IEEE Access
Multidisciplinary ⋮ Rapid Review ⋮ Open Access Journal

Received 4 June 2022, accepted 16 July 2022, date of publication 27 July 2022, date of current version 5 August 2022.

Digital Object Identifier 10.1109/ACCESS.2022.3194273

RESEARCH ARTICLE

MSE-RPL: Mobility Support Enhancement in RPL for IoT Mobile Applications

AMIRREZA VAEZIAN AND YOUSEF DARMANI
Department of Computer Engineering, K. N. Toosi University of Technology, Tehran 16317-14191, Iran
Corresponding author: Yousef Darmani (darmani@kntu.ac.ir)

ABSTRACT Due to the expansion of IoT applications which causes the generation of a massive amount of data, data routing is one of the most important challenges in these networks. The Routing Protocol for Low-power and Lossy Networks (RPL) was developed to cope with the Low-power and lossy network constraints, which play a significant role in IoT networks. Although most IoT applications involve mobility and topology change that makes mobility support a substantial need to prevent disconnection of nodes and data loss, the RPL is designed for static networks. This article proposes a mobility support method called MSE as an extension of RPL. The MSE supports mobility of all nodes except the root node, and it provides a seamless connection during the mobility. It also manages a situation when a physical obstacle settles between two paired nodes in a dynamic environment. To this end, it uses a dynamic trickle timer with two different ranges, a neighbor link quality table, a function to select the best parent in case of mobility, confidence, critical zones, and a blacklist. Simulations in multiple scenarios indicate that MSE, despite causing a slight increase in signaling cost and power consumption, significantly reduces hand-off delay, increases Packet Delivery Ratio, reduces the number of lost data packets, and outperforms both RPL as a reactive and mRPL as a proactive protocol regarding mobility.

INDEX TERMS IoT, low-power, mobility support, multi-hop routing, RPL, WSN.

I. INTRODUCTION

With the advent of the Internet of Things (IoT) and the growth of Low-power and lossy networks (LLN) applications, the massive amount of data generated from these networks needs to be managed and routed. LLN, which makes up the bulk of the IoT, has some limitations in memory, energy, and processing power and also includes lossy links. These issues make data routing a challenge. Thus, a routing protocol compatible with these networks will be required. As a result, the IETF introduced the Routing Protocol for Low-power and lossy networks (RPL).

Due to the node constraints in IoT networks, especially the energy limitation on the one hand and broader radio coverage, on the other hand, most of the proposed methods are designed for multi-hop routing. In order to exchange data among sensors and the Internet, multi-hop infrastructure-based routing methods are required. Hence two main approaches were offered: the first approach denotes that data could be routed like ad-hoc networks until it is delivered to the Internet and uses methods such as AODV, DSR, and Flooding [1]. In contrast, the second approach argues that the traditional routing protocols should be used to govern data in IoT networks [2]. Akkaya *et al.* define three general methods for multi-hop routing as follows: Data-centric protocols in which data are transmitted through a few particular nodes, Location-based (geographic) protocols in which routing is based on the geographical location of the nodes, and Hierarchical protocols that determine a cluster structure for nodes in an area and each node must send its data to the root node to be routed [3]. Presenting the 6LoWPAN, which can be considered as a gateway for communication between LLN devices and IP protocols, many methods were introduced to increase its compatibility with low-power and lossy networks such as CTP, Hydro, Hilow, and Dymo-low. Unfortunately, none of them fully comply with all the LLN restrictions. Finally, the

The associate editor coordinating the review of this manuscript and approving it for publication was Salekul Islam.

RPL was introduced specially for this purpose [4], [5]. The RPL, which uses the IPv6 is based on the distance vector method [6].

The IEEE proposed a compatible standard for PHY and MAC layers of the LoWPAN-based devices called IEEE 802.15.4. The IEFT introduced the 6LoWPAN as a gateway between the IEEE 802.15.4 standard and the IP protocols so that LLN devices could connect to the Internet and send their IPv6 packets to lower layers based on IEEE 802.15.4. The RPL, based on 6LoWPAN, is deployed in the third layer. The transport and application layers use UDP and CoAP protocols, respectively [5], [7].

The RPL is designed for static topologies and environments and performs well in these conditions. Conversely, many LLN applications involve node mobility and topology changes that RPL is unable to manage, leading to node disconnection and loss of packet and energy. In addition, other RPL defections in the mobile environment will be discussed in detail in sec 2.2. Therefore, a routing method is needed to support the node mobility and topology changes. Further, the widespread use of the Internet of Mobile Things (IoMT) has a significant impact on the use of E-health, Building automation, and Smart cities [8]–[10].

Many methods have been proposed to improve mobility support in RPL. However, most of them restrict the mobility only to the leaf nodes according to the tree-like structure of RPL. Besides, they have some incompatibilities with the LLN constraints in many cases. In this paper, we introduce a method that supports mobility of all nodes and at any depth of the DAG coping with LLN constraints. Our contributions are as follows:

•Supporting mobility of all nodes except the sink node (dynamic topology)

•Providing a seamless and continuous connection during the mobility

•Improving the performance of RPL in the presence of an obstacle between two nodes (dynamic environment)

•Implementing a dynamic trickle timer

The remainder of this paper is organized as follows: Section II represents an overview of RPL and a brief description of the mobility issue. Section III reviews some relevant works that suggested improving the mobility support of RPL. The proposed MSE method is explained in section IV. The performance of the proposed method has been evaluated and compared with the previous works in section V. Finally, the conclusion and future perspectives are addressed in Section VI.

II. RPL OVERVIEW AND MOBILITY CHALLENGE

RPL is a multi-hop and hierarchical routing protocol that is basically designed for static IoT networks regarding LLN constraints.

A. RPL OVERVIEW

RPL suggests a tree-like structure called Directed Acyclic Graph (DAG), which harbors a hierarchical IPv6 routing to route data through the devices in an area. In a typical routing mode, nodes forward their packets to the sink node to be sent over the internet. Hence this structure is also called Destination Oriented DAG (DODAG). The RPL selects the routing paths according to the desired Quality of Service defined in each application using an OF function in each node. There are three types of nodes in DAG: 1) Root node, a gateway for data transmission between nodes of the DAG and the Internet. This node is usually more potent than the other nodes of the DAG in its hardware aspect. 2) Router nodes, including the DAG middle nodes, could generate and route the data packets. 3) Host nodes are the leaf nodes of the DAG, which can only generate data without the ability to route data [4], [11], [12].

In order to create and maintain the DAG, some control messages are defined in RPL, the three crucial of which are as follows: 1) a node sends a DIS message as a request for joining the DAG. 2) a node sends a DAO message to its parent after joining the DAG to create the reverse path. 3) The DIO message is sent periodically as a heartbeat message to maintain the DAG, and it is scheduled by a timer called trickle timer. This timer starts with a short period which is usually set to 4 seconds in the beginning, and doubles each time until an error occurs (e.g., node disconnection) or reaches its maximum interval, usually around 17 minutes. Then the timer will be reset. Using this timer, it detects the disconnection if a node does not receive a DIO message after a particular time. It then starts sending the DIS message to reconnect; This procedure is called the self-healing mechanism. In addition, in case of any disconnection, routes will be updated and corrected by the correction mechanism defined in the RPL [13]–[15].

B. MOBILITY DISSECTION

In many IoT applications, some mobile nodes carry delay and loss-sensitive data. In some cases, static and mobile nodes (MNs) are already known and can be programmed separately. In other applications, however, the mobility of a node is not determinable in the beginning, and each node can start moving at any time. Fig.1 presents a comprehensive model of mobility in DAG. As illustrated in the figure, the MN could simultaneously act as a child node and a parent node. Thus, two separate connections should be considered: 1) connection between MN and its parent, and 2) connection between MN and its children. In both mentioned connections, if a child node fails to find an alternative parent to switch with immediately, that would lead to disconnection and losing energy and data accordingly. Similarly, if a parent node that senses the mobility does not stop sending to the child node(s), that could have the same result. Data and energy loss in nodes would rise as the DAG depth below the MN grows or as the number of nodes below the MN increases.

C. RPL BEHAVIOR IN PRESENCE OF MOBILITY

RPL is designed and developed for static networks, and it has a reactive approach to dealing with any error or disconnection in DAG.

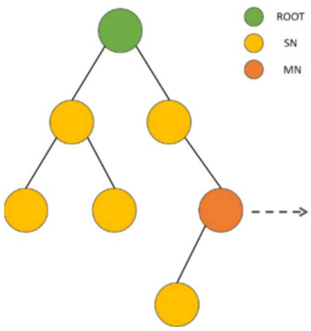

FIGURE 1. A comprehensive model of mobility in DAG.

When a node starts moving, it will detect the disconnection when it has not received a DIO message from its parent after a particular time. At this time, the self-healing mechanism will be activated to re-establish the connection. Nevertheless, this procedure is prolonged and has a high Hand-off delay that will cause data and energy loss, while many applications contain delay and loss-sensitive data. On the other hand, reducing the trickle timer period to raise the DIO sending rate can also increase network overhead and energy consumption. As a result, it seems that RPL cannot perform well in terms of mobility.

D. IMPACT OF THE MOBILE ENVIRONMENT ON RPL

In most IoT applications, the movement of a physical object is inevitable. These mobile entities sometimes settle between two paired nodes and cause a disconnection. In these situations, the self-healing and correction mechanisms of RPL perform very slowly and cause packet and energy loss. Further, if a disconnected node from DAG has children, the number of disconnected nodes would increase while they are considered as connected nodes by RPL; Hence, it produces a worse result.

III. RELATED WORKS

Some methods have been proposed to support mobility in RPL, among which some produce a cross-layer method, including layer 2, to detect or prevent node disconnection [16]. However, other methods could be studied under two general groups considering their approach toward mobility; The first group is made up of the reactive methods. In this group, a relatively significant period is spent detecting mobility. Therefore, disconnection is unavoidable, which involves losing data packets and the energy of nodes. The second group consists of proactive methods in which the mobility is detected and managed before disconnection occurs. Packet and energy loss in this group are relatively lower.

A. REACTIVE METHODS

In [17], the authors presented a layered architecture around the root node called Corona. Each Corona was a layer determined by its Corona_ID, which was the number of hops to the root node. In this method, each node mentions its Corona_ID in its DIO messages. After receiving DIO messages from its new neighbors, a moving MN connects itself to the neighbor with the lowest Corona_ID. This method has reduced the hand-off delay but did not solve the disconnection issue and its consequences.

Authors in [18] proposed a mobility supporting layer called Momoro. When MN does not receive the acknowledgment of its sent message after a particular time, it retries another time, and if it does not receive that again, it will start to flood a request message to its neighbors. After receiving replies, it will connect to the neighbor with the highest link quality measured based on link quality metrics using a pre-defined fuzzy estimator. This method speeds up mobility detection. However, flooding messages produce a high network overhead, and also, involves reactive methods problems.

Moreover, the authors in [19] designed a method called MobiRPL in which they suggested an adaptive time out for DIO to detect the disconnection more quickly. They also produced an adaptive probing mechanism to check the connectivity and a proactive discovery mechanism to discover new parents by sending specific DIS messages. However, the method cannot prevent disconnection, and the adaptive probing mechanism produces a high network overhead. Similarly, other methods have proposed a dynamic time-out period for DIO or DIS messages to reduce the parent change delay in mobile topologies after disconnection occurs [20].

B. PROACTIVE METHODS

A routing method based on geographical information for VANET was suggested in [21]. The authors introduced a new structure called Tiny-DODAG, which was created and removed quickly between a vehicle and a roadside node.

They also introduced two particular areas; The area with the smaller radius was used for data transmission between the vehicle and roadside node, and the area with the larger radius was used for establishing or terminating the connection. Although this method avoids disconnection and packet loss, it was designed especially for VANET, where the LLN constraints were not considered.

In [22], the authors defined Time-To-Reside (TTR) as a new metric to select the best-preferred parent, which employs relative-velocity and locational information of nodes to provide a longer connection period and more reliable routing paths. In order to obtain the required information, the authors suggest using different hardware or software techniques like the Global Positioning System (GPS) which consumes more energy from nodes in LLN networks.

In [23], the Authors developed a great mobility support algorithm called Smart-Hop, in which an MN measures the Received Signal Strength Indicator (RSSI) upon receiving any message. It continues the data transmission phase if the RSSI exceeds a particular threshold. Otherwise, the MN detects the mobility and chance of disconnection. Hence it starts sending a burst of DIS messages to discover new neighbors and measure their link quality.

After receiving its replies, it would connect to a neighbor with the highest Average RSSI (ARSSI). In addition, if the

TABLE 1. Comparison among mentioned methods towards mobility.

	Methods	Description	Strengths	Weaknesses
Reactive	Co-RPL [17]	- Presents routing solution by defining corona mechanism	- Switches the parent node faster than the RPL - Reduces hand-off delay	- Does not detect disconnection - Inefficient in mobile environments
Reactive	Momoro [18]	- Defines the Momoro mobility supporting layer and a fuzzy estimator	- Detects the mobility and switches the parent node	- Does not prevent disconnection - Produces a high overhead
Reactive	MobiRPL [19]	- Uses adaptive time out, adaptive probing, and proactive discovery mechanisms	- Detects the disconnection after an adaptive time out	- Does not prevent disconnection - Produces a high overhead and signaling cost
Proactive	GI-RPL [21]	- Defines transmission and communication zones	- Prevents mobile node disconnection	- Incompatible with the LLN limitations
Proactive	mRPL [23]	- Defines the smart-hop algorithm	- Detects the mobility before node disconnection - Prevents node disconnection and packet loss	- Produces high overhead and energy consumption - assumes mobility is limited to leaf nodes
Proactive	EMA-RPL [24]	- Defines a triple-phase approach to deal with the mobility	- Detects mobility before disconnection and prevents packet loss - Produces lower overhead and energy consumption	- assumes mobility is limited to leaf nodes - Inefficient in the presence of a physical obstacle between pairing nodes - May not detect mobility with a relatively low data packet rate
Proactive	ARMOR [22]	- Defines Time-To-Reside (TTR) metric to select the best-preferred parent	- Establishes a long-lasting, reliable path - Provides a longer connection period	- Requires specific hardware (e.g., GPS Module) or software techniques to obtain the node locational information

MN notices that the link quality to none of the neighbors is not good enough, it would continue the neighbor discovery phase. This method detects mobility before disconnection but produces a high overhead and consumes considerable energy from the nodes [24].

Authors in [24] offer a three-phase approach. The first is the data transmission phase. In this phase, if a parent node perceives the link quality as less than a pre-defined threshold, it detects the mobility before disconnection and enters into the next phase. In the second phase, the parent node asks the MN to send three DIS periodically and asks other DAG members to send back the measured ARSSI of the MN DIS messages. Finally, in the third phase, the parent node selects the node with the highest ARSSI and notifies it to establish a connection to the MN. This method includes lower energy consumption than the previous method, but both have considered mobility, limited to the leaf node(s) in the DAG. Also, they cannot detect mobility with a very low packet rate. Moreover, they give a poor performance in the presence of an obstacle between pair of nodes exchanging data and lead to losing data packets and consumed energy of nodes.

In addition, most of the mentioned methods manipulate the standard format of RPL controlling messages suggested by the IETF [25]. Table 1 presents a summary of the methods stated above and also illustrates a comparison among them.

IV. MSE-RPL: MOBILITY SUPPORT ENHANCEMENT IN RPL

In our proposed method, called MSE, we improve the RPL mobility support by adding our defined tools and mechanisms. Consequently, the MSE method could be used in both static and dynamic environments encompassing node movements and topology changes. We assume that the data is sensitive to delay and data loss. This assumption, in addition to the possibility of mobility for all nodes in the network, causes this method to be considered and tested under more complex conditions than the protocols mentioned in sec III. In other words, a higher Packet Delivery Ratio (PDR), a lower Hand-off delay, and a lower packet loss are needed. Furthermore, the power consumption of the method must be considered. In this section, our contributions will be briefly explained, and after defining new tools and mechanisms, our proposed method will be discussed.

A. CONTRIBUTIONS

1) SUPPORTING MOBILITY OF ALL NODES

In the mRPL and EMA-RPL, the mobility is limited to leaf nodes. In contrast, in many real applications, mobility could occur at any time and for any nodes in the DAG, including leaf and middle nodes, except the root node, which has a different structure.

2) PROVIDING A SEAMLESS AND CONTINUOUS CONNECTION DURING THE MOBILITY

Sensors-generated data must be sent immediately due to the delay sensitivity of the data. In the mRPL, the MN would not update its parent until it reaches a new parent with very high link quality. Therefore, it could be disconnected from the network for a long time in many cases and cause a high hand-off delay. In contrast, in the MSE method, the MN always keeps the address of the best-preferred parent, and it updates its parent immediately using two defined zones, a neighbor link quality table, and a dynamic trickle timer which will be discussed below. Thus, it produces a very low hand-off delay.

3) IMPROVED PERFORMANCE OF RPL IN THE PRESENCE OF AN OBSTACLE BETWEEN TWO NODES

The presence of an obstacle between two nodes is a usual event in IoT applications. It potentially could cause disconnection, which has not been solved in methods mentioned in sec III. In this situation, a good solution for a child node could be to find a new parent and change it if a preferred parent is found. Also, the parent node must stop sending packets to the child node. Otherwise, a disconnection could cause packet and energy loss.

4) IMPLEMENTING A DYNAMIC TRICKLE TIMER

As mentioned in section II, the trickle timer determines the time interval between DIO messages. It would be set by two variables I_{MIN} and $I_{DOUBLINGS}$ typically valued with 12 and 8, respectively. Then the range of the trickle timer will be set from $2^{I_{MIN}}$ to $2^{I_{MIN}+I_{DOUBLINGS}}$. This range might be appropriate for static networks and avoid causing overhead and energy loss. However, it might not detect the mobility in mobile networks by using a relatively large DIO interval, which could cause disconnection, high hand-off delay, high power consumption, and packet loss. Thus, two solutions could be suggested. The first is to reduce the trickle timer range and increase the DIO sending rate in mobile applications. For instance, I_{MIN} and $I_{DOUBLINGS}$ could be set to 12 and 2, then the trickle will start with an interval of 4 seconds and overflows with an interval of 16 seconds approximately. In this way, the mobility would be detected immediately, but a high network overhead and power consumption will be sacrificed. This will not be reasonable, especially when the mobility is not predictable in the application or when it only occurs in small parts of the network. Indeed, there is a tradeoff between the earlier mobility detection and the node power consumption.

The second is to define a dynamic trickle timer consisting of a Basic trickle for static states and a Mobile Mode trickle (MM trickle) for mobile states. The Basic trickle should have a large range, and the MM trickle should set into a relatively small range. In the beginning, the Basic trickle will be activated by default. The MM trickle will be activated only in nodes that sense the mobility around themselves, and they would readjust it to the Basic trickle after sensing a static state again. In this way, the MM trickle will be activated just in case of mobility and in a limited network area, while the Basic trickle is used in the other parts. Thus, utilizing a dynamic trickle timer will cause detecting the mobility immediately without making a high overhead, hand-off delay, and energy loss.

TABLE 2. Some applicational ranges for the trickle timer.

I_{min}	$I_{doubling}$	DIO_{min} (s)	DIO_{max} (s)
12	8	4.096	1048.576
12	2	4.096	16384
12	1	4.096	8.192
10	2	1.024	4.096
8	1	0.256	0.512

The ranges of basic trickle and MM trickle should be adjusted according to some parameters like node maximum speed, the radio transmission range of the nodes, and some other application attributes, and they could be different in each application. Table 2 shows some applicable ranges of trickle timer based on the I_{MIN} and $I_{DOUBLINGS}$ values.

B. TOOLS AND MECHANISMS

1) DIO WAITING TIMER

In each node, a timer is defined for the parent node called Parent DIO Waiting timer (PDW), and n timers are defined for n child nodes called Child DIO Waiting timer (CDW). These timers are defined due to the possibility of the entrance of a physical obstacle between each pair of parent-child nodes. Both nodes will detect the disconnection in case of not receiving a DIO message from the other side after a specific time. Then they would take the necessary measures to prevent packet and energy loss. The PDW timer would be reset upon receiving any DIO from the parent node. After the timeout, the node will stop sending, and it will disconnect itself from the DAG.

Similarly, the CDW timer will be reset upon receiving any DIO from the child node, and after the timeout, it will stop sending to the child node and removes the route to which. Furthermore, these timers will have the same function when a node moves out of the radio transmission range of the other node quickly, which might cause loss of the last packet and disconnection. It is noteworthy that the maximum number of CDW timers in each node is equal to the maximum definable number of children for a node in the application that could differ in any application, and it must be already determined and programmed in each node. F Further, whether the Basic or MM trickle is currently active, a node DIO-waiting timer interval is set based on the current state of the dynamic trickle timer.

2) CRITICAL AND CONFIDENCE ZONES

Two important areas are defined around each node: the confidence zone and the critical zone. In the confidence zone, despite the movement of the nodes, the link quality is satisfactory. The critical zone is made up of the area between the confidence zone and the maximum radio transmission range of the node. In the critical zone, the link quality is relatively lower, and if a node has mobility, the other side of the connection might not receive the following message. The RSSI determines the link quality in the MSE method, and a threshold is defined based on the RSSI value to determine the boundary between the confidence zone and the critical zone. Similarly, some previous methods utilized the concept of RSSI-based defined zones in different ways and functionalities [19], [21].

3) NEIGHBORS AND CANDIDATE PARENTS LIST

In the MSE method, each node has an address list of its current neighbors at any time. Each node distinguishes the candidate parent nodes among the List of neighbors to select the best candidate parent and connect to it immediately in critical situations. The List of neighbors is updated upon appearing a new neighbor around. The maximum size of the neighbor list is equal to the number of all nodes in the network. However, it is primarily programmed in a reasonably small fixed size according to the memory limitation of nodes.

4) NEIGHBOR LINK QUALITY TABLE

In the proposed method, a table is defined in each node called the neighbor link quality table, presenting the link quality to each neighbor. Each row in the table consists of n RSSI values measured from the last n DIO messages received from a neighbor. The RSSI values might have a high variance caused by the behavior of radio waves in the environment (e.g., reflection, scattering, diffraction, refraction, etc.). Therefore, we use the total trend of changes among n RSSI recorded for a particular neighbor to reduce the impact of the variance. For this purpose, some functions could be used, including Moving Average (MA), Exponential Moving Average (EMA), etc.

Although estimating the general trend of change in RSSI values will be more accurate by increasing the size of n, it will occupy more memory in nodes and vice versa. Thus, there is a trade-off between being more accurate and occupying less memory, and it will be determined in each application according to the memory of nodes. In addition, some other parameters such as nodes speed and radio transmission range are involved in determining the size of n.

5) BEST PARENT SELECTION FUNCTION (MOBILE MODE)

Typically, the best parent is the candidate parent with the lowest RANK. The RANK is defined based on the needed QoS in each application, and the OF calculates it. Nevertheless, in the MSE, after sensing any mobility, the best parent would be the candidate parent with the highest link quality based on RSSI values recorded in the related row of the neighbor link quality table. Because selecting a parent with a shorter distance to the root node and a lower received RSSI may lead to disconnection and packet and energy loss; Because when the parent starts moving, the mobility may not be detected. A node will need to select a new parent when its parent or the node itself has mobility. Also, when a node is disconnected from the DAG and receives a new DIO, it considers whether the sender is the best parent or not. According to the possibility of being mobile for all nodes, four priority levels are defined as presented in Algorithm 1. We have prioritized being the best parent to the candidate node with an approximately static RSSI trend according to the corresponding row of the neighbor link quality table because a connection to a static node would be more reliable than that to a mobile node. If more than one static node existed, the best parent would be the node with the highest Average RSSI. The second and third priority would also be given respectively to the node that is getting closer slowly and the node that got a little far and then stopped moving. Finally, the last priority would be for a node that is getting close quickly while there is no other node around. If there were more than one oncoming node, the best parent would be the node with the lowest Average RSSI (the furthest node) to establish a more durable connection.

Algorithm 1 Best Parent Selection Function (Mobile Mode)

```
for all nodes in candidate_parents:
  if RSSI_trend(node) ≈ static then //1st Priority
    output = node_with_max_AvgRSSI
    Break
  else if RSSI_trend(node) is slowly_increasing then //2nd Priority
    output = node_with_min_AvgRSSI
    Break
  else if RSSI_trend(node) is decrease_static then //3rd priority
    output = node_with_max_AvgRSSI
    Break
  else if RSSI_trend(node) is quickly_increasing then //4th Priority
    output = node_with_min_AvgRSSI
  Endif
Endfor
return output
```

6) BLACKLIST

There are some critical situations where nodes need to stop sending packets to a particular node. In these conditions, the node address should be added to a defined Blacklist then no packet will be sent to that. Conversely, a node address will be removed from the Blacklist if it appears in the confidence zone after a while, and then communication could be resumed.

C. METHODOLOGY

Generally, the function of our proposed method can be divided into two parts: 1) The MSE function regarding the mobility of nodes (dynamic topology). 2) The MSE function in case of the entrance of an obstacle between a pair of parent and child nodes (dynamic environment). The flowchart of the first part is illustrated in Fig. 2; According to the figure, after the DAG is created, each node, upon receiving any DIO, will measure the RSSI and add it to the sender address corresponding row in the neighbor link quality table. The receiver would consider if the sender node were connected to the DAG, the measured RSSI was indicating the critical zone, and the trend of stored RSSI values from the sender was decreasing, then it would recognize the mobility and chance of disconnection. From this stage on, the roles of both the parent and the child nodes would be different. The parent node should remove the routes to the sender node and add the sender address to the Blacklist to prevent losing data and

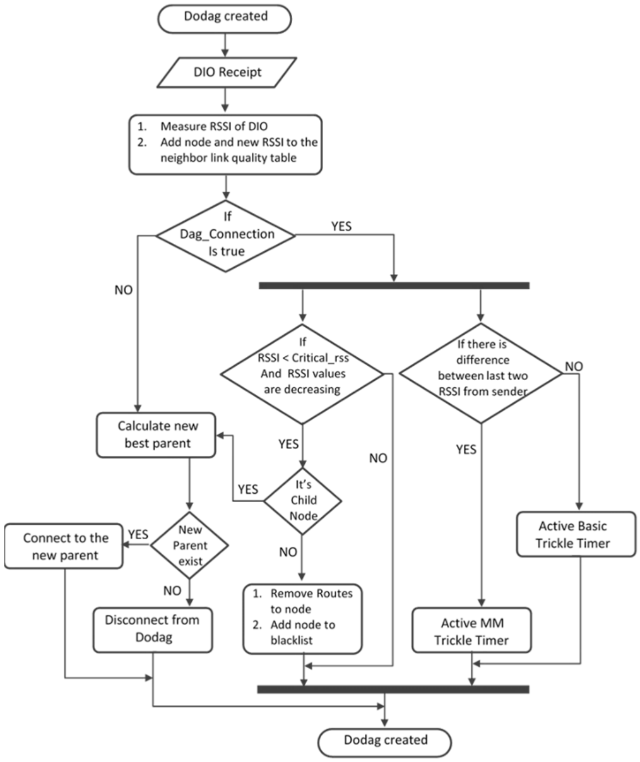

FIGURE 2. The MSE function flowchart in case of mobility.

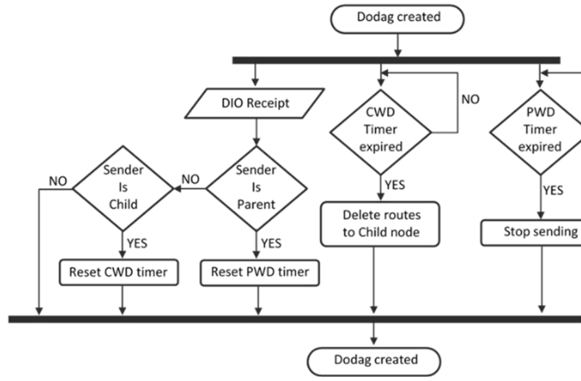

FIGURE 3. The MSE function flowchart in case of entrance of an obstacle.

energy. On the other hand, the child node should look for a new parent. Hence, it will call the best parent selection function. The node will connect to the new parent if an appropriate parent has been found. Otherwise, it would stop sending and disconnect from the DAG. In addition, after measuring RSSI, if the node senses a significant change between the two-last received RSSI, it will detect the mobility. Then, it activates the MM Trickle to follow the mobile node with a higher rate of DIO messages. Furthermore, after receiving a DIO, If the node were disconnected from DAG, it will check whether the sender is an appropriate parent to connect to or not. This stage would be done by calling the best parent selection function considering four priority levels, as mentioned previously.

"Stop sending packets" by a node before its disconnection is stated above to prevent establishing a wrong connection between the node that has already disconnected from the DAG, and a wanderer disconnected node moving across the network, which would cause more data and energy loss.

The MSE function flowchart in case of entrancing an obstacle is presented in Fig. 3. Accordingly, After the DAG was created, the PDW and CDW(s) timers would be activated for the parent and child nodes, respectively. They would be reset upon receiving each DIO from the related sender. If the PDW timer expires, the node should stop sending packets and is disconnected from the DAG, and if the CDW timer expires, the node must remove the routes to the related child node. As noted above, the MSE method does not make any change or manipulation in the standard RPL control message format, and it makes the proposed method compatible with the standard RPL.

D. SYSTEMATIC REVIEW

In a systematic approach to the general functionality of the MSE method, each node could be studied under some states. Given that the task of a child node is different from a parent node, two separate State Transition Diagrams (STD)s are illustrated below. As shown in Fig. 4, a child node could be in one of four defined states at each time. At first, when the DAG is just created, all nodes are in the "Static" state. Then, each node will activate the MM trickle timer and enter into the "Mobility Monitoring" state to track and monitor the mobility just after noticing mobility around itself by sensing a change in the RSSI values received from one of the neighboring nodes DIO messages. Nevertheless, nodes will return to the "Static" state and activate their Basic Trickle timer again whenever they find their surrounding static. Meanwhile, in the "Mobility Monitoring" state, if a node notices the distance to its responding node indicates the critical zone, it would enter into the "Parent selection" state by calling the best parent selection function. Then, if a new parent has existed, it would return to the previous state again after connecting to the new parent. Otherwise, it will stop sending and enter into the "DAG Disconnection" state. Then it must wait until it receives a DIO from neighbors to return to the previous state and do the same action.

Notably, a child node is connected to the DAG in all states except one. If the PDW timer expires, the child node enters the "DAG Disconnection" state. Then, it must stop sending packets until a new DIO is received.

The state transition diagram of a parent node is presented in Fig. 5; A parent node is always in one of the two shown states. This STD is very similar to the first two states of the child node STD, but the parent node is connected to the DAG in all states. Further, if a CDW timer expires in both states, the parent node will remove the routes to the related child node and add the child node address to the Blacklist. Consider if a middle node in the DAG starts moving, both of the above STDs would be run concurrently since a middle node is a child and a parent node simultaneously. Hence, as only non-child nodes could be parent candidates, MSE prevents the routing loops. Further, it is remarkable that in the MSE, nodes detect the mobility regardless of which neighboring node has

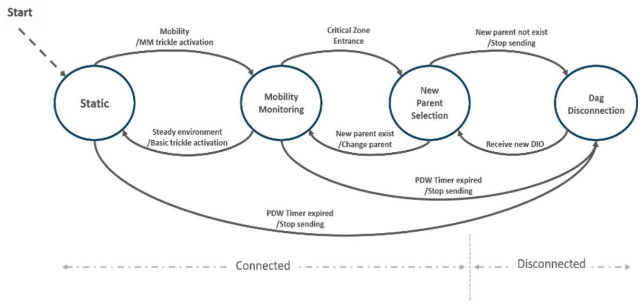

FIGURE 4. State Transition Diagram of proposed method for child node.

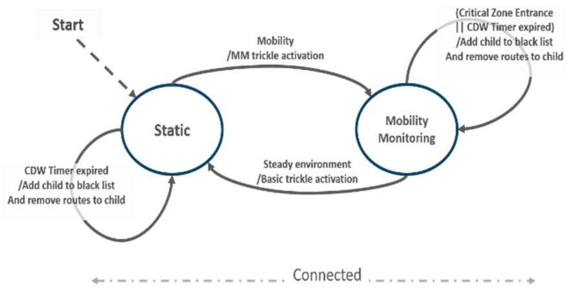

FIGURE 5. State Transition Diagram of proposed method for parent node.

started moving, including parent and child nodes and other neighboring nodes.

In order to review the time domain of the proposed method, two Timing diagrams will describe the two general aspects of the MSE functions mentioned previously. The first one presents the MSE function in the case of mobility. As illustrated in Fig. 6, firstly, the DAG will be created in the static state, and data transmission (Tx) will start. Nodes will add the measured RSSI to their neighbor link quality table upon receiving any DIO. Each node would activate the MM trickle timer when there was a significant gap between the last two RSSI received from a neighbor. Then, the node will change its state to the Mobility Monitoring state to follow the mobile node while the sending rate of DIO is increased in that area of the network. Meanwhile, whenever an entrance to the critical zone was detected, both parent and child nodes would begin to do their tasks separately to handle the mobility.

If the child node has already received any DIO from a neighboring node, it selects the new parent using the best parent selection function. Then it switches its parent by sending a DAO message to the new parent immediately and resumes data Tx with the new parent with a very low Hand-off delay. Otherwise, the child node would stop sending and would be disconnected from the DAG. It is also mentioned in the figure that the Basic trickle timer will be activated again when nodes discover a static environment around. Moments of change of states for parent node and child node are noted separately by numbers 0 to 3. For instance, number 1 indicates the moment that a node changes its state from "data Tx" to "Mobility Monitoring".

The function of the MSE regarding the entrance of an obstacle has illustrated in Fig. 7. At first, the DAG is created in the static state, and nodes will start to send data. Nodes will reset the related DIO-waiting timer after receiving any DIO. When an obstacle settles between the parent and the child nodes, it may cause the data packets to be absorbed in the obstacle and not be received by the destination.

At this time, both parent and child nodes will wait until the expiration of their DIO-waiting timers. Then the parent node will remove the routes to the child node and add the child node address to the Blacklist. Meanwhile, the child node will stop sending and be disconnected from DAG utill a new DIO is received from a candidate parent. After selecting the best parent, it will connect to that by sending a DAO and resume the data transmission with the new parent. As shown in Fig. 7, the DIO-waiting timers will reduce the packet and energy loss, and also, they will reduce the network overhead, but some packets will be lost before their expiration.

V. PERFORMANCE EVALUATION

The performance of our proposed method will be evaluated and compared with the Standard RPL as a reactive method and the mRPL as a proactive method in the same condition.

A. SIMULATION SETUP

To implement and evaluate the MSE method, we used the Contiki 3.0 operating system (OS) and Cooja simulator. Some main motivations for opting for the Cooja simulator are: (a) containing the open-source implementation of RPL, (b) supporting node mobility and dynamic topology via the mobility plugin [26], and (c) reserving compatibility with the mRPL implementation setup [23]. The implementation of MSE has consisted of some variable parameters which should be determined in each application. We assumed simplified and applicable values for most of the parameters. Additionally, our simulations are a schematic version of an actual implementation. Table 3 presents values of the fixed parameters that are constant in all simulations. Furthermore, some other float parameters will be determined in each simulation. Since the direction of data packets is upward in all simulations, the Blacklist and CDW timer are left unused. Notably, duty cycling is applied in MSE-RPL.

The Basic Trickle and MM trickle timers are adjusted depending on some other parameters in the application. For example, suppose the node speed is 2 m/s, the transmission range is 50 m, and the confidence zone radius is 40 m; In that case, the maximum delay between receiving two DIO messages would be 20 seconds to detect the mobility. Since the upper bound of the Basic trickle is set in the base-2 numeral system, the maximum adjustable interval is16 seconds. For this purpose, we used the trickle 12_2. Similarly, if the node speed was 5 m/s, then the trickle 12_1 could be an appropriate option.

B. EVALUATION METRICS

Our main purpose in designing the MSE was to produce a method with the highest packet delivery ratio in the shortest possible time due to data sensitivity, and the least amount of generation and propagation of control messages and energy

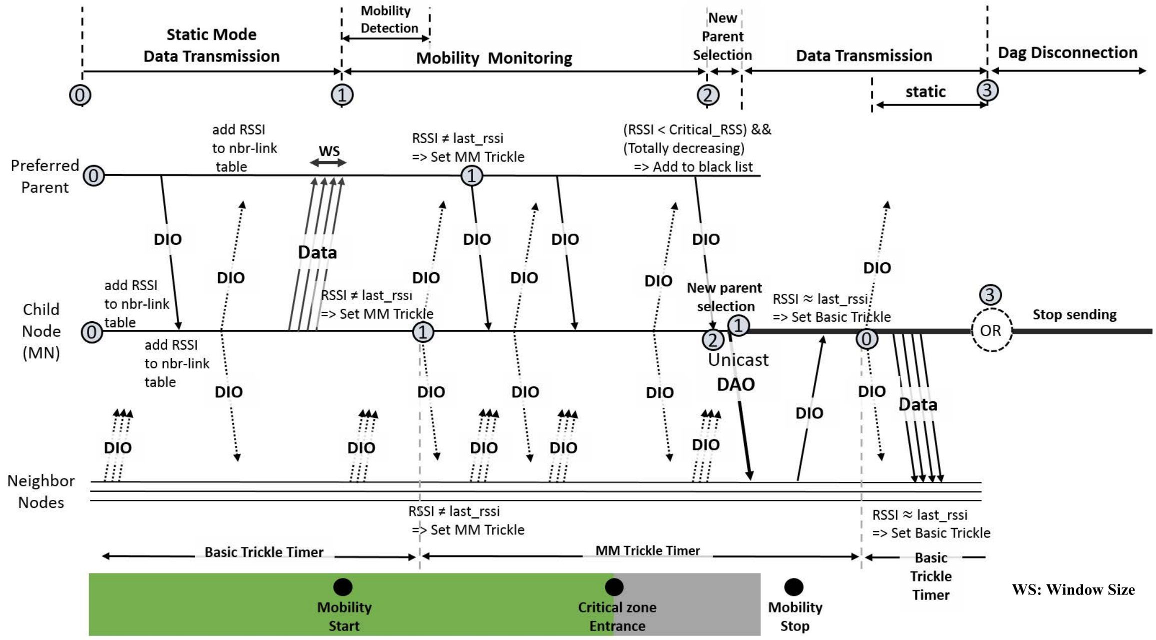

FIGURE 6. Timing diagram of the MSE function in case of mobility.

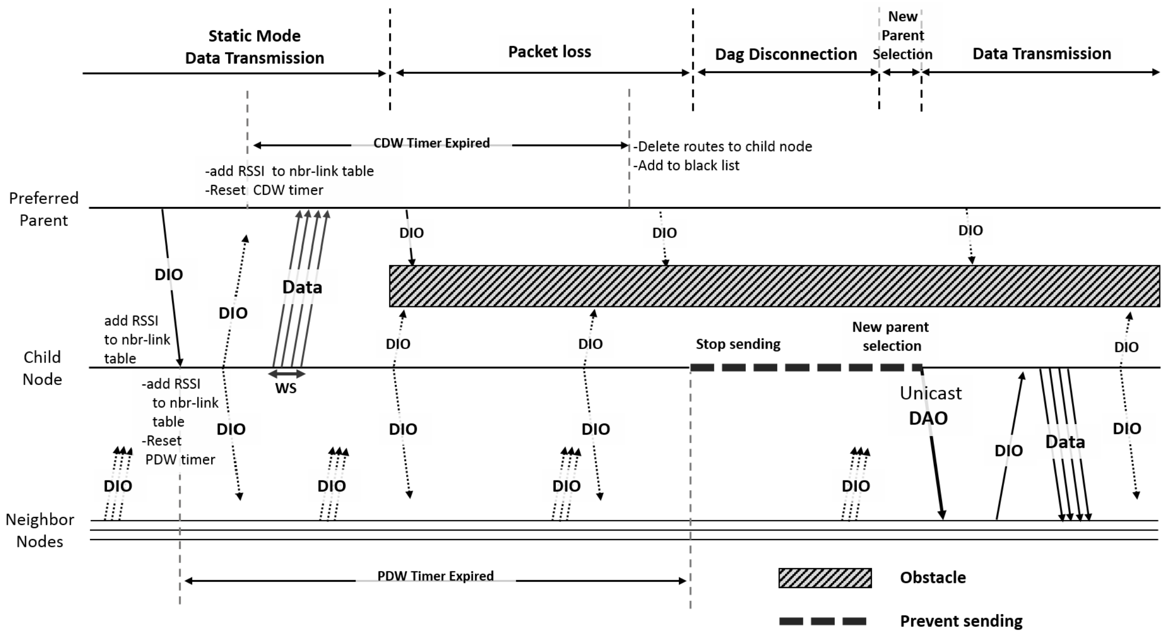

FIGURE 7. Timing diagram of the MSE function regarding entrance of an obstacle.

loss, in a dynamic situation. Hence, the relevant evaluation metrics are as follows:

Signaling cost: The total number of bytes of control messages propagated in the network to maintain the DAG and manage the mobility, including DIO, DIS, DAO, and all acknowledgment packets.

Power consumption: Due to using low-power nodes, studying the consumed power in implementing the proposed method is very important. The power of a node is constantly consumed by being in one of four modes of sending, receiving, processing, and Low-power Mode (LPM). The highest power is consumed in sending mode. Similar to what has been done in previous articles, the consumed power of a node is measured by the time that it has spent in each mentioned mode and the consumed current of that mode according to the Z1 datasheet shown in table 4 [27]–[29].

Packet Delivery Ratio (PDR): The ratio of the total number of received data packets to the total number of sent data packets.

Lost packets: The number of data packets that have not been delivered to the destination or delivered to a node that had been disconnected from the DAG before. As the

TABLE 3. Fixed parameters in simulations.

Radio transmission range	50 m
Confidence zone radius	40 m
Radio interference range	50 m
Critical zone entrance threshold (based on RSSI)	-80 dBm
Mobility model	Random walk model
Node speed	2 m/s
Mote type	Z1 mote
Neighbor link quality table size	Number of nodes in the network
Number of recorded RSSI from each neighbor	3
DIO Waiting timer interval	Time space to receive next 2nd DIO
Basic Trickle timer range	Trickle 12_2 (4.096 sec – 16.384 sec)
MM Trickle timer range	Trickle 10_2 (1.024 sec – 4.096 sec)

TABLE 4. Current consumption in different modes.

Node modes	Current consumption (mA)
Send	17.4
Receive	18.8
Process	0.426
LPM	0.02

number of nodes increases in a simulation or as simulation time passes, the PDR will get closer to 100%, and it could not describe the lost data packets well, while the data are loss-sensitive. Then, the lost packets metric will be helpful.

Hand-off delay: It is defined as the delay in updating the parent node in case of mobility or a disconnection. The three methods that have been used in this section suggest different procedures for updating parents. The standard RPL uses a timer to update the parent and usually does it after a relatively long time, sometimes causing a disconnection. In the mRPL method, the parent will update after broadcasting bursts of DIS, receiving the replies, measuring the ARSSI, and finally connecting to the most appropriate neighbor. The best parent in the MSE method is always maintained in each node. Thus, the hand-off process can be summarized by delivering a DAO message to the new parent.

C. SCENARIOS AND RESULTS

There are four experiments designed to evaluate the performance of the proposed method toward its four contributions. Each experiment consists of one scenario except the 2nd, which includes two scenarios. The performance of the MSE method will be evaluated and compared with the RPL and mRPL protocols using the mentioned metrics. Meanwhile, different data packet rates, from very low to high rates, are applied.

TABLE 5. Simulation parameters.

Number of nodes	7
Data packet interval	10 s
Simulation time	390 s
Movement starting time of node 6	56 s after beginning

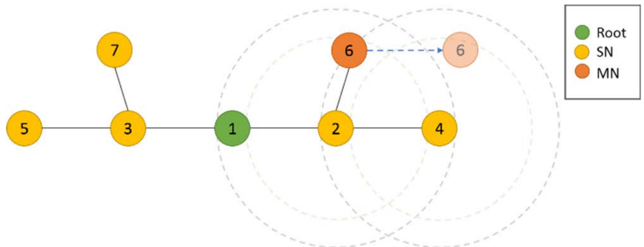

FIGURE 8. Dynamic trickle timers.

1) SCENARIO 1) DYNAMIC TRICKLE TIMER

This scenario is designed for evaluating and comparing the performance of different trickle timers in the MSE method. Fig. 8 shows the topology of the network. All nodes are initially static, and after the DAG is created, node 6 starts moving to the position shown in the figure with a constant speed and gets out of the radio transmission range of node 2. Then it enters the confidence zone of node 4, and it stops moving. Simulation parameters are given in table 5, and the proposed method is implemented with five different timers as follows:

- Static trickle 12_8
- Static trickle 12_2
- Static trickle 10_2
- Dynamic trickle: basic trickle 12_8, MM trickle 10_2
- Dynamic trickle: basic trickle 12_2, MM trickle 10_2

As given in Fig. 9, the signaling cost and the consumed power of nodes rise as the trickle timer range gets shorter. On the other hand, it makes a higher PDR, fewer lost packets, and shorter hand-off delay. Further, the hand-off delay of 10_2 setting is higher than that of the 12_2, which is caused by the node duty cycling. In other words, sometimes, the DIO packet of a sender is not received by the slept receiver. Accordingly, a delay may occur in detecting the mobility and establishing a connection between MN and new PP even with a relatively high rate of sending DIO packets. The average consumed power and signaling cost produced from each trickle timer are presented in Fig. 10. As shown, the highest power consumption of nodes is in the sending mode. Hence, considering the low data packet rate, the highest signaling cost is consumed for sending the DIO.

In addition, the mobility response time has been examined in this experiment which determines the time-space between the beginning of mobility and parent update. As given in Fig. 9, the mobility response time is shorter with the short-range trickle timers because of the better monitoring of MN with a higher rate of DIO. Moreover, the static trickle timers with a relatively long range have missed the DIO sent by the MN and could not have recognized the mobility

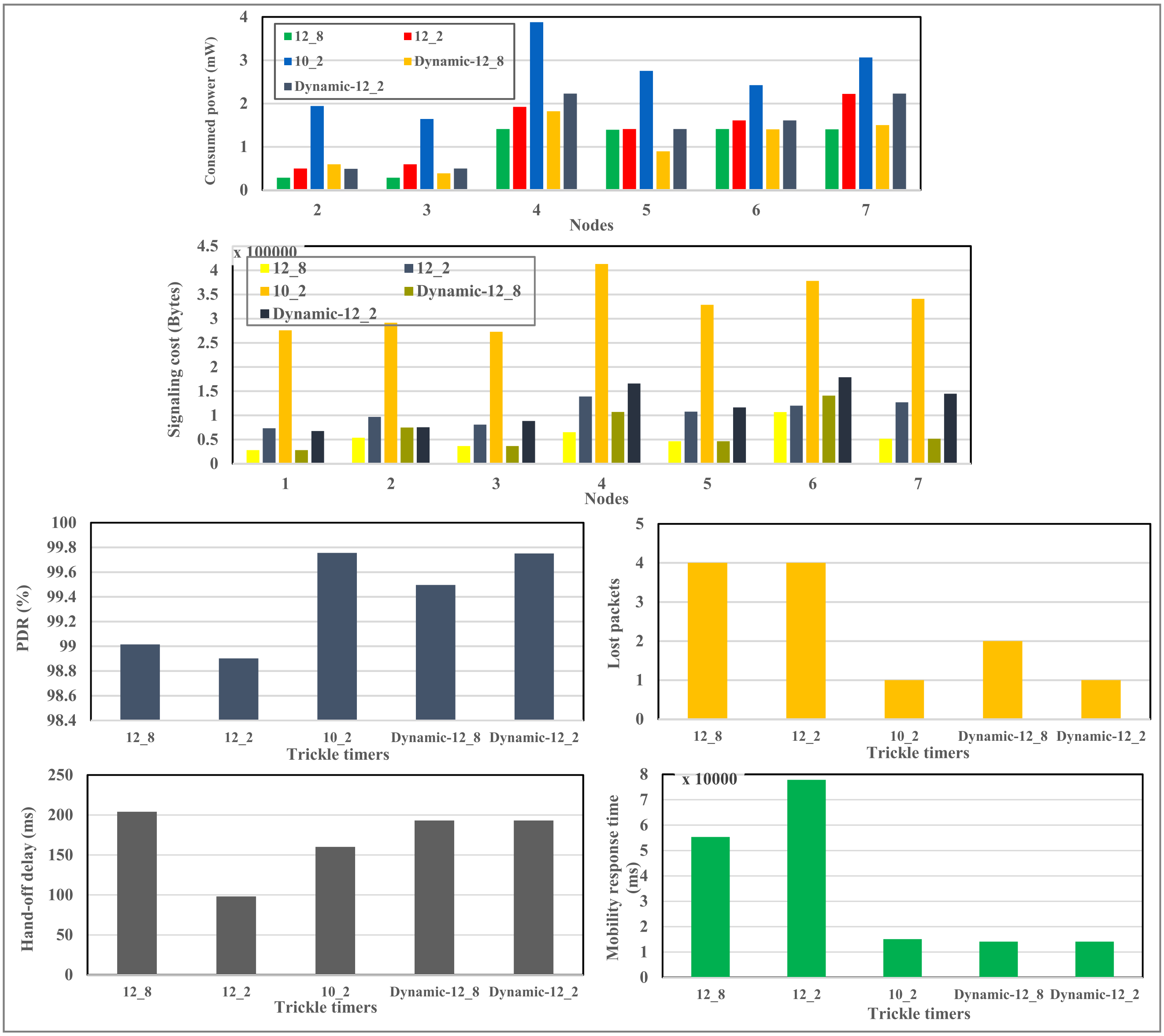

FIGURE 9. Signaling cost, consumed power, PDR, lost packets, hand-off delay and mobility response time.

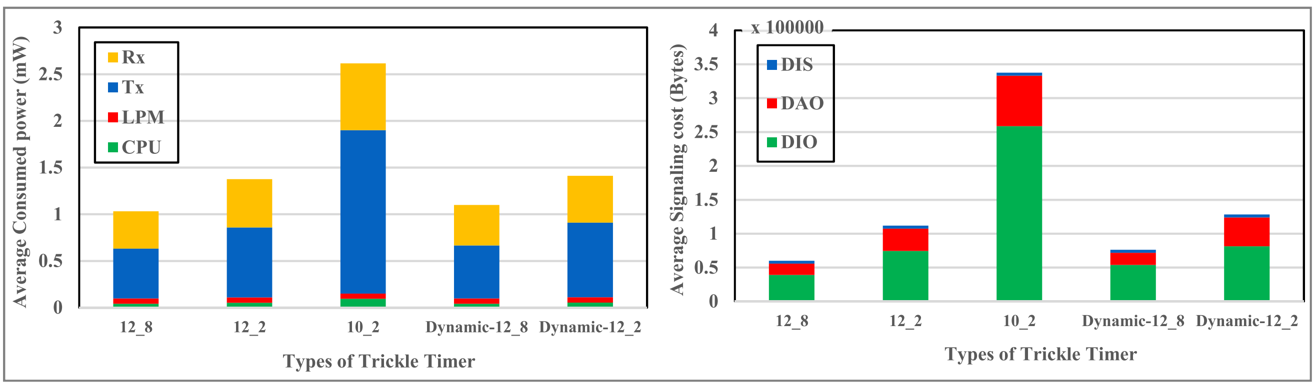

FIGURE 10. The average signaling cost and average power consumption.

and caused the disconnection. Note that we will use the trickle 12_2 as the Basic trickle timer and trickle 10_2 as the MM trickle in the MSE. Therefore, it could produce more signaling cost and power consumption than the RPL and mRPL protocols which use static trickle 12_8 in the following experiments.

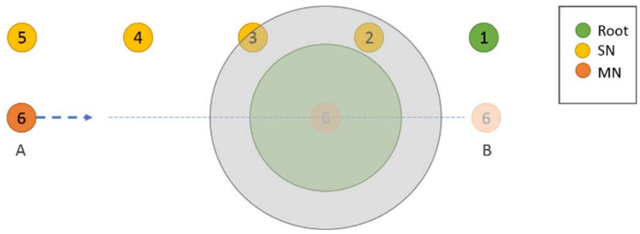

FIGURE 11. Keeping connection during the mobility.

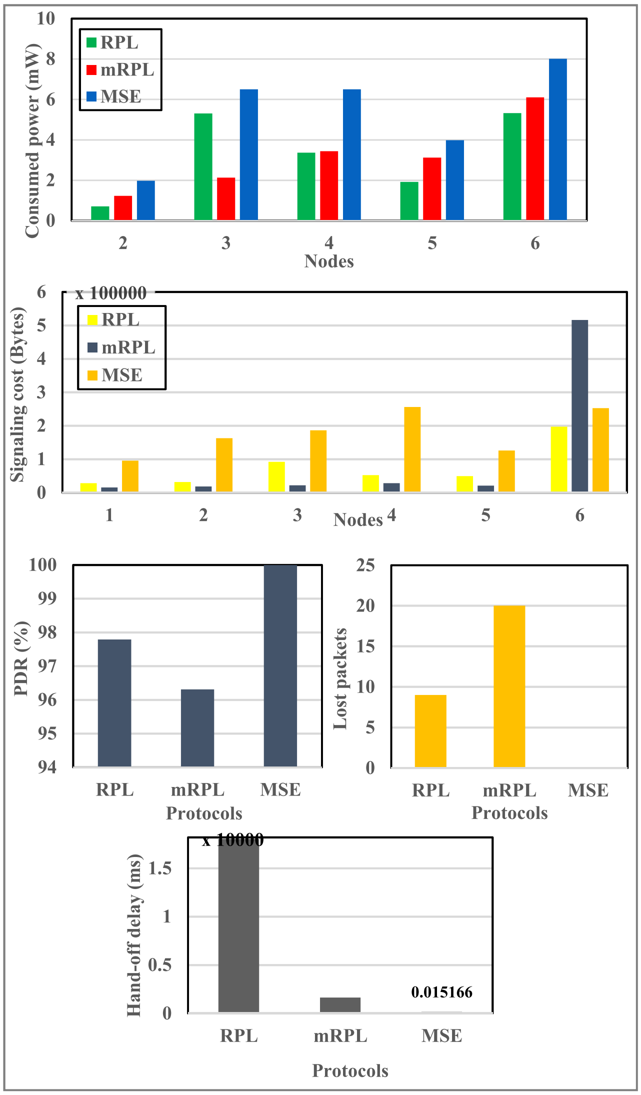

FIGURE 12. Signaling cost, consumed power, PDR, lost packets and hand-off delay.

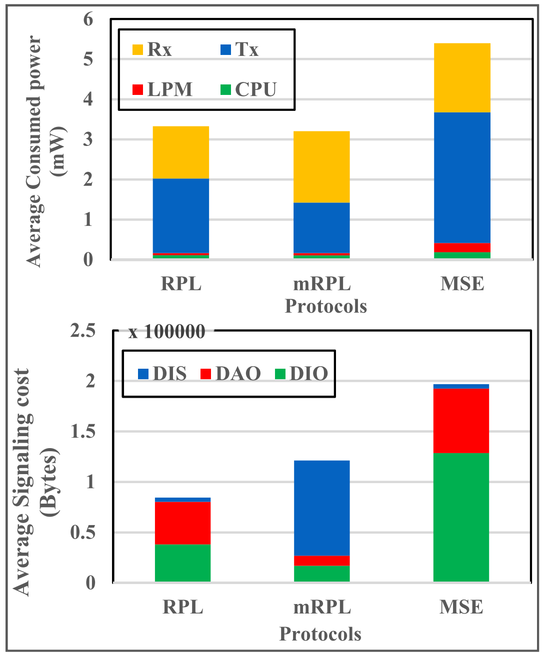

FIGURE 13. The average signaling cost and average power consumption.

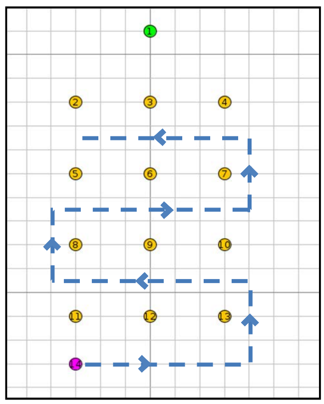

FIGURE 14. Keeping connection during mobility, complex topology.

TABLE 6. Simulation parameters.

Number of nodes	6
Data packet interval	5 s
Simulation time	180 s
Movement starting time of node 6	22 s after beginning

2) SCENARIO 2) KEEPING A SEAMLESS CONNECTION DURING THE MOBILITY

The second experiment involves two scenarios designed to evaluate the performance of the proposed method in providing a seamless connection during the mobility, compared with the mRPL and the standard RPL. A short hand-off delay and high PDR metrics play key roles in this experiment, and the signaling cost and power consumption show the cost to achieve them. These two scenarios are repeated from the mRPL article with some bit of manipulation [23].

As illustrated in Fig. 11, the MN starts moving from A to B with a constant speed and then returns to its first position. In the MSE method, MN connects to an SN as it gets closer to that and disconnects from a node as it gets far from that, then it updates its parent immediately. Other simulation setups are given in table 6.

As given in Fig. 12, the dynamic trickle in MSE implementation produces higher signaling cost and power consumption. However, it causes a very low hand-off delay and 100% PDR without any lost data packet compared with the RPL and mRPL. Moreover, the ratio of mobile node connection time to total node mobility time in MSE is the highest one, with

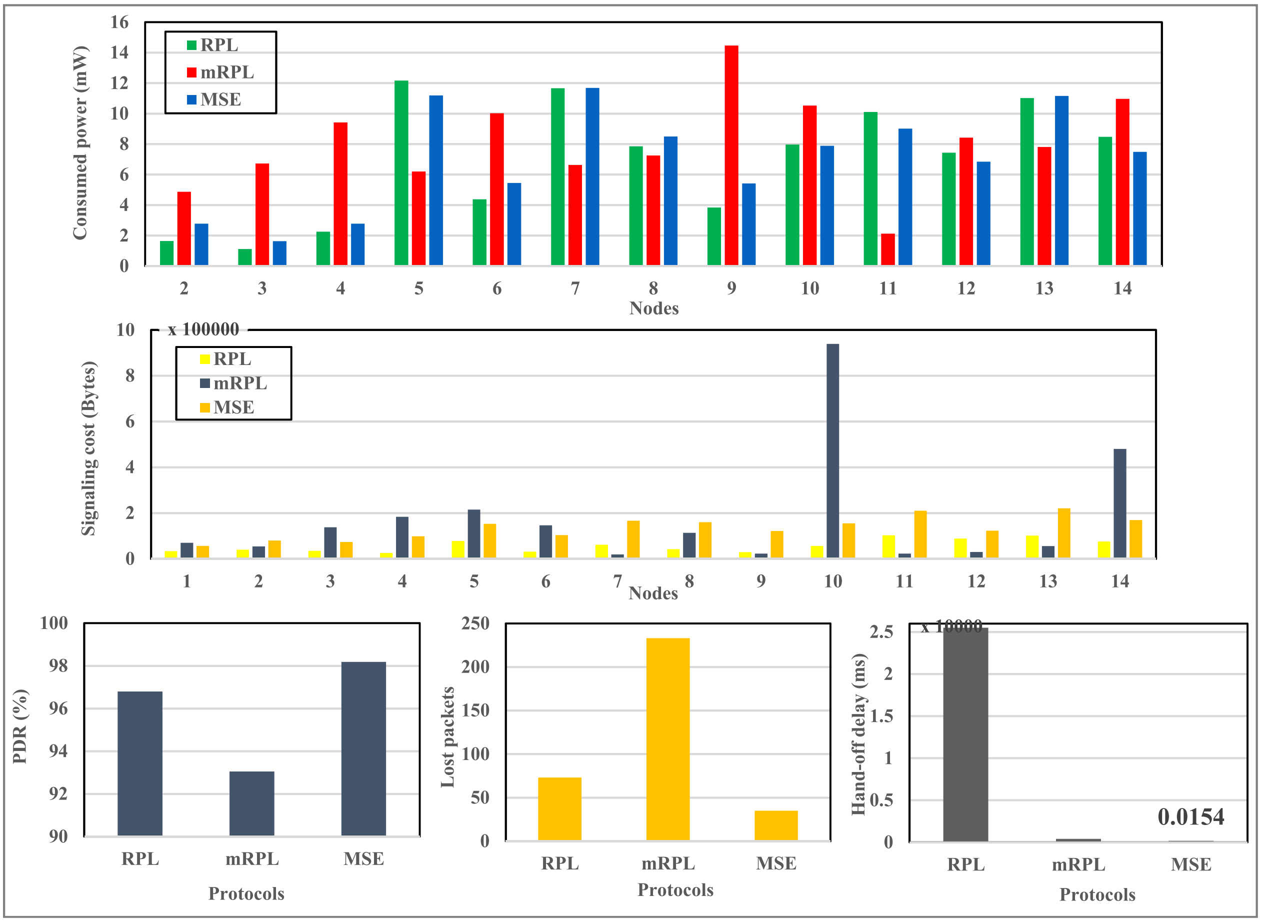

FIGURE 15. Signaling cost, consumed power, PDR, lost packets and hand-off delay.

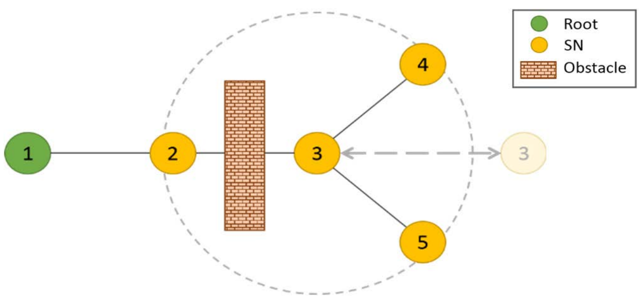

FIGURE 16. Obstacle entrance (dynamic environment).

99.15% being connected compared with 80% in mRPL and 55.7% in standard RPL.

Sending a burst of DIS and receiving many replies in mRPL produces a relatively high network overhead and power consumption. Besides, the mRPL uses the RSSI of data packets to detect mobility. Thus, it could not detect the mobility on time in applications with a low data packet rate like this experiment, and it produces a high signaling cost and power consumption after a particular time, using some predefined timers. As shown in Fig. 13, the maximum average of signaling is costed by sending DIO in the MSE and DIS in mRPL methods. Also, the maximum consumed power in MSE is used in sending mode, which is mainly related to DIO messages considering the low data packet rate.

TABLE 7. Simulation parameters.

Number of nodes	14
Data packet interval	2 s
Simulation time	220 s
Movement starting time of node 14	22.5 s after beginning

3) SCENARIO 3) KEEPING CONNECTION DURING MOBILITY, COMPLEX TOPOLOGY

This scenario is the complex form of the previous scenario, including a longer way for MN, more nodes, and a higher data packet rate.

As illustrated in Fig. 14, node 14 starts moving along the marked path and traverses the entire network. Other simulation parameters are given in Table 7.

According to Fig. 15, we observed that the average power consumption is 7.0586 mW for MSE compared to 8.1068 mW for mRPL and 6.9148 mW for RPL. Moreover, the signaling cost is on average 135110.7 bytes for MSE compared to an average of 177850 bytes for mRPL and 57380.1 bytes for RPL. Further, as shown in Fig. 15, MSE has a better performance by providing a higher PDR, shorter hand-off delay, and fewer lost packets than the other protocols.

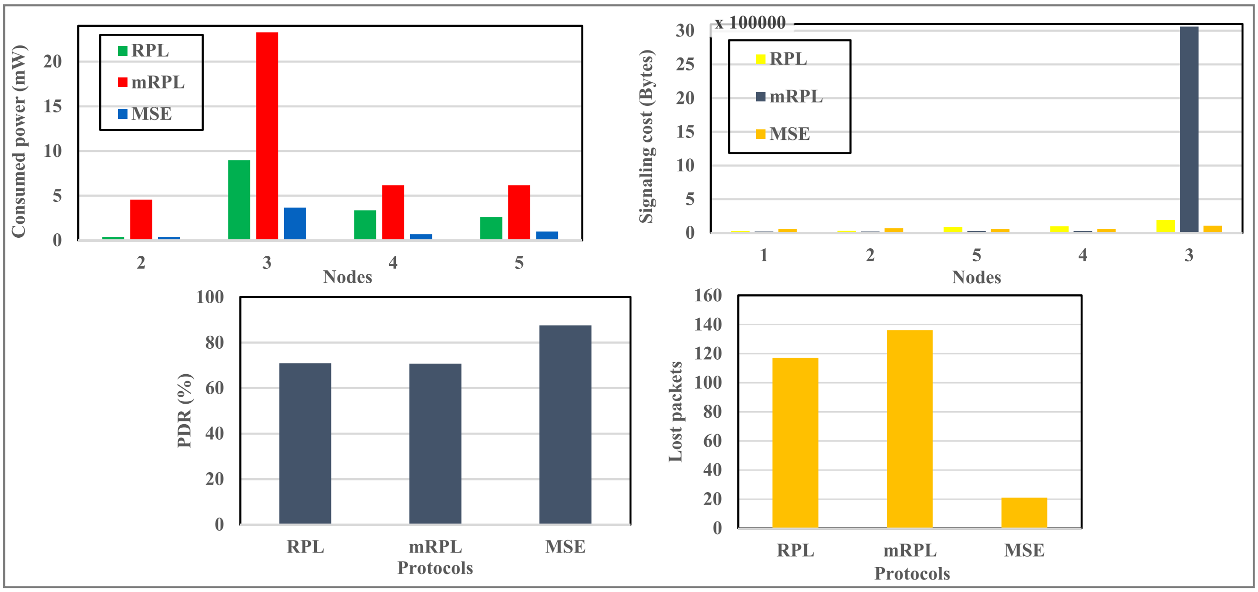

FIGURE 17. Signaling cost, consumed power, PDR and lost packets.

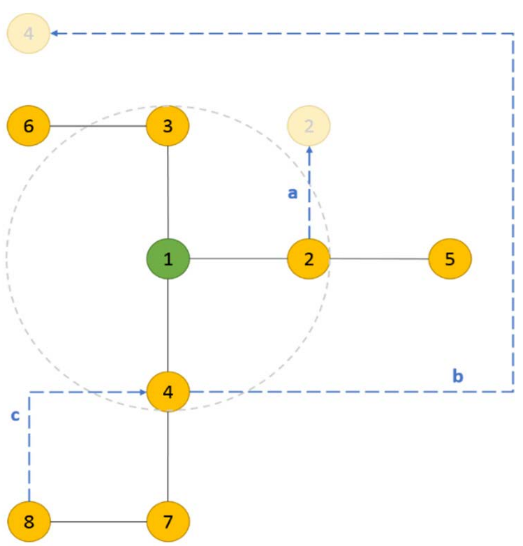

FIGURE 18. Multi-level mobility.

4) SCENARIO 4) OBSTACLE ENTRANCE (DYNAMIC ENVIRONMENT)

This scenario is designed to evaluate the performance of the MSE when an obstacle settles between two paired nodes. As shown in Fig. 16, a physical obstacle enters and settles between nodes 2 and 3 and avoids receiving packets from each other. Since there is no tool to play the role of an obstacle in Cooja, we immediately moved node 3 to the position shown in the figure to simulate the obstacle's entrance. We returned it after 180 milliseconds to evaluate the performance of MSE-RPL in immediate reconnecting the disconnected nodes to the DAG just after the obstacle leaves. Note that the link quality in the second position of node 3 is the same as its first one, while it is out of the radio transmission range of node 2. In this condition, it is expected to stop losing data packets by nodes 3, 4, and 5. The simulation setup is given in table 8.

TABLE 8. Simulation parameters.

Number of nodes	5
Data packet interval	5 s
Simulation time	330 s
Obstacle entrance time	22.5 s after beginning
Obstacle presence time	180 ms

TABLE 9. Simulation parameters.

Number of nodes	8
Data packet rate	0.1 pkt/sec and 30 pkt/sec
Simulation time	300 s
Movement starting of the node 2	22.5 s after beginning
Movement starting of the nodes 4 and 8	180 ms

As shown in Fig. 17, in MSE, nodes 3, 4, and 5 stopped sending and are disconnected from the DAG using the PDW timer 29.7 seconds after the obstacle entrance. Two other protocols could not detect the disconnection, and all nodes kept sending after the obstacle entrance. Thus, MSE produced the least consumed power with an average of 1.4287 mW compared with 10.0352 mW for mRPL and 3.83337 mW for RPL. Similarly, the signaling cost was minimum in MSE with an average of 73280.25 bytes compared with 875208.25 bytes for mRPL and 103285.5 bytes for RPL.

In addition, the PDR in MSE is 17.5% higher than other protocols. A high signaling cost and consumed power of mRPL is caused by sending a burst of DIS from node 3 after a particular time.

5) SCENARIO 5) MULTI-LEVEL MOBILITY (DYNAMIC TOPOLOGY)

The main contribution of our proposed method is to support the mobility of all nodes except the root node, at each level of

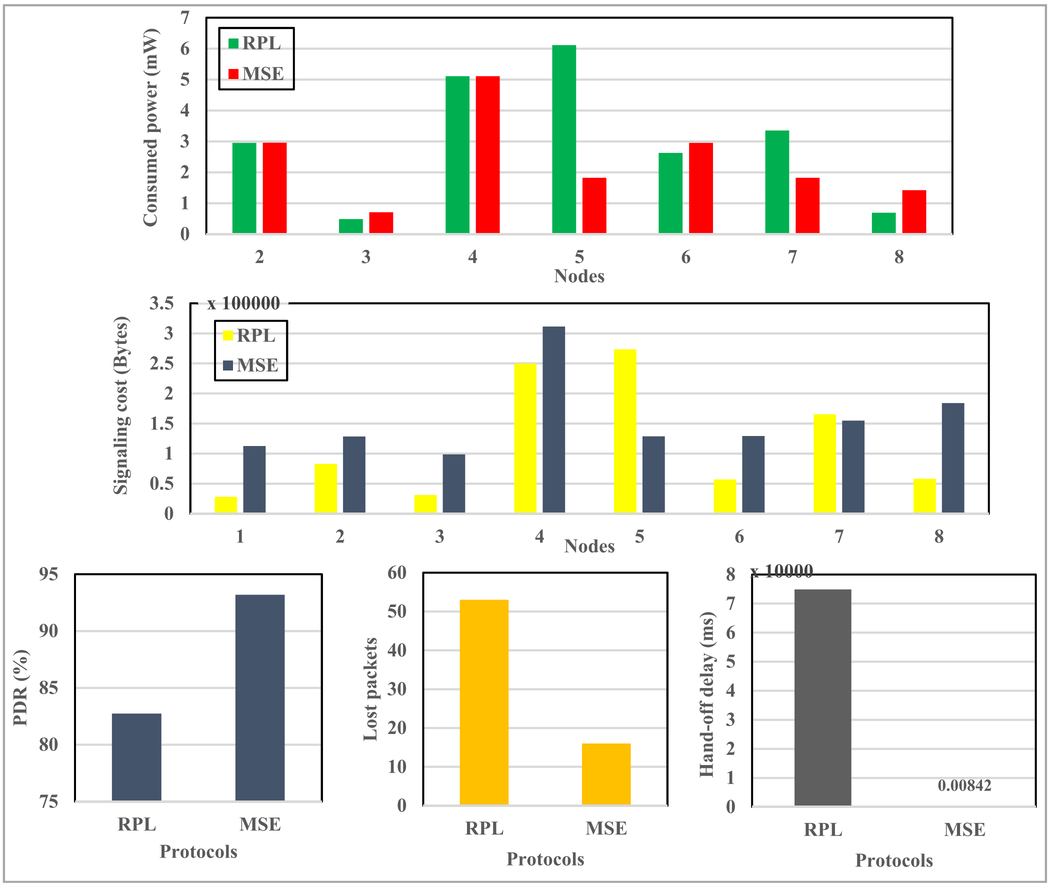

FIGURE 19. Signaling cost, consumed power, PDR, lost packets and hand-off delay.

the DAG and with any number of children. Therefore, this scenario is designed to evaluate the performance of MSE in a dynamic topology compared with the RPL and mRPL protocols. Our main focus in designing this scenario was the mobility of middle nodes, which have some child nodes.

As illustrated in Fig. 18, node 2 starts moving and follows path (a) to the marked position. Then nodes 4 and 8 start moving simultaneously on paths (b) and (c), respectively. Finally, node 8 will settle on the previous position of node 4, and the 4 will stop moving next to node 6. Hence, two middle nodes and one leaf node are mobile while the others are static nodes. There are three practical challenges embedded in this scenario: (i) node 5 must be disconnected from the DAG after node 2 moves away, (ii) node 4 must keep its connection to the DAG while it moves beside nodes 2, 3, and 6, while it should not connect to node 5 when it is disconnected, and (iii) nodes 7 as a parent node, and 8 as a child node should change their roles when the node 8 settles on the previous position of node 4. We ran this scenario with two different data packet rates. Other simulation parameters are given in Table 9.

- Results of low data packet rate (0.1 pkt/sec)

Performing this scenario by mRPL with a low data packet rate caused sending burst of DIS messages and an enormous network overhead; thus, we omitted its results from this simulation. Fig. 19 portrays the result of what MSE and RPL have performed in this scenario. As shown, MSE produced less signaling cost with an average of 162110.2 bytes than 131076.2 bytes for RPL. Similarly, the power consumed in MSE with an average of 2.3958 mW is less than RPL with an average of 3.0457 mW. Furthermore, the PDR is 10.4% higher in MSE than that in RPL. A fewer number of lost packets and a Distinctive shorter hand-off delay are also observable in the figure.

- Results of high data packet rate (30 pkt/sec)

Fig. 20 shows that the MSE has produced the least consumed power with an average of 3.8583 mW compared with 13.0314 mW for mRPL and 10.0834 mW for RPL. The signaling cost in MSE with an average of 22370.5 bytes is a bit higher than RPL with an average of 15710.5 bytes, but it is significantly lower than mRPL with an average of 238657.7 bytes. Besides, as shown in the figure, the MSE had a better performance by a 100% PDR without any lost packet and a shorter hand-off delay than the mRPL and RPL protocols. The considerable signaling cost produced by the mRPL has been caused by sending a burst of DIS messages.

Since the hand-off process offered in the mRPL had failed in some of the above scenarios, we repeated scenarios 2 and 3 in the designed condition of the mRPL article [23], regardless of the duty cycling of nodes (only in one experiment).

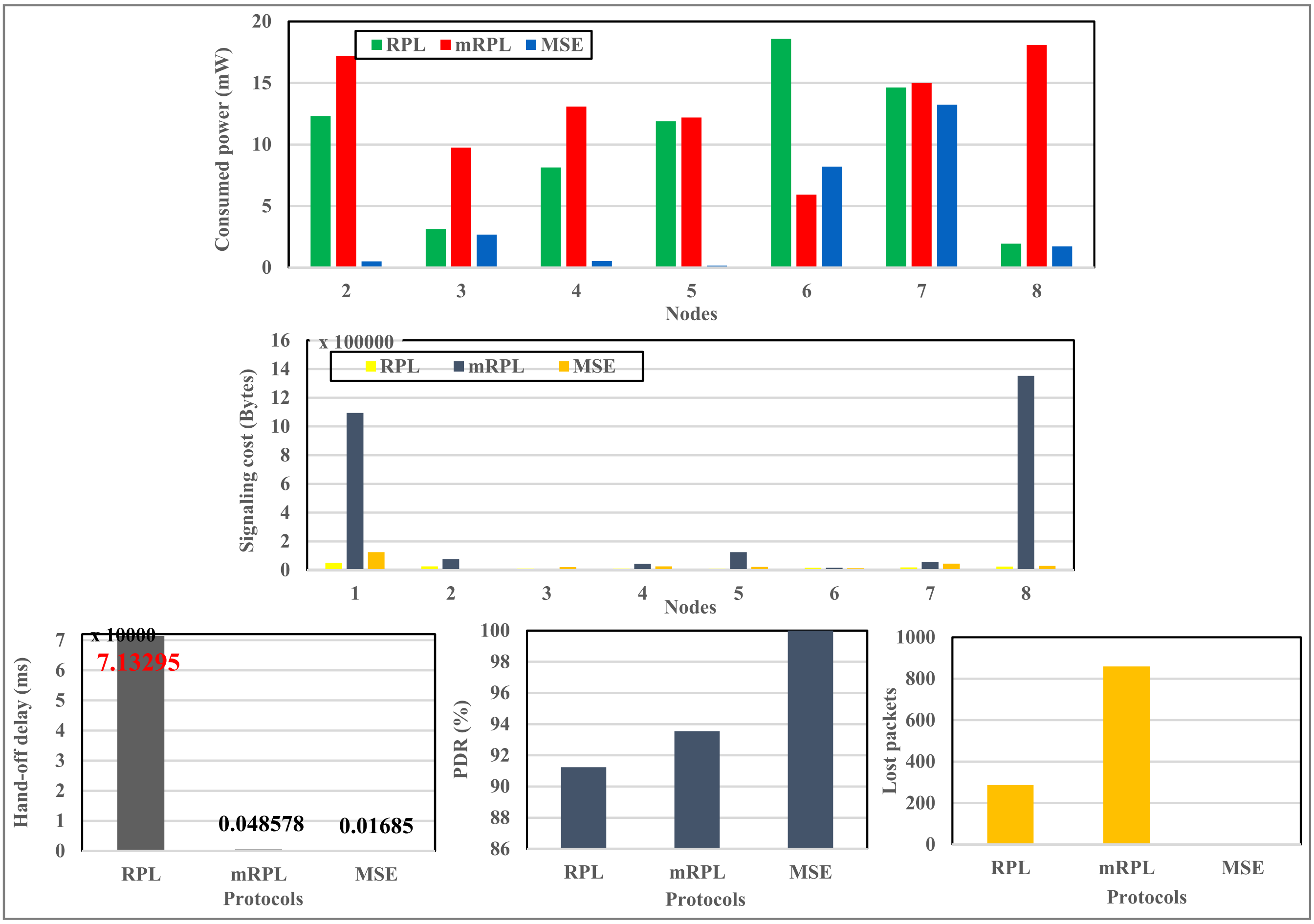

FIGURE 20. Signaling cost, consumed power, PDR, lost packets and hand-off delay.

Both mRPL and MSE methods completed all hand-off processes successfully. Finally, we calculated the average of the hand-off delays produced by each method. The result showed that MSE had a significantly shorter hand-off delay with an average of 3 milliseconds than 110.5 milliseconds for the mRPL. That is because the hand-off process in the mRPL includes sending a burst of DIS messages, receiving replies, selecting the best parent with the highest link quality, and sending DAO to that, while the best candidate parent is always predetermined in the MSE, and the hand-off process could be summarized to sending a DAO message to the best parent when it is needed.

VI. CONCLUSION AND FUTURE WORKS

We proposed a method to improve the mobility support in RPL, called MSE, which can manage the mobility of nodes in the Internet of Things mobile applications. MSE supports the mobility of all nodes except the root node and provides a seamless connection during the mobility for mobile nodes using a dynamic trickle timer with two different ranges, DIO-waiting timers, a neighbor link quality table, a function to select the best parent in case of mobility, confidence and critical zones and a blacklist. MSE can manage a situation when a physical obstacle settles between two paired nodes in a dynamic environment. Some variable parameters and tools are used in the MSE, a fine-tuned formulation of which could bring about high performance in a fully dynamic topology and the environment with any packet rate, despite the delay and loss-sensitive data.

We have evaluated the performance of our proposed method and compared it to the mRPL and the standard RPL in multiple varied scenarios simulated in the Contiki / Cooja simulator. The simulation results showed that the MSE, by a slight increase in signaling cost and power consumption, significantly reduces hand-off delay, increases Packet Delivery Ratio, and reduces the number of lost data packets. We are working on more complicated scenarios with a higher number of nodes using more powerful computing resources to evaluate our proposed method on some actual testbeds and apply further comprehensive evaluations. We are also designing some scenarios to compare the performance of MSE-RPL with that of other related protocols.

MSE may have an inappropriate performance in applications with severe limitations in node memory or where the maximum speed of nodes is relatively high. In the future, some different link quality metrics other than the Received Signal Strength Indicator (RSSI) could be exerted. Besides, a combination of multiple metrics could be used to select the best parent, including the node energy, network congestion, and other metrics that could be exclusively set in each application.

DECLARATION OF COMPETING INTEREST

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this article.

REFERENCES

[1] J. Broch, D. A. Maltz, D. B. Johnson, Y.-C. Hu, and J. Jetcheva, ''A performance comparison of multi-hop wireless ad hoc network routing protocols,'' in *Proc. ACM MobiCom*, 1998, p. 109.

[2] S. Biswas and R. Morris, ''ExOR: Opportunistic multi-hop routing for wireless networks,'' *ACM SIGCOMM Comput. Commun. Rev.*, vol. 35, no. 4, pp. 133–144, 2005.

[3] K. Akkaya and M. Younis, ''A survey on routing protocols for wireless sensor networks,'' *Ad Hoc Netw.*, vol. 3, no. 3, pp. 325–349, May 2005.

[4] O. Gaddour and A. Koubâa, ''RPL in a nutshell: A survey,'' *Comput. Netw.*, vol. 56, no. 14, pp. 3163–3178, 2012.

[5] B. Ghaleb, A. Y. Al-Dubai, E. Ekonomou, A. Alsarhan, Y. Nasser, L. M. Mackenzie, and A. Boukerche, ''A survey of limitations and enhancements of the IPv6 routing protocol for low-power and lossy networks: A focus on core operations,'' *IEEE Commun. Surveys Tuts.*, vol. 21, no. 2, pp. 1607–1635, 2nd Quart., 2018.

[6] A. Oliveira and T. Vazão, ''Low-power and lossy networks under mobility: A survey,'' *Comput. Netw.*, vol. 107, pp. 339–352, Oct. 2016.

[7] O. Iova, P. Picco, T. Istomin, and C. Kiraly, ''RPL: The routing standard for the Internet of Things... Or is it?'' *IEEE Commun. Mag.*, vol. 54, no. 12, pp. 16–22, Dec. 2016.

[8] J. M. L. P. Caldeira, J. J. P. C. Rodrigues, and P. Lorenz, ''Toward ubiquitous mobility solutions for body sensor networks on healthcare,'' *IEEE Commun. Mag.*, vol. 50, no. 5, pp. 108–115, May 2012.

[9] J. Martocci, P. De Mil, N. Riou, and W. Vermeylen, *Building Automation Routing Requirements in Low-Power and Lossy Networks*, document RFC 5867, 2010.

[10] T. Dohler, T. Watteyne, T. Winter, and D. Barthel, *Routing Requirements for Urban Low-Power and Lossy Networks*, document RFC 5548, 2009.

[11] T. Winter, *RPL: IPv6 Routing Protocol for Low-Power and Lossy Networks*, document RFC 6550, Internet Engineering Task Force (IETF), 2012.

[12] N. Pradeska, W. Widyawan, W. Najib, and S. S. Kusumawardani, ''Performance analysis of objective function MRHOF and OF0 in routing protocol RPL IPV6 over low power wireless personal area networks (6LoWPAN),'' in *Proc. 8th Int. Conf. Inf. Technol. Electr. Eng. (ICITEE)*, Yogyakarta, Indonesia, Oct. 2016, pp. 1–6.

[13] H. Kharrufa, H. A. A. Al-Kashoash, and A. H. Kemp, ''RPL-based routing protocols in IoT applications: A review,'' *IEEE Sensors J.*, vol. 19, no. 15, pp. 5952–5967, Apr. 2019.

[14] S. Goyal and T. Chand, ''Improved trickle algorithm for routing protocol for low power and lossy networks,'' *IEEE Sensors J.*, vol. 18, no. 5, pp. 2178–2183, Mar. 2018.

[15] P. O. Kamgueu, E. Nataf, and T. D. Ndie, ''Survey on RPL enhancements: A focus on topology, security and mobility,'' *Comput. Commun.*, vol. 120, pp. 10–21, May 2018.

[16] C. Cobârzan, J. Montavont, and T. Noel, ''Integrating mobility in RPL,'' in *Proc. Eur. Conf. Wireless Sensor Netw.*, 2015, pp. 135–150.

[17] O. Gaddour, A. Koubaa, R. Rangarajan, O. Cheikhrouhou, E. Tovar, and M. Abid, ''Co-RPL: RPL routing for mobile low power wireless sensor networks using corona mechanism,'' in *Proc. 9th IEEE Int. Symp. Ind. Embedded Syst. (SIES)*, Jun. 2014, pp. 200–209.

[18] J. Ko and M. Chang, ''MoMoRo: Providing mobility support for low-power wireless applications,'' *IEEE Syst. J.*, vol. 9, no. 2, pp. 585–594, Jun. 2015.

[19] H. Kim, H.-S. Kim, and S. Bahk, ''MobiRPL: Adaptive, robust, and RSSI-based mobile routing in low power and lossy networks,'' *J. Commun. Netw.*, vol. 24, no. 3, pp. 365–383, Jun. 2022.

[20] Z. Shah, A. Levula, K. Khurshid, J. Ahmed, I. Ullah, and S. Singh, ''Routing protocols for mobile Internet of Things (IoT): A survey on challenges and solutions,'' *Electronics*, vol. 10, no. 19, p. 2320, Sep. 2021.

[21] B. Tian, K. M. Hou, H. Shi, X. Liu, X. Diao, J. Li, Y. Chen, and J.-P. Chanet, ''Application of modified RPL under VANET-WSN communication architecture,'' in *Proc. Int. Conf. Comput. Inf. Sci.*, Jun. 2013, pp. 1467–1470.

[22] A. Mohammadsalehi, B. Safaei, A. M. H. Monazzah, L. Bauer, J. Henkel, and A. Ejlali, ''ARMOR: A reliable and mobility-aware RPL for mobile Internet of Things infrastructures,'' *IEEE Internet Things J.*, vol. 9, no. 2, pp. 1503–1516, Jan. 2022.

[23] H. Fotouhi, D. Moreira, and M. Alves, ''mRPL: Boosting mobility in the Internet of Things,'' *Ad Hoc Netw.*, vol. 26, pp. 17–35, Mar. 2015.

[24] M. Bouaziz, A. Rachedi, A. Belghith, M. Berbineau, and S. Al-Ahmadi, ''EMA-RPL: Energy and mobility aware routing for the internet of mobile things,'' *Future Gener. Comput. Syst.*, vol. 97, pp. 247–258, Aug. 2019.

[25] J. V. V. Sobral, J. J. P. C. Rodrigues, R. A. L. Rabêlo, J. Al-Muhtadi, and V. Korotaev, ''Routing protocols for low power and lossy networks in Internet of Things applications,'' *Sensors*, vol. 19, no. 9, p. 2144, 2019.

[26] P. Deshpande. (Nov. 8, 2014). *ANRG*. [Online]. Available: https://anrg.usc.edu/contiki/index.php/Mobility_of_Nodes_in_Cooja

[27] *Zolertia Z1 Datasheet*. (Mar. 11, 2010). [Online]. Available: http://wiki.zolertia.com/wiki/images/e/e8/Z1_RevC_Datasheet.pdf

[28] A. Musaddiq, Y. B. Zikria, Z. Nain, and S. W. Kim, ''Routing protocol for low-power and lossy networks for heterogeneous traffic network,'' *EURASIP J. Wireless Commun. Netw.*, vol. 2020, no. 1, pp. 1–23, Dec. 2020.

[29] B. H. Patel and P. Shah, ''RPL routing protocol performance under sinkhole and selective forwarding attack: Experimental and simulated evaluation,'' *Telkomnika*, vol. 18, no. 4, pp. 1849–1856, 2020.

AMIRREZA VAEZIAN received the B.Sc. degree in computer engineering from the Shahrood University of Technology, Semnan, Iran, in 2014, and the M.Sc. degree in information technology engineering from the K. N. Toosi University of Technology (KNTU), Tehran, Iran, in 2021. He is a member of the Network Laboratory, Department of Computer Engineering, where he works on the mobile IoT applications. His main research interests include the Internet of Things (IoT), embedded systems, cyber-physical systems (CPS), edge computing, fog computing, and machine learning.

YOUSEF DARMANI received the Ph.D. degree in computer networking from the University of Adelaide, Adelaide, Australia, in 2004. He is currently an Associate Professor and the Dean of the Computer Engineering Faculty, K. N. Toosi University of Technology. His main research interests include VOIP, real time communication over the internet, wireless and ad-hoc networks and their protocols, and computer hardware.