IET Networks

Research Article

Proposing a new method for improving RPL to support mobility in the Internet of things

ISSN 2047-4954
Received on 28th August 2019
Revised 13th December 2019
Accepted on 6th January 2020
E-First on 13th February 2020
doi: 10.1049/iet-net.2019.0152
www.ietdl.org

Samaneh Hoghooghi[1], Reza Javidan[1] ✉
[1]Department of Computer Engineering and Information Technology, Shiraz University of Technology, Shiraz, Iran
✉ *E-mail: rezajavidan@gmail.com*

Abstract: The internet engineering task force working group has proposed IPv6 Routing Protocol for Low-power and Lossy (RPL) networks for the energy-constrained networks on the Internet of things. This protocol has limitations and does not pay particular attention to mobile nodes. We introduce a new method to improve the RPL protocol with the aim to provide continuous connectivity for mobile nodes. Using additional fields to RPl control packets, a parent is responsible for mobility detection and informs a mobile node for maintaining the energy of it and reduces network overheads. A mobile node chooses a new parent based on metrics namely expected transmission count, received signal strength indicator and residual energy, before disconnection from previous parent. Memory has also been defined that maintains a list of previous parents and when selecting a new parent, a mobile node gets help from this list to select stable parents. They used the Cooja simulator to evaluate the performance of the proposed protocol and the results compared with the RPL and mRPL from the aspect of packet delivery ratio, network overhead, end-to-end delay, energy consumption, and end-to-end throughput under different scenarios. The simulation results showed the effectiveness of the proposed method.

1 Introduction

Unprecedented growth in the number of objects which interact with each other through the Internet is the idea of the Internet of things (IoT) [1, 2]. Various environments can be mentioned that IoT can play a significant role and improve the quality of our lives. These applications include transportation, health care, industrial automation, smart building, smart city etc. [3, 4]. A subset of the wireless sensor network (WSN) is a low-power and lossy network (LLN), a collection of interconnected [5] embedded devices such as smart objects. Smart objects interact with the physical world by obtaining information from the physical world with their sensors and by affecting the physical world with their actuators [6]. Sensors are typically small and inexpensive. This means they often have limited power, CPU and memory [4], and they operate in constrained environments with short communication ranges, limited transmission speeds, low bandwidth, frame size limitations and a high rate of packet failure [6, 7]. LLNs are attracted by academia and industries because of the low cost and improvements that can be made in the quality of service (QoS) [1]. With the emergence of LLNs, new ways of communication were needed. IEEE 802.15.4 is a standard that was developed to provide a framework and the lower layers in the Open System Interconnection model for low-power and lossy wireless connectivity networks. IEEE 802.15.4 provides the medium access control and physical layers [8]. Maximum transmission unit (MTU) in this standard is 127 B; in contrast, IPv6 requires the MTU to be at least 1280 B [9]. The 6LOWPAN technology was proposed by the Internet engineering task force (IETF) working group to combine IPv6 protocol and IEEE 802.15.4 [8]. IPv6 packets offered a more important address space that helps to increase the number of nodes. 6LoWPAN is used to create a set of headers that allow for large IPv6 address/headers into smaller compressed header from 40 to 2–11 B [10].

The IPv6 routeing protocol for low-power and lossy (RPL) networks has been introduced by IETF-routeing over low-power and lossy link (ROLL) group in 2012 as a proposed standard routeing protocol for LLNs [11]. Since then, several types of research have reported that RPL has many limitations and weaknesses. One of these weaknesses has lacked the support of mobility [5]. Nowadays, mobility is a major factor present in human and non-human lives [12]. Numerous applications require to support mobility; for example, health care application keeps continuous monitoring of a patient's vital signs while the patient is moving [13]. However, mobility can cause some problems such as disconnection of nodes, data loss, increase delay, consume more energy and decrease application performances [14]. RPL is very efficient in static WSNs and it supports reduced mobility [15].

The needs of applications for using protocols that support mobility and also the low efficiency of RPL protocol have motivated us to improve this protocol. The protocol has no limitations in terms of the presence of mobile nodes in the routeing topology. However, studies have revealed that when a mobile node is present in the network, the connection continuity between the mobile node and its parent node is lost, which leads to packet loss and an increase in Internet Control Message Protocol (ICMP) control messages, and consequently increases network overhead. Furthermore, the nodes should process the received messages, which cause energy wastage.

In the present paper, the RPL protocol is improved to enhance its support for mobility in the network. The main goal of the present paper is proposing a reliable method for mobility detection and parent selection based on the metrics, namely expected transmission count (ETX), received signal strength indicator (RSSI) and residual energy.

The main contributions of this paper can be summarised as follows:

- The parent node is responsible for checking the communication link through additional fields in control packets. When the link becomes weak, it will inform the mobile node, so the mobile node consumes less energy, and the network overhead is reduced while providing seamless connectivity.
- The mobile node selects its parent node based on three parameters (ETX, RSSI and residual energy).
- Memory space is assigned to mobile nodes to store the nodes' addresses that participated in the parent node selection process. In the next steps, those nodes are assigned with a lower priority during the selection process. The idea behind this mechanism is to use this information at the next step to avoid more hand-off processes, select the stable parent and stable link and maintain

maximum connectivity time between the mobile nodes and its parent to avoid data losses and to reduce the delay.

- We have defined two different scenarios so we can extend simulations to validate the proposed protocol.
- This paper is structured as follows: in Section 2, we present an overview of the standard RPL, and then works that proposed a solution to support mobility for the RPL are discussed. In Section 3, the framework of the proposed protocol is explained. Section 4 discusses the proposed protocol in details. In Section 5, the performance of the proposed protocol is evaluated using Cooja simulator. Finally, in Section 6, the conclusions are presented and references are mentioned in Section 7.

2 Background and related works

ROLLs are a working group of IETF that handles IoT routeing topics. RPL is a routeing protocol based on IPv6 for resource-limited embedded devices defined by IETF in March 2012 [11]. We describe the RPL protocol and related works of mobility management protocols in this section.

2.1 RPL overview

RPL employs a distance vector routeing algorithm to fulfil routeing requirements through building a robust topology. Moreover, RPL supports three traffic flows [5] such as multipoint-to-point (MP2P), point-to-multipoint (P2MP) and point-to-point (P2P) [15, 16]. RPL is based on the topological concept of directed acyclic graphs (DAGs). The DAG defines a cluster-tree structure that specifies the default routes between nodes. More specifically, each DAG is rooted at a single destination (i.e. sinks or border router) and is referred to as a destination-oriented DAG (DODAG) in RPL [17]. RPL uses the term instance to refer to multiple DODAGs that share the same routeing policies and mechanisms. This routeing protocol uses the term upward routes to refer to routes that carry the traffic from each node to the DODAG root (i.e. MP2P), whereas routes that carry the traffic from the DODAG root to other nodes (i.e. P2MP) are called the downward routes. To build the upward routes, each node must select one of its neighbours as its preferred parent (next hop) toward the root but they have no information about related children.

As illustrated in Fig. 1, to exchange routeing information, RPL introduces four ICMPv6-type control packets [5, 18]:

- *DODAG information object (DIO):* The DIO packet is created by the DODAG root to construct a new DAG and then sent it in multicast through the DODAG. The DIO message is used to carry the relevant information and configuration parameters that enable a node to discover an RPL instance, join a specific DODAG, select a set of candidate parents according to its objective function (OF) and routeing information (e.g. RANK, DODAG ID) [19], and maintain the DODAG.
- *Destination advertisement object (DAO):* DAO messages are sent by each node to populate the routeing tables with prefixes of their children and to advertise their addresses and prefixes to their parents and the DODAG root so that the downward routes from the DODAG root to its associated nodes can be built.
- *DODAG information solicitation (DIS):* A newly joined node that does not have a route multicasts the DIS message to solicit a DIO from an RPL node. A node may use DIS to probe its neighbourhood for alongside DODAGs.
- *DAO acknowledgement (DAO-ACK):* The DAO-ACK is sent as a unicast packet by a DAO recipient to the DAO sender to acknowledge the reception of that DAO message.

RPL uses an adaptive timer mechanism called the 'trickle timer' [20]. This mechanism controls the sending rate of DIO messages. The interval of this timer exponentially increases as the network conditions are considered stable, which results in fewer DIO messages being sent in the network. As inconsistencies are detected, the nodes reset the trickle timer and send DIOs more often. The frequency of the DIO messages depends on the stability of the network and the frequency is increased, where the

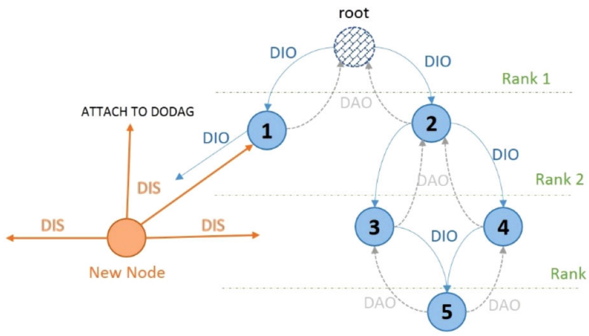

Fig. 1 *Illustration of the RPL mechanism*

inconsistency is detected [21]. In other words, as the network becomes stable, the number of RPL messages decreases. When an inconsistency is detected (such as mobility, network loop or preferred parent change) the timers are reset to fix the issue quickly. This mechanism may negatively affect the ability of RPL to detect any topology changes using routeing control packets quickly.

2.2 Related works

In typical applications of IoT, it is common that things frequently move. Several works have tried to propose a solution to support mobility in RPL; a number of these solutions are discussed.

Mobility-enhanced RPL (ME-RPL) [22] focuses on the dynamic allocation of DIS interval to manage the changes in topology based on the frequently preferred parent switching. If the number of the preferred parent changes increases, the mobile node broadcasts more DIS messages in a small period, which, in turn, increases the DIO messages, and finally, it causes network overhead and congestion.

Somaa *et al.* [23] proposed a new method called the Bayesian model mobility prediction to provide mobile nodes in WSNs. A new metric was designed for the node status and information lost. However, this protocol requires exchanging a large amount of data between the network nodes. Similarly, in [24], Barcelo *et al.* proposed by Kalman positioning RPL (KP-RPL) to support mobile nodes. In the KP-RPL, for a mobile node, a confidence region is defined using standard Kalman filter, and then a new parent is selected based on ETX and estimated its position. Since KP-RPL uses a linear model, Bouaziz *et al.* [25] provided seamless connectivity by using the extended Kalman filter (EKP-mRPL) to forecast the new parent for the mobile node with a non-linear strategy based on the EKP. KP-RPL and EKP-mRPL are both position-based routeing and that position is estimated through a confidence region defined by the standard Kalman filter.

Fotouhi *et al.* [26] proposed mRPL to support the mobile nodes that use a proactive mechanism in RPL called smart-HOP. Mobility detection is based on the RSSI when it reaches a threshold value, then it entered the discovery phase. Although, the delay time for each hand-off is decreased compared with RPL, it is suitable for a network where a significant amount of server node is available. Owing to using a hard hand-off method, the packet loss during hand-off rises. As a result, the authors proposed another mechanism (mRPL+) [27] that the mobile node chooses a new attach point before disconnecting from the previous parent.

Anand and Tahiliani [28] proposed the mRPL++ protocol to extend and improve the mRPL. It chooses the best parent from neighbours based on RSSI and ETX. Nevertheless, mRPL++ requires larger signalling overhead and consumes more energy. This protocol has mRPL weaknesses. Also, it is not efficient in some cases and leads to an increased handover delay.

An energy-efficient and mobility-aware routeing protocol called EC-mRPL is proposed by Bouaziz *et al.* [29]. This protocol aims to improve the functionality of the mRPL and reduce the energy consumption of the mobile node. The basic opinion was to reduce mobile node involvement to save energy. Besides, reducing the number of exchanged control messages between the parent and mobile nodes helps reduce the link overload, which, in turn, decreases the data loss.

Sanshi and Jaidhar [30] proposed an algorithm that fixed nodes and mobile nodes used multiple routeing metrics for supporting mobility in LLNs. When the mobile node changed its location, it selected a new preferred parent in the direction of the movement based on metrics in OFs. The end-to-end delay in the EM-RPL was high, which affected the performance of the protocol.

Murali and Jamalipour [31] proposed the algorithm that supports random mobility of the nodes in RPL and chooses the best parent from the preferred parent list based on the multiple metrics such as ETX, expected life time (ELT), RSSI and so on. The authors proposed a dynamic trickle algorithm for trickle timer to allocate timer dynamically based on the random set of neighbour nodes under mobility. The authors have not measured the amount of network overhead. This parameter is important because it affects energy consumption.

A comparison of the previous works and our proposed protocol is presented in Table 1.

Table 1 Comparison of previous works and our proposed protocol

Protocol	OF	Mobility detection method	Additional layer	Overhead
RPL	ETX	use trickle timer	no	medium
ME-RPL	ETX	dynamic DIS interval	no	high
Bayesian model Mobility Prediction-RPL	new metric was designed	mobility prediction using Bayesian model	no	medium
KP-RPL	ETX, RSSI	KP and blacklisting	no	high
EKP-mRPL	RSSI	EKP	no	medium
mRPL	RSSI	control message and hard hand-off	no	high
mRPL +	RSSI	control message and hard and soft hand-off	no	high
mRPL + +	RSSI, ETX	control message and hand-off	no	high
EC-mRPL	RSSI	mobility prediction with the control message	no	high
EM-RPL	ETX, RSSI, energy, timers	use timer	no	medium
Murali	ETX, ELT, RSSI, distance	use dynamic trickle (D_Trickle) algorithm	no	medium
proposed protocol	ETX, RSSI, energy	use control message and additional timer to handle mobility	no	medium

3 Problem formulation

RPL routeing protocol, employed in IoT applications, has some shortcoming when used in large-scale networks including mobile nodes. There is an increasing growth in the use of mobile tools. Thereby, an increasing need for mobility support in low-power networks and applications that require this capability such as health care monitoring, smart city, industrial automation and home automation. Communication links in such networks are not reliable and stable, which adversely impact the QoSs. Achieving a high QoS is much harder in the presence of the mobile node. Indeed, mobility is not a main requirement of the RPL and devices are static. Therefore, the proposed protocol has been designed to provide continuous connectivity for mobile nodes and find the best parent using a different number of metrics and timers. The metrics are used to select the best parent when the mobile node gets away from transmission range, while both the timers maintain connectivity when a mobile node leaves the communication range and prevent congestions. We also have a list of candidate parents that will be updated for the later step. First, we state the assumptions and terms used in this paper:

(a) The proposed protocol focuses on four steps, as shown in Fig. 2. In the first step, the mobile node transfers the data with the parent. In this step, the value of Average received signal strength indicator (ARSSI) is measured by the parent node. When the parent node detects any mobility, it informs the mobile node to enter the next step. In the next phase, the mobile node should find a proper parent to transfer its data and designate a new route. In the last phase, the mobile node updates the routes in the routeing table. In the default RPL protocol, the parent does not check the connection between itself and its child, so it starts sending the DIS when the node is transferred and there is disconnection but the proposed protocol continuously checks the connection between itself and its child and it creates new connections when the connection is weak.

(b) To maintain the network's stability, mobile nodes act as tree leaves, and they are not involved in creating paths for sending data to the root. With this assumption, the control messages decrease since no DIO message is sent to the leaf nodes.

(c) The parent node periodically measures the ARSSI value and compares it with the threshold. If the measured value is higher than the threshold value, the parent continues communicating and the mobile node is not informed about the ARSSI value. On the contrary, when the value is less than the threshold, a DIO message is sent to the mobile node, forcing to enter the next phase. Therefore, the parent node carries out two tasks: sending or

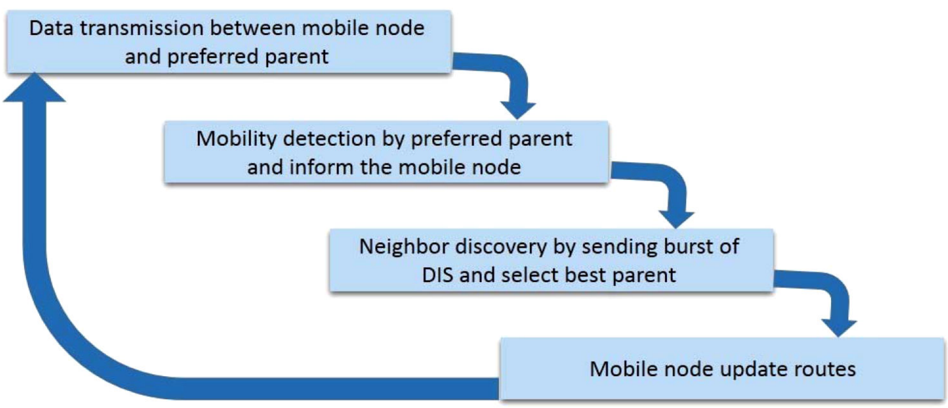

Fig. 2 *Proposed protocol steps*

```
begin
if count of send data == m then
    timer reset
    if receive DIO with flag=1 packet then
        go to Neighbor Discovery and parent selection
    else if  expired timer then
        go to Discovery Neighbor and parent selection
    else
        Continue Data Transmission
    end
end
```

Fig. 3 *Algorithm 1: data transfer phase with movement detection*

```
begin
If receive DIO (flag=1) then
    Broadcast DIS packets
    If count of DIS packets == C then
        Reset timer
        If timer expire then
            For i in the neighbors list do
                If receive DIO (flag=2) then
                    R_i = BestParent(node_i)
                    R_j = BestParent(node_j)
                    Select_Parent = R_i < R_j ? Node_i : Node_j
                end
            Go to update phase
        end
    else
        Continue data transmission
    end
end
BestParent(Node_i)
begin
R = [α(ETX/max_ETX) + β(RSSI/max_RSSI) + ɣ (1-residual_Energy /total_Energy]
return R;
end
```

Fig. 4 *Algorithm 2: neighbour discovery and choosing the best parent*

```
begin
Mobile Node sends DAO to previous parent
If receive DAO ACK then
    Send DAO to new parent
end
```

Fig. 5 *Algorithm 3: update phase*

receiving information to/from the mobile node, and mobility detection.

(d) To preserve consistency with the RPL and enable nodes to communicate with one another, our proposed protocol employs the same control messages used by the RPL but with additional fields to deal with mobility. The RSSI field is included to hold the ARSSI value and the flag field to distinguish the main messages of the RPL from the mobility-related messages.

(e) Mobile node uses DAO to update downward routes.

(f) Mobile node uses ETX, RSSI and the remaining energy for choosing a new parent. To calculate the link quality, we use the following formula:

$$R = \alpha\left(\frac{\text{ETX}}{\text{max ETX}}\right) + \beta\left(\frac{\text{rssi}}{\text{max rssi}}\right) + \gamma\left(1 - \frac{\text{residual energy}}{\text{total energy}}\right) \quad (1)$$

where $\alpha + \beta + \gamma = 1$. So they can be altered according to the priority of the metrics.

ETX: ETX is the number of expected transmissions of a packet necessary for it to be received without error at its destination [30].

RSSI: RSSI is a measurement of the power present in a received signal from the parent.

Residual energy: The energy issue is one of the key factors in the IoT because the sensor node works with the battery. The sensor node is treated as disabling when its energy is exhausted so a black hole is created in the network topology. To overcome this issue, it is critical to take the remaining energy of the node when selecting the parent node. The residual energy is calculated according to the equation below [30]:

$$\text{residual}_{\text{energy}} = \text{total}_{\text{energy}} - \text{depletion}_{\text{energy}} \quad (2)$$

where depletion energy is calculated according to the equation below [30]:

$$\text{depletion}_{\text{energy}} = V \times \left(I_{\text{ap}}T_{\text{ap}} + I_{\text{lp}}T_{\text{lp}} + I_{\text{tx}}T_{\text{tx}} + I_{\text{rx}}T_{\text{rx}}\right) \quad (3)$$

V is the supply voltage, I_{ap}, I_{lp}, I_{tx} and I_{rx} are the current consumptions in milliamperes when the node is in the active, idle, transmit and receive modes. T_{ap}, T_{lp}, T_{tx} and T_{rx} are the periods when the node is in the active, idle, transmit and receive modes [30].

In the next section, the proposed protocol has been discussed in details.

4 Proposed method

The main aim of the proposed protocol is to provide continuous connectivity while the network overhead and energy consumption are reduced. When mobile nodes transmit data with its parent and move away from it, the proposed protocol follows some steps, as illustrated in Fig. 2, to deal with this change in the network. We describe them as follows:

- *Data transfer between the mobile node and parent phase:* Fig. 2 illustrates, in this phase, the mobile node and its preferred parent transmit data. The parent node periodically calculates the ARSSI value after receiving *'m'* data packets. If the measured value exceeds the threshold, the parent continues to transfer data; otherwise, it puts the ARSSI value in the DIO message (flag = 1) and sends it to the mobile node. When the value *'m'* is very small, it imposes some extra processing on the parent node hence increasing energy consumption. Conversely, when *'m'* is large, any loss in the network link can lead to multiple packet losses. Therefore, in this paper, *'m'* is experimentally determined to be 5 [26]. The ARSSI value should not be less than a certain value because it indicates an inappropriate link status, and if the data is sent packet loss will increase. Therefore, when the parent measures the ARSSI, it compares it with the threshold value, and if the ARSSI is higher it will continue to communicate.
- *Mobility detection phase:* The mobile node waits for the message from the parent after sending a specified number of data. When the mobile node receives the DIO (flag = 1) message from the parent, it detects mobility and broadcasts the burst of DIS packets. It then goes to the neighbour discovery and best parent selection phase. The algorithm of data transfer and movement detection phase is described in Algorithm 1 (see Fig. 3).
- *Neighbour discovery and best parent selection:* The mobile node replicates sending the DIS packets several times and then waits to receive DIO (flag = 2) packets from neighbours. It also sets a timer and receives DIO packets within a specified time and receives no packets after the timer expires. The mobile node begins to compare RSSI, ETX and energy to select the best parent after receiving the DIO. Algorithm 2 (Fig. 4) shows the procedure steps of the mobile node to discover its neighbours and choose the best parent.
- *Routes update phase:* When the mobile node chooses a new parent node, the routes need to be updated in the routeing table, so it sends the DAO packet to update the backward route. This is described in Algorithm 3 (Fig. 5).

Fig. 6 shows the timing diagram for the proposed protocol.

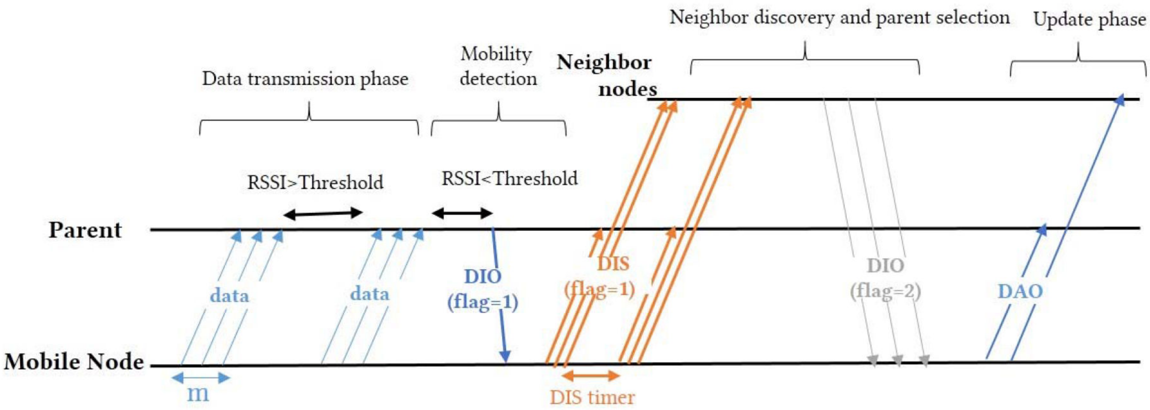

Fig. 6 *Timing diagram for proposing a new method*

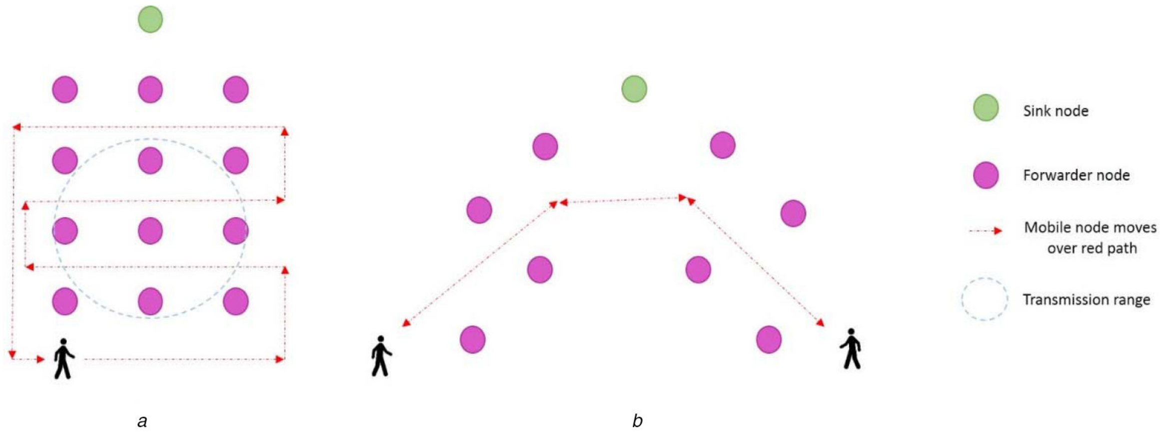

Fig. 7 *Two simulated scenarios*
(a) The first scenario: mobile node traverses the path colored in red and returns to its initial point, ***(b)*** The second scenario: two mobile nodes, both moving toward the root at first, then moving away from it

Table 2 Cooja simulation parameters

Parameter	Values
operating system	Contiki-2.7
simulator	Cooja
radio medium	Unit Disk Graph Medium
mote type	Tmote-Sky
transmission range	50 m
speed	(0.5, 1, 2) m/s
simulation area	[100 m, 175 m], [200 m, 125 m]
simulation time	300 s, 250 s

The mobile node transfers data with its parent node. The parent node measures the RSSI value when the mobile node sends *'m'* data packets toward its parent. When the measured value is higher than the threshold, the communications go on. Otherwise, it assumes the occurrence of mobility and informs the mobile node by sending a DIO message. The mobile node becomes aware of its mobility on receiving the DIO message, hence entering the next phase. In this phase, the mobile node broadcasts DIS messages. On receiving the DIS messages, the neighbouring nodes respond through DIO messages. The mobile node then compares the received DIO packets to select the best parent. The parent node with mobility detection capability reduces the communicated control messages and lowers the interference of mobile nodes. After mobility detection by the parent, the mobile node is forced to select a new parent node.

5 Performance evaluation and results

We select Cooja for simulations created by Instant Contiki 2.7 [32] to evaluate the proposed protocol. The main reasons for selecting Contiki Cooja are: (i) the availability of an RPL/6LoWPAN implementation, that is, reasonably mature and widely used, (ii) the availability of a mobility plugin in Cooja that enables to evaluate protocol in a repeatable environment and (iii) Instant Contiki provides transfer control protocol//Internet protocol stack and it is an open-source and lightweight system designed for resource-constrained embedded devices to develop complex networks. We have designed two scenarios, which we have simulated accordingly as shown in Fig. 7.

In the first scenario, Scenario (a) in Fig. 7, the mobile node traverses the path coloured in red and returns to its initial point. In the second scenario, Scenario (b) in Fig. 7, two mobile nodes are shown, both moving toward the root at first, then moving away from it.

The experimental parameters for both scenarios are shown in Table 2.

We compare the performance of our proposed protocol with the standard RPL [17] and mRPL [26].

Different performance metrics were considered to evaluate the performance of the proposed protocol:

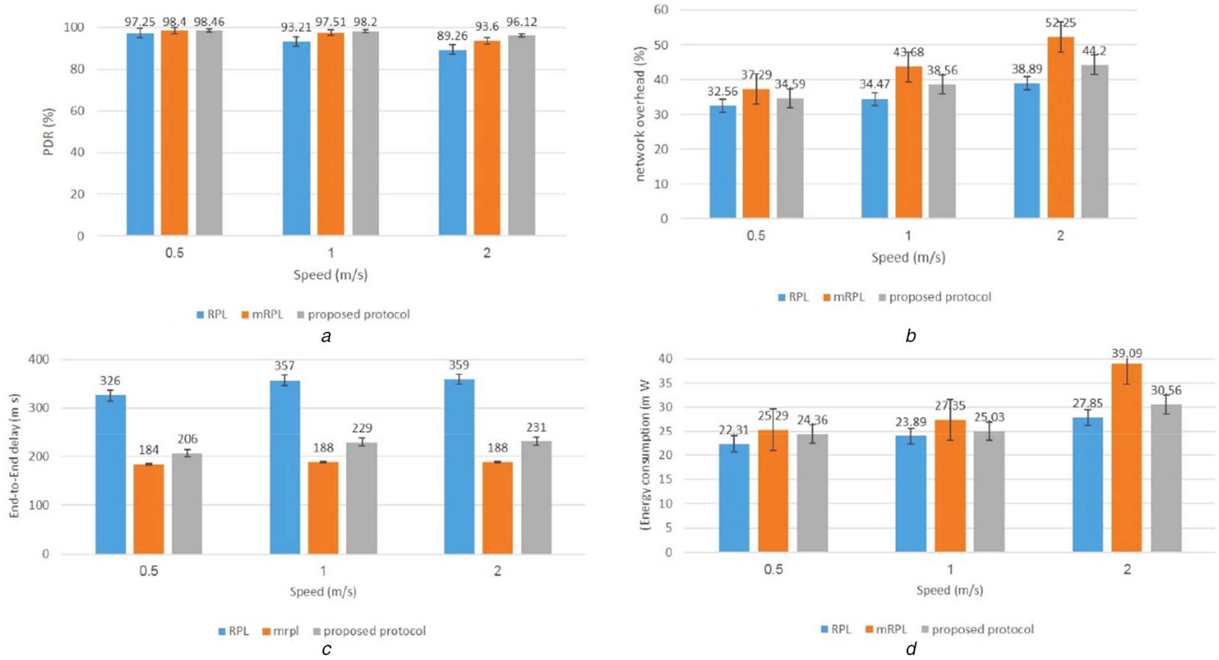

Fig. 8 *Simulation results of the first scenario*
(a) PDR, ***(b)*** Network overhead, ***(c)*** End-to-end delay, ***(d)*** Energy consumption according to mobile node speed for the first scenario

- *Packet delivery ratio (PDR):* The ratio of the total number of packets delivered successfully to the sink node to the total number of packets sent by the forwarder node [30].
- *Network overhead:* The amount of additional packets such as DIO, DIS and DAO ICMPv6 packets divided by the sum of additional packets plus data messages [5].
- *End-to-end delay:* The average time it takes to transfer the whole message across a network from the sender node to sink node successfully.
- *Energy consumption:* The nodes in the IoT are usually battery powered, and hence energy is a scarce resource [33]. It is calculated as follows (4) and (5) [34].
- *End-to-end throughput* is expressed by (6) (see (4))

$$\text{power (mW)} = \text{energy(mJ)}/(\text{time(s)}) \tag{5}$$

$$\text{end} - \text{to} - \text{end throughput} = \frac{(\text{total\_recieve\_date} \times \text{ data\_length } \times 8)}{\text{simulation\_time}} \tag{6}$$

5.1 Simulation results of the first scenario

Fig. 8 shows a comparison between the proposed protocol and RPL and mRPL by changing the speed of the mobile node to 0.5, 1 and 2 m/s. In Fig. 8*a*, we can see that the PDR decreases when the mobile node's speed increases. RPL responds very slowly to topology changes so the PDR of the RPL reaches 89.26% when the mobile node's speed is 2 m/s. The PDR of the proposed protocol is 96.12%, which is a sign of better functionality by avoiding packet loss and maintaining seamless connectivity with the RPL but it is not substantially different from the mRPL protocol, since the proposed protocol and mRPL are active and react quickly to topology changes.

Control messages such as DIO, DIS and DAO are essential for establishing, sustaining and rebuilding routes in the network. We have defined some additional control packets to enable mobility management. As shown in Fig. 8*b*, the number of control messages transmitted in the proposed protocol is higher than of the RPL but less than the mRPL. Fig. 8*c* shows the end-to-end delay according to the mobile node's speed. As said before, our proposed protocol and mRPL are proactive protocol to deal with mobility; therefore, the end-to-end delay in the proposed protocol and mRPL is less than the RPL. Fig. 8*d* shows that the power consumption of the network increases when the mobile node's speed increases. The mRPL consumed more power because the nodes process the burst control messages for several times by keeping the radio on.

Fig. 9 depicts the comparison of the end-to-end throughput of RPL, mRPL and the proposed protocol according to the speed of the mobile node. It is clear the throughput decreases when the mobile node's speed increases. The end-to-end throughput of the proposed protocol is greater than the RPL and mRPL.

5.2 Simulation results of the second scenario

Fig. 10 shows the simulation results obtained from Fig. 7*b*, when there are two mobile nodes in the network.

The RPL reacts very slowly to topology changes, and when two mobile nodes are present in the network topology changes take place more often, so the packets sent to the destination are lost and the PDR decreases (about 53%). Consequently, the delay increases because of rebuilding a new route to the destination.

The proposed protocol reacts quickly to topology changes, so it shows better performance when there are two mobile nodes in the network by increasing reliability. As shown in Fig. 10*a*, the PDR for the proposed protocol is above 80%, proving the preservation of network connections compared with the mRPL and RPL standards. Fig. 10*c* shows the end-to-end delay according to the speed of mobile nodes. It shows that proposed protocol packets reach the destination faster than the RPL and mRPL. The energy consumption in the proposed protocol is higher than that of the RPL protocol because the nodes perform more processing operations to preserve their connections. However, when compared with mRPL protocol, our protocol has less energy consumption.

As shown in Fig. 11, we conclude that the proposed protocol improves the end-to-end throughput of the network under mobility.

Under mobility, the link quality is weak and it leads to the frequent selection of parent. The proposed protocol selects the stable parent and stable link for mobile nodes to send packet successfully to the root. Therefore, the proposed protocol has a higher throughput.

$$\text{energy (mJ)} = \frac{(\text{transmit} \times 19.5(\text{mA}) + \text{listen} \times 21.8(\text{mA}) + \text{CPU} \times 1.8(\text{mA}) + \text{LPM} \times 0.0545(\text{mA})) \times 3\,\text{V}}{4096 \times 8} \tag{4}$$

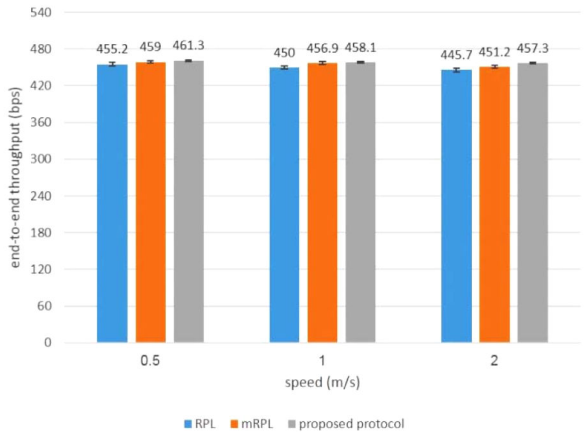

Fig. 9 *End-to-end throughput to mobile node speed for the first scenario*

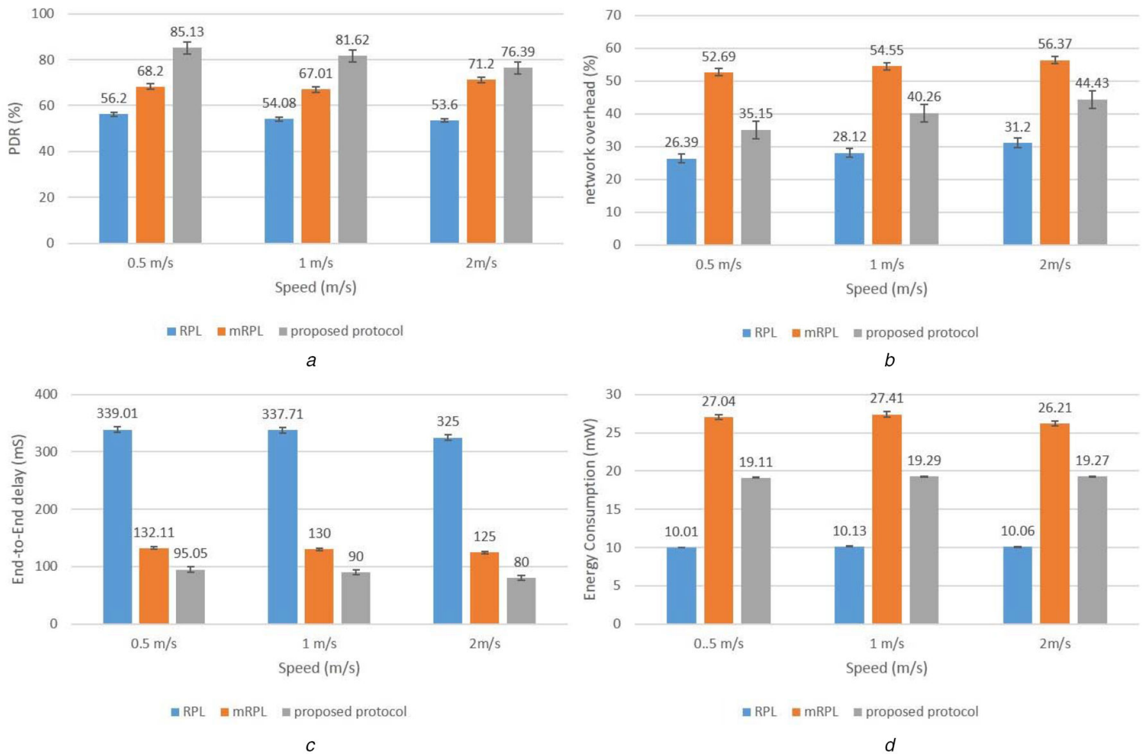

Fig. 10 *Simulation results of the second scenario*
(a) PDR, ***(b)*** Network overhead, ***(c)*** End-to-end delay, ***(d)*** Energy consumption according to mobile nodes speed for the second scenario

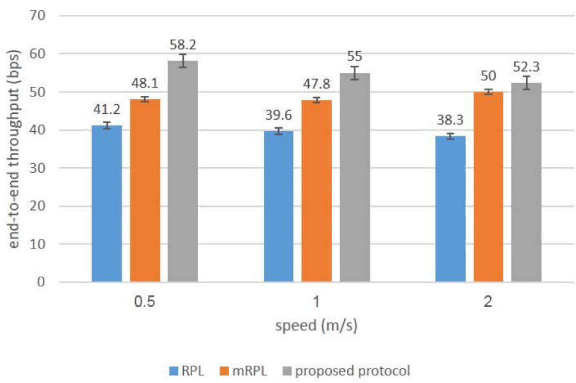

Fig. 11 *End-to-end throughput to mobile node speed for the second scenario*

6 Conclusion

In this paper, we have proposed a new method to improve the RPL protocol under mobility condition. Our approach has the aim to provide continuous connectivity for mobile nodes and select the best parent. The proposition is based on a proactive protocol to predict new attachment of mobile nodes based on the ETX, RSSI and residual energy, before their disconnection. The parent uses the additional field in control packets of RPL to detect mobility and informs mobile node to reduce the involvement of it to keep its energy and fewer control messages are communicated. Under mobility, the link quality is weak and it leads to the frequent selection of parent. A memory is defined to keep a set of the preferred parent and the mobile node gets help from this list to select stable parents to avoid the frequent selection of parent. We analysed the proposed protocol with the RPL and mRPL based on PDR, throughput, network overhead, end-to-end delay and energy consumption. The results showed the effectiveness of the proposed method. To conclude, the proposed algorithms proved to be an efficient algorithm in terms of energy efficiency when compared with the previous mRPL algorithms as it is most necessary for energy-constrained devices to survive long in the network. Our proposed protocol complies with the RPL protocol, and it can be easily implemented in the real world. However, in real-world environments, many factors influence the value of RSSI and are susceptible to interference, which can affect the result and performance of the proposed protocol. To resolve this problem, we need to look for a better parameter to replace the RSSI so that it can have the best performance about the restrictions of nodes.

7 References

[1] Al-Fuqaha, A., Guizani, M., Mohammadi, M.*, et al.*: 'Internet of things: a survey on enabling technologies, protocols, and applications', *IEEE Commun. Surv. Tutor.*, 2015, **17**, (4), pp. 2347–2376

[2] Atzori, L., Iera, A., Morabito, G.: 'The Internet of things: a survey', *Comput. Netw.*, 2010, **54**, (15), pp. 2787–2805

[3] Kritsis, K., Papadopoulos, G., Gallais, A.*, et al.*: 'A tutorial on performance evaluation and validation methodology for low-power and lossy networks', *IEEE Commun. Surv. Tutor.*, 2018, **20**, (3), pp. 1799–1825

[4] Bouaziz, M., Rachedi, A.: 'A survey on mobility management protocols in wireless sensor networks based on 6LoWPAN technology', *Comput. Commun.*, 2016, **74**, pp. 3–15

[5] Hoghooghi, S., Nasr Esfahani, R.: 'Mobility-aware parent selection for routing protocol in wireless sensor networks using RPL'. 2019 Fifth Int. Conf. Web Research (ICWR), Tehran, Iran, April 2019, pp. 24–25

[6] Ghaleb, B., Al-Dubai, A., Ekonomou, E.*, et al.*: 'A survey of limitations and enhancements of the IPv6 routing protocol for low-power and lossy networks: a focus on core operations', *IEEE Commun. Surv. Tutor.*, 2019, **21**, (2), pp. 1607–1635

[7] Akpakwu, G., Silva, B.J., Hancke, G.*, et al.*: 'A survey on 5G networks for the Internet of things: communication technologies and challenges', *IEEE Access*, 2018, **6**, pp. 3619–3647

[8] Hanes, D., Salgueiro, G., Grossetete, P.*, et al.*: '*IoT fundamentals networking technologies, protocols, and use cases for the Internet of things*' (Cisco Press, USA, 2017)

[9] Oliveira, A., Vazao, T.: 'Low-power and lossy networks under mobility: a survey', *Elsevier – Comput. Netw.*, 2016, **107**, pp. 339–352

[10] Hasbollah, A., Ariffin, S., Hamini, M.: 'Performance analysis for 6loWPAN IEEE 802.15.4 with IPv6 network'. TENCON 2009–2009 IEEE Region 10 Conf., Singapore, November 2009, pp. 23–26

[11] Kim, H.-S., Ko, J., Culler, D.E.*, et al.*: 'Challenging the IPv6 routing protocol for low-power and lossy networks (RPL): a survey', *IEEE Commun. Surv. Tutor.*, 2017, **19**, (4), pp. 2502–2525

[12] Santos, B.P., Goussevskaia, O., Vieira, L.F.M.*, et al.*: 'Mobile matrix: routing under mobility in IoT, IoMT, and social IoT', *Ad Hoc Netw.*, 2018, **78**, pp. 84–98

[13] Tokognon, C.A., Gao, B., Yun Tian, G.*, et al.*: 'Structural health monitoring framework based on Internet of things: a survey', *IEEE Internet Things J.*, 2017, **4**, (3), pp. 619–635

[14] Sneha, K., Prasad, B.G.: 'An efficient hand-off optimization-based RPL routing protocol for optimal route selection in mobility enabled LLNs'. Int. Conf. Global Trends in Signal Processing, Information Computing and Communication, Jalgaon, India, 2016

[15] OlivierKamgueu, P., Nataf, E., Djotio Ndie, T.: 'Survey on RPL enhancements: a focus on topology, security and mobility', *Elsevier – Comput. Commun.*, 2018, **120**, pp. 10–21

[16] Gaddour, O., Koubâa, A.: 'RPL in a nutshell: a survey', *Elsevier*, 2012, **56**, (14), pp. 3163–3178

[17] Winter, T., Thubert, P., Hui, J.*, et al.*: 'RPL: IPv6 routing protocol for low-power and lossy networks'. Available at https://tools.ietf.org/html/rfc6550, accessed 01 August 2019

[18] Lamaazi, H., Benamar, N., Jara, A.: 'RPL-based networks in static and mobile environment: a performance assessment analysis', *J. King Saud Univ., Comput. Inf. Sci.*, 2017, **30**, (3), pp. 320–333

[19] Parsaei, M.R., Parnian, A.R., Rostami, M.*, et al.*: 'RPL load balancing in Internet of things', *IIUM Eng. J.*, 2017, **18**, (2), pp. 137–150

[20] Levis, P., Patel, N., Culler, D.*, et al.*: 'Trickle: a self-regulating algorithm for code propagation and maintenance in wireless sensor networks'. First Conf. Symp. Networked Systems Design and Implementation, San Francisco, CA, United States, 2004

[21] Bahramlou, A., Javidan, R.: 'Adaptive timing model for improving routing and data aggregation in Internet of things networks using RPL', *IET Netw.*, 2018, **7**, (5), pp. 306–312

[22] El Korbi, I., Ben Brahim, M., Adjih, C.*, et al.*: 'Mobility-enhanced RPL for wireless sensor networks'. 2012 Third Int. Conf. Network of the Future (NOF), Gammarth, Tunisia, February 2013

[23] Somaa, F., Korbi, I.E., Adjih, C.*, et al.*: 'A modified RPL for wireless sensor networks with Bayesian inference mobility prediction'. 2016 Int. Wireless Communications and Mobile Computing Conf. (IWCMC), Paphos, Cyprus, Sep 2016, pp. 5–9

[24] Barcelo, M., Correa, A., Lopez Vicario, J.*, et al.*: 'Addressing mobility in RPL with position assisted metrics', *IEEE Sens. J.*, 2015, **16**, (7), pp. 2151–2161

[25] Bouaziz, M., Rachedi, A., Belghith, A.: 'EKF-MRPL: advanced mobility support routing protocol for Internet of mobile things: movement prediction approach', *Elsevier – Future Gener. Comput. Syst.*, 2017, **93**, pp. 822–832

[26] Fotouhi, H., Moreira, D., Alves, M.: 'mRPL: boosting mobility in the Internet of things', *Elsevier – Ad hoc Netw.*, 2015, **26**, pp. 17–35

[27] Fotouhi, H., Moreira, D., Alves, M.*, et al.*: 'mRPL+: a mobility management framework in RPL/6LoWPAN', *Elsevier: Comput. Commun.*, 2017, **104**, pp. 34–54

[28] Anand, M.C.R., Tahiliani, M.P.: 'mRPL++: smarter-HOP for optimizing mobility in RPL'. 2016 IEEE Region 10 Symp. (TENSYMP), Bali, Indonesia, July 2016

[29] Bouaziz, M., Rachedi, A., Belghith, A.: 'EC-MRPL: an energy-efficient and mobility support routing protocol for Internet of Mobile things'. 14th IEEE Annual Consumer Communications & Networking Conf. (CCNC), Las Vegas, NV, USA, July 2017

[30] Sanshi, S., Jaidhar, C.D.: 'Enhanced mobility routing protocol for wireless sensor network', *Wirel. Netw.*, 2018, **56**, pp. 615–684

[31] Murali, S., Jamalipour, A.: 'Mobility-aware energy-efficient parent selection algorithm for low-power and lossy networks', *IEEE Internet Things J.*, 2019, **6**, (2), pp. 2593–2601

[32] 'Contiki: the open-source OS for the Internet of things'. Available at http://www.contiki-os.org/, accessed 14 December 2018

[33] Chang, L., Lee, T., Chen, S.*, et al.*: 'Energy-efficient oriented routing algorithm in wireless sensor networks'. 2013 IEEE Int. Conf. Systems, Man, and Cybernetics, Manchester, UK, October 2013, pp. 13–16

[34] Han, S.: 'Internet of things technology'. Available at http://thingschat.blogspot.com/2015/04/contiki-os-using-powertrace-and.html, accessed 15 October 2018