



Performance evaluation of the routing protocol for low power and loopy networks in the context of node mobility

Ibtissem Zaatouri, Francoise Sailhan, Stephane Rovedakis, Awatef Ben Fradj Guiloufi, Nouha Alyaoui, Abdennaceur Kachouri

► To cite this version:

Ibtissem Zaatouri, Francoise Sailhan, Stephane Rovedakis, Awatef Ben Fradj Guiloufi, Nouha Alyaoui, et al.. Performance evaluation of the routing protocol for low power and loopy networks in the context of node mobility. 16th International Multi-Conference on Systems, Signals & Devices, Mar 2019, Istanbul, Turkey. pp.1-6. hal-02468566

HAL Id: hal-02468566

<https://hal.science/hal-02468566>

Submitted on 4 Jan 2022

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

Performance Evaluation of Mobility-Aware Routing Protocol for Low power and Lossy Networks

Ibtissem Zaatouri^{*}, Francoise Sailhan[†], Stephane Rovedakis[‡], Awatef Guiloufi[‡], Nouha Alyaoui[‡], Abdennaceur Kachouri[‡]

^{*}National Engineering School of Gabes, LETI, Tunisia

[†]National Conservatory of Arts and Crafts, CEDRIC Laboratory, France

[‡]National Engineering School of Sfax, LETI, Tunisia

Email: *ibtissem.zaatouri@hotmail.com

[†]firstname.familyname@cnam.fr

[‡]benfradj_awatef@yahoo.fr

[‡]alyaoui_nouha@yahoo.fr

[‡]abdennaceur.kachouri@enis.rnu.tn

Abstract—Wireless Sensor Networks (WSNs) is a class of low power and Lossy networks (LLN) which are useful to monitor a physical/environmental phenomenon. The Routing Protocol for Low Power and lossy networks (RPL) is a standardized protocol specifically designed to route the data over LLN networks. In particular, to take into account the specific features of LLN the RPL protocol computes the best routes following one or several metrics (e.g., the residual energy of nodes or the link quality) based on the state of the nodes (a.k.a sensors) or links. Nevertheless, RPL was originally designed to handle static networks ; when node(s) move(s), the performances of RPL drop down. This issue has motivated a great deal of interest in ensuring that sensors may route the sensing data even though they move from place to place, establishing new links and moving away from the previously established links. In this paper, we present RPL and the solutions that permit to deal with mobile nodes. We introduce a simulation-based evaluation of these solutions, which is focused on the energy consumption and the packet losses. The comparison between RPL and MRPL shows that RPL consumes less energy due to many packet losses. In counterpart, RPL does not provide a continuous connectivity to mobile sensor contrary to MRPL that keeps connection.

Index Terms—IoT, RPL, Mobility.

I. INTRODUCTION

Nowadays, Wireless Sensor Networks (WSNs) are used by many applications pertaining e.g. to the industrial or military domains [1]. The specific features of WSNs (wireless communication, small size, low price) stimulate their deployment everywhere. However, sensors are characterized by a relatively limited capacity expressed in terms of energy, memory, computing power and throughput. These distinctive features require to design a routing protocol that is adapted to WSN [2]. In particular, RPL [3] (Routing Protocol for Low Power and Lossy networks) is one of the best candidates that was proposed by IETF ROLL

(Routing Over Low Power and Lossy Network) working group. RPL was originally designed to support data collection in static networks (i.e., without any consideration for mobility). Nevertheless, new trend seeks to handle more complex applications that involve mobile users equipped with sensors. In this paper, we present the versions of RPL that accommodate mobility. We further compare RPL with one version of RPL, which is adapted for mobility and is called MRPL [4].

The rest of the paper is organized as follows: Section II provides an overview of RPL with regards to mobility, followed in Section III by the related work. Then, Section IV introduces a comparison between RPL and MRPL. Finally, a synthesis is presented in the conclusion with a description of the future works.

II. OVERVIEW OF RPL

RPL [5] is a proactive and distance-vector protocol that aims at building a DODAG (Destination Oriented Directed Acyclic Graph), which is further used to disseminate data packets. A DODAG is a directed acyclic graph rooted at a particular node, called sink or root. RPL supports three types of traffic: point-to-multipoint a.k.a from sink to all nodes, multipoint-to-point a.k.a from all nodes to sink and point-to-point a.k.a from node to node. RPL attempts to deal with sensors having either sufficient or insufficient storage. In the former case (versus later case), routing information are stored (versus not stored) by the intermediate sensors. During the construction of the DODAG, each node identifies a set of parents so that a path towards the DODAG root (also called a upward path) can be established. This selection is operated based on a predefined Objective Function (OF) that implements one or a combination of metrics.

Establishment of upward route- In order to build

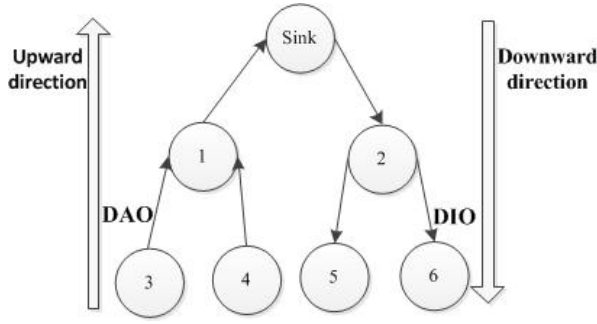


Fig. 1. DAG composed of a single DODAG.

the DODAG (see Figure 1), the sink periodically sends a so-called DIO message (DODAG Information Object) to its neighbors, which includes the following parameters: a rank¹ corresponds to the position of the node in the DODAG, a routing metric is used to determine the best path to the sink, as defined by the OF, a DODAG id, and a path cost (associated to a path toward the sink, as given by the OF). Based on these parameters, the sink neighbors establish the sink as parent and join the DODAG. Each node belonging to a DODAG sends a DIO to its neighbours. In order to join a DODAG, each node has to select as parent one of its neighbors that belongs to the DODAG. Such a parent selection is governed by an Objective Function (OF), which relies on the routing metric(s). This selection is periodically repeated so that each node includes in its routing table its (actual) parent(s) and its relative position (i.e., rank and path cost). In order to keep to a minimum the amount of DIO messages that are exchanged so as to maintain and update the DODAG, RPL relies on the Trickle algorithm [5] which tunes the time period between the emission of two consecutive DIO messages. This algorithm increases (versus decreases) this time period when the network remains stable (versus when changes are detected). Rather than waiting for a DIO message, any node may also request a DIO message by sending a DIS (DODAG Information Solicitation) message to its neighbours.

Once the DODAG is established (using DIO and/or DIS messages), any data packet can be routed, hop-by-hop, towards the sink: this constitutes a multipoint-to-point interaction pattern. In order to support a point-to-multipoint interaction pattern wherein the sink can send data packets to the sensors, downward routes are computed.

Establishment of Downward Route - In order to establish the downward route, a DAO (Destination Advertisement Object) message is propagated from the leafs towards the sink. This DAO message contains the network prefix so that the sink can further establish which network (and which constituting sensors) is under its responsibility.

¹A rank is a strictly increasing value in the downward direction of the DODAG

RPL relies on a Neighbor Discovery mechanism [6] so as to discover new/unavailable neighbors. This mechanism uses the following ICMPv6 control messages:

- *Neighbor Solicitation* checks if a neighbor is still accessible.
- *Neighbor Advertisement* is sent periodically to announce link changes.
- *Router Solicitation* requests the neighbours of a node.
- *Router Advertisement* corresponds to a response to a route solicitation. It includes link information.

On one hand, sending the above messages very frequently involves a high responsiveness given topology changes, but induces a high overhead; on the other hand, sending less frequently these messages implies a low responsiveness and a reduced traffic.

RPL shortcomings - RPL was originally designed to handle static sensors that communicate using a wireless and low data rate network, which may be subject to high loss rates and instabilities. Thus, RPL assumes that topological changes are mostly due to sensors disconnections and to wireless instabilities. In order to detect sensors disconnections and link instabilities, DIO messages are periodically sent. The periodicity is defined by the Trickle algorithm [7], which is tuned based on the number of consistent DIO messages received: if there are no update, the periodicity is increased, otherwise active dissemination is performed by setting this period to the minimum value. With the Trickle algorithm, the time is divided into intervals of size I (with $0 < I < I_{max}$). Any node listens during the first half of the I interval and then transmits during the second half. The transmission takes place if the number of inconsistent messages received is less than a given threshold. If the message cannot be sent within I , then this interval I is doubled (until I_{max} is reached) and the node sends again the message. But, when an inconsistent message is detected, Trickle sets I to its minimum value. The Trickle algorithm presents two major shortcomings. First, the listen period increases the transmission delay and henceforth the convergence time of RPL. Consequently, the time induced to detect inconsistency [8] is increased. When nodes are mobile, Trickle also delays the selection of a new parent for mobile nodes, which results in high packet loss, energy consumption and end-to-end latency. Thus, the trickle algorithm and the RPL protocol should be adapted to accommodate mobile nodes.

III. RELATED WORK: TOWARDS MOBILE-AWARE RPL

Dealing with the node mobility involves performing the three following tasks:

- **Detection of the node mobility** consists in determining whether a node is mobile. There are three options to detect the mobility. First, a mobile node may simply announce its mobility

(Reverse Trickle [9]). Second, the mobility can be established/anticipated based on the node history (ME-RPL [9]). Third, the network is monitored so as to determine whether a node is mobile and if the disconnection with parent will take place. For this purpose, different parameters can be used. For instance, the jitter between two DIO messages can be analyzed by the mobile node (MRPL [4]). The RSSI can also be analyzed by the mobile node (MRPL, Mod-RPL [10]) or by the parent (EC-MRPL [11]) so as to determine when the mobile node will get out of range. In these cases, the nodes need to overhear. When the presence of a mobile node is detected, the periodicity of the DIO messages (CO-RPL, Mod-RPL) and DIS messages (CO-RPL, ME-RPL) is increased so as to maintain the connectivity with the mobile node.

- **Getting connected to the best parent** - the selection of the best parent (among potential neighbour(s)) can be performed by the mobile node (Reverse Trickle) or by the previous parent (EC-MRPL). This selection is usually based on the (best) RSSI (CO-RPL, MRPL). Note that in order to ensure the stability of the DODAG, mobile nodes should not act as a parent (MRPL, ME-RPL, EC-MRPL, Reverse Trickle).
- **Forwarding the data traffic to the newly selected parent** - while the node is connected to another parent, the data traffic should be forwarded by the previous parent to the new parent (EC-MRPL, MRPL+).

In the following, we detail the solutions that have been proposed so that RPL supports mobile sensors. With the CO-RPL protocol [12], the network is divided into some circular regions, named *coronas*, which are characterized by a certain radius (corresponding typically to the maximum transmission range) and which are centered around the DODAG root. Each corona is identified by a corona ID. This corona ID presents the distance from mobile node to the sink. Each node belongs to one corona. This notion of corona permits to select the parents among the closest sensors (i.e., on same corona id). In practice, the node selects as the best parent, the candidate that has the lowest corona id, and if the potential candidates have the same corona id, then the one having the best link quality is selected. In order to deal with the sensor mobility without flooding the network with DIO messages, CO-RPL adapts the transmission rate of DIO messages according to the sensor speed. In addition, a mobile node that joins the DODAG immediately broadcasts a new DIO message, ignoring the Trickle timer. CO-RPL prevents packet loss by (i) quickly detecting sensor disconnection and (ii) searching for an alternative path: when a node does not receive a DIO message from its parent, it sends DIS messages to inform its children, in order to stop forwarding data packets until an alternative parent is found. This protocol is evaluated using the Cooja [13] simulator. All the nodes follow a random mobility

model (except the sink which is static): the destination, the speed and the direction of each mobile node is chosen independently and randomly. Thus, CO-RPL induces (i) a significant communication overhead due to the high-periodical broadcast rate associated with DIO messages, (ii) congestion and (iii) an increase of the energy consumption.

In [11], the authors propose a new proactive protocol named EC-MRPL, which prevents mobile sensor from acting as RPL routers: only static nodes should act as parents so as to guarantee the stability of the core RPL DODAG. The parent of a mobile node is responsible for monitoring the connection with the mobile node and for choosing an alternative parent if necessary. EC-MRPL enhances the routes stability by using mobile sensors only as leaf nodes. However, the failure of the parent causes a loss of connection in the network. This protocol is tested with the Cooja simulator with a static sink and a single mobile sensor that follows a random walk model with horizontal moves from the one corner to the other with a constant speed ($v=2$ m/s).

In [4], authors introduce the MRPL protocol wherein mobile sensors use the DIO messages to monitor (i) the link quality with its parent and (ii) the jitter to establish that the mobile sensor moves away. In such a case, the mobile sensor triggers a hard hand-off which consists in choosing as parent the node having the highest RSSI. This hard hand-off mechanism consists of the two following phases:

- **Monitoring phase** during which the mobile node monitors the quality of link with its parents. The quality is determined when the mobile node receives packet emitted by its parent. If the Received Signal Strength Indicator (RSSI) value is lower than a given threshold, the next phase starts.
- **Parent selection phase** - the mobile node selects as parent the node characterised by the highest RSSI value.

Unfortunately, in order to establish the RSSI, many DIO message should be frequently sent. With EC-MRPL and MRPL, mobile sensors act only as leaf nodes to reduce instabilities and path changes. Simulations and real experiments are performed. During simulations, one mobile node travels 15 times between the left and bottom corner and the right bottom corner with a constant speed of 2 m/s in a row in first scenario and in two parallel rows in the second. Real experiments are achieved with one person equipped with a sensor.

MRPL+ [14] supports both a hard and soft hand-off. The hard hand-off takes place when a sensor does not receive a DIO message from its parent within a given time period or when the sensor observes a low amount of data traffic. In such a case, the connection between the mobile node and the preferred parent is canceled and then a new connection is established (see MRPL for detail). With soft hand-off, the connection between the mobile node and the new parent is activated before the current link is dropped so as to avoid losing data

traffic. The soft hand-off, which constitutes the default option, necessitates an overhearing mechanism so that mobile nodes monitor the link with their parent and measure RSSI. A mobile node is defined as mobile (i) if the link with the parent is inactive during a given period of time and/or (ii) the RSSI strength varies. The mobile node is placed in an overhearing state to monitor the link quality and to discover potential alternative parents. Such an overhearing process is especially energy-consuming. MRPL+ introduces the soft hand-off that enables continuous connection between the mobile node and its parents. This reduces the packet loss in comparison to EC-MRPL.

With ME-RPL (Mobility Enhanced RPL) [15], mobile nodes announce their mobility by adding a mobile flag to the DIO message. Thus, sensors do not monitor the link state to determine that a node is mobile. Only static nodes can be parents. Authors suppose that the mobility of the node in the near future can be established based on its past mobility. So, if a node has been stable (versus unstable) for a while, the DIS interval increases (versus decreases). Overall, DIS messages are used to add nodes to a DODAG and to refresh the connection to a preferred parent; the frequency of DIS messages is regulated following the history of the mobile nodes. This protocol is tested based on the Cooja simulator with a static sink and one mobile node that moves depending on linear and grid topologies. However, the variation of the DIS interval implies that the disconnection of the mobile nodes is not rapidly detected.

Mod-RPL [10] customizes the trickle timer to accommodate mobility. As a first step upon that goal, any mobile node relies on the RSSI so as to determine when it gets out of communication range from its preferred parent. Based on this estimation, the mobile node adjusts the DIO interval. This protocol is tested with Cooja simulator considering a static sink, 10 senders and a varying number of mobile nodes (ranging from 20 to 100). The velocity of mobile nodes varies from 0 to 10 m/s. This protocol reduces overhead but it is tricky to determine when a mobile node is out of range of the parent when the parent is also mobile.

In [9], the authors introduce the Reverse Trickle timer to deal with mobile nodes. As with ME-RPL [15], mobile nodes advertise the fact that they are mobile. But in this case a mobile node can only act as a leaf node. This work supposes that the more the mobile node remains connected to the same parent, the more it is likely to move outside the coverage of the parent. Therefore, if a node advertises its mobility, its parent reduces drastically the (Reverse) Trickle timer. In a nutshell, this timer starts with the maximum allowed value and is halved when a DIO message is received. Once the minimum value is reached, the Reverse Trickle timer is reset (i.e., is given the maximum value). Simulations have been performed using the WSNNet simulator by considering a scenario with only

one mobile node moving along a linear trajectory with a constant speed of 2 m/s. Moreover, real experiment has been performed with 100 static nodes and 10 mobile nodes moving randomly inside the area covered by the fixed nodes. The DODAG root is located in the middle of the network area. This protocol reduces the packet loss, but the mobile node stability does not always decrease over time since it is deeply related to the model of mobility.

Table 1 summarizes the characteristics of the aforementioned works that accommodate mobility.

IV. COMPARISON BETWEEN RPL AND MRPL

Our objective is to compare RPL and MRPL with regards to mobility. We performed 1-hour simulations with the Cooja network simulator. We consider two scenarios (Figures 2 and 3) in which 12 sensors are static (including the sink) and one is mobile. Sensors correspond to Tmote Sky nodes [16] that run Contiki 2.7 OS and move following the random way point mobility model. The nodes move: (i) horizontally at 2 m/s, from the left and bottom corner to the right bottom corner, and then, returns to the original position in the first scenario, and (ii) from the bottom to the top corner and then returns to its original position in the second scenario.

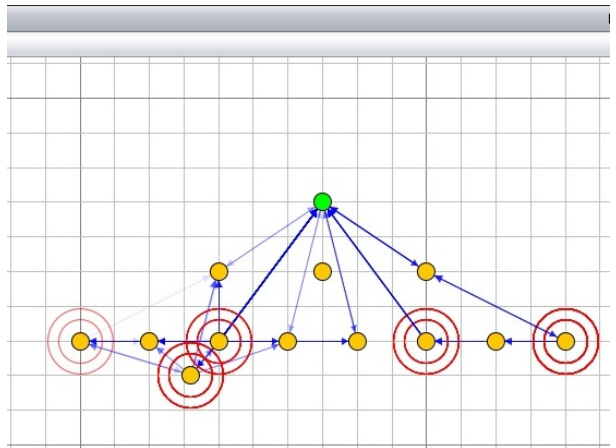


Fig. 2. Scenario i - Simulation Environment

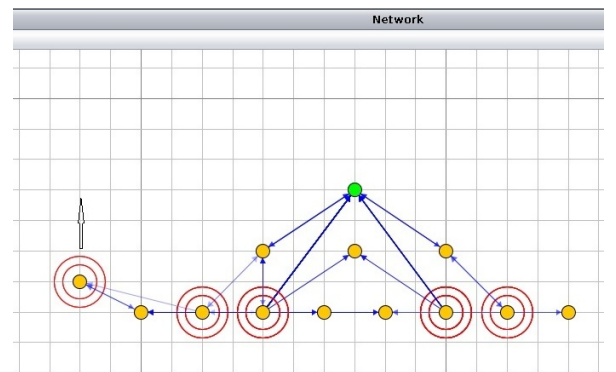


Fig. 3. Scenario ii - Simulation Environment

The comparison between RPL and MRPL is performed according to the following two criteria:

TABLE I
RPL METRIC IN MOBILE ENVIRONMENTS

Protocol	Mechanisms	Mobility detection	Nodes	Results
CO-RPL [12]	Path recovery RSSI Beaconsing	Corona	1 static sink Mobile senders	Low packet loss High energy consumption High Control messages High overhead Low stability
EC-MRPL [11]	Only static nodes act as router Distributing the consumption among different static nodes	Static node Detects the mobility	1 static sink Static senders 1 mobile sender	Low energy consumption Low Control messages Low Overhead
MRPL [4]	Hard hand-off	Timers RSSI Beaconsing	1 static sink Static senders 1 mobile sender	High energy consumption High Control messages Low Overhead
MRPL+ [14]	Hard and soft hand-off	Timers RSSI Beaconsing Overhearing	1 static sink static senders 1 mobile sender	High energy consumption High Control messages Low Overhead High stability
ME-RPL [15]	Identification of mobile node	Adaptive DIS interval	1 static sink Static senders 1 mobile sender	High Control messages High Energy consumption Low packet loss
Mod-RPL [10]	TL timer MN localization	RSSI Beaconsing	1 Sink statique 10 static senders [20-100] mobile senders	Low Control messages Low Energy consumption
Reverse Trickle [9]	Reverse-Trickle	DAO mobility flag	1 static sink Static senders 1 mobile sender	Low Control messages Low Energy consumption Low Overhead

- **packet loss** is defined as the number of dropped packets during the simulation.
- **energy consumption** (E_i) is calculated as follows:

$$E_i = \frac{(T_{CPU} \times I_{CPU} + T_{RX} \times I_{RX} + T_{TX} \times I_{TX}) \times V}{f} \quad (1)$$

where I_{CPU} , I_{TX} , I_{RX} represent respectively the energy consumed during the time during which the CPU runs (T_{CPU}), the sensor transmits (T_{TX}), the sensor listens (T_{RX}). V represents the initial battery voltage level and f represents the number of ticks per second. In practice, $I_{CPU} = 1.8$ mA, $I_{TX} = 19.5$ mA, $I_{RX} = 21.8$ mA, $V = 3.6$ V, $f = 32\,768$ Hz.

We present in Figures 4 and 5 the results with regards to the two scenarios, considering a ratio of 6 data packets or 12 data packets per minute for each sensor.

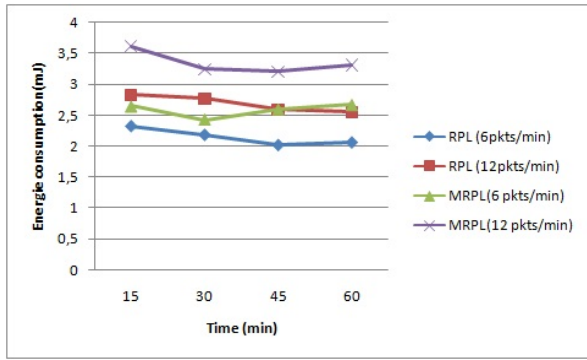
Energy consumption - Figures 4(a) and 5(a) present the energy consumption of RPL and MRPL under Scenario (i) and (ii) respectively. As expected, both RPL and MRPL consume more energy when the packet transmission rate increases. The increase is explained by the additional amount of data packets transmitted. RPL is characterized by the lowest energy consumption because RPL induces less control messages. In counterpart, RPL does not provide a continuous connectivity to mobile sensor(s): this is suggested by the high rate of lost packets of RPL. MRPL is characterized by a significant energy consumption because a mobile node overhears so as to receive any packet sent by the neighbors and detect any change in the topology. In order to detect any topology changes, the mobile node sends many control messages (i.e., DIS and DIO messages). With Scenario

(i), RPL and MRPL consume more energy compared to Scenario (ii). This result can be explained by the fact that in Scenario (i), nodes move horizontally keeping almost the same distance to the sink (3 hops), while in Scenario (ii), the nodes move vertically and hence approaches the sink (the number of hops decreases and therefore the energy consumption decreases)

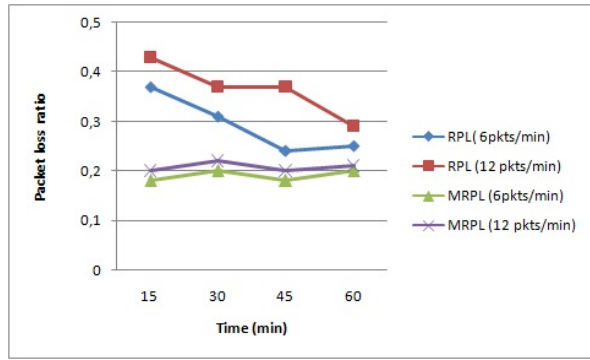
Packet loss - Figures 4(b) and 5(b) present the packet loss under Scenarios (i) and (ii) respectively. The packet loss ratio increases with the packet transmission rate due to the saturation of the wireless medium. However, the packet loss decreases over time, especially for RPL. RPL is characterized by the highest packet loss because RPL is characterized by a low responsiveness to topology changes: mobile sensors receive infrequent control messages and are often not connected to any parent. In the contrary, with MRPL, the mobile sensor sends data to the previous parent during the discovery of a new parent, which allows to maintain a higher delivery ratio. The packet loss ratio for RPL and MRPL in Scenario (i) is more important than in Scenario (ii) since the mobile node is located near the sink in Scenario (ii).

V. CONCLUSION

This paper reviews recent advances that support sensor mobility with regards to the RPL routing protocol. This work presents a description of the challenges encountered by RPL and the recent mechanisms proposed in the state of the art to enhance RPL for mobility. A performance evaluation of RPL and MRPL has been undertaken under two particular node mobility scenarios with respect to two parameters: packet loss and energy consumption. MRPL successfully deals with packet loss but is characterized by a high energy

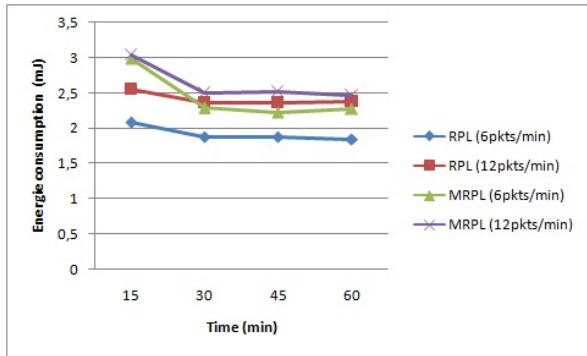


(a) Energy Consumption

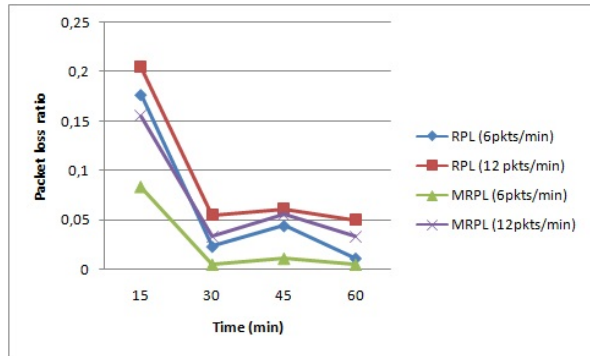


(b) Packet Loss Ratio

Fig. 4. Impact of the data transmission rate on the energy consumption (a) and packet loss (b) for RPL and MRPL under Scenario (i).



(a) Energy Consumption



(b) Packet Loss Ratio

Fig. 5. Impact of the data transmission rate on the energy consumption (a) and packet loss (b) for RPL and MRPL under Scenario (ii).

consumption compared to RPL. This can be explained by the fact that MRPL sends (and overhear) more control messages to detect network changes. However, MRPL is characterized by a low packet loss ratio because MRPL performs a soft hand-off: when a new parent need to be selected, data are forwarded to the previous parent.

REFERENCES

- [1] JA.Stankovic, AD.Wood, and T.He. Realistic applications for wireless sensor networks. In *Theoretical aspects of distributed computing in sensor networks*, pages 835–863. Springer, 2011.
- [2] D.Carels, E.De Poorter, I.Moerman, and P.Demeester. Rpl mobility support for point-to-point traffic flows towards mobile nodes. *International Journal of Distributed Sensor Networks*, 11(6):470349, 2015.
- [3] T.Winter, P.Thubert, A.Brandt, J.Hui, R.Kelsey, P.Levis, K.Pister, R.Struik, JP.Vasseur, and R.Alexander. Rpl: Ipv6 routing protocol for low-power and lossy networks. Technical report, Internet Engineering Task Force (IETF), 2012.
- [4] H.Fotouhi, D.Moreira, and M.Alves. mrpl: Boosting mobility in the internet of things. *Ad Hoc Networks*, 26:17–35, 2015.
- [5] A.Oliveira and T. Vazo. Low-power and lossy networks under mobility: A survey. *Computer Networks*, 107:339–352, 2016.
- [6] T.Narten, E.Nordmark, W.Simpson, and H.Soliman. Neighbor discovery for ip version 6 (ipv6). Technical report, Neighbor discovery for IP version 6 (IPv6), 2007.
- [7] M.Vuini, M.Krl, B.Jonglez, T.Coladon, and B.Tourancheau. Trickle-d: high fairness and low transmission load with dynamic redundancy. *IEEE internet of things journal*, 4(5):1477–1488, 2017.
- [8] B.Ghaleb, A.Al-Dubai, and E.Ekonomou. E-trickle: Enhanced trickle algorithm for low-power and lossy networks. In *IEEE International Conference on Computer and Information Technology; Ubiquitous Computing and Communications; Dependable, Autonomic and Secure Computing; Pervasive Intelligence and Computing (CIT/IUCC/DASC/PICOM)*, pages 1123–1129. IEEE, 2015.
- [9] C.Cobzarzan, J.Montavont, and T.Noel. Analysis and performance evaluation of rpl under mobility. In *IEEE Symposium on Computers and Communication (ISCC)*, pages 23–26, 2014.
- [10] F.Gara, L.BenSaad, E.BenHamida, B.Tourancheau, and R.BenAyed. An adaptive timer for rpl to handle mobility in wireless sensor networks. In *International Conference on Wireless Communications and Mobile Computing Conference (IWCMC)*, pages 678–683. IEEE, 2016.
- [11] M.Bouaziz, A.Rachedi, and A.Belghith. Ec-mrpl: An energy-efficient and mobility support routing protocol for internet of mobile things. In *14th IEEE Consumer Communications & Networking Conference (CCNC)*, pages 19–24, 2017.
- [12] 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 *9th IEEE International Symposium on Industrial Embedded Systems (SIES)*, 2014.
- [13] Fredrik Osterlind, Adam Dunkels, Joakim Eriksson, Niclas Finne, and Thiemo Voigt. Cross-level sensor network simulation with cooja. In *Local computer networks, proceedings 2006 31st IEEE conference on*, pages 641–648. IEEE, 2006.
- [14] H.Fotouhi, D.Moreira, M.Alves, and P.Meumeu Yonsi. mrpl+: A mobility management framework in rpl/6lowpan. *Computer Communications*, 104:34–54, 2017.
- [15] I.El Korbi, M.Ben Brahim, C.Adjih, and L.Azouz Saidane. Mobility enhanced rpl for wireless sensor networks. In *Third International Conference on the Network of the Future (NOF)*, pages 1–8, 2012.
- [16] PO.Kamgueu, E.Nataf, and TN.Djotio. On design and deployment of fuzzy-based metric for routing in low-power and lossy networks. In *IEEE 40th Local Computer Networks Conference Workshops (LCN Workshops)*, pages 789–795. IEEE, 2015.