

Mobility Aware RPL (MARPL): Providing Mobility Support for RPL Protocol

Vinicius de Figueiredo Marques¹, Janine Kniess¹

¹Programa de Pós-Graduação em Computação Aplicada –
Universidade do Estado de Santa Catarina (UDESC)

vinicius.marques@edu.udesc.br, janine@udesc.br

Abstract. *Low Power and Lossy Network (LLN) is a common type of wireless network in IoT applications. LLN communication patterns usually require an efficient routing protocol. The IPv6 Routing Protocol for Low-Power and Lossy Network (RPL) is considered to be the standard routing protocol for LLNs. However, RPL was developed for static networks and node mobility decreases RPL overall performance. These are the aims of the Mobility Aware RPL (MARPL), presented in this paper. MARPL provides a mobility management based on neighbor variability. MARPL is composed by mechanisms for mobility and link disconnection detection, and DIO and DIS transmission adjustment. Performance evaluation obtained through simulations confirms the effectiveness of MARPL regarding packet delivery rate, overhead and residual energy when compared to RPL, mRPL and D-RPL.*

1. Introduction

Internet of Things (IoT) is a concept that aims to include wireless connectivity for day-to-day devices in order to enable many forms of smart applications. Such as smart cities, farming and transportation. The data collect by IoT applications could potentially enhance decision making [Oppitz and Tomsu 2018]. An IoT device is commonly composed by a sensing and wireless communication component. The sensing component is responsible for data collection. The communication component might differs in terms of radio range and transmission power [Sethi and Sarangi 2017]. The main characteristics of radio technologies incorporated in IoT devices are short transmission range and low power consumption, since many IoT devices are battery powered [Sethi and Sarangi 2017].

Low Power and Lossy Network (LLN) is a common type of network formed by IoT devices [Paul and Saraswathi 2017]. LLN devices operate with low range radio technology, such as the IEEE 802.15.4, in order to be energy efficient [Gubbi et al. 2013]. The low range radio technology demands a hop-by-hop communication model. Thus, it requires an efficient routing protocol. IPv6 Routing Protocol for Low-Power and Lossy Networks (RPL) is a standard routing protocol for LLN designed by the IETF group [RFC 6550 2012]. The RPL design was motivated by the lack of a proper routing protocol for LLNs. RPL is compatible with IPv6 through the IPv6 over Low-Power Wireless Personal Area Networks (6LoWPAN) [RFC 6550 2012]. Therefore, RPL enabled devices can be addressed on the internet.

IoT applications, such as smart health and smart city, may have both static and mobile devices in the network [Iova et al. 2016]. Accordingly, the complexity of network management is increased. Mobility requires a resilient routing protocol to handle frequent

topology changes [Zhao et al. 2017]. A routing protocol for LLN should have efficient mechanisms for a rapid mobility detection, so it may diminish packet loss. The RPL routing protocol was initially designed for static LLN topologies. Therefore it faces some issues in mobile LLNs such as low packet delivery rate. The RPL limitations in mobile LLNs are due to the lack of mechanisms for mobility management. Nevertheless, RPL can be enhanced to support node mobility [Gara et al. 2016].

This paper presents the Mobility Aware RPL (MARPL). MARPL is a mobility support for the RPL routing protocol based on neighbor variability. MARPL detects node mobility and disconnection through neighborhood monitoring. MARPL utilizes a new metric named Neighbor Variability, representing node mobility, for mobility detection, route selection and control packet dissemination adjustments. MARPL has the following main contributions:

- A metric related to node mobility called Neighbor Variability;
- A mechanism for mobility detection through the proposed metric, Neighbor Variability;
- A mechanism to detect route disconnection;
- A control packet transmission adjustment mechanism.
- A route selection mechanism that takes into account node mobility.

The road map of this paper is organized as follows. Section 2 details RPL and the issues faced by it in mobile LLNs. Section 3 presents RPL mobility support proposals of the literature. Section 4 depicts MARPL, proposed by the authors of this paper. MARPL performance analysis results are shown at Section 5. Finally, Section 6 presents the conclusions and future work.

2. Relevant Aspects and Mobility Issues of RPL Protocol

RPL was originally developed for static networks. However, mobility support is a requirement for a plenty of IoT application such as smart health and cities [Iova et al. 2016]. RPL faces performance challenges when there are mobile nodes in the topology such as: packet loss. Nevertheless, it's possible to enhance RPL for mobility support [Gara et al. 2016].

Devices executing RPL can perform three types of roles in a LLN: a root, a router or a host node [RFC 6550 2012]. A root node receives all data collected inside the LLN. A router and host node are responsible for data collection, but only a router node can forward packets towards the root node (i.e. a root node in the LLN). The RPL topology is based on a Directed Acyclic Graph (DAG). DAG is a graph with no cycles and all its edges are oriented toward one or more root nodes [RFC 6550 2012].

A Destination-Oriented DAG (DODAG) is a DAG rooted at a single root node. RPL has three control packet: DODAG Information Object (DIO), Destination Advertisement Object (DAO) and a DODAG Information Solicitation (DIS). DIO carries metrics and restrictions to route selection. Therefore, is used by RPL to construct and maintain a DODAG. When a node joins a DODAG, it does so by selecting a neighbor node with the best route towards the root node. In RPL, route quality is expressed by its rank value. A node rank is related to its distance to the DODAG root and it's calculated based on DIO metrics. Therefore, the best route is the route with smaller rank [RFC 6550 2012]. After joining a DODAG, a node sends a DAO message to the selected neighbor with best rank,

referred as preferred parent in the RPL terminology. DIS is utilized for a node to request DIO messages from its neighbors to assess the possible routes towards the root node.

RPL has a proactive route discovery and topology construction through periodic DIO transmissions. The construction of a RPL DODAG is initiated by DODAG root nodes [RFC 6550 2012]. DIO control packet dissemination in RPL is controlled by the Trickle algorithm [RFC 6206 2011].

Trickle is an algorithm for DIO dissemination in RPL in a simple, robust and scalable manner [RFC 6206 2011]. Trickle has two mechanisms to achieve it: (i) when an inconsistency in the network is detected (e.g. loop), Trickle increases the signaling rate of messages as a way to solve it. By contrast, Trickle exponentially decreases control message transmissions when the network is stable to save node energy; (ii) Trickle suppress DIO transmissions when its content is considered trivial [RFC 6206 2011].

In RPL, DIO propagation by the Trickle algorithm is configured through three parameters: I_{min} , I_{max} and k . I_{min} specifies the minimum period of time the suppression of DIO transmission can last. I_{max} regulates the maximum period of DIO suppression. k is the redundancy factor that is used to verify if a message can be transmitted at a specific time [RFC 6206 2011]. Trickle's transmission suppression is adjusted by the variable I . I value is selected randomly in the closed set $[I_{min}, I_{max}]$ and grows exponentially until it reaches I_{max} [RFC 6206 2011]. In MARPL, the Trickle adjustment is applied in the variable I , reducing it by half at any moment a mobile node is identified in the neighbor.

The suppression period regulated by the Trickle algorithm influences the performance of the RPL routing protocol in mobile LLNs. When the suppression period is too long, nodes may not be able to detect a preferred parent disconnection efficiently since there are less control messages being propagated in the network [Gara et al. 2016]. Another issue with long periods of DIO suppression is that the DODAG will take longer to update.

It was found in the analysis of related work that most of the issues the RPL routing protocol faces when dealing with mobility are: the lack of efficient mechanisms for DIO and DIS control messages transmission for a rapid preferred parent unavailability detection; a better mechanism for mobility detection or prevention; and preferred parent selection that takes node mobility into consideration.

3. Related Work

As stated at Section 2, the main issues regarding RPL faces when dealing with mobility are the lack of a mobility detection mechanism and preferred parent unavailability identification.

[Cobarzan et al. 2014] argued that when a node disconnects from its preferred parent because of mobility, it might wait for too long to receive a DIO message. Therefore, increasing disconnection time and packet loss. In RPL, a node can update its preferred parent by receiving a DIO message from another candidate parent with lower rank [RFC 6550 2012]. [Cobarzan et al. 2014] proposes a reverse Trickle algorithm that the DIO suppression time starts short and increases over time. The main rationale behind this idea is that mobile node connects to a preferred parent and it remains connected for a considerable amount of time. A drawback of [Cobarzan et al. 2014] proposal is that it de-

depends on the existence of static nodes. In contrast, MARPL makes no distinction between mobiles and static nodes, since mobile nodes can stay static for a period of time.

[Fotouhi et al. 2015] proposed a RPL extension named Mobile Compliant RPL (mRPL). mRPL utilizes a handoff mechanism called SmartHop [Fotouhi et al. 2012]. The authors showed that mRPL enables the exchange of preferred parent efficiently with low overhead and power consumption. mRPL is a proactive preferred parent unavailability prediction mechanism. Therefore, it enables frequent control message exchange in order to assess if the preferred parent is still connected. Nevertheless, it is expected that mRPL increases the network overhead in order to perform a fast disconnection identification. [Ko and Chang 2015] argues that the utilization of proactive handoff mechanisms in LLN (e.g. such as mRPL) is too aggressive because it generally performs detailed link analysis to detect preferred parent unavailability.

[Ko and Chang 2015] proposes MoRoRo, a mobility support mechanism for RPL. With MoRoRo, the node mobility can be detected based on packet loss rate by the increase of control message to assess link quality. [Gara et al. 2016] argues that packet loss is usual in LLN for its lossy links. Therefore, mobility detection through packet loss may not be efficient in some scenarios.

[Bouaziz et al. 2017] proposed an enhancement for mRPL through a mobility prediction mechanism. Such mechanism seeks to solve two issues: high Received Signal Strength Indication (RSSI) interference from the environment and increased energy consumption by the overhead caused by a proactive node disconnection approach, such as mRPL. [Bouaziz et al. 2017] mechanism is based on the following assumption: static nodes are required in the topology and their positions are known by the mobile nodes before the network starts to execute. This assumption may not be realistic for every LLN application scenario. Besides that, the authors' proposal has another drawback related to the processing power required by the static nodes in order to process the mobility predicting model.

[Kharrufa et al. 2017] proposed a reverse-Trickle timer based on the RSSI called Dynamic RPL (D-RPL). Every node executing D-RPL measures RSSI values from the last two packet messages from every single local-link neighbor node. It could be a control or data packet. Upon reception of a new packet, a node measures the RSSI and compares it to the last measurement from the same neighbor node. If the new RSSI plus a redundancy constant K_{RSSI} is lesser than the last RSSI, the reverse-Trickle timer is executed. The node also sends a local-link multicast DIS to all its neighbors. Otherwise, the default Trickle is executed. As D-RPL, MARPL also utilizes a reversed Trickle approach. Nevertheless, MARPL considers the proposed metric Neighbor Variability to do so.

Section 4 presents more details about MARPL, a proposal of mobility support for RPL. D-RPL and mRPL were implemented and compared to MARPL. These protocols were chosen for comparison because they are similar to MARPL in some terms. Details about the obtained results are shown at Section 5.

4. Mobility Aware RPL

This section details the Mobility Aware RPL (MARPL) protocol. The design of MARPL encompasses a mobility detection and a preferred parent unavailability detection mechanisms. Also, an enhancement to the RPL trickle timer.

As mentioned before, many IoT application domains requires a LLN with both static and mobile nodes. Thus, the routing protocol should be resilient enough to handle constant topology changes caused by node mobility. MARPL is compatible with the canonical RPL. Therefore, both MARPL and RPL can coexist in the same LLN. MARPL is composed by three mechanisms: (i) **mobility detection through the metric Neighbor Variability (γ)**; (ii) **preferred parent unavailability detection** and; (iii) **DIO and DIS transmission adjustment**.

MARPL utilizes a timer called $T_{monitoring}$ for mobility detection and γ calculation. $T_{monitoring}$ is calculated by the Equation 1. $T_{monitoring}$ operates based on the *Data Packet Period* (DPP) (i.e. the frequency of measurements made by the sensor). It varies based on application characteristics.

$$T_{monitoring} = DPP \quad (1)$$

Algorithm 1 MARPL Protocol

```

1: procedure MOBILITY_MONITORING
2:   start  $T_{monitoring}$ 
3:   if received a packet then ▷ data or control packet
4:     update sender IP,  $\gamma$  and RSSI in the neighbor table ▷ if control packet
5:     if packet is a DIS or DAO from child node then
6:       TRICKLE_ADJUSTMENT
7:     if packet is DIO then
8:        $neighbor_{new\_rank} \leftarrow neighbor_{old\_rank} + \gamma$ 
9:     if  $T_{monitoring}$  expires then
10:      if  $max\{var\{\Delta p_i\}_{i=1}^y\} > K_\gamma$  then ▷ if there's a greater variance then  $K_\gamma$ 
11:         $K_\gamma = max\{var\{\Delta p_i\}_{i=1}^y\}$  ▷ update  $K_\gamma$  with the greatest  $var$ 
12:         $\gamma \leftarrow var\{\Delta p_i\}_{i=1}^y / K_\gamma$ 
13:      if received no packet from the preferred parent then ▷ data or control packet
14:        if  $\gamma > 0$  then
15:          send DIS to neighbors
16:        restart  $T_{monitoring}$ 
17: procedure TRICKLE_ADJUSTMENT
18:    $\gamma_{packet} \leftarrow \gamma$  from control packet ▷ DIO, DIS or DAO
19:   if  $\gamma_{packet} > \gamma$  then
20:      $I \leftarrow I/2$ 
21:     if  $I < I_{min}$  then
22:        $I \leftarrow I_{min}$ 

```

Algorithm 1 illustrates MARPL. When a node enters a DODAG, MARPL starts the $T_{monitoring}$ timer (line 2) in order to monitor the node mobility. At every data or control packet reception (line 3), MARPL updates its neighbor table with the packet sender IP, the proposed metric Neighbor Variability (γ) and the Received Signal Strength Indication (RSSI) (line 4). If the received packet is a DIS or DAO control message, MARPL analyses if it's necessary to adjust the Trickle timer (line 6). If the received packet is a DIO message, MARPL updates the neighbor's rank with its γ (line 8) by the Equation 2.

$$neighbor_{new_rank} = neighbor_{old_rank} + \gamma \quad (2)$$

In RPL, a preferred parent is selected by its rank value. The rank value is related by the distance of the preferred parent candidate to the root node. A rank value is calculated by a RPL objective function. MARPL updates the rank of a candidate parent with its γ value since γ is a metric related to the node mobility. The metric γ is updated by Equation 3. Thus, it's expected that by using γ in the preferred parent selection, static nodes will have greater probability to be selected. Equation 2 shows how the rank value is updated. MARPL utilizes two weight parameters: α for the rank calculated by the RPL objective function and β for the Neighbor Variability metric.

The metric γ is derived by the variance of all the positive RSSI differences (i.e. $\Delta p_i \mid \Delta p_i > 0$) from every neighbor node $i \mid i \in [1, y]$, y being all neighbors with two consecutive RSSI measurements, over a threshold K_γ . The RSSI (p) difference for every neighbor (i) is calculated by Equation 4. K_γ is the maximum variance ever calculated by the node during its execution as depicted in Equation 5 (Algorithm 1, lines 10 and 11). Only positive Δp_i is considered in this calculation because it indicates that two nodes are approaching to each other.

The first step for Trickle's adjustment (Algorithm 1 line 17) is to read metric γ from the control packet (i.e. identified by γ_{packet} at line 18). If γ_{packet} is greater than the node's γ (line 19), the Trickle's variable I is reduced by half (line 20). It's important to make sure that $I \geq I_{min}$ (line 21 and 22), since the following is a requirement for Trickle to work: $I \in [I_{min}, I_{max}]$.

The MARPL Trickle adjustment mechanism is inspired by [Kharrufa et al. 2017] and is executed in order to temporary increase DIO transmissions. It's expected that such increase might improve MARPL overall performance in terms of packet delivery rate.

At every $T_{monitoring}$ expiration (Algorithm 1, line 9), MARPL updates the proposed metric Neighbor Variability (γ) (line 12). Using γ , a sensor node can identify its mobility. $T_{monitoring}$ execution time is adjusted by the frequency of sensor data generation. Consequently, it's expected that $T_{monitoring}$ execution time is sufficient to enable a node to received packets from all of its neighbors.

$$\gamma = \frac{var\{\mid \Delta p_i \mid\}_{i=1}^y \mid y > 0, \Delta p_i > 0, K_\gamma > 0}{K_\gamma} \quad (3)$$

$$\Delta p_i = p_{i-1} - p_i \quad (4)$$

$$K_\gamma = max\{var\{\Delta p_i\}_{i=1}^y\} \quad (5)$$

For a better understanding of the rationale behind metric γ calculation, consider Figure 1. Figure 1 depicts an example of γ calculation. There are three different $T_{monitoring}$ periods: Figure 1.a, 1.b and 1.c. As can be seen in Figure 1.a, the mobile node **A** initially has three neighbors. Node **A**'s γ value is set to 0 since there's no entries in its neighbor table with two consecutive RSSI measurements (p_{i-1} and p_i). Since there hasn't

been calculated any variance yet, K_γ has no value. In Figure 1.b, node **A** moved and there are RSSI differences of neighbor **C** and **D**, besides the two new neighbors **E** and **F**. The differences of neighbors **C** and **D** are positive (50 and 10 respectively), therefore γ can be calculated. Since it's the first time the variance (var) is calculated, K_γ will be set with the initial variance value, 400. In Figure 1.b, the variance and K_γ have the same, 400. Thus, γ is updated to 1. In Figure 1.c, node **A** moved again. This time, it has four neighbors with RSSI difference. Nevertheless, only two of them (**E** and **F**) has positive difference (50 and 15 respectively). At Figure 1.c the variance is 306. Since $306 < 400$, K_γ is not updated and γ is updated to 0.76. It's possible to assess that at Figure 1.c, node **A** moved less then at Figure 1.b.

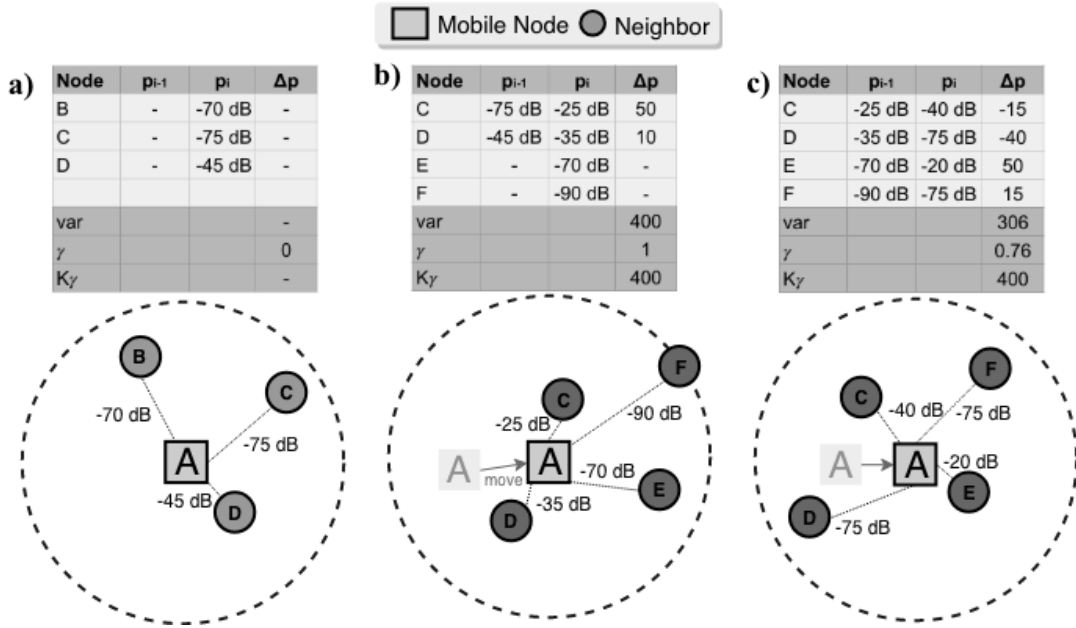


Figure 1. MARPL γ calculation example.

Hand-off detection is critical in topologies with mobile nodes since disconnections will be frequent. RPL does not specify any preferred parent unavailability mechanism [RFC 6550 2012]. The RPL specification suggests the use of an external mechanism for this task [RFC 6550 2012]. Hence, if no packet from the preferred parent was received after $T_{monitoring}$ expiration (Algorithm 1, line 13) and node's $\gamma > 0$ (line 14), MARPL sends a DIS message to all the neighbor sensor nodes (line 15) to assess information about the available candidate parents. After that, $T_{monitoring}$ is started again (line 16).

4.1. RPL and MARPL Comparison

Figure 2.a illustrates the RPL default operation in mobile LLNs. There are four nodes in the example: one root (**A**), two static (**B** and **D**) and one mobile (**C**). Assuming that mobile node **C** has node **B** as its preferred parent, node **C** still inside the radio range of node **B** in the periods of time **I**, **II** and **III**. Assuming that node **C** moves at moment **III** and disconnects from node **B** at moment **IV**, all data packet sent by node **C** throughout period **IV** is lost. This is because RPL doesn't have any mobility management mechanism. Therefore, node **C** will send packets towards node **B** until it receives a DIO messages from another node with smaller rank than **B**. As Figure 2.a depicts, node **C** re-enters the

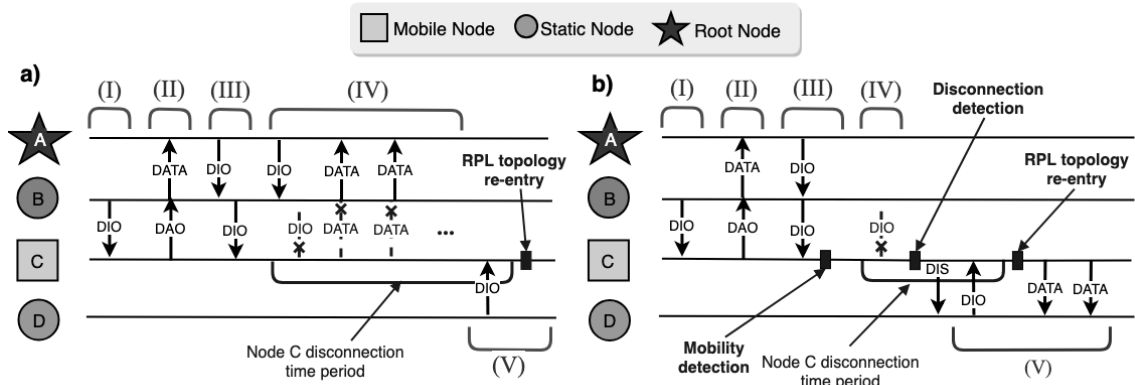


Figure 2. (a) RPL message exchange without mobility support. (b) MARPL mobility support for RPL.

DODAG only at the period **V** when it receives a DIO message from node **D**. By the means of simplicity, in this example, it is assumed that the rank of node **D** is smaller than node **B**.

The behavior of MARPL is illustrated in Figure 2.b. MARPL detects node **C** mobility at moment **III** and its disconnection at moment **IV**. The canonical RPL specification does not specify how DIS control messages can be used to detect or diminish the issues caused by node mobility. MARPL uses DIS messages when it detects the unavailability of the preferred parent to request DIO messages from the potential new parents in the neighborhood of the node, see Figure 2.b moment **IV**. Therefore, at moment **V**, described in Figure 2.b, node **C** could re-enter the DODAG. MARPL could improve the RPL performance in terms of packet delivery rate as a consequence. For example, compare the **C** disconnection time period in Figure 2.b against Figure 2.a.

Section 5 presents a performance analysis of MARPL against the canonical RPL specification [RFC 6550 2012], mRPL [Fotouhi et al. 2015] and D-RPL [Kharrufa et al. 2017].

5. Simulation Results and Analysis

This section presents a performance analysis of MARPL compared to the protocols, RPL [RFC 6550 2012], mRPL [Fotouhi et al. 2015] and D-RPL [Kharrufa et al. 2017]. mRPL and D-RPL were chosen by the following reasons: mRPL [Fotouhi et al. 2015] is a proactive RPL mobility support proposal. In other words, it tries to identify as fast as possible the preferred parent unavailability to re-establish DODAG connection. mRPL aims to achieve this by making the sensor node to send periodic DIS messages to its preferred parent, while monitoring the reception of DIO messages in return. In contrast, MARPL doesn't try to monitor link disconnections since it utilizes Trickle's adjustments based on node mobility to diminish disconnections. D-RPL, proposed by [Kharrufa et al. 2017], utilizes a reverse Trickle adjustment so that each time a sensor node identifies a RSSI difference. Differently, MARPL's Trickle adjustment stands on the node's neighbor variability monitoring.

A total of 20 simulations were executed using the Cooja simulator [Osterlind et al. 2006] for each routing protocol. Statistical analysis was also performed.

Cooja is a simulation tool for the Contiki Operational System [Dunkels et al. 2011]. Contiki was designed to execute in low powered devices commonly utilized in LLNs. The Contiki LLN networking stack is compatible to 6LoWPAN and the IEEE 802.15.4 radio technology.

Table 1 presents the parameters used in the simulations. The Steady-State Random Waypoint [Navidi and Camp 2004] mobility model was used to simulate node mobility. The Steady-State Random Waypoint model extends the Random Waypoint model to enable a period of time of pause for the node [Navidi and Camp 2004]. The simulation scenario is composed by 1 root node and 30 router nodes. All nodes are mobile, but they can remain static for a period of time. The area has 200 meters of width and 200 meters of height. Every node is randomly distributed in the simulation area and a new simulation seed is generated at every execution.

Table 1. Cooja Simulation Parameters.

Parameters	Values
Number of nodes	30 mobile routers e 1 root node
Tipo do dispositivo simulado	Zolertia Z1
Radio	CC2420 [Instruments 2007]
Simulation Time	10 minutes
Node placement	Random
Mobility Model	Steady-State Random Waypoint
Maximum node speed	3 m/s
Maximum pause time	40 seconds
Data generation rate	1 every 8 seconds
Transmission medium	<i>Unit Disk Graph Medium</i> (UDGM)
Radio transmission range	50m
Simulation area	200m x 200m
$T_{monitoring}$	8 seconds

The simulation analysis was performed in terms of: (i) **Packet Delivery Rate (PDR)**: that means the rate of received data packets over sent data packets; (ii) **Packet Delivery Delay (PDD)**: the time needed for a data packet to travel from the router to the root node; (iii) **DODAG Disconnections**: the number of DODAG disconnections caused by the node mobility. This metric enables to evaluate how good a protocol could prevent disconnections; (iv) **DODAG Reconnection Delay**: the time needed for a DODAG disconnection to be solved; (v) **Overhead**: the rate of control packets over the total of packets transmitted to the network; (vi) **Residual Energy**: it was simulated the utilization of a battery for each device in the LLN with 700 Joules of initial energy. The Residual Energy is the percentage of energy in the battery after the network execution.

It is valuable to notice that statistical significance tests were performed for every set of simulation analysis. Therefore, even though some results have different means, they are very likely to belong to similar distributions. The first step for the statistical significance test is to assess if the result set for every routing protocol belongs to a normal distribution. It is assessed by the Shapiro-Wilk test [Shapiro and Wilk 1965]. Thus, Wilcoxon signed-rank test [Wilcoxon et al. 1970] is performed if one of the result distribution can't be assumed to be normally distributed. Student's t-test [Student 1908] is

performed otherwise.

Figure 3 shows the results of PDR and PDD. It's noticeable that MARPL had the greater PDR when comparing it to other evaluated protocols, 23.13%. This MARPL PDR result was possible because of the Trickle adjustments and the utilization of the proposed metric Neighbor Variability (γ) by the preferred parent selection procedure. The PDRs of RPL and mRPL protocols have no statistical difference ($p = 0.296$). Therefore, it's possible to conclude that RPL and mRPL might present similar PDRs in further experiments.

In terms of PDD, RPL had smaller delay on average, 25121 milliseconds. This is because RPL had the smallest overhead. The increase of control message exchange enable the topology to be constantly updated. But, it delays the transmission of data packets because the increase of packet collisions. mRPL, D-RPL and MARPL delay increase is expected because the increase of control message exchange caused by the mobility support proposals. D-RPL had the second smaller PDD in average, 29987 milliseconds. MARPL's PDD was on average 33454 milliseconds. Nevertheless, by our statistical analysis, it is not possible to affirm that D-RPL delay is different from obtained by MARPL ($p = 0.338$). Thus, we conclude that MARPL had a better overall performance in these experiments as a result of its PDR. MARPL has better PDR because of its mobility management mechanisms.

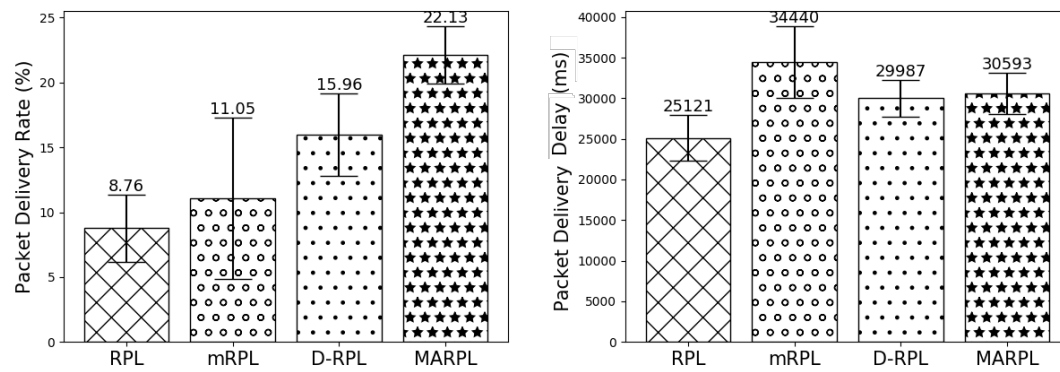


Figure 3. (a) Packet Delivery Rate (b) Packet Delivery Delay.

Figure 4 shows the average overhead and residual energy measured in the simulations. As expected, proposals for RPL mobility support increase overhead because they transmit more control packets in order to detect or predict link disconnections. Therefore, RPL presents the smaller overhead. MARPL had the smallest overhead with 22.67%. D-RPL had an overhead of 24.40%. mRPL had the greatest overhead of 35.96%. It's arguable that mRPL causes more overhead due to its attempts to rapidly solve link disconnections through a frequent exchange of control messages. Among the RPL mobility support proposals, MARPL overhead is the lowest since it only increases control message exchange when it's necessary to update the topology due to link disconnections. This MARPL behavior had direct impact on its PDR results presented in Figure 3. MARPL had the greatest residual energy, 86.35%. Even greater than RPL with 85.73%. mRPL had the smallest residual energy since its overhead was the greatest. We conclude that proactive link disconnection mechanism such as mRPL, significantly increases overhead and

reduces residual energy, notwithstanding having small packet delivery rate and greater delay.

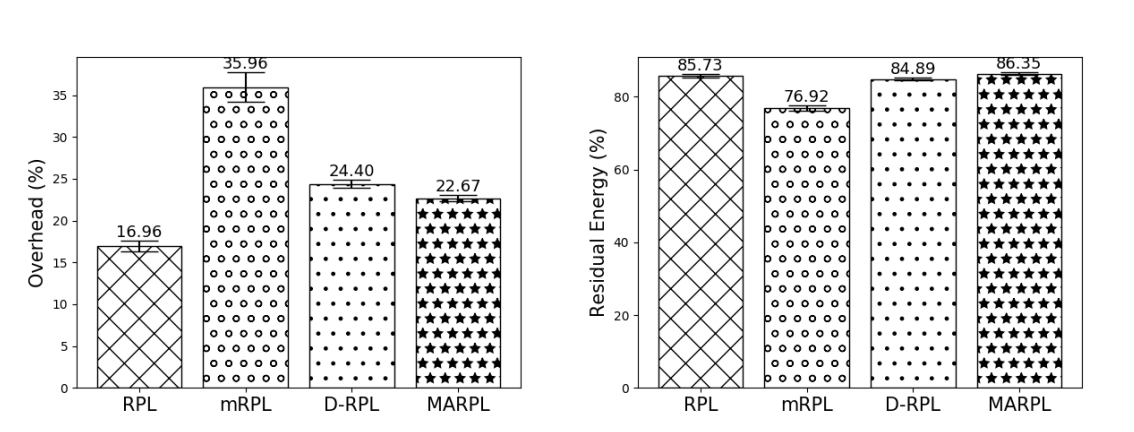


Figure 4. (a) Overhead (b) Residual Energy.

Figure 5 presents the results of DODAG Reconnection Delay. mRPL had the smallest delay on average, 6366 milliseconds. MARPL had the greatest delay with 12901. D-RPL and RPL presented statistically similar results, $p = 0.697$. Even though mRPL had the smallest reconnection delay, it had presented the worst performance in terms of PDD, overhead and residual energy.

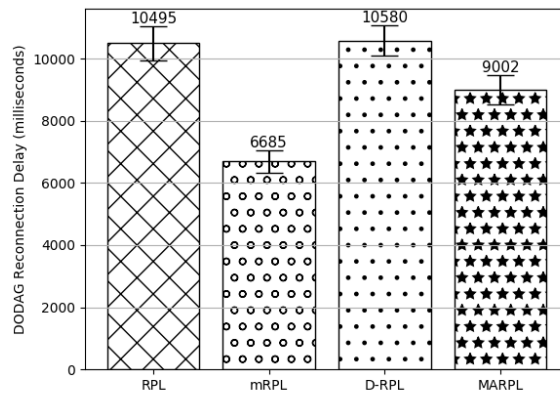


Figure 5. DODAG Reconnection Delay.

6. Conclusion

RPL is a routing protocol for Low Power and Lossy Networks (LLNs), a common type of network formed by Internet of Things (IoT) applications. Mobility support is a requirement for a wide range of IoT applications. Regardless the fact that RPL was initially intended for static LLNs (i.e. LLNs composed only by static devices) it can be enhanced to include mobility support capabilities. Node mobility increases link disconnections and consequently packet loss. RPL faces a series of issues when dealing with mobile nodes. Natively, RPL doesn't have a way to detect node mobility, nor a way of identifying when

a route is not available anymore. This paper discussed the issues RPL faces when dealing with mobile LLNs and approaches to solve them.

This paper depicted the Mobility Aware RPL (MARPL). MARPL intends to enable mobility support for the RPL routing protocol. MARPL is composed by three mechanisms: (i) mobility detection; (ii) preferred parent unavailability detection and; (iii) DIO and DIS transmission adjustment. MARPL was compared against the canonical RPL [RFC 6550 2012] and more two RPL mobility support proposals found in the literature: mRPL [Fotouhi et al. 2015] and D-RPL [Kharrufa et al. 2017]. This paper also acquaint the statistical analysis performed upon the simulation results.

The results indicates that MARPL has better performance in terms of overhead, residual rnergy and packet delivery rate. MARPL boosts DODAG updates when necessary to minimize DODAG disconnection consequences. Since MARPL has better PDR in comparison against RPL, mRPL and D-RPL, DODAG updates based on node mobility enables greater packet delivery rate.

This paper concludes that further studies can be done to improve MARPL transmission delay while maintaining high PDR, overhead and residual energy. D-RPL and MARPL had similar packet delivery delay. RPL outperforms all the three proposals in terms of packet delivery delay. Therefore, it is possible to conclude that RPL mobility support increases mobile LLN PDR and overhead, since more control packets are transmitted to detect or predict DODAG disconnections. Higher overhead in comparison against the canonical RPL is expected since node mobility management requires an increase of control messages. Therefore, enabling the topology to update faster. We conclude that it's necessary for the RPL mobility management to only increase control message transmissions when there are mobile nodes in the topology, in order to update the DODAG. MARPL only increases control message transmission when a node has mobile nodes connected to it in its neighborhood.

References

- Bouaziz, M., Rachedi, A., and Belghith, A. (2017). EKF-MRPL advanced mobility support routing protocol for internet of mobile things: Movement prediction approach. *Future Generation Computer Systems*, pages 19–24.
- Cobarzan, C., Montavont, J., and Noel, T. (2014). Analysis and performance evaluation of RPL under mobility. *Computers and Communication (ISCC)*, pages 1–6.
- Dunkels, A., Schmidt, O., Finne, N., Eriksson, J., Österlind, F., Tsiftes, N., and Durvy, M. (2011). The Contiki OS: The operating system for the internet of things. *Online at <http://www.contikios.org>. Accessed at December 2018.*
- Fotouhi, H., Moreira, D., and Alves, M. (2015). mRPL: Boosting mobility in the internet of things. *Ad Hoc Networks*, pages 17–35.
- Fotouhi, H., Zuniga, M., Alves, M., Koubâa, A., and Marrón, P. (2012). Smart-Hop: A reliable handoff mechanism for mobile wireless sensor networks. In *European Conference on Wireless Sensor Networks*, pages 131–146, Berlin, Germany. Springer.
- Gara, F., Saad, L. B., Hamida, E. B., Tourancheau, B., and Ayed, R. B. (2016). An adaptive timer for RPL to handle mobility in wireless sensor networks. In *Wireless*

- Communications and Mobile Computing Conference (IWCMC), 2016 International*, pages 678–683, Paphos, Cyprus.
- Gubbi, J., Buyya, R., Marusic, S., and Palaniswami, M. (2013). Internet of Things (IoT): A vision, architectural elements, and future directions. *Future generation computer systems*, 29(7):1645–1660.
- Instruments, T. (2007). CC2420 datasheet. *Reference SWRS041B*, pages 1–93. <http://www.ti.com/product/CC2420>, December.
- Iova, O., Picco, P., Istomin, T., and Kiraly, C. (2016). Rpl: The routing standard for the internet of things... or is it? *IEEE Communications Magazine*, 54(12):16–22.
- Kharrufa, H., Al-Kashoash, H., Al-Nidawi, Y., Mosquera, M. Q., and Kemp, A. H. (2017). Dynamic RPL for multi-hop routing in IoT applications. In *Wireless On-demand Network Systems and Services (WONS)*, pages 100–103, Jackson, WY, USA. IEEE.
- Ko, J. and Chang, M. (2015). MoMoRo Providing Mobility Support for Low-Power Wireless Applications. *IEEE Systems Journal*, 9(2):585–594.
- Navidi, W. and Camp, T. (2004). Stationary distributions for the random waypoint mobility model. *IEEE transactions on Mobile Computing*, (1):99–108.
- Oppitz, M. and Tomsu, P. (2018). Internet of things. In *Inventing the Cloud Century*, pages 435–469. Springer.
- Osterlind, F., Dunkels, A., Eriksson, J., Finne, N., and Voigt, T. (2006). Cross-level sensor network simulation with cooja. In *2006 31st IEEE Conference on Local Computer Networks*, pages 641–648, Tampa, FL, USA. IEEE.
- Paul, P. V. and Saraswathi, R. (2017). The internet of things—a comprehensive survey. In *Computation of Power, Energy Information and Communication (ICCPEIC), 2017 International Conference on*, pages 421–426, Melmaruvathur, India. IEEE.
- RFC 6206 (2011). The trickle algorithm. *No. RFC 6206*, pages 1–13.
- RFC 6550 (2012). RPL: IPv6 Routing Protocol for Low-Power and Lossy Networks. *No. RFC 6550*, pages 1–157.
- Sethi, P. and Sarangi, S. R. (2017). Internet of things: architectures, protocols, and applications. *Journal of Electrical and Computer Engineering*, 2017:1–26.
- Shapiro, S. and Wilk, M. (1965). An analysis of variance test for normality. *Biometrika*, 52(3):591–611.
- Student (1908). The probable error of a mean. *Biometrika*, pages 1–25.
- Wilcoxon, F., Katti, S., and Wilcox, R. A. (1970). Critical values and probability levels for the wilcoxon rank sum test and the wilcoxon signed rank test. *Selected tables in mathematical statistics*, 1:171–259.
- Zhao, M., Kumar, A., Chong, P. H. J., and Lu, R. (2017). A comprehensive study of RPL and P2P-RPL routing protocols: Implementation, challenges and opportunities. *Peer-to-Peer Networking and Applications*, 10(5):1232–1256.