm, Levy flights and global optimization. In: Research and Development in Intelligent Systems XXVI, pp. 209–218. Springer, London (2010)
49. Liu, X., et al.: Adapting feature selection algorithms for the classification of chinese texts. Systems **11**(9), 483 (2023). https://doi.org/10.3390/systems11090483
50. Zheng, Y., Li, L., Qian, L., Cheng, B., Hou, W., Zhuang, Y.: Sine-SSA-BP ship trajectory prediction based on chaotic mapping improved sparrow search algorithm. Sensors **23**(2), 704 (2023). https://doi.org/10.3390/s23020704
51. Kniess, J., de Figueiredo Marques, V.: MARPL: A crosslayer approach for internet of things based on neighbor variability for mobility support in RPL. Trans. Emerg. Telecommun. Technol. **31**(12), e3931 (2020). https://doi.org/10.1002/ett.3931
52. Safaei, B., et al.: Impacts of mobility models on RPL-based mobile IoT infrastructures: an evaluative comparison and survey. IEEE Access. **8**, 167779–167829 (2020). https://doi.org/10.1109/ACCESS.2020.3022793
53. Mohammadsalehi, A., Safaei, B., Monazzah, A.M.H., Bauer, L., Henkel, J., Ejlali, A.: ARMOR: a reliable and mobility-aware RPL for mobile internet of things infrastructures. IEEE Internet Things J. (2021). https://doi.org/10.1109/JIOT.2021.3088346
54. Murali, S., Jamalipour, A.: Mobility-aware energy-efficient parent selection algorithm for low power and lossy networks. IEEE Internet Things J. **6**(2), 2593–2601 (2018). https://doi.org/10.1109/JIOT.2018.2872443
55. Zhang, J., Zhu, C., Zheng, L., Xu, K.: ROSEFusion: random optimization for online dense reconstruction under fast camera motion. ACM Trans. Graph. (TOG) **40**(4), 1–17 (2021). https://doi.org/10.1145/3450626.3459676
56. Zhang, J., Tang, Y., Wang, H., Xu, K.: ASRO-DIO: active subspace random optimization based depth inertial odometry. IEEE Trans. Robot. **39**(2), 1496–1508 (2022). https://doi.org/10.1109/TRO.2022.3208503
57. Shi, J., et al.: Optimal adaptive waveform design utilizing an end-to-end learning-based pre-equalization neural network in an UVLC system. J. Lightwave Technol. **41**(6), 1626–1636 (2022). https://doi.org/10.1109/JLT.2022.3225335
58. Rabet, I., et al.: Pushing IoT mobility management to the edge: granting RPL accurate localization and routing. In: 2021 IEEE

7th World Forum on Internet of Things (WF-IoT), pp. 338–343. IEEE (2021). https://doi.org/10.1109/WF-IoT51360.2021.9595872
59. Rabet, I., et al.: SDMob: SDN-based mobility management for IoT networks. J. Sens. Actuator Netw. **11**(1), 8 (2022). https://doi.org/10.3390/jsan11010008
60. Hussain, T., Yang, B., Rahman, H.U., Iqbal, A., Ali, F.: Improving source location privacy in social internet of things using a hybrid phantom routing technique. Comput. Secur. **123**, 102917 (2022). https://doi.org/10.1016/j.cose.2022.102917
61. Deak, G., Curran, K., Condell, J.: Wireless sensor networks-smoothing algorithms for RSSI-based device-free passive localisation. In: The Tenth International Conference on Information Technology and Telecommunications (IT&T 2010), pp. 99–107 (2010)
62. Österlind, F., Dunkels, A., Eriksson, J., Finne, N., Voigt, T.: Cross-level sensor network simulation with cooja. In: First IEEE International Workshop on Practical Issues in Building Sensor Network Applications (SenseApp 2006), pp. 641–648. IEEE, Tampa (2006). https://doi.org/10.1109/LCN.2006.322172
63. Dunkels, A., Gronvall, B., Voigt, T.: Contiki-a lightweight and flexible operating system for tiny networked sensors. In: 29th Annual IEEE International Conference on Local Computer Networks, pp. 455–462. IEEE (2004). https://doi.org/10.1109/LCN.2004.38
64. Johnson, D.B., Maltz, D.A.: Dynamic source routing in ad hoc wireless networks. In: Mobile Computing, pp. 153–181. Springer, Cham (1996)



Ali Seyfollahi received his M.Sc. in Information Technology Engineering, majoring in Computer Networks, from Islamic Azad University, Tabriz Branch, Iran (IAUT), with first-class honors. He completed his bachelor's degree in Computer Networks Technology Engineering as a distinguished graduate from the University of Applied Science and Technology, East Azerbaijan Telecommunication Training Center, Tabriz, Iran, in 2016. He is currently a research assistant for computer networks in Dr. Ali Ghaffari's Lab at IAUT. He has published his original research, review, and conference papers in prestigious journals and venues and reviewed numerous papers for peer-reviewed journals. His research interests are Computer Networks, IoT, Wireless and Mobile Networks, Routing, network and system security, and AI/ML/DL applications.

Md Mainuddin a Computer Science Ph.D. at Florida State University, possesses strong expertise in the identification and analysis of compromised devices in smart networks. Utilizing cutting-edge Machine Learning and Deep Learning techniques, he is dedicated to enhancing IoT security. Additionally, he is an accomplished web developer with over ten years of experience in designing, developing, and maintaining web applications. With a profound understanding of AWS Cloud-based software development, he has been involved in numerous projects encompassing front-end, back-end, and full-stack development, adapting to a wide range of business needs.



Tania Taami received her M.Sc. degree in Computer Engineering at the Science and Research Branch, IAU, Tehran, IRAN. Currently, she is a Ph.D. candidate in Computer Science at Florida State University, Florida, USA. Her research interests include Computer Networks, the Internet of Things, Serverless Computing, and Cloud/Fog Computing.

Ali Ghaffari received his B.Sc., M.Sc., and Ph.D. degrees in computer engineering from the University of Tehran and IAU (Islamic Azad University), TEHRAN, IRAN in 1994, 2002, and 2011 respectively. He has served as a reviewer for some high-ranked journals such as IEEE Transactions on Mobile Computing, IEEE/ACM Transactions on Networking, IEEE Transactions on Network and Service Management, Applied Soft Computing, Ad Hoc networks, Future Generation Computer System (FGCS), Journal of Ambient Intelligent and Humanized Computing (AIHC) and computer networks. Dr. Ghaffari has been featured among the World's Top 2% Scientists List in computer science, according to a conducted study by US-based Stanford University in 2020, 2021, and 2023, and Top 1% Scientists List in computer science, according to Clarivate analytics in 2022. As an associate professor of computer engineering at Islamic Azad University, Tabriz branch, IRAN, and research professor at Istinye University, Turkey, his research interests are mainly in the field of software-defined network (SDN), Wireless Sensor Networks (WSNs), Mobile Ad Hoc Networks (MANETs), Vehicular Ad Hoc Networks (VANETs), networks security and Quality of Service (QoS). He has published more than 200 international conferences and reviewed journal papers.